

45/2004-05

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

SUBDIRECCIÓN DE TERCER CICLO Y POSTGRADO

Reunido el día de la fecha, el Tribunal nombrado por el Excmo. Sr. Magfco. de esta Universidad, y finalizada la defensa y discusión de esta tesis doctoral, los señores miembros del Tribunal, emiten la siguiente calificación global:

é>¿^hiMi¿'Xte^€ ¿(JiM .-//^¿ic p^ Uñci¿^i'VA'^<^'^d'

Votos favorables:

^

Las Palmas de Gran Canaria, a 21 de mayo de 2005

El Presidente: Dr. D. Carlos Artemip Coello Coello

El Secretario: Dr D. ~~Bias~~ Galván González

El Vocal: Dr. D. Jacques Periaux

El Vocal: Dr. D. Francisco Herrera Triguero

El Vocal: Dr. D. francisco Chirino Godoy

^pCl><V\kv:H

El Doctorando: D. David Juan Greiner Sánchez

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Escuela Técnica Superior Ingenieros Industriales

Departamentos de
Informática y Sistemas, Matemática Aplicada e Ingeniería Civil

Programa de Doctorado:
Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería



Tesis Doctoral

'Optimización multiobjetivo de pórticos metálicos
mediante algoritmos evolutivos'

David Juan Greiner Sánchez

2005

Don Luis Mazorra Manrique de Lara, SECRETARIO DEL DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS DE LA UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA,

CERTIFICA,

Que el Consejo del Departamento, en su sesión ordinaria de fecha 11 de marzo de 2005, tomó el acuerdo de dar el consentimiento para su tramitación a la tesis doctoral titulada "**Optimización Multiobjetivo de Pórticos Metálicos mediante Algoritmos Evolutivos.**" presentada por el doctorando Don David Juan Greiner Sánchez (DNI 42.877.798) y dirigida por los Doctores Don Gabriel Winter Althaus y Don José María Emperador Alzóla.

Y para que así conste, y a efectos de lo previsto en el Artº 73.2 del Reglamento de Estudios de Doctorado de esta Universidad, firmo la presente en Las Palmas de Gran Canaria, a catorce de marzo de dos mil cinco.

Anexo II



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Departamento: Informática y Sistemas

Programa de Doctorado:
Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería

Título de la Tesis

'Optimización multiobjetivo de pórticos metálicos
mediante algoritmos evolutivos'

Tesis Doctoral presentada por D. David Juan Grelner Sánchez

Dirigida por el Dr. D. Gabriel Winter Althaus

Codirigida por el Dr. D. José María Emperador Alzóla

El/ia Director/a,

El/ia Codirector/a

El/ia Doctorando/a,

Las Palmas de Gran Canaria, a 30 de Marzo de 2005

A mi familia, especialmente a Juan y Mercedes

Agradecimientos

Sirvan las siguientes líneas como un reconocimiento a todas las personas que de un modo u otro han contribuido al desarrollo de este trabajo.

Al Dr. Gabriel Winter Althaus, por ser mentor, consejero, orientador y estimulador a lo largo de todo el proceso de esta tesis.

Al Dr. José María Emperador Alzóla, por su apoyo y sugerencias mostradas en el desarrollo de esta tesis, especialmente en el ámbito estructural.

Al Dr. Blas Gal van González, por sus sugerencias, apoyo continuo, consejos y capacidad de crear un entorno de trabajo cómodo y amigable, además de haber compartido sus códigos de diseño de sistemas de seguridad empleados en esta tesis.

Al Dr. Pedro Cuesta Moreno, quien hace ya muchos años, me dio la oportunidad de iniciarme en la investigación aún siendo un estudiante de Ingeniería Industrial, al permitir mis pasos iniciales como colaborador del CEANI, ahora división del Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI).

A los compañeros del Instituto de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería, así como del departamento de Ingeniería Civil de la Universidad de Las Palmas de Gran Canaria, que han mostrado su ánimo y apoyo a lo largo de la realización de este trabajo.

Agradecer también la financiación aportada por el Ministerio de Educación, Cultura y Deportes del Gobierno de España, bajo el soporte de la beca de Formación de Profesorado Universitario (FPU) AP2001-3797 disfrutada durante los años 2002 y 2003. Asimismo esta tesis está enmarcada dentro del proyecto DPI2001-3570 titulado *^Subsistemas de Software Inteligente del Despacho, Deslastre y Reposición de Cargas en Sistemas Eléctricos de Potencia'*.

RESUMEN.....	1
CAPÍTULO 1. INTRODUCCIÓN.....	2
1.1 ANTECEDENTES.....	2
1.2 OPTIMIZACIÓN MULTIOBJETIVO DE ESTRUCTURAS PLANAS CON NUDOS RÍGIDOS Y PERFILES METÁLICOS.....	3
1.3 METODOLOGÍA DE OPTIMIZACIÓN PROPUESTA.....	4
1.4 ORGANIZACIÓN DE LA TESIS.....	5
BIBLIOGRAFÍA.....	6
CAPÍTULO 2. ALGORITMOS EVOLUTIVOS MULTICRITERIO.....	7
2.1 ALGORITMOS EVOLUTIVOS.....	7
2.1.1 <i>Introducción. Mirando hacia nuestros orígenes: Sabia Naturaleza</i>	7
2.1.2 <i>La Computación Evolutiva: Un modo Natural de resolver y optimizar problemas</i>	8
2.1.2 A ¿Dónde se contextualizan ?.....	9
2.1.2.2 Aplicaciones.....	9
2.1.3 <i>Reseña histórica</i>	10
2.1.4 <i>Codificación del problema: el cromosoma</i>	10
2.1.5 <i>Operadores básicos</i>	11
2.1.5.1 La Selección.....	11
2.1.5.2 El Cruce.....	14
2.1.5.3 La Mutación.....	15
2.1.6 <i>Otros operadores genéticos</i>	16
2.1.7 <i>Equilibrio Exploración -Explotación</i>	17
2.2 ALGORITMOS EVOLUTIVOS MULTICRITERIO.....	20
2.2.1 <i>Introducción</i>	20
2.2.2 <i>Definiciones</i>	22
2.2.3 <i>Sumario de Estado del Arte de los algoritmos evolutivos multiobjetivo</i>	24
2.2.3.1 Primeros Algoritmos.....	24
2.2.3.2 El elitismo en la optimización evolutiva multiobjetivo.....	25
2.2.4 <i>Métricas Comparativas</i>	28
2.2.5 <i>Casos Test</i>	30
BIBLIOGRAFÍA.....	37
CAPÍTULO 3. OPTIMIZACIÓN MULTICRITERIO ESTRUCTURAL.....	42
3.1 ESTADO DEL ARTE EN INGENIERÍA ESTRUCTURAL-CIVIL DE ALGORITMOS EVOLUTIVOS Y DE OPTIMIZACIÓN MULTIOBJETIVO.....	42
3.1.1 <i>Optimización Estructural con Algoritmos Evolutivos</i>	42
3.1.2 <i>Optimización Estructural Multiobjetivo</i>	50
3.1.2.1 Optimización Clásica.....	50
3.1.2.2 Optimización Evolutiva.....	53
3.2 DEFINICIÓN DEL PROBLEMA ESTRUCTURAL.....	57
3.2.1 <i>Introducción</i>	57
3.2.2 <i>Etapas de Cálculo del Método Directo de la Rigidez</i>	57
3.2.3 <i>La Matriz de Rigidez. Características</i>	58
3.2.4 <i>La Matriz de Rigidez. Ensamblaje y resolución</i>	59
3.2.5 <i>El Ancho de Banda</i>	60
3.2.6 <i>Optimización del Ancho de Banda. Renumeración nodal</i>	61
3.2.7 <i>Resolución Matemática del Sistema de Ecuaciones</i>	62
3.2.8 <i>Determinación de esfuerzos y tensiones a lo largo de todas las barras</i>	64

3.2.9	<i>Efecto de Pandeo</i>	65
3.3	OPTIMIZACIÓN DEL PROBLEMA ESTRUCTURAL.....	68
3.4	CODIFICACIÓN DE SOLUCIONES CANDIDATAS.....	68
3.5	FUNCIONES OBJETIVO.....	69
3.5.1	<i>Función Objetivo 1: Minimización del peso estructural restringido</i>	69
3.5.2	<i>Función Objetivo 2: Minimización del número de tipos de perfiles distintos</i>	71
3.6	ACOPLAMIENTO EVALUADOR ESTRUCTURAL - ALGORITMO GENÉTICO.....	72
3.7	OTRAS CONSIDERACIONES DE OPTIMIZACIÓN.....	73
3.7.1	<i>Introducción</i>	73
3.7.2	<i>Consideraciones respecto a la optimización de tiempo de cálculo</i>	73
3.7.3	<i>Costes computacionales</i>	73
3.7.4	<i>Generación de números pseudoaleatorios</i>	75
3.8	CASOS TEST ESTRUCTURALES.....	77
3.8.1	<i>Caso X</i>	77
3.8.2	<i>Caso Y</i>	79
	BIBLIOGRAFÍA.....	83
CAPÍTULO 4. CONTRIBUCIONES.....		96
4.1	INTRODUCCIÓN.....	96
4.2	UN OPERADOR TRUNCAMIENTO PARA PROBLEMAS BIFUNCIONALES EN EL ALGORITMO SPEA2 ..	97
4.2.1	<i>Introducción</i>	97
4.2.2	<i>Una Propuesta Eficiente del Operador Truncamiento en el SPEA2</i>	97
4.2.3	<i>Resultados Experimentales. Casos Test</i>	99
4.2.4	<i>Conclusiones</i>	103
4.3	ALGORITMOS DENSEA Y DENSEA-NSH.....	104
4.3.1	<i>Introducción</i>	104
4.3.2	<i>Algoritmo DENSEA</i>	105
4.3.3	<i>Algoritmo DENSEA-NSH</i>	108
4.4	LA CODIFICACIÓN EN OPTIMIZACIÓN MULTI OBJETIVO ESTRUCTURAL: CÓDIGO BINARIO FRENTE A CÓDIGO GRAY.....	109
4.4.1	<i>El Código Gray</i>	109
4.4.2	<i>Caso Test X. Comparativa Código Binario / Gray</i>	113
4.4.3	<i>Caso Test Y. Comparativa Código Binario / Gray</i>	114
4.5	EL OPERADOR ELITISMO EN OPTIMIZACIÓN MULTI OBJETIVO ESTRUCTURAL.....	122
4.5.1	<i>Caso Test X. Comparativa Elitismo /No Elitismo</i>	122
4.5.2	<i>Caso Test Y. Comparativa Elitismo / No Elitismo</i>	123
4.6	LA PROBABILIDAD DE MUTACIÓN EN OPTIMIZACIÓN MULTI OBJETIVO ESTRUCTURAL.....	126
4.6.1	<i>Caso Test X. Comparativa de Probabilidades de Mutación</i>	125
4.6.2	<i>Caso Test Y. Comparativa de Probabilidades de Mutación</i>	152
4.7	COMPARATIVA ENTRE ALGORITMOS EVOLUTIVOS MULTI OBJETIVO EN OPTIMIZACIÓN ESTRUCTURAL.....	139
4.7.1	<i>Caso Test X. Comparativa entre Algoritmos</i>	139
4.7.2	<i>Caso Test Y. Comparativa entre Algoritmos</i>	143
4.7.3	<i>Análisis de Resultados. La Diversidad Poblacional</i>	149
4.8	LA MULTI OBJETIVIZACIÓN DEL PROBLEMA DE OPTIMIZACIÓN ESTRUCTURAL DE MASA RESTRINGIDA.....	152
	<i>Caso Test X</i>	154
4.8.1	<i>Caso Test Y</i>	158
4.8.2	<i>Análisis de Resultados</i>	161

4.9 ANÁLISIS DEL TRATAMIENTO POBLACIONAL: GENERACIONAL FRENTE A ESTADO-ESTACIONARIO EN OPTIMIZACIÓN MULTI OBJETIVO ESTRUCTURAL.....	163
4.10 COMPARACIÓN DE LOS ALGORITMOS DENSEA FRENTE A DENSEA-NSH EN OPTIMIZACIÓN MULTI OBJETIVO ESTRUCTURAL.....	176
4.10.1 Caso Test X.....	176
4.10.2 Caso Test Y.....	179
4.11 EL OPERADOR RENACIMIENTO AUTOADAPTATIVO EN OPTIMIZACIÓN ESTRUCTURAL. ANÁLISIS FRENTE AL RENACIMIENTO.....	185
4.11.1 Influencia del tamaño de la población y la reducción espacial.....	187
4.11.2 Influencia del parámetro de saturación de la población (RP).....	191
4.11.3 Comparando Renacimiento frente a Renacimiento AutoAdaptativo.....	196
4.11.4 Introducción de soluciones de alta calidad monocriterio en la población inicial multicriterio.....	200
ANEXO. OTRAS TABLAS Y FIGURAS.....	201
BIBLIOGRAFÍA.....	270
CAPÍTULO 5. APLICACIONES A OTROS PROBLEMAS DE OPTIMIZACIÓN MULTI OBJETIVO EN INGENIERÍA INDUSTRIAL.....	274
5.1 INTRODUCCIÓN. OBJETIVOS DEL ESTUDIO.....	274
5.2 PROBLEMA DE OPTIMIZACIÓN MULTI OBJETIVO EN DISEÑO DE SISTEMAS DE SEGURIDAD.....	274
5.2.1 Introducción.....	274
5.2.2 Diseño Óptimo de Sistemas de Seguridad.....	276
5.2.2.1 El Modelo del Sistema.....	276
5.2.2.2 El Método de Optimización.....	278
5.2.3 La Aproximación Evolutiva.....	278
5.2.4 Casos Test.....	280
5.2.4.1 Descripción.....	280
5.2.4.2 Consideraciones Evolutivas.....	283
5.2.5 Análisis de Resultados.....	283
5.2.5.1 Analizando los Algoritmos con Diferentes Probabilidades de Mutación.....	284
5.2.5.2 Analizando los Algoritmos con Probabilidad de Mutación Fija.....	286
5.2.6 Conclusiones.....	288
5.3 PROBLEMA DE OPTIMIZACIÓN MULTI OBJETIVO EN DESPACHO DE CARGAS EN SISTEMAS DE GENERACIÓN DE ENERGÍA ELÉCTRICA.....	289
5.3.1 Despacho de cargas Multiobjetivo.....	289
5.3.2 Esquema de la codificación y funciones objetivo.....	289
5.3.3 Función Objetivo: Coste Económico.....	290
5.3.4 Incorporación de Funciones Objetivo Medioambientales.....	290
5.3.5 Algoritmo Evolutivo.....	291
5.3.6 Aplicación test.....	292
5.3.7 Resultados.....	295
5.3.8 Conclusiones.....	300
5.4 CONCLUSIONES.....	301
BIBLIOGRAFÍA.....	301
CAPÍTULO 6. CONCLUSIONES Y LÍNEAS FUTURAS.....	306
6.1 INTRODUCCIÓN.....	306
6.2 CONCLUSIONES. CONSECUENCIAS.....	307
6.3 LÍNEAS FUTURAS DE INVESTIGACIÓN.....	312
BIBLIOGRAFÍA.....	313

Resumen

En el presente trabajo se ha desarrollado un método muy eficiente para la optimización multicriterio del diseño de pórticos metálicos, minimizando simultáneamente tanto la masa restringida de la estructura como el número de distintos tipos de perfiles que conforman la misma. Ambos requisitos responden a criterios económicos de coste material y constructivo, respectivamente.

La metodología de resolución ha sido en el contexto de los métodos evolutivos multicriterio o multiobjetivo (métodos estocásticos globales inspirados en mecanismos de la evolución natural) que permiten optimizar según varios criterios en conflicto y resolver problemas antes vedados a los métodos clásicos. Son idóneos para el problema abordado, que requiere la optimización global, discreta y multicriterio.

Tras un estudio del estado del arte tanto de los métodos de optimización multicriterio, especialmente los evolutivos, como de la optimización evolutiva estructural, se ha realizado un análisis sobre varios casos test estructurales considerando los algoritmos más eficaces (trece) y distintos aspectos en la optimización (codificación, elitismo, mutación y estrategia poblacional). Este estudio no sólo nos permite obtener directrices acerca de una optimización eficaz, sino que también incluye la propuesta de un nuevo algoritmo muy competitivo con respecto a los comparados. Además, en relación con el problema estructural de minimización de la masa restringida se realizan dos aportaciones de mejora: la multiobjetivización y el operador renacimiento autoadaptativo.

Finalmente se resuelven dos problemas de optimización multicriterio de interés en la ingeniería industrial: el diseño de sistemas de seguridad (el sistema de contención por inyección en spray de una central nuclear de generación eléctrica, contemplando la minimización del coste y la minimización de la no-disponibilidad) y el despacho de carga en sistemas de producción de energía eléctrica (contemplando la minimización del coste y la minimización de la emisión de contaminantes atmosféricos) a los que se aplican algunas de las aportaciones previas.

Capítulo 1. Introducción

*Concede a tu espíritu el hábito de la duda,
y a tu corazón, el de la tolerancia.*

GEORGCHRISTOPHLICHTENBERG (1742-1799)

1.1 Antecedentes

Este trabajo se desarrolla en el contexto de la investigación de la división de Computación Evolutiva y Aplicaciones (CEANI) del Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI) de la Universidad de Las Palmas de Gran Canaria (ULPGC). El centro, dirigido por el profesor Gabriel Winter, es partícipe desde mediados de los años 90 en los foros internacionales de investigación de computación evolutiva y más concretamente en su aplicación a problemas de interés industrial. Se puede citar la organización del congreso EUROGEN 1995 en la ULPGC [1.12], congreso pionero de una serie de periodicidad bianual que continúa en la actualidad (EUROGEN 2005 en Munich, Alemania); así como la co-coordinación de la red europea de investigación INGenet [1.11][1.14], financiada por las Direcciones Generales XII y XIII de la Unión Europea entre 1997 y 2002. Esta red se constituye con la finalidad de establecer casos test de diseño óptimo en aplicaciones industriales en los más diversos campos de la ingeniería: aeronáutica, de transporte o eléctrica; y el desarrollo eficaz de estrategias de resolución óptima mediante algoritmos evolutivos.

A finales de la década de 1980 y con mayor profusión ya en la década de los años 90, especialmente a partir de su último lustro, se ha considerado en la resolución óptima de aplicaciones de los más diversos campos de las ciencias e ingeniería la denominada computación evolutiva ('evolutionary computation'). Gran parte de las aplicaciones pioneras realizadas en esos años, son recogidas entre otros, en Goldberg (1989) [1.6] y Davis (1991) [1.1].

En el año 1986 se realiza el primer trabajo de aplicación de algoritmos genéticos a la optimización estructural (Goldberg y Samtani) [1.7], concretamente aplicado a la optimización de celosías de mínimo peso mediante variables continuas. A principios de los años 90 se desarrollan trabajos pioneros en esta área de optimización estructural mediante computación evolutiva por parte del profesor de la Escuela Técnica Superior de Ingenieros Industriales, Miguel Galante Guille' [1.4][1.5] en la ULPGC. La presente tesis se constituye como continuación de una de sus propuestas de investigación futura recogidas en (1996) [1.3]: la optimización multiobjetivo de estructuras de barras considerando el mínimo peso y el número de distintos tipos de perfiles, mediante métodos innovadores de optimización evolutivos multicriterio basados en el concepto de la no-dominancia de Pareto.

1.2 Optimización Multiobjetivo de Estructuras planas con nudos rígidos y perfiles metálicos

El objeto del presente trabajo es el estudio de un método eficaz que permita resolver la optimización multicriterio del diseño de pórticos metálicos minimizando simultáneamente tanto el peso restringido de la estructura como el número de distintos perfiles que conforman la misma, respondiendo ambos requisitos a criterios económicos de coste material y constructivo, respectivamente.

La optimización estructural consistente en la minimización del peso en pórticos metálicos exige la satisfacción de ciertas características como son: una optimización global que evite el caer en mínimos locales, frecuentes en el espacio de búsqueda del problema; se requiere además un método que permita la optimización con variables discretas, dado que al considerar el problema mediante perfiles reales una continuidad de las variables no está garantizada.

Asimismo, al abordar una optimización multiobjetivo, considerando una segunda función objetivo que será el número de distintos tipos de perfiles de la estructura considerada, se requiere un método eficaz capaz de abordar una optimización multicriterio.

Estas tres características de la optimización: globalidad, discreta y multicriterio, hacen de los algoritmos evolutivos un método de optimización que cumple con todas ellas, siendo métodos estocásticos globales (dado que trabajan con una población de individuos en lugar de con una única solución candidata) inspirados en mecanismos de la evolución natural de los seres vivos [1.2][1.6][1.9]. Admiten trabajar con variables discretas, además de haberse desarrollado recientemente eficaces métodos multicriterio de optimización.

Estos métodos evolutivos multiobjetivo han sido aplicados en la última década a múltiples campos de ingenierías (medioambiental, naval, hidráulica, eléctrica, electrónica, de telecomunicaciones y redes, robótica y de control, estructural, mecánica, civil y de construcción, de transporte o aeronáutica), ciencias (geografía, química, física, medicina, ecología, informática) y aplicaciones industriales (diseño y manufactura, gestión, organización) resolviendo múltiples problemas de optimización antes vedados a los métodos clásicos.

La obtención de un método que permita el cálculo y diseño óptimo estructural mediante consideraciones reales, incorporando los requerimientos de la normativa española y aportando soluciones reales con perfiles normalizados disponibles en el mercado que considere tanto aspectos de coste material como constructivo, como es el problema abordado en este trabajo, adquiere mayor importancia en el caso de la localización de carácter aislado del archipiélago canario y su dependencia del suministro marítimo en perfiles de construcción. Consideramos la optimización del caso de aplicación bajo el supuesto real que sucede frecuentemente, de disponer únicamente de una parte de los perfiles reales. Esta situación puede ocurrir con frecuencia en la Comunidad Autónoma de Canarias, dado su carácter archipelágico y la limitación de comunicación y transporte que ello implica. El traslado de un tipo específico de perfil, por vía marítima desde la península, puede resultar económicamente más costoso que la elección óptima entre los existentes en stock en el mercado. Más aún, la implicación en términos de tiempo de transporte del mismo y la consiguiente espera en la obra puede ser un factor perjudicial de coste añadido no despreciable a tener en cuenta. Los algoritmos evolutivos multicriterio, permiten efectuar la optimización incluso en el supuesto de que tengamos un restringido abanico de perfiles a nuestra disposición.

1.3 Metodología de Optimización propuesta

La optimización estructural respondiendo a criterios de peso mínimo restringido que permita la viabilidad de la estructura con unos límites tensionales, de desplazamiento y atendiendo a criterios de pandeo, es un problema que se caracteriza por poseer multitud de mínimos locales [1.8]. Un método de optimización global es conveniente por tanto para obtener resultados óptimos. Además, de cara a una aplicación directa real, el uso de perfiles normalizados disponibles en el mercado es imperativo, debiéndose la optimización realizarse bajo este criterio de dominio discreto. Se requiere para abordar el problema un método global y que admita dominios discretos.

Los métodos o algoritmos evolutivos son métodos de optimización global capaces de encontrar soluciones óptimas entre una gran diversidad de soluciones candidatas, escapando fácilmente de la atracción de óptimos locales. De este modo, la computación evolutiva (evolutionary computation) destaca sobremedida en la actualidad como una potente herramienta de optimización en los más diversos campos de las ciencias y la ingeniería. Son métodos estocásticos de optimización global, inspirados en mecanismos de la evolución natural de los seres vivos. Los recursos y el rigor de las matemáticas han dotado progresivamente a los algoritmos genéticos de demandados fundamentos teóricos. Se puede afirmar que es en la última década cuando se consigue obtener con rigor los primeros fundamentos matemáticos. Nix y Vose en 1992 [1.10], establecieron un modelo que da una representación exacta de los estados poblacionales esperados en el tiempo del Algoritmo Genético Simple. Este modelo es una cadena de Markov, con estados definidos por el número de duplicados de cada individuo en la población. Estos métodos admiten además, a diferencia de los métodos clásicos, dominios discretos tal como requiere el problema abordado.

En los últimos años, especialmente desde mediados de los años 1990 han sido desarrollados métodos de optimización evolutivos multiobjetivo o multicriterio, permitiendo el obtener soluciones optimizadas según más de un criterio u objetivo, aún en el caso de estar estos objetivos en conflicto.

En este contexto, y partiendo del estado del arte en la materia, se ha realizado un estudio de un método eficaz que permita la optimización multicriterio del diseño de pórticos metálicos minimizando tanto el peso restringido de la estructura, como el número de distintos perfiles que conforman la misma simultáneamente, respondiendo ambos requisitos a criterios económicos tanto de coste material, como constructivo respectivamente.

Ha sido necesaria la programación de un algoritmo de cálculo de pórticos metálicos que partiendo de los datos necesarios para definir la estructura nos proporciona las tensiones y desplazamientos de la misma permitiendo verificar su viabilidad.

Se han analizado y programado igualmente aquellos algoritmos o métodos multiobjetivo evolutivos más destacados de los seleccionados a partir del estado del arte en función de claras ventajas y mayores eficiencias, proponiendo posibles mejoras en la adaptación al problema a resolver con el objetivo de obtener los resultados óptimos en el menor tiempo posible.

El módulo de cálculo se ha acoplado a los distintos algoritmos evolutivos multiobjetivo seleccionados entre los más eficientes y variantes que podrían proponerse, que nos han permitido obtener una frontera de Pareto final con las estructuras de mínimo peso restringido para cada uno de los diferentes números máximos de perfiles distintos; proporcionándose al diseñador un abanico de elección según sus preferencias dentro del conjunto de soluciones óptimas obtenidas.

Mediante estudios estadísticos se han comparado los resultados obtenidos con los diferentes métodos y variantes en distintas estructuras test o de verificación y validación de resultados. Ello se ha efectuado a través de métricas que permiten evaluar no sólo los írentes finales obtenidos sino también la evolución de la convergencia de las poblaciones de soluciones de los distintos métodos durante todo el proceso evolutivo.

Se pretende por tanto, un estudio del estado del arte de los métodos evolutivos multicriterio, la selección de los más eficaces, y la aplicación eficaz y eficiente adaptada a medida en el área de cálculo de pórticos y estructuras metálicas, contemplando la implementación computacional necesaria para el desarrollo expuesto.

1.4 Organización de la tesis

La tesis doctoral se estructura en seis capítulos, cada uno de ellos con su propia bibliografía al final de los mismos, recogándose una bibliografía total unificada tras el sexto capítulo.

El capítulo primero aborda los antecedentes del trabajo, presenta una descripción general de los objetivos abordados y la metodología propuesta, resume las contribuciones originales y presenta una descripción de la organización de la obra escrita.

El segundo capítulo trata acerca de los algoritmos evolutivos multicriterio. Inicialmente contextualiza los algoritmos evolutivos, centrándose particularmente en los algoritmos genéticos. Recoge una breve reseña histórica de los mismos, así como sus operadores más importantes. Posteriormente expone una revisión del estado del arte de los algoritmos evolutivos multicriterio describiendo la evolución histórica de estos métodos, así como sus características. Se tratan las métricas empleadas en la evaluación del desempeño de estos algoritmos, abordando finalmente algunos casos test mediante uno de los algoritmos de última generación destacados.

El capítulo tercero describe la optimización multicriterio estructural, recogiendo inicialmente un exhaustivo estado del arte en ingeniería estructural-civil de los algoritmos evolutivos y de optimización multicriterio. Expone las características del problema estructural abordado, enfatizando aquellos aspectos relevantes de cara a la optimización con algoritmos evolutivos. Finalmente se muestran los dos casos test estructurales considerados, incluyendo la detallada descripción de las soluciones óptimas obtenidas.

El cuarto capítulo contiene las contribuciones originales aportadas en el presente trabajo. En el ámbito de los algoritmos evolutivos multicriterio se realiza una adaptación eficiente de uno de los operadores encargados del mantenimiento de la diversidad, concretamente en el algoritmo SPEA2 [1.13] para problemas de optimización bicriterio, como el estructural afrontado. El problema estructural multicriterio abordado es resuelto de forma satisfactoria, habiéndose realizado un exhaustivo estudio con trece algoritmos evolutivos multiobjetivo diferentes, y distintos y variados parámetros y estrategias con el fin de obtener una configuración eficiente en su resolución. Se ha aportado el diseño de nuevos algoritmos (DENSEA y DENSEA-NSH) que mejoran los resultados de los anteriormente existentes para este problema estructural considerado. Los análisis realizados desglosados según diferentes aspectos de la optimización (codificación, probabilidad de mutación, tipo de algoritmo, tratamiento poblacional, etc.) quedan expuestos en este capítulo cuarto. Además se realiza una exhaustiva comparación del operador renacimiento propuesto en Galante [1.3], frente a un

operador mejorado que se ha denominado renacimiento autoadaptativo en el caso de estructuras de nudos rígidos.

En el capítulo quinto se tratan dos aplicaciones industriales objeto de investigación en el CEANI tratando de aplicar algunas de las aportaciones del cuarto capítulo a otros problemas de diseño óptimo multicriterio. Concretamente, el diseño óptimo de sistemas de seguridad comprendiendo la minimización del coste simultáneamente con la minimización de la no-disponibilidad; y la optimización multiobjetivo del despacho de cargas en sistemas de generación de energía eléctrica, considerando la minimización del coste simultáneamente con la minimización de las emisiones contaminantes.

En el sexto capítulo se presentan las conclusiones derivadas de la aplicación práctica de los métodos propuestos y se especifican líneas abiertas para futuros estudios.

Bibliografía

- [1.1] L. Davis, "Handbook of Genetic Algorithms", VanNostrand Reinhold, New York, (1991).
- [1.2] K.A. De Jong, "An Analysis of the Behaviour of a Class of Genetic Adaptive Systems", *PhD Thesis dissertation*. (1975) Univ. of Michigan, Ann Arbor, MI.
- [1.3] M. Galante. 'Genetic Algorithms as an approach to optimise real-world trusses' *International Journal for Numerical Methods in Engineering*, J. Wiley, vol 39, pp 361-382 (1996).
- [1.4] M. Galante, "Un algoritmo genético simple para la optimización de estructuras planas articuladas", *Métodos numéricos para cálculo y diseño en ingeniería*, 9-2 (1993) pp 179-200.
- [1.5] M. Galante, M. Cerrolaza, "Un algoritmo genético para optimizar estructuras esbeltas reales", // *Congreso de Métodos Numéricos en Ingeniería*, La Coruña, (1993).
- [1.6] D.E. Goldberg, *Genetic Algorithms in Search, optimization and Machine Learning* Reading, Massachusetts, (1989), Addison-Wesley.
- [1.7] D.E. Goldberg, M.P. Samtani, "Engineering Optimization vía genetic algorithm". *Proceedings Ninth Conference on Electronic Computation*, ASCE, New York, NY, (1986), pp. 471-482.
- [1.8] S. Hernández, "Del diseño convencional al diseño óptimo. Posibilidades y variantes. Parte I. Análisis de sensibilidad y optimización local y global". *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, Yo\9-\, (1993) pp. 91-110.
- [1.9] H.J. HoUand, *Adaptation in Natural and Artificial Systems*". University of Michigan, 1975.
- [1.10] A.E. Nix, M. D. Vose, "Modelling genetic algorithms with Markov chains" *Annals of Mathematics and Artificial Intelligence* 5, (1992), pp. 79-88
- [1.11] J. Periaux, B. Galván, B. González, A. Blesa, JI Jiménez, G. Winter, "Red Europea INGEnet y Base de Datos de Casos Test en Aplicaciones de interés Industrial", *Actas del II Congreso español sobre Metaheurísticas, Algoritmos evolutivos y bioinspirados*, (2003) Gijón, pp. 62-69.
- [1.12] G. Winter, J. Periaux, M. Galán, P. Cuesta (Eds.) *Genetic Algorithms in Engineering and Computer Science*, John Wiley & Sons, (1995).
- [1.13] E. Zitzler, M. Laumanns, L. Thiele "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization", *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, John Wiley & Sons. (2001).
- [1.14] <http://ceani.ulpgc.es/ingenetcd> (consultado en fecha de Febrero 2005)

Capítulo 2. Algoritmos Evolutivos Multicriterio.

2.1 Algoritmos Evolutivos

*Ninguna nube inútil,
Ni la fuga de un pájaro,
Estremece tu ardiente
resplandor azulado.*

*Así sobre la tierra
Cantas y ríes, cielo.
Como un impetuoso
Y sagrado aleteo.*

De "Primeras Poesías "
LUIS CERNUDA (1902-1963)

2.1.1 Introducción. Mirando hacia nuestros orígenes: Sabia Naturaleza

La selva amazónica contiene una cantidad enorme de variedad biológica: miles de especies vegetales y animales. Pero no tenemos que andar muy lejos para que la naturaleza nos demuestre todo su poder: Un litro de agua de mar contiene en promedio cientos de miles de protozoos y miles de organismos constituyentes del plancton. Ambos ejemplos pueden servir de muestra testimonial de cómo a través de muchos millones de años, la evolución natural conduce a especializaciones y variedades que van desde el más simple organismo unicelular, hasta el más complejo ente pluricelular, siendo capaz de conquistar y adaptarse a los más variados hábitats: desde las profundidades del mar, las cimas de las montañas o las grietas volcánicas.

Los más diversos fenómenos estudiados por el hombre e instrumentalizados a su propio servicio, como la electricidad o el magnetismo, están presentes en la misma. El vuelo de las aves migratorias o el nado de las tortugas marinas, capaces de orientarse y recorrer distancias de miles de kilómetros, se basa en parte en un sensor magnético situado en su cerebro a modo de brújula. Un pez africano es capaz de poseer visión eléctrica, creando un campo eléctrico que es distorsionado cuando una presa entra en él, capturándola. La electricidad es empleada en el mundo acuático como medio de defensa, ataque, orientación y comunicación entre individuos. Los propios impulsos nerviosos con los que el cerebro coordina nuestro cuerpo y da órdenes son eléctricos.

El vuelo de las aves desde los tiempos de la antigüedad, no ha dejado de inspirar a la humanidad en sus ansias de volar. Así, el vuelo de una libélula es un símil biológico del principio empleado por los helicópteros.

El hombre, en su empeño por imitar la naturaleza, trata de reproducir la base de la energía del sol, la fusión nuclear, confinando en un núcleo magnéticamente plasma a temperaturas elevadísimas: Una fuente de energía puntera, la fusión nuclear.

Incluso la estructura del césped, tan fácilmente recuperable tras ser pisoteada ininidad de veces, es objeto de estudio por parte de ingenieros para tratar de imitar su comportamiento en estructuras aprovechables para el hombre.

Todo ello, no constituye sino una pequeña muestra a modo de ejemplo, de la sabiduría de la Naturaleza.

En este productivo afán de imitar a la Naturaleza, surge un modo de resolver y optimizar problemas basado en la evolución natural que ha conducido a resultados no obtenidos previamente hasta el momento por cualquier otro método conocido con anterioridad y que se configura actualmente como un instrumento de gran potencia para las ciencias y las ingenierías: la computación evolutiva.

2.1.2 La Computación Evolutiva: Un modo Natural de resolver y optimizar problemas

La posibilidad que ha dado a la humanidad el uso de los ordenadores, mediante los cuales efectuar un elevado número de operaciones en un corto periodo de tiempo es viable, ha transformado nuestro mundo. No sólo penetrando en la mayoría de los sectores de la sociedad: finanzas, logística, sanidad, pequeño y mediano comercio, etc; sino incluso transformándolo también científica y tecnológicamente. Así, como ejemplo de lo más reciente, podemos citar la titánica tarea de desciframiento del genoma humano. Han permitido métodos de análisis y simulación como los elementos finitos, empleados en la modelización de automóviles o del sector aeronáutico.

Entre estos métodos surgidos al amparo del poder tecnológico que ha generado el ordenador, podemos englobar la computación evolutiva. Ésta se basa en los principios de la evolución natural: la supervivencia de los mejor adaptados al entorno, que con el transcurso del tiempo, conduce a la predominancia de los mejores individuos.. Análogamente, la posibilidad de resolver un problema un número elevado de veces mediante los ordenadores, permite efectuar mediante la adecuada codificación del mismo, una evolución que conduzca a soluciones óptimas dentro de los requisitos adecuadamente definidos. Es la progresión del conjunto de soluciones candidatas inicialmente, mediante una analogía con el proceso evolutivo, la que va transformando esa población y conduciéndola hacia una calidad elevada de soluciones. Éstas pueden ser optimizadas desde el punto de vista de varios criterios (multicriterio), siendo el diseñador, ingeniero, o tomador de decisiones, el que finalmente discierna qué solución candidata tomar.

Los algoritmos genéticos son métodos de búsqueda global, adaptativos, iterativos y robustos inspirados en líneas muy generales en el proceso de evolución de la naturaleza.

La evolución es una hipótesis científica que consiste en considerar que unas especies provienen de otras mediante una secuencia de alteraciones o cambios en su mensaje genético o ADN, que se han ido sucediendo a lo largo del tiempo. De los distintos seres vivos producidos por la recombinación natural y la mutación, se forman una variedad de individuos, y en ellos actúa la selección natural, primando las formas más adaptadas al entorno natural en que se encuentran (predominancia o supervivencia de los más aptos). La similitud de este proceso con la secuencia seguida por los algoritmos genéticos se muestra a continuación:

Se codifica la información de un problema en una población de cadenas o arrays (cromosomas) y mediante el intercambio de información entre ellas (cruce) o la variación aleatoria de esa información (mutación), se aplica una evaluación de la solución definida por las cadenas relacionada con el problema tratado, midiéndose así su adaptabilidad al entorno (evaluación de la función objetivo o aptitud). Se efectúa la elección de los individuos en función de su mejor adaptación (selección) y se

genera una nueva población sobre la que se vuelve a aplicar este proceso iterativo, y así progresivamente va mejorando la calidad media de los cromosomas (representativos de soluciones del problema tratado).

La diferencia principal entre los algoritmos genéticos y otros métodos de búsqueda -como métodos de gradiente, búsqueda aleatoria controlada o cristalización simulada entre otros- es que su búsqueda es conducida utilizando la información de una población de estructuras o individuos en lugar de seguir una dirección o considerar una única estructura; la información necesaria que requieren acerca del problema a resolver es la evaluación del mismo mediante la función objetivo o aptitud. Se muestra a continuación el esquema del proceso genérico llevado a cabo por un algoritmo genético:

```
crear la población inicial (P);
evaluar la población inicial (P);
mientras (NO condición de terminación)
{
    P1 = seleccionar probabilísticamente
    P2 = cruzar probabilísticamente(P1);
    P3 = mutar probabilísticamente(P2);
    evaluar(P3);
    P4 = nuevapoblación(P,P3);
    P = P4 ;
    verificar condición de terminación;
}
```

Esquema de proceso genérico de algoritmo genético

2.1.2.1 ¿Dónde se contextualizan ?

Los algoritmos genéticos quedan enmarcados dentro del área de conocimiento denominado inteligencia computacional (Computational Intelligence), la cual a su vez se estructura en las siguientes vertientes [2.2]:

- Sistemas difusos (Fuzzy Systems)
- Redes Neuronales (Neural Networks)
- Algoritmos evolutivos (Evolutionary Algorithms):
 - Programación evolutiva (Evolutionary Programming)
 - Cristalización simulada (Simulated Annealing)
 - Estrategias evolutivas (Evolution Strategies)
 - Algoritmos genéticos (Genetic Algorithms), que engloba a su vez a los sistemas clasificadores (Classifier Systems) y a la programación genética (Genetic Programming).

2.1.2.2 Aplicaciones

Su empleo en amplios campos de la ciencia y la tecnología ha permitido la obtención de soluciones innovadoras. Entre otros, se pueden citar los siguientes: Aeronáutica, Electrónica, Economía,

Investigación Operativa, Redes Neuronales, Reconocimiento y Proceso de Imágenes, Biología, Teoría de la decisión o Ingeniería estructural.

2.1.3 Reseña histórica

Los orígenes de los algoritmos evolutivos van ligados al desarrollo de las computadoras, permitiendo la realización de gran número de operaciones en un limitado periodo temporal de modo antes impensable.

Citando a J. H. Holland [2.35]: 'Las primeras tentativas realizadas para imbricar evolución e informática, a finales del decenio de 1950 y comienzos del siguiente, no llegaron muy allá porque se fundaron en las nociones biológicas de la época y fiaron más en la mutación que en el apareamiento para engendrar nuevas combinaciones génicas. Por estas últimas fechas, H.J. Bremerman, de la Universidad de California en Berkeley, añadió una especie de apareamiento: las características de la descendencia estaban determinadas por un resumen de los correspondientes genes de los padres. El proceso resultaba empero muy restringido, al ser sólo aplicable a las características que pudieran sumarse de forma coherente y significativa'. Eso ocurría en 1958, cuando también Friedberg trabajó sobre la programación automática, empleándose representaciones binarias.

En los inicios de los 60 surge la programación evolutiva en San Diego a través de L. J. Fogel concretamente en 1962, aplicada a máquinas de estado finito y empleando únicamente el operador de mutación. En 1965, Schwefel y Rechenberg (Berlín) plantean las bases de las estrategias evolutivas empleando mutación, y la estrategia operativa fue un único individuo (no aún una población). Es J. H. Holland (Universidad de Michigan, E.E.U.U.) quien se puede considerar el fiandador de los algoritmos genéticos mediante sus tres operadores fundamentales evolutivos: selección, cruce y mutación, a finales de los años 60 y durante los primeros años 70. Su objetivo era desarrollar la teoría y procedimientos necesarios para generar programas y máquinas con capacidad de adaptación a entornos dinámicos. Aglutina dos fundamentos básicos de los algoritmos genéticos: la capacidad de *codificar estructuras* complejas en simples cadenas binarias (*cromosomas*) y la potencia de simples *transformaciones* en estas cadenas (*operadores genéticos*) para *mejorar tales estructuras*. Se relatan a continuación ambos aspectos de modo más detallado orientados a la optimización funcional.

2.1.4 Codificación del problema: el cromosoma

La representación binaria del cromosoma, entendida como la codificación mediante unos y ceros, es la tradicional y la que ha dominado la investigación en algoritmos genéticos. Permite la mayor descomposición de las características del problema (parámetros), al ser el alfabeto de menor cardinalidad y con ello la mejor búsqueda de similitudes en la codificación que permitan una progresiva evolución y mejora. Es la implementada en la optimización estructural desarrollada, siguiendo las directrices de [2.28] entre otras; este aspecto se abordará en mayor detalle en el capítulo tercero.

A modo de ejemplo se describe a continuación cómo se codificaría un problema que resultara definido con cuatro variables en el intervalo (0, 25.6) y de precisión requerida para cada uno de ellos de 0.1. Necesitaríamos 256 puntos para cubrir todo el intervalo, pues: $25.6/0.1 = 256$, por lo que en términos

binarios se requiere una cadena de 8 bits ($2^8=256$) para cada variable. Entonces, 8 bits/variable * 4 variables = 32 bits. Así pues, una solución aleatoria resultaría definida por un cromosoma compuesto por treinta y dos unos y ceros aleatoriamente dispuestos (el genotipo), donde por grupos de ocho genes o bits del mismo se obtendrían los cuatro valores de las variables (el fenotipo) a partir de los cuales se podría evaluar esa solución.

Se define el espacio de búsqueda del algoritmo genético como el número total de puntos posibles a explorar. En el caso del ejemplo anterior, éste sería de $2^{32}= 4294967296 = 4.3 \cdot 10^9$.

Se debe señalar respecto de la codificación binaria las carencias a la hora de abordar problemas multidimensionales con gran número de parámetros a codificar y elevada precisión numérica requerida para los mismos, al traducirse este conjunto de características en una longitud binaria de cromosoma que puede hacerlo inabordable aún para el algoritmo genético (dando lugar a espacios de búsqueda abismales, como puede ser 10^{1000}).

Una posible alternativa a la codificación binaria es la codificación real o en coma flotante. Ésta ha mostrado experimentalmente buenos resultados, a pesar de adolecer de los fundamentos teóricos iniciales que acredita el algoritmo genético tradicional de codificación binaria; de hecho, otros tipos de algoritmos evolutivos, como las estrategias evolutivas [2.2], emplean tradicionalmente codificación en coma flotante.

2.1.5 Operadores básicos

Los tres operadores básicos que llevan a cabo la evolución poblacional del algoritmo genético son la selección, cruce y mutación. Su interacción conjunta se describe a continuación sucintamente:

Consideramos una población progenitora, y habiendo asignado a cada uno de los individuos su función objetivo (medida cuantitativa de la calidad de la solución que representa cada uno), actúa el *operador de selección* para escoger un número determinado de individuos -primando de algún modo a los de mejor adaptación- que constituyan una población filial intermedia.

Sobre esta población filial (cuya función objetivo media se prevé probabilísticamente superior a la de la población progenitora) interviene el *operador de cruce*, intercambiando la información genotípica de los cromosomas (los bits) y produciendo nuevos individuos.

Estos nuevos individuos son modificados de modo aleatorio por el *operador de mutación*, causando pequeñas variaciones en el genotipo.

Se construye de este modo una población descendiente, cuyos individuos, unidos en cierto número según la estrategia seguida, con individuos de la progenitora, constituyen la población sobre la que se realizará la siguiente iteración del ciclo de selección, cruce y mutación hasta el final de la evolución.

Se muestra a continuación una descripción de cada uno de los operadores, mostrando sus posibles variaciones y su operativa en el desarrollo de la población hacia el óptimo buscado, así como una posible manera de implementarlos.

2.1.5.1 La Selección

Se distinguen dos etapas en la actuación del operador selección :

1. • Cómo se asignan las probabilidades individuales de selección?; Debe asignarse a cada individuo una probabilidad de ser escogido para formar parte de la población intermedia, normalmente primando aquellos con mejor función objetivo. Entre los modos de asignación, encontramos :

* *Selección proporcional*: esta selección fue propuesta por Holland (1975). En ella se asignan probabilidades proporcionales a la función objetivo directamente. La probabilidad del individuo j en la generación t será:

$$P_{i,j} = \frac{f_j}{\sum_{i=1}^n f_i} \quad (2.1)$$

para un total de n individuos formando parte de la población.

Se define la presión de selección como la relación entre la función objetivo máxima y la media de las funciones objetivo de los individuos de la población.

Pudiera suceder que en las etapas iniciales, la existencia de unos pocos individuos de mucho mejor función objetivo que el resto, llegara a dominar la población debido a la asignación a los mismos de probabilidades relativamente altas respecto al resto, impidiendo la exploración de otras posibles soluciones aún mejores (presión de selección excesivamente alta); este hecho recibe la denominación de convergencia prematura.

Análogamente, en las etapas finales de evolución, cuando por aplicación de la misma la mayoría de individuos de la población sean bastante similares entre sí, las probabilidades relativas entre ellos pudieran ser muy similares, no primándose debidamente la selección de aquellos que destacan aunque sea levemente sobre el resto (presión de selección excesivamente baja); este hecho se denomina estancamiento.

Para tratar de evitar o menguar estos dos posibles efectos (convergencia prematura y estancamiento), se han propuesto los siguientes modos de asignación citados a continuación.

* *Selección escalada*: La función objetivo sufre una transformación escalada, con el fin de situarla en un intervalo que permita una presión de selección que proporcione una evolución hacia el óptimo de forma adecuada. Este escalado puede variar con el transcurso de las generaciones, disminuyendo o aumentando la presión de selección en función de que esta resulte excesivamente alta o baja respectivamente. Existen distintos tipos de escalado posible (Forrest, 1985), entre los que se pueden citar, el escalado exponencial, el lineal o ventana.

* *Selección en función del orden* : la probabilidad de selección es asignada en función del orden de los individuos de la población, establecido por jerarquía de valores de la función aptitud u objetivo. De forma genérica (Baker 1985), la expresión de selección lineal basada en la ordenación se representa a continuación:

$$P_{i,j} = \frac{1}{X} \left(\frac{f_j - f_{\min}}{f_{\max} - f_{\min}} \right)^{j-1} \quad (2.2)$$

siendo :

X : número de individuos contenidos en una generación

f_{\min} = 2 - f_{\max} , siendo f_{\max} valor de : $1 < f_{\max} < 2$.

La presión de selección permanece constante por consiguiente a lo largo de toda la evolución, al depender únicamente de la ordenación, que en cada iteración es la misma (varía el orden de los individuos entre sí, pero no los valores de la ordenación). Los valores de las probabilidades tampoco son necesarios recalcularlos cada iteración, al ser directamente asociados al orden de cada individuo, a diferencia de las dos estrategias de selección anteriores, donde bastaba la incorporación de un único nuevo individuo para tener que recalcular nuevamente las de toda la población al completo.

Es posible modificar esta selección según la ordenación y hacerla exponencial, logarítmica, etc.

* *Selección uniforme* : a todos los individuos se asigna la misma probabilidad de ser seleccionados independientemente de su función aptitud. La selección -como ganancia de valor medio de funciones objetivo en la población- se realiza a posteriori, tras el cruce y mutación, al mezclar la población progenitora con la filial, denominándose *selección por truncamiento*. Esta es una importación directa de las estrategias evolutivas y su selección ($l+1$) donde los X progenitores y los l hijos compiten entre sí, seleccionándose entre ambos a los A , de mejor función aptitud que serán incluidos en la población descendiente.

2. ¿Cómo se selecciona en función de las probabilidades individuales? ; Una vez asignadas las probabilidades de los individuos, deben escogerse los individuos en función de ellas. Esto se puede realizar de diversos modos, siendo los procedimientos siguientes los habituales en el campo de conocimiento de los algoritmos genéticos :

* *Selección de Ruleta (Roulette Wheel Selection RWS)* : Propuesto por De Jong (1975), este procedimiento requiere tantos números aleatorios como individuos deseamos seleccionar de la población progenitora. Una vez obtenidas las probabilidades de cada individuo, éstas se reasignan de modo acumulativo. Se genera un número aleatorio entre 0 y 1 y el primer individuo con probabilidad mayor al número generado es el escogido. Este proceso se itera hasta completar el total de individuos seleccionados. La varianza de este método es mayor que el de los dos siguientes a continuación expuestos, si bien el valor de esperanza de cada uno de los individuos es igual para todos ellos [2.47].

* *Selección de Resto Estocástico con Reemplazamiento (Stochastic Remainder Selection with Replacement SRSWR)* : Propuesto por Brindle (1981), Booker (1982), este procedimiento origina tantas copias de cada individuo j de la población progenitora como el valor del producto entre el número de individuos total a seleccionar por la probabilidad del individuo j : $n \cdot p(j)$. El resto sobrante en cada multiplicación para cada individuo (parte no entera) se trata como en la selección de ruleta y se efectúan lanzamientos aleatorios para rellenar los individuos restantes en la población intermedia inmediata.

* *Selección Universal Estocástica (Stochastic Universal Selection SUS)* : Propuesto por J. E. Baker (1987), Grefenstette & Baker (1989), este procedimiento requiere únicamente un número aleatorio entre 0 y 1 para generar todos los individuos escogidos para la selección. Si este total es de n , partiendo de ese lanzamiento aleatorio se reparte uniformemente en intervalos de longitud $1/n$ los siguientes elegidos, y se opera del mismo modo que en la RWS.

Por último, se señala un método más de selección, divergente de los dos apartados anteriores descritos: la *selección por torneo* (tournament selection), propuesta por Brindle (1981). Para la inserción de cada

individuo en la población intermedia como seleccionado, se procede del siguiente modo: son escogidos al azar un grupo reducido (población de torneo), normalmente dos, de individuos de la población progenitora, resultando elegido el de mayor función adaptación entre ellos, que pasa a formar parte de la población intermedia. Este proceso se repite hasta completar el número total de individuos a incluir en esta población intermedia. La selección por torneo destaca por ser fácilmente paralelizable.

Esta selección por torneo (de dos individuos) resulta ser equivalente en eficacia como promedio al modo en que asigna las probabilidades individuales la selección lineal por ordenación, tal como pone de manifiesto D. Goldberg en [2.31].

2.1.5.2 El Cruce

Este operador permite el intercambio de información entre los individuos de la población, recombinando los cromosomas y creando de ese modo nuevos individuos. Desde la población de individuos escogida tras la selección, se escogen dos de ellos y se aplica el cruce de acuerdo a la probabilidad de cruce p_c (si el número aleatorio generado es menor o igual que p_c se cruzan ambos y no en caso contrario). Tras él, se efectúa la mutación de estos individuos y pasan a formar parte de la población filial.

El *operador de cruce* [2.34] sirve un propósito doble bajo la perspectiva de ser un intercambiador de información genética. Primero, *testea la correlación de un conjunto de bits o genes con la asignación de una elevada función aptitud*. Ello lo realiza permitiendo a esos bits cruzarse con otros conjuntos de bits y la selección obra según la calidad de esos nuevos individuos. Segundo, *permite materializar la alta aptitud de esos subgrupos de bits en un mismo individuo*, al poderse reunir los subgrupos de bits de alta aptitud provenientes de dos progenitores en un mismo individuo. Ello supone la existencia de la hipótesis de que ciertos grupos de genes en un individuo implican una elevada función aptitud. Los grupos de genes son denominados bloques de construcción. Por tanto, bajo selección y cruce buenos grupos de bloques de construcción se propagan y materializan en un individuo.

Entre las posibilidades de cruce de dos cadenas binarias se destacan :

* *Cruce de un punto* :

Propuesto por J. HoUand, recombina la información codificada binariamente en el cromosoma mediante un punto de cruce entre ambos.

* *Cruce de dos puntos* :

Recombina la información codificada binariamente en el cromosoma mediante dos puntos de cruce entre ambos.

* *Cruce uniforme* :

Propuesto por Syswerda (1989), recombina la información codificada binariamente en el cromosoma mediante tantos posibles puntos de cruce como espacios entre genes existen, determinando los puntos a cruzar mediante una probabilidad normalmente uniforme.

Comparando el modo de actuar de los tipos de cruce de un punto y el uniforme, los más opuestos entre sí, en función de estudios teóricos y experimentales (Frantz, 1972 ; Goldberg & Bridges, 1990; Goldberg et al, 1992) se puede concluir que el cruce de un punto favorece la preservación y propagación de bloques de construcción de corta longitud, mientras que los de mayor longitud son

fácilmente rotos. Sin embargo, el cruce uniforme no tiene esta predisposición contra los bloques de mayor longitud, si bien los de longitud media se ven desfavorecidos. Asumiendo la hipótesis de que los buenos bloques de construcción presentan estrecha relación entre sí (los constituyen genes cercanos del cromosoma), el cruce de un punto da buenos resultados; no obstante, cuando la longitud de los mismos crece, puede aparecer tendencia a converger hacia óptimos locales. La veracidad o no de dicha hipótesis dependerá del problema tratado en cuestión. El cruce uniforme es más disruptivo que el de un punto o el de dos puntos, resultando un instrumento de ayuda en poblaciones de pequeño tamaño para mantener su limitada capacidad de acumular información y su tendencia a mayor homogeneidad.

El cruce se puede realizar mediante la adecuada generación de la máscara de cruce en función del tipo de cruce escogido y combinando ésta con los progenitores para engendrar los descendientes, eficazmente mediante operaciones a nivel de bits. Estas son posibles en el lenguaje de programación C/C++ a un coste relativamente reducido.

2.1.5.3 La Mutación

El *operador mutación* es un instrumento para incrementar la diversidad poblacional, aplicándose tras el cruce a cada uno de los descendientes. Consiste en la inversión de los genes designados a mutar (si éste es un 0 pasa a ser un 1 y viceversa). Esto se traduce en un incremento de diversidad en la población y permite aumentar la exploración en el espacio de búsqueda hacia posibles nuevos entornos. Se recomiendan porcentajes bajos de mutación en general: p_m , y el modo más extendido de aplicación de la misma es la mutación uniforme, donde cada bit tiene la misma probabilidad p_m de mutar que el resto de los que componen el cromosoma, aplicándose el operador a todos y cada uno de ellos.

Siendo un útil instrumento de infundir diversidad poblacional, debe advertirse que tasas de p_m excesivamente elevadas pueden aproximar su evolución a una búsqueda aleatoria intensamente explorativa, disminuyendo la eficacia y eficiencia del algoritmo.

La operación de mutación se puede implementar fácilmente mediante la función binaria XOR entre el cromosoma y una máscara cuyo contenido en 0s equivale a posiciones no mutadas y cuyo contenido en 1s equivale a posiciones mutadas. De este modo generando en la máscara los 1s con probabilidad p_m para cada posición, se puede realizar la misma.

Un modo de generación propuesto con probabilidades determinadas y a un coste relativamente bajo puede ser mediante la sucesiva combinación de cadenas de bits aleatorias a las que se aplican sucesivos ANDs y por tanto, cuyo resultado va reduciendo progresivamente el contenido de 1s en la misma, hasta alcanzar una máscara con la deseada probabilidad de mutación en cada posición. Se muestra a continuación el núcleo de dicha subrutina:

```

    masmut = (unsigned long int *)malloc (niiment*sizeof (unsigned long
int));
    for (j=0; j<nuinindivfil; j++)
    {
        for (k=0 ;k<niiment,•]<;++)
        {
            *(masmut+k) = ~0;
            for (i=0; i<nprob; i++)
            {
                *(masmut+k) &= random32();
            }
            *( (nuevmiembro+j)->fraccromos +k) ^= *(masmut+k);
        }
    }
    free(masmut);

```

Algoritmo de Mutación de la población descendiente

donde:

masmut : máscara de mutación.

nument: número de enteros largos contenidos en el cromosoma.

numindivfil : número de individuos de la población descendiente.

nprob : número dependiente de la probabilidad de mutación.

random32 () : generador de números pseudoaleatorios.

(nuevmiembro+j)->fraccromos +k : puntero dirigido a la fracción k del cromosoma perteneciente al individuo j de la población descendiente (nuevmiembro).

2.1.6 Otros operadores genéticos

Se citan a continuación algunos operadores genéticos interesantes.

Entre ellos, como instrumento de introducción de diversidad en la población se puede citar :

- *Eliminación de duplicados* : La población inicial del algoritmo genético contiene individuos generalmente todos distintos. Además, cada individuo nuevo insertado en la población durante la evolución puede ser contrastado para evitar que sea copia de otro ya existente, en cuyo caso será rechazada su inclusión en la misma. Tiene el inconveniente de su elevado coste computacional en general.
- *Reinicialización* : Este operador, sugerido en [2.30] permite tras la convergencia del algoritmo genético, ejecutar una nueva evolución partiendo de una población aleatoria nueva a la que se inserta el mejor individuo de la evolución finalizada. De ese modo se mantiene a través de ese individuo la información obtenida de la evolución inicial y al mismo tiempo se posee gran diversidad de genes aportados por los nuevos cromosomas. Conjugado con una reducción del

intervalo admisible de los parámetros codificados en el cromosoma (lo que implica una reducción del espacio de búsqueda) fue empleado por Galante en optimización estructural de celosías [2.28], y denominado *renacimiento*.

Además se destaca el operador de *elitismo* ; pretende evitar que durante el trascurso de la evolución, algún individuo óptimo pueda no generar descendencia o ser ésta muy reducida no mejorando al progenitor y así perderse de ese modo su valiosa información genética. Recordemos que el algoritmo genético hace uso de operadores estocásticos y por tanto este hecho, aunque no probable con fi-ecuencia, sí es posible en la práctica. Mediante elitismo el mejor (o varios de los mejores) individuos de la población en una generación son directamente incluidos en la población filial, impidiéndose así su extinción y manteniendo la información evolutiva obtenida. El elitismo permite emplear probabilidades de cruce más altas.

El tamaño de la población genética es también un parámetro de diversidad. Así pues, a mayor tamaño poblacional, aumenta la diversidad contenida en ésta. Sin embargo, se debe señalar que el aumento de ese tamaño, también ralentiza la evolución del algoritmo genético, siendo necesarias mayor número de evaluaciones para alcanzar la convergencia y con ello, mayor tiempo de cálculo ; y viceversa, con su disminución ésta se alcanza más rápidamente, si bien una excesiva disminución puede dar lugar a convergencia de soluciones de calidad mediocre.

2.1.7 Equilibrio Exploración - Explotación

El equilibrio entre la *exploración* y la *explotación* es la tarea principal a conseguir en la resolución de problemas. Una excesiva explotación de buena información obtenida, conlleva el dejar de lado la exploración del espacio de búsqueda en pos de posibles mejores estrategias o soluciones, al ser la población finita.

El comportamiento de la evolución de un algoritmo genético durante el tratamiento de un problema debe ser abordado desde esa óptica, tal como plantea HoUand entre otros (por ejemplo en [2.35]).

El modo en que son definidos sus operadores y los parámetros que los gobiernan, así como la representación genotípica empleada, todos ellos contribuyen al deseado éxito de un algoritmo genético determinado. Entre otros factores intervinientes se pueden citar el tamaño de la población, el tipo de selección, tipo de cruce, probabilidades de cruce y mutación, etc.

Un modo de abordar el equilibrio exploración - explotación en los algoritmos genéticos es a través de dos factores de primordial influencia en su evolución (si no los únicos [2.70]): *la presión de selección* y la *diversidad poblacional*. Ambos están inversamente relacionados : una alta presión de selección implica una rápida pérdida de diversidad en la población, puesto que la búsqueda genética se focaliza intensamente en los mejores miembros de la misma; por el contrario, el mantenimiento de la diversidad poblacional es capaz de contrarrestar los efectos de una excesiva presión de selección. La mayoría de los parámetros que son empleados para ajustar las posibles estrategias de búsqueda genética son en realidad términos indirectos de actuación sobre los dos antes expuestos.

En un espacio de búsqueda particular (para un problema concreto), si la estructura impuesta por los operadores permite una exitosa alineación con algunas características inherentes a la orografía del problema, el algoritmo mostrará un buen rendimiento. Igualmente, una divergencia entre los operadores y la orografía del problema puede conducir a resultados pobres. Este hecho ha sido demostrado para el conjunto de los métodos de búsqueda (Wolpert & Macready, 1995) en su *teorema*

NFL : 'Todos los algoritmos de búsqueda presentan el mismo rendimiento frente al conjunto universal de todas las funciones discretas'. Este resultado también enfatiza la importancia de encontrar un algoritmo de búsqueda que se ajuste adecuadamente a un determinado problema si se pretende obtener un buen rendimiento.

Así pues es importante entender cómo las posibles estrategias afectan a la búsqueda del algoritmo genético y cómo introducen cierto modo de predisposición en la trayectoria favoreciendo la búsqueda con cierto tipo de algoritmo. Examinando el problema (en nuestro caso éste será la optimización estructural) desde distintas estrategias puede ser posible el calibrado del éxito de los resultados.

Un algoritmo genético base genérico se muestra en el cuadro siguiente:

```
/* Inicialización del generador aleatorio MWC de 32 bits */
    inirangen();
/* Cálculo de la transformación inversa del GrayCode */
    caldegray();
/* Cálculo de la probabilidad de selección en función de
ordenación */
    rankprob=(double *)malloc(numind*sizeof(double));
    calcprobsel();
/* Creación de la población inicial, evaluación y ordenación */
    creapobinicial(indiv);
    evalúa(indiv);
    ordena(indiv);
/* Bucle principal donde se desarrollan las sucesivas
generaciones */
    final = FALSO;
    for(i=0; i<generacion && !final; i++)
    {
        selecciona(indiv,tempindiv);
        cruza(tempindiv,nuevindiv);
        muta(nuevindiv);
        evalúa(nuevindiv);
        nuevapob(indiv,nuevindiv);
        ordena(indiv);
        final = saturac(indiv) ;
        salidadatos(indiv,&final,i,generación);
    }
    free(rankprob) ;
```

Algoritmo principal del Algoritmo Genético

Donde las variables son :

rankprob : puntero que almacena las probabilidades de selección de cada individuo en función de la ordenación

numind : número de individuos de la población progenitora

npobin : número de individuos de la población descendiente

indiv : población progenitora

`tempindiv` : población intermedia
`nuevindiv` : población descendiente
`final` : condición de terminación de la evolución
`generación` : número máximo de generaciones de la evolución

Donde las funciones son :

`inirangen()` : inicializador del generador de números pseudoaleatorios de 32 bits implementado
`caldegray()` : realiza el cálculo del puntero global donde se almacenan las correspondencias biunívocas que permiten efectuar la transformación código Gray - código binario.
`calcpobsel()` : calcula la probabilidad de selección de los individuos en función de su ordenación en la población.
`creapobinicial(indiv)` : crea la población inicial del algoritmo genético.
`evalúa(indiv)` : realiza la evaluación de los `numind` cromosomas (soluciones candidatas del problema) de la población `indiv`.
`ordena(indiv)` : ordena los `numind` individuos de la población `indiv`.
`selecciona(indiv, tempindiv)` : efectúa la selección de `npobin` individuos de la población progenitora `indiv` que pasan a formar parte de la población intermedia `tempindiv`.
`cruza(tempindiv, nuevindiv)` : realiza el cruce de los `npobin` individuos de la población intermedia `tempindiv` pasando a formar parte éstos de la población filial `nuevindiv`.
`muta(nuevindiv)` : muta los `npobin` individuos de la población filial `nuevindiv` de acuerdo con la probabilidad de mutación establecida.
`nuevapob(indiv, nuevindiv)` : crea a partir de las poblaciones progenitora `indiv` y filial `nuevindiv` la población descendiente final `indiv`.
`saturac(indiv)` : devuelve el valor verdadero si la población alcanza la condición de saturación fijada y falso en caso contrario.
`salidadatos(indiv, &final, i, generación)` : calcula las estadísticas de la evolución periódicamente, escribiéndolas a ficheros de salida.

2.2 Algoritmos Evolutivos Multicriterio.

La peor decisión es la indecisión.

BENJAMÍN FRANKLIN (1706-1790)

2.2.1 Introducción

En la década de los 90, y a raíz de un breve párrafo en Goldberg [2.30] donde se hacía mención a la posibilidad de *optimización multicriterio mediante algoritmos evolutivos* basándose en el concepto de ordenación no-dominada de Pareto, y más intensamente en los últimos años han surgido numerosos métodos evolutivos multicriterio. Reportes del estado del arte reciente de los mismos pueden ser consultados en Coello (2001)(2002) [2.7][2.10], Deb (2001) [2.14], Roudenko (2004) [2.59], Sánchez Carpena (2002) [2.60], VanVeldhuizen (1998)(2000) [2.66][2.68] o Winter (2002) [2.72]. En la optimización multicriterio existe un conjunto de soluciones óptimas -donde el aumento de uno de los criterios conlleva la disminución del otro- que se denomina frontera de Pareto. Ello ocurre cuando nos encontramos con funciones que están en conflicto.

Por ejemplo, se puede pretender maximizar la eficiencia exergética de una central de ciclo combinado (cociente entre potencia neta frente a flujo exergético de combustible) y minimizar simultáneamente el ratio de coste total de la operación (ratio de coste de inversión de los componentes más ratio de coste de combustible y agua) para una producción eléctrica constante (por ejemplo 100 MW) de la central [2.3]. Este ejemplo supone un compromiso entre el coste total de la planta frente al rendimiento de la misma. A mejor rendimiento, el coste total aumenta y viceversa. Será labor del diseñador o proyectista (tomador de decisiones), la elección de la solución que mejor se adapte a sus necesidades o requerimientos de entre esta frontera.

La importancia que han adquirido los algoritmos evolutivos multicriterio o multiobjetivo, métodos globales de optimización, parte de que estos carecen de las desventajas de que hacen gala los algoritmos multicriterio clásicos (en su mayoría métodos locales), lo que les ha dotado de un rango de aplicación a múltiples problemas reales que estaban vedados a aquellos por sus limitaciones. Entre ellas se pueden citar [2.16]:

La necesaria reiteración del algoritmo para obtener varias soluciones pertenecientes al frente final; no requieren condiciones de continuidad o diferenciabilidad de la función a optimizar; el requerir cierto tipo de información sobre el problema tratado; la dispersión de las soluciones de Pareto localizadas depende de la eficiencia del optimizador monocriterio empleado; sensibilidad a la forma del frente de Pareto (no localización de zonas no-convexas); en caso de incluir incertidumbres o estocacidades los métodos clásicos no son viables y la incapacidad de abordar problemas de dominio discreto.

Reportes del estado del arte de estos algoritmos multicriterio clásicos pueden ser encontrados en Miettinen [2.50][2.51] o Marler y Arora [2.46]. En el caso de que tanto la función objetivo como las restricciones sean lineales, se podrán aplicar métodos multicriterio de optimización lineal, como el método multicriterio simplex (Zeleny, 1976 [2.73]). En caso contrario, se deberán aplicar métodos no lineales.

Estos métodos pueden clasificarse según el momento de participación del tomador de decisiones ('decision maker') en el proceso de elección de la solución:

- Métodos A Priori: el tomador de decisiones debe especificar inicialmente cuáles son sus preferencias o prioridades entre los criterios a optimizar, o bien los objetivos a alcanzar. Éstos no siempre son conocidas. Entre ellos se pueden citar: Método de la función de utilidad (método de criterio global ponderado -también denominados programación por compromiso o métodos del punto utópico-, método de agregación ponderada, método min-max ponderado, criterio exponencial ponderado [2.1], criterio de producto ponderado), ordenación lexicográfica, programación por metas (goal programming [2.6]).
- Métodos Interactivos o Progresivos: el tomador de decisiones interactúa con el método, proporcionándole información durante su evolución a partir de la información que éste va generando. Entre ellos se pueden citar: Método Geoffrion-Dyer-Feinberg [2.29], Método de Tchebycheff [2.65], Método del punto de referencia [2.71], Método GUESS [2.5], Método de satisfacción del intercambio (STOM) [2.52], Búsqueda light beam [2.39] o Método NIMBUS[2.51].
- Métodos A Posteriori: La frontera de Pareto es generada, y es después de esta generación cuando el tomador de decisiones elige la solución que más se adapta a sus necesidades de entre las óptimas. Entre ellos podemos citar: Programación física (physical programming) [2.49], Método de intersección de frontera normal (NBI: normal boundary intersection method) [2.13], Método de restricción normal (NC: normal constraint method) [2.48]. Dentro de estos métodos podemos encontrar también los algoritmos evolutivos multicriterio basados en muestreo de Pareto según el criterio de no-dominancia, además de otros métodos empleados con optimización evolutiva como técnicas de muestreo independiente (se ejecuta múltiples veces un optimizador que cada vez nos da una solución del frente) y técnicas de selección de criterios (distintas subpoblaciones optimizan distintos criterios).

Muchos de estos métodos han sido acoplados como función de selección en algoritmos evolutivos convencionales, siendo recogida una amplia recopilación de sus aplicaciones en VanVeldhuizen y Lamont [2.68].

Otros autores realizan la clasificación de estos métodos en base a otros criterios, como pueden ser J. Hom [2.36], que divide los métodos en agregativos (aquellos que agrupan los objetivos o criterios en un único escalar que constituye la función objetivo), y no agregativos (basados en la no-dominancia de Pareto o no).

Los métodos multicriterio evolutivos pueden resolver aplicaciones reales que estaban vedadas a los métodos clásicos por sus limitaciones. Son numerosas las aplicaciones que han sido desarrolladas mediante ellos, en múltiples vertientes de las ciencias e ingeniería. Puede consultarse un detallado estudio de las mismas en Coello et al. [2.10], comprendiendo aplicaciones en: Ingenierías (medioambiental, naval, hidráulica, eléctrica, electrónica, de telecomunicaciones y redes, robótica y de control, estructural, mecánica, civil y de construcción, de transporte o aeronáutica), ciencias (geografía, química, física, medicina, ecología, informática) y aplicaciones industriales (diseño y manufactura, gestión, organización).

Existe una página web que aglutina las referencias más importantes en el campo de la optimización multicriterio evolutiva mantenida por C.A. Coello Coello:

<http://www.lania.mx/~ccoello/EMOO/EMOObib.htm>. que en su última actualización en Noviembre de 2004, contenía 1884 referencias bibliográficas ordenadas alfabéticamente por autores.

2.2.2 Definiciones

En ingeniería, optimizar un problema desde la óptica de varios criterios u objetivos es una necesidad frecuente. A diferencia de la optimización cara a un único criterio, donde se persigue la obtención del óptimo global o una solución muy cercana al mismo, en la optimización multicriterio sucede con frecuencia que una mejora en uno de los criterios implique un empeoramiento en el otro objetivo. No existe una solución aislada pretendida, sino un conjunto de soluciones donde el favorecimiento de uno de los criterios conlleva el perjuicio del otro. A este conjunto de soluciones óptimas se le denomina superficie o frontera de Pareto. Este nombre procede del economista italiano Vilfredo Pareto (1848-1923) [2.56], que postuló el modo eficiente de asignación de los recursos, que porta su nombre: 'los recursos se asignan eficientemente en el sentido de Pareto cuando no es posible mejorar el bienestar de ninguna persona sin empeorar el de otra'. Con ella, será labor del diseñador o proyectista (tomador de decisiones), la elección de la solución que mejor se adapte a sus necesidades o requerimientos.

En el caso que nos ocupa, es de interés el minimizar el peso restringido de una estructura, directamente relacionado con el coste material, y simultáneamente minimizar el número de distintos tipos de perfiles de la misma, relacionado con el coste constructivo. Así pues, si de la óptima estructura en peso restringido deseamos disminuir el número de distintos perfiles en ella, sólo será posible efectuarlo a costa de aumentar el peso. Se concluye que no existe una solución aislada pretendida, sino un conjunto de soluciones donde el aumento de uno de los criterios conlleva la disminución del otro.

En el caso de optimización multicriterio, para comparar dos soluciones entre sí, no es suficiente con comparar los valores de sus funciones objetivo o aptitud, puesto que tendremos tantos valores de ella como criterios sean necesarios por individuo (así, dos valores definirían una solución en el caso de dos criterios; es el caso estructural que nos ocupa). Un modo de realizarlo es mediante el establecimiento de una jerarquía en función del concepto de dominancia entre soluciones.

Una solución x se dice que domina a otra y (o bien que y está dominada por x , o que x es no dominada) si se cumplen las dos condiciones siguientes:

1. La solución x no es peor que y en todos los criterios. En el caso que nos ocupa de optimización estructural para dos criterios, ambos minimizados, equivale a decir que la solución x es menor o igual que la solución y para ambos criterios tratados (peso restringido y número de distintos tipos de perfiles).
2. La solución x es estrictamente mejor que la solución y en al menos un criterio. En el caso que nos ocupa de optimización estructural para dos criterios, ambos minimizados, equivale a decir que la solución x es menor que la solución y ; al menos en uno de los dos criterios tratados.

Así, en el conjunto de todas las soluciones posibles, las soluciones no dominadas son las que constituyen la frontera de Pareto.

A modo de ejemplo, en la figura 2.1 se muestran las seis soluciones dominadas mediante cruces y mediante puntos las cuatro no dominadas (que serían la frontera de Pareto de los diez puntos mostrados); los ejes muestran los valores de dos criterios ficticios a minimizar:

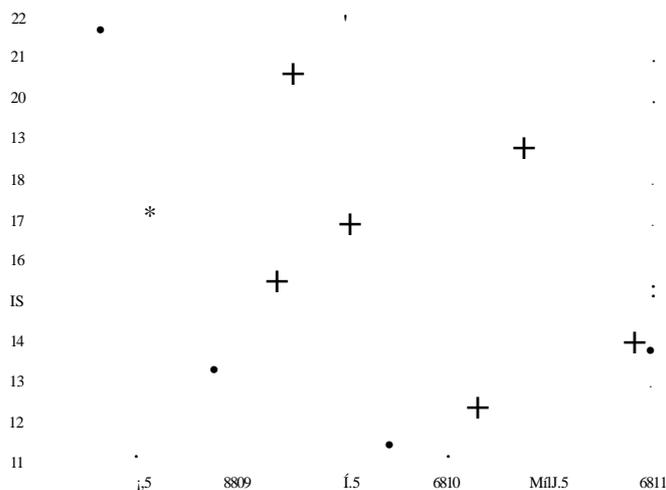


Figura 2.1. Representación de soluciones no-dominadas (puntos) en el espacio funcional.

Los dos requisitos que debe ser capaz de llevar a cabo un algoritmo evolutivo multicriterio para obtener resultados eficaces son: conducir la búsqueda hacia la frontera de Pareto y mantener la diversidad de la población en el entorno de esta frontera con amplitud y uniformidad [2.14]. Se puede apreciar gráficamente la orientación de la búsqueda en un problema de minimización de dos criterios en la figura 2.2.

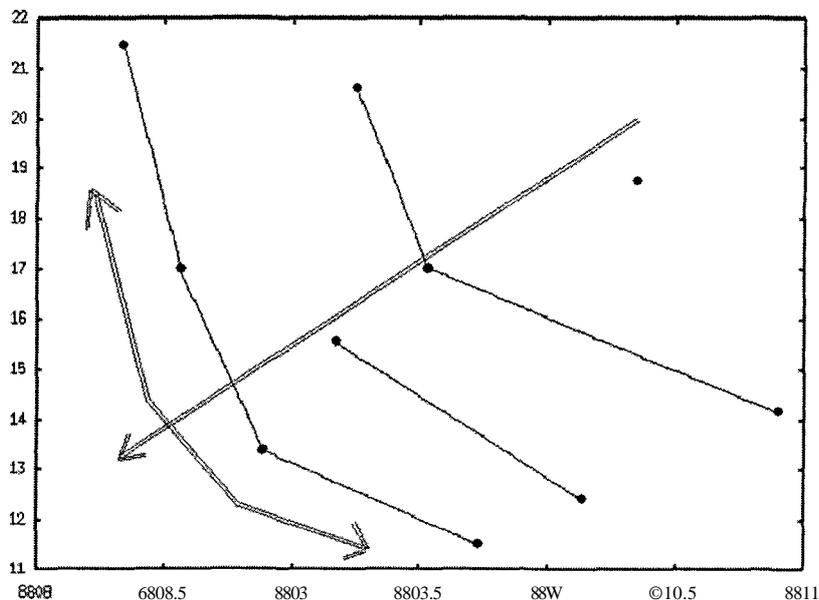


Figura 2.2. Orientación del proceso de búsqueda en un problema de minimización de dos criterios.

Se describe a continuación una breve reseña de la evolución histórica de los algoritmos evolutivos multiobjetivo, incluyendo las ultimas tendencias.

2.2.3 Sumario de Estado del Arte de los algoritmos evolutivos multiobjetivo

2.2.3A Primeros Algoritmos

Se citan a continuación algunos algoritmos evolutivos desarrollados tratando de materializar ambos requerimientos. Inicialmente son enumeradas las primeras aproximaciones, no basadas en el criterio de no-dominancia de Pareto para efectuar la selección:

- **Algoritmo genético por evaluación vectorial** (Vector Evaluated Genetic Algorithm VEGA): debido a Schaffer (1984) [2.61]. Este algoritmo es el primer algoritmo evolutivo que aborda la optimización multicriterio o multiobjetivo. Particiona la población en tantas subpoblaciones como funciones distintas se optimizan, seleccionando cada una de las particiones de acuerdo a su función, posteriormente se crea la población descendiente mezclando el resultado de selección de esas particiones. Se observan tendencias en el frente solución final de converger hacia los extremos del mismo, encontrándose que las soluciones intermedias de la frontera pueden no ser correctamente localizadas.
- **Algoritmo VOES:** Este algoritmo VOES (Vector-Optimized Evolution Strategie) fue propuesto en 1990 por Kursawe [2.43] para la optimización multicriterio. Empleaba una codificación doble, portando cada individuo dos soluciones diferentes, con carácter dominante y recesivo, siendo la evaluación individual producto de una ponderación de ambos caracteres. Se van almacenando las soluciones no-dominadas obtenidas, pero la selección no se realiza considerando este conjunto.

Algoritmo HLGA: Otro algoritmo evolutivo multicriterio pionero es el sugerido por Hajela y Lin [2.33] que fue aplicado a calculo estructural: el algoritmo HLGA. Este algoritmo no se basa en el criterio de nodominancia para realizar la selección, sino que la establece basándose en un método similar a la programación por compromiso (método de pesos ponderado), con la novedad de introducir vectores de pesos independientes y tratando de mantener la diversidad entre ellos, bien mediante técnicas de subespeciación (niching) cuando se asigna un vector de pesos distinto a cada individuo, bien asignando vectores de pesos predefinidos a distintas subpoblaciones.

David Goldberg en 1989 [2.30] propone la idea de realizar la optimización multiobjetivo con algoritmos evolutivos mediante una selección basada en el criterio de no-dominancia de Pareto:

"... One way to achieve equal reproductive potential for all points at the same level is through a nondominated sorting procedure. This procedure is similar to single-criterion ranking selection procedures; however, the population is ranked on the basis of nondomination ..."

("Un modo de alcanzar igual potencial reproductor para todos los puntos a igual nivel es a través de un procedimiento de ordenación de no-dominancia. Este procedimiento es similar a los procedimientos de selección por ordenación monocriterio; sin embargo, la población es ordenada sobre la base de la no-dominancia".)

A partir de esta sugerencia, a mediados de los años 90 surgieron algunos métodos que llevaron su idea a la práctica, entre los que podemos citar los siguientes:

- **Algoritmo genético multiobjetivo de Fonseca y Fleming** (Multi Objective Genetic Algorithm MOGA): debido a Fonseca y Fleming (1993) [2.27]: la ordenación de cada individuo está basada en el número de individuos por los que un individuo es dominado. La distribución de individuos a lo largo de la frontera de Pareto para mantener diversidad es inducida mediante compartición de la aptitud (fitness sharing).

- **Algoritmo genético de ordenación no-dominada** (Non-dominated sorting genetic algorithm NSGA): debido a Srinivas y Deb (1993) [2.63][2.64]: la ordenación de cada individuo está basada en la ordenación del frente al que pertenece según el criterio de no-dominancia de Pareto. La distribución de los individuos a lo largo del mismo se basa también en la compartición de aptitud.

- **Algoritmo genético de Pareto anichado** (Niche Pareto Genetic Algorithm NPGA): debido a Horn y Nafayloitis (1993) [2.36][2.37]; emplea selección por torneo en el frente, siendo el resultado final bastante sensible al tamaño escogido de la población de torneo, además depende también del factor de compartición, siendo por tanto dependiente de dos parámetros.

Estos algoritmos han sido aplicados a gran cantidad de problemas científicos y de la ingeniería mostrando la potencialidad de la computación evolutiva en un amplio número de campos. No obstante, aún en ciertos problemas presentaban dificultades para obtener resultados óptimos, siendo susceptibles de mejora.

2.2.3.2 El elitismo en la optimización evolutiva multiobjetivo

Un operador que ha demostrado aportar una gran mejora en el rendimiento de los algoritmos evolutivos multiobjetivo es el operador elitismo. Así lo muestran estudios realizados, entre ellos se pueden citar los de Purshouse y Fleming (2002) [2.58] o el de Zitzler et al. (2000) [2.75]. El elitismo mantiene el conocimiento adquirido durante la ejecución del algoritmo y lo materializa conservando los individuos con mejor función objetivo en la población o en una población auxiliar. Además, los algoritmos mencionados en el apartado anterior 2.2.3.1 poseen como debilidad adicional la dependencia del factor de compartición de aptitud para distribuir con diversidad los individuos a lo largo de la frontera de no dominados.

La inclusión del operador elitismo en los algoritmos multicriterio ha conllevado a que el operador encargado del mantenimiento de la diversidad sea un operador 'eliminador', y asuma el ir descartando soluciones candidatas de entre las soluciones no-dominadas obtenidas en cada fase del algoritmo, para dejar aquellas más homogéneamente repartidas de acuerdo al criterio empleado por el mismo.

En el esfuerzo por desarrollar algoritmos que carecieran de dependencia paramétrica además de incorporar el operador elitismo, se dio lugar a los tres algoritmos siguientes:

- SPEA (Strength Pareto Evolutionary Algorithm), Zitzler y Thiele (1999) [2.79]: Almacena las

soluciones del mejor frente obtenido en una población auxiliar (elitismo). La ordenación de cada individuo está basada en el factor de fuerza (strength factor). Es calculado para los miembros de la población auxiliar como un cociente cuyo numerador es proporcional al número de individuos de la población que domina y cuyo denominador es el número de individuos de esta población más uno. Un método de clustering (el método de ligadura ponderada) en el espacio de las funciones objetivo es utilizado para preservar diversidad en la población, evitando el uso de cualquier parámetro como la compartición de aptitud.

- **PAES** (Pareto Archived Evolution Strategy), Knowles y Come (1999) [2.42]: Almacena las soluciones del mejor frente buscado en una población auxiliar externa (elitismo). Un método de subespeciación novedoso es introducido en este algoritmo para inducir diversidad. El dominio en función objetivo es subdividido en hipercubos por una malla que determina la densidad de individuos; las zonas con menor densidad son favorecidas en detrimento de las de mayor densidad de puntos. Esta técnica depende sólo del parámetro número de divisiones. El algoritmo PAES fue concebido inicialmente como un método de búsqueda multiobjetivo local (1+1)-PAES, habiendo sido ampliado posteriormente a (A+|x)-PAES. No hace uso de parámetros variables en la búsqueda como las estrategias evolutivas clásicas. Además, la ordenación de cada individuo nuevamente generado es asignada comparando su dominancia o no-dominancia con respecto al archivo o población externa además de con relación a la densidad de la malla a la que pertenece.

- **NSGA-II**; Deb, Agrawal, Pratap y Meyarivan (2000) [2.17][2.21]: Mantiene las soluciones del mejor frente hallado incluyéndolas dentro de la siguiente generación (elitismo). La ordenación de cada individuo está basada en el criterio de no-dominancia correspondiente al NSGA. Además, esta propuesta introduce un algoritmo más rápido para ordenar la población que requiere $O(mN^2)$ cálculos en lugar de $O(mN^3)$ requeridos por el anterior, siendo m el número de objetivos y N el tamaño de la población. Una distancia de crowding es evaluada, considerando el tamaño del mayor paralelogramo que circunda cada individuo sin incluir cualquier otro de la población. Este parámetro mantiene la diversidad en la población y de ese modo los individuos pertenecientes al mismo frente y con mayor distancia crowding le son asignados una mejor evaluación respecto a aquellos con menor distancia crowding, evitando así el uso del factor de compartición de aptitud. Una muestra de la incidencia que está adquiriendo la optimización evolutiva multicriterio en la comunidad científica actualmente se puede inferir a partir de que la referencia [2.21] ha sido catalogada como el más rápido artículo con importancia ('fast breaking paper') en febrero de 2004 por el ISI (Institute for Scientific Information).

En *un* área tan dinámica como la optimización multiobjetivo, nuevas aportaciones fueron propuestas:

- **PESA** (Pareto Envelope-based Selection Algorithm), Come, Knowles y Gates (2000) [2.12]: Almacena las soluciones del mejor frente encontrado en una población auxiliar (elitismo). Como en PAES, el mecanismo de subespeciación se basa en la división de hipercubos del espacio funcional; además, el criterio de selección se apoya en este concepto. En un conjunto de funciones test compitiendo con PAES y SPEA, el algoritmo PESA aparenta tener los mejores resultados globales.

PESA-II, Come, Jerram, Knowles y Oates (2001) [2.11]: El algoritmo PESA-II está basado en el PESA, pero es introducido el concepto de selección basada en la región, en contraposición a la selección por criterios individuales del resto de algoritmos. Se continúa empleando el concepto de división del espacio en hipercubos, pero es el hipercubo el que es seleccionado y no el individuo. Aquellos hipercubos con más número de individuos son penalizados frente a los de población más escasa. Tras la selección del hipercubo, se elige uno de sus miembros aleatoriamente. De ese modo la distribución de probabilidades de selección en la población favorece en mayor medida a los individuos más aislados. En una comparativa de sus autores en seis funciones test frente a PAES, PESA y SPEA, el algoritmo PESA-II parece obtener los mejores resultados conjuntos.

SPEA2; Zitzler, Laumanns y Thiele (2001) [2.76][2.77]: El algoritmo SPEA2 se basa en el SPEA, y está orientado a eliminar las posibles debilidades de su antecesor. La función objetivo de asignación se ha mejorado, teniendo en cuenta para cada individuo tanto el número de individuos que domina como el número de individuos por los que es dominado. Asimismo, una estimación de la densidad de vecinos se añade a esta función, permitiendo una orientación más exacta a lo largo del proceso de búsqueda. El archivo externo elitista adquiere tamaño fijo. Además, se sustituye la técnica de clustering -el operador encargado de mantener la diversidad-, por un operador de truncamiento que evita la posibilidad de eliminar las soluciones extremas de entre las no-dominadas localizadas, preservando así durante la evolución del algoritmo, el rango máximo de las soluciones no-dominadas. En una comparativa de sus autores en cinco funciones test frente a PESA, SPEA y NSGA-II, este último conjuntamente con el SPEA2, obtienen los resultados más competitivos.

NSGA-II + elitismo controlado; Deb y Goel (2001) [2.19]: La introducción de un elitismo controlado en el algoritmo NSGA-II produce un mejor equilibrio entre exploración y explotación. No únicamente regulando la tasa de mutación (parámetro asociado a la exploración), sino regulando también la 'tasa de elitismo' con el elitismo controlado (parámetro asociado a la explotación) puede ser un modo de obtener resultados mejorados. El elitismo controlado limita el máximo número de individuos perteneciente a cada frente por una función geométrica decreciente gobernada por un factor de reducción r , permitiendo introducir mayor diversidad en la población. Este mecanismo se propone para introducir en otros algoritmos evolutivos multiobjetivo. El elitismo controlado pretende por tanto proporcionar un control cuantitativo sobre la presión de selección en el algoritmo. Su actividad se centra en dos factores: suaviza el elitismo del NSGA-II y limita el máximo número de individuos perteneciente a cada frente. El elitismo en el NSGA-II se encuentra presente por partida doble. En una primera etapa se selecciona la población progenitora de la siguiente generación, a partir de los mejores individuos (según el criterio de no-dominancia), pertenecientes a la agregación de las poblaciones progenitora y filial de la generación actual. Una vez creada esta población, se efectúa una selección por tomo para determinar qué individuos serán cruzados y mutados por el criterio de no-dominancia. Esto representa una segunda aplicación de un cierto elitismo

probabilístico, que aumenta la presión de selección. Como el objetivo del elitismo controlado es proporcionar un control sobre esta presión de selección, este segundo paso es eliminado en el algoritmo. Es decir, los individuos que serán cruzados y mutados son seleccionados aleatoriamente de la población progenitora. Además, se impone la existencia de un cierto número de frentes en la población, evitándose la posible rápida convergencia a una población donde sólo se encuentren presentes individuos no-dominados. Se garantiza entonces un cierto número de soluciones pertenecientes a frentes sucesivos al primero no-dominado mediante una función geométrica decreciente gobernada por un factor de reducción r , permitiendo así introducir mayor diversidad en la población. Siendo K el número de frentes a mantener impuesto, N el número total de individuos de la población y r el factor de reducción ($r < 1$) seleccionado, entonces el número de individuos n_i en cada frente i viene dado por la ecuación [2.19]:

$$n_i = N \frac{1 - r}{1 - r^i} \quad (2.3)$$

- s-MOEA: Deb, Mohán, Mishra (2003) [2.21]: Este es un algoritmo evolutivo multicriterio con reemplazamiento poblacional de estado estacionario, cuyo criterio de selección está basado en el concepto de s-dominancia sugerido en Laumanns et al. [2.44]. Es comparado frente al algoritmo NSGAII y a un algoritmo NSGAII al que se ha cambiado su operador distribuidor en el frente (distancia crowding) por el operador truncamiento del SPEA, con unos resultados que indican un buen compromiso del algoritmo s-MOEA entre la velocidad de ejecución del algoritmo y la distribución final de soluciones obtenida sobre un conjunto de varias funciones test clásicas en optimización evolutiva multicriterio.

Dentro de los algoritmos evolutivos multicriterio cabe mencionar también otra aproximación basada en la teoría de juegos no-cooperativa, concretamente en el equilibrio de Nash (1950) [2.53][2.54], en contraste con la teoría de juegos cooperativa que ofrece la aproximación de Pareto. Los algoritmos genéticos basados en Nash (Periaux y Sefrioui) [2.57][2.62] son aplicados a varias funciones test y a optimización en problemas de dinámica de fluidos e ingeniería aeronáutica, permitiendo localizar un punto de equilibrio en cada ejecución.

En este trabajo realizaremos una selección de preferencias a posteriori, empleando algoritmos basados en el concepto de no-dominancia de Pareto, encuadrando nuestro trabajo dentro del campo de los algoritmos evolutivos de última generación.

2.2.4 Métricas Comparativas

En el caso de la optimización multicriterio, a la hora de comparar métodos diferentes, esto se puede realizar considerando los distintos objetivos a alcanzar en toda optimización multiobjetivo: conducir la

búsqueda hacia la frontera de Pareto y mantener la diversidad de la población en el entorno de esta frontera con amplitud y uniformidad. Con dicho objetivo se han propuesto varias métricas de evaluación de eficacia de los algoritmos. Existen métricas que evalúan cada uno de los tres aspectos mencionados con anterioridad de forma independiente: acercamiento al frente, amplitud de cobertura del frente y uniformidad en la distribución de las soluciones; también existen métricas que tratan de aglutinar varios de estos aspectos en un único escalar (como la denominada hipervolumen [2.81]). Dentro de las propuestas más difundidas podemos destacar las de VanVeldhuizen [2.68], y Zitzler [2.81], si bien análisis más exhaustivos acerca de estas métricas pueden ser consultados en Roudenko [2.59] o Deb [2.14]. El adecuado uso de estas métricas para una comparación eficaz de los métodos ha originado publicaciones en el área, entre las que destacamos [2.78], [2.41] y [2.19]. En esta última referencia (Deb y Jain, 2002) se destaca la idoneidad de realizar los estudios comparativos entre distintos algoritmos mediante 'métricas en movimiento' (rurning metrics), es decir, representando los valores que adoptan las métricas a lo largo de la evolución del algoritmo. Esta sugerencia ha sido recogida en el presente trabajo, habiéndose considerado la evolución de dos métricas: una representativa del acercamiento al frente óptimo de Pareto, y otra representativa de la amplitud del frente de soluciones no-dominadas obtenido.

La métrica representativa del acercamiento al frente óptimo escogida, es la métrica denominada métrica MI* de Zitzler [2.81]. Para el cálculo de esta métrica, perteneciente a las métricas escaladas, es preciso conocer a priori el frente óptimo de Pareto, y su expresión es la siguiente:

$$MI^*(U) = \sum_{j=1}^n \max_{u \in U} \{ |u_j - y_j^*| \} \quad (2.4)$$

La segunda métrica es representativa de la amplitud de cobertura del frente de soluciones no-dominadas y en el caso de las funciones test matemáticas resueltas en el siguiente apartado, la métrica escogida es la métrica M3* de Zitzler [2.81], que calcula los rangos de las funciones objetivo en el conjunto de Pareto conocido. La expresión de la métrica es:

$$M3^*(U) = \sum_{i=1}^m \max_{u \in U} \{ |u_i - v_i| \} \quad (2.5)$$

En las expresiones anteriores (ecuaciones 2.4 y 2.5) Y_p es el conjunto de Pareto real y U es el conjunto de Pareto conocido en una generación cualquiera.

En la resolución del problema estructural multicriterio abordado, puesto que la segunda función objetivo (número de distintos tipos de perfiles) es una función discreta, en este caso, al ser un problema de dominio discreto, se ha escogido la amplitud de intervalo de la variable discreta, es decir, el número de distintos tipos de perfiles como métrica indicadora de la amplitud del frente.

2.2.5 Casos Test

Se han considerado tres funciones test clásicas de algoritmos evolutivos multicriterio. Cada una de ellas refleja alguna de las dificultades inherentes a los métodos clásicos que se ven superadas por este tipo de métodos. Todas son funciones biobjetivo donde se pretende minimizar cada uno de los dos criterios. Como primera función se ha tomado una función cuyo frente óptimo es no-convexo. Como segunda función se considera una función con frente de Pareto discontinuo, dividido en cuatro sectores. Finalmente, como tercera función se muestra una función con un frente local que puede entorpecer la localización del global. Cada una de ellas se expone a continuación:

1. MOP2 de Van Veldhuizen. Frente de Pareto no-convexo continuo [2.67].

La expresión matemática de la función es la siguiente:

$$\begin{aligned} \min /1(x) &= 1 - e^{-x_1} \\ \min /2(x) &= 1 - e^{-x_2} \\ -2 < x_j, x_2 < 2 \end{aligned} \quad (2.6)$$

Se puede observar en la figura 2.3 la representación del frente óptimo de Pareto correspondiente a la función descrita.

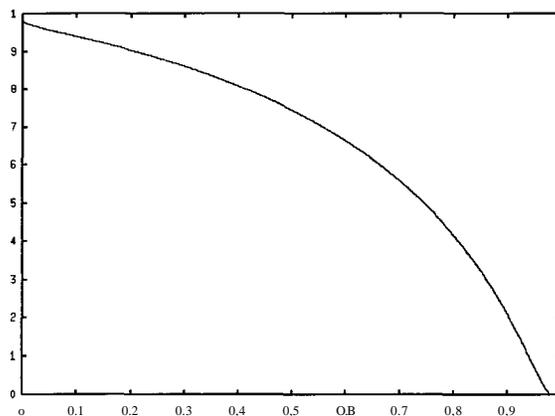


Figura 2,3. Frente de Pareto óptimo para la función MOP2.

2. MCGA2. Frente de Pareto discontinuo [2.8].

La expresión matemática de la función es la siguiente:

$$\begin{aligned}
 \min f(x) &= Y^{\wedge} - 10e^{-2A/X; +X, +1} \\
 \min & \quad 10.8 \\
 -5 < x_j, X_2, X_3 < 5
 \end{aligned}
 \tag{2.7}$$

Se puede observar en la figura 2.4 la representación del frente óptimo de Pareto correspondiente a la función descrita.

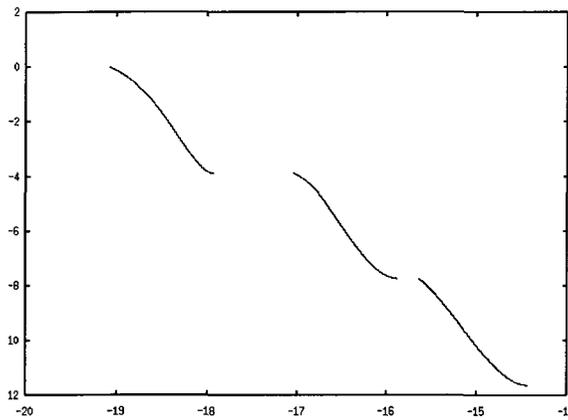


Figura 2.4. Frente de Pareto óptimo para la función MCGA2.

3. MOGA 3 de Deb. Frente de Pareto local y global [2.15].

La expresión matemática de la función es la siguiente:

$$\begin{aligned}
 \min f(x) &= x_i \\
 \min f_2(x) &= \frac{g(x^2)}{x_i} \\
 g(x_2) &= 2 - e^{-0.004x_2} - 0.8e^{-0.6x_2} \\
 0.1 < x_i < 1.1 & ; 0 < x_2 < 1
 \end{aligned}
 \tag{2.8}$$

La función $g(x)$ se representa en la figura 2.5, en el intervalo considerado (0:1), donde se puede observar cómo existe un mínimo local amplio de más fácil localización que el mínimo global situado en la coordenada de valor 0.2.

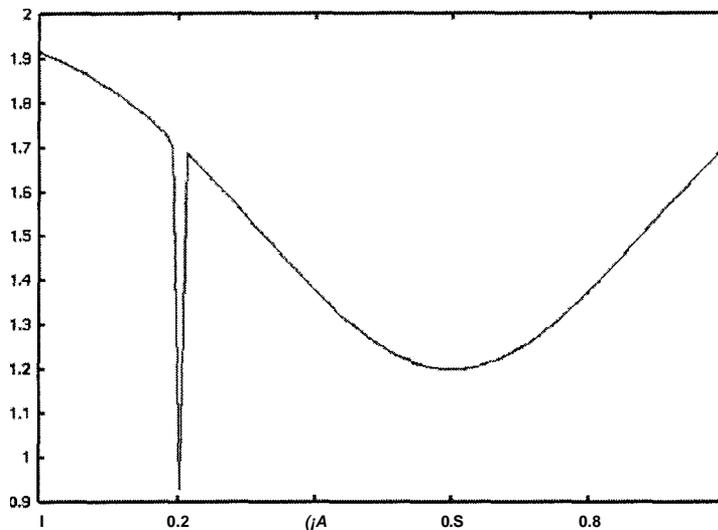


Figura 2.5. Función $g(x)$ integrante de la función test MOGAS.

Se puede observar en la figura 2.6 la representación del frente óptimo de Pareto correspondiente a la función descrita, donde se han representado los dos frentes originados (local y global) como consecuencia de la topología de la función $g(x)$ representada en la figura 2.5.

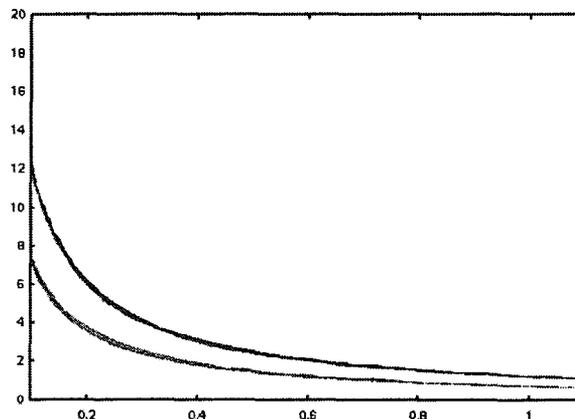


Figura 2.6. Frente de Pareto óptimo para la función MOGAS y el frente local de soluciones no-dominadas originado.

Se muestra a continuación el rendimiento en estas funciones test de un algoritmo evolutivo multicriterio de última generación, el NSGA-II, que conjuntamente con el SPEA2 son los algoritmos más extendidos en uso reciente y los que mejores resultados empíricos han obtenido en ensayos comparativos [2.76] [2.77]. Se ha escogido para cada uno de los ensayos una población de 80 individuos, habiéndose realizado 30 ensayos independientes en cada una de las tres funciones. La longitud de los individuos es de 16 bits por cada una de las variables. El cruce empleado es el cruce uniforme estándar y la probabilidad de mutación utilizada es del 3%. Se muestran los resultados promedios sobre los 30 ensayos de las métricas de Zitzler descritas en el apartado anterior 2.2.4, MI* y M3* en su evolución a lo largo de las generaciones, representado tanto la media (izquierda) como la desviación típica (derecha) de los resultados (en las figuras 2.8, 2.9, 2.11, 2.12, 2.14 y 2.15). Igualmente, para cada una de las funciones se representa la población final (tras 30.000 evaluaciones) en términos del espacio funcional de una de las ejecuciones (en las figuras 2.7, 2.10 y 2.13). Se puede observar cómo el algoritmo es capaz de alcanzar el frente óptimo de Pareto y de distribuir homogéneamente las soluciones en dicho frente en una tánica ejecución para cada uno de los casos test.

Resultados para la primera función test MOP2:

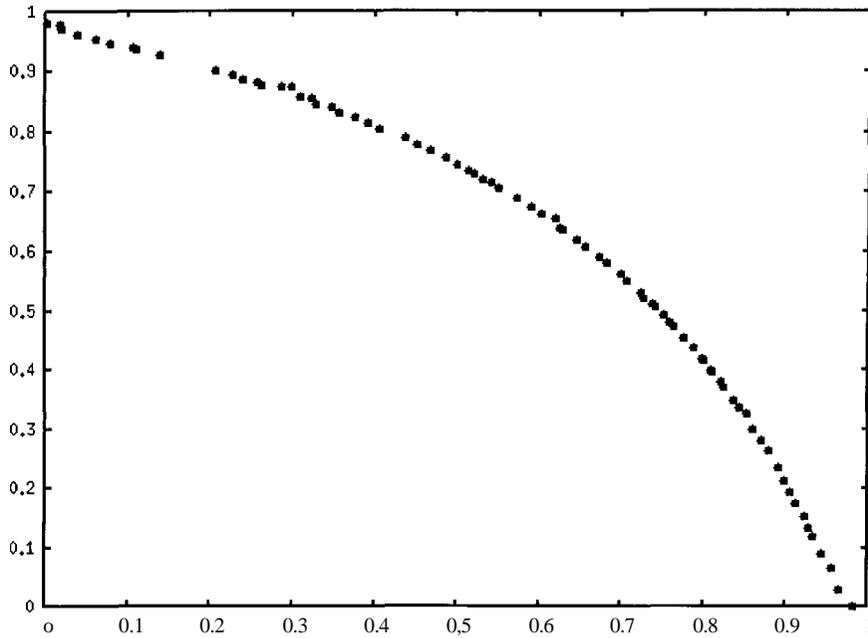


Figura 2.7. Distribución de frente final de una de las ejecuciones en la función MOP2 mediante el algoritmo NSGA-II.

Tabla 2.1. Resultados finales de las métricas sobre 30 ejecuciones para el test MOP2.

	NSGA-II
MI* promedio	0.099407
MI* desv. tipi.	0.019292
M3* promedio	1.388194
M3* desv. tipi.	2.61×10^{-4}

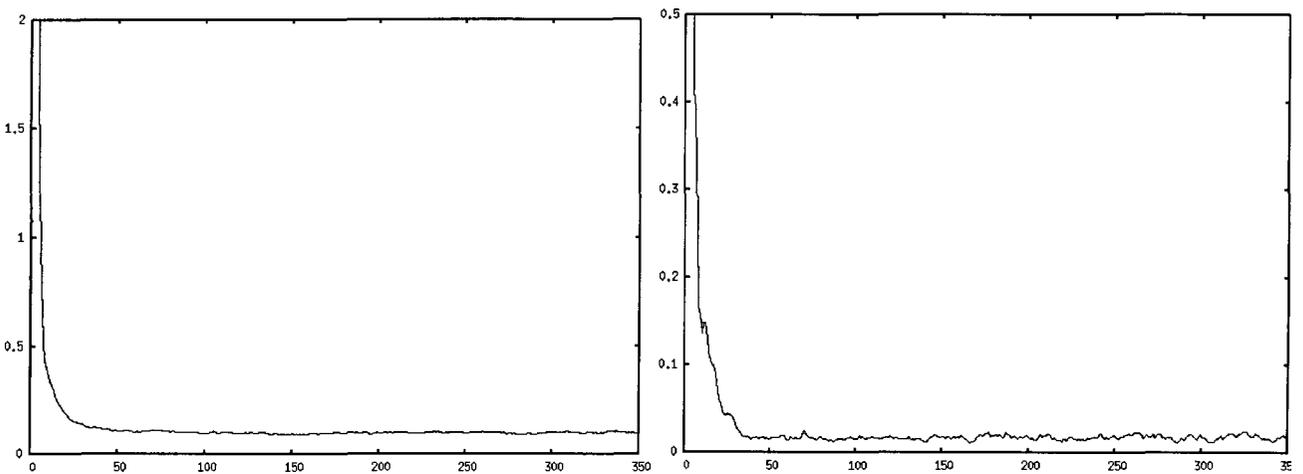


Figura 2.8. Evolución del promedio y desviación típica de la métrica MI* para el test MOP2, de 0 a 350 generaciones.

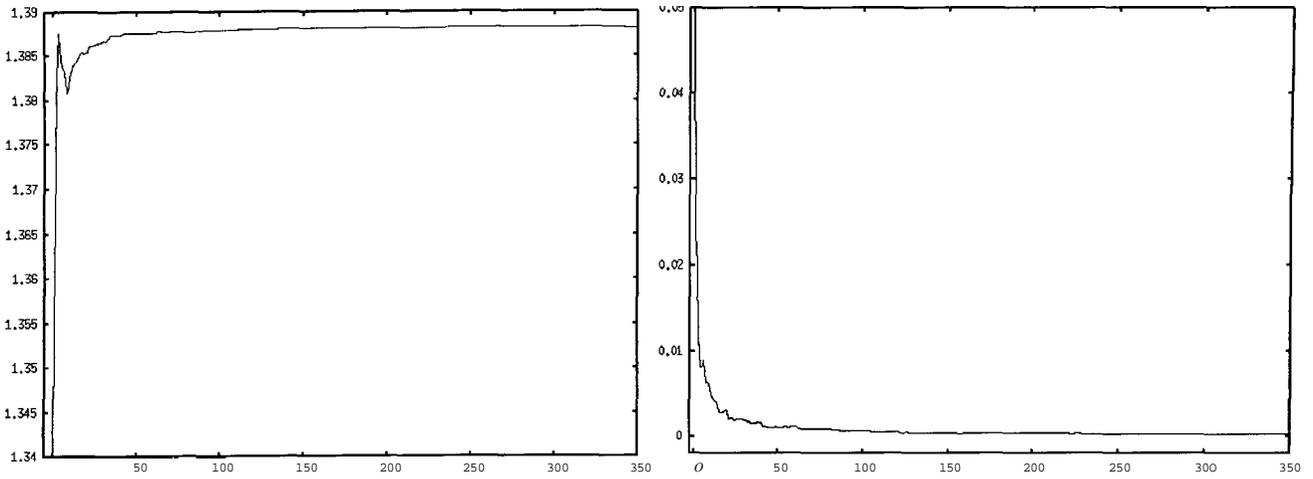


Figura 2.9. Evolución del promedio y desviación típica de la métrica M3* para el test MOP2, de 0 a 350 generaciones.

Resultados para la segunda función test MCGA2:

Tabla 2.2. Resultados finales de las métricas sobre 30 ejecuciones para el test MCGA2.

	NSGA-II
MI* promedio	0.109676
MI* desv. típi.	0.014556
M3* promedio	12.88384
M3* desv. Típi.	0.014442

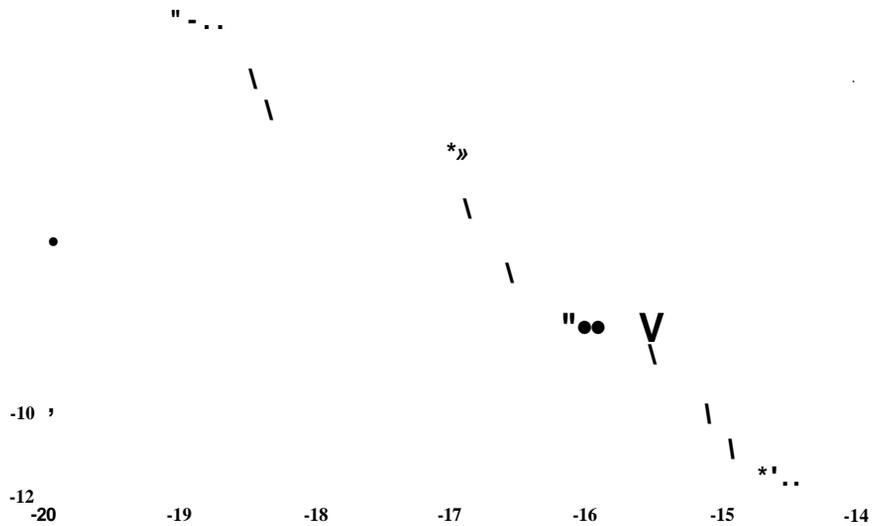


Figura 2.10. Distribución de frente final de una de las ejecuciones en la función MCGA2 con el algoritmo NSGAII.

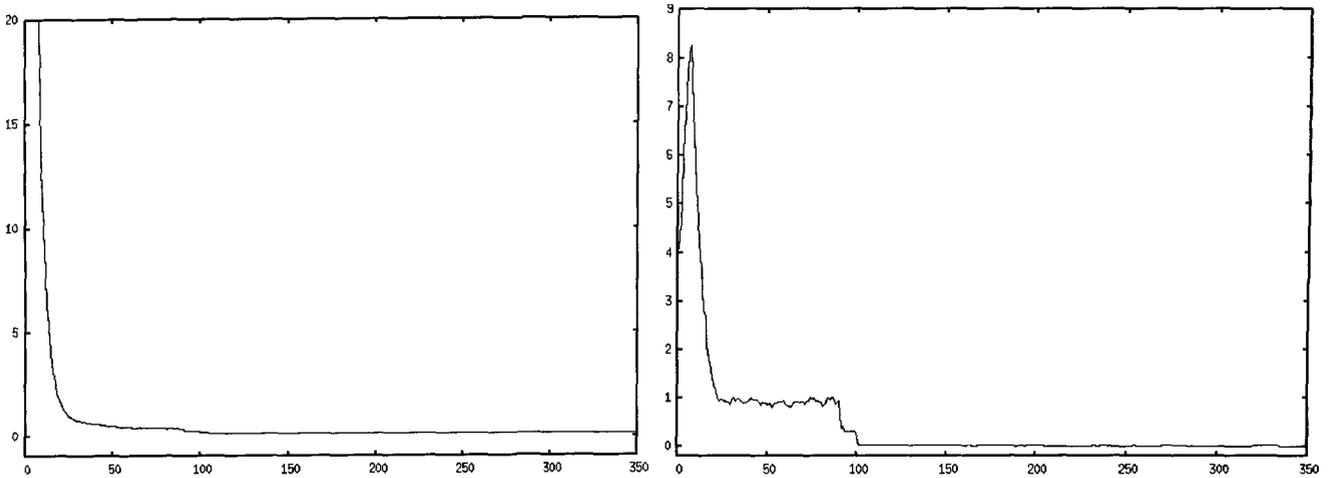


Figura 2.11. Evolución del promedio y desviación típica de la métrica MI* para el test MCGA2, de 0 a 350 generaciones.

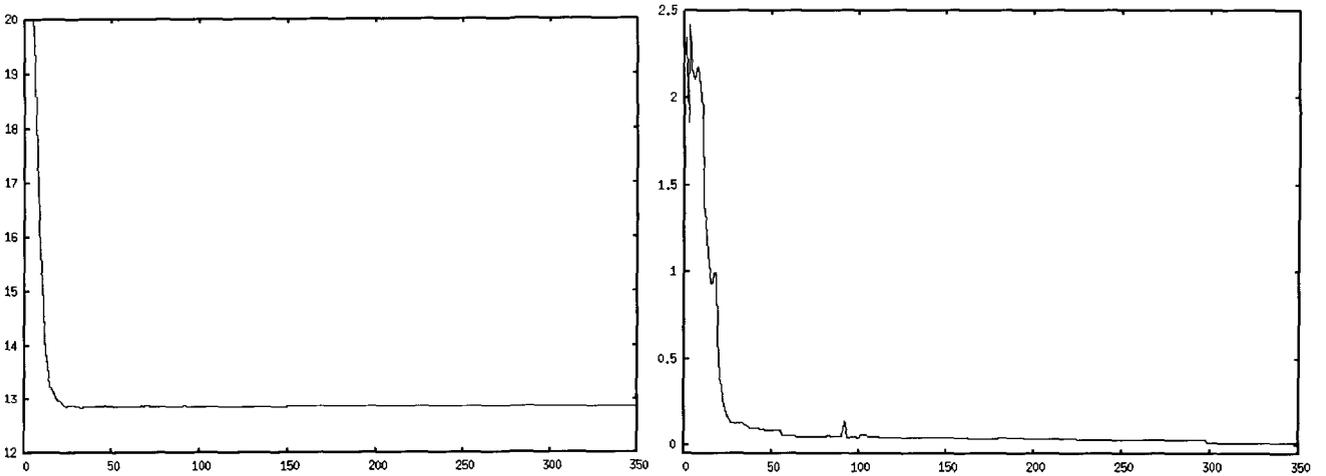


Figura 2.12. Evolución del promedio y desviación típica de la métrica M3* para el test MCGA2, de 0 a 350 generaciones.

Resultados para la tercera función test M0GA3:

Tabla 2.3. Resultados finales de las métricas sobre 30 ejecuciones para el test MOGA3.

	NSGA-II
MI* promedio	0.064784
MI* desv. típi.	0.276932
M3* promedio	6.657222
M3* desv. típi.	0.881066

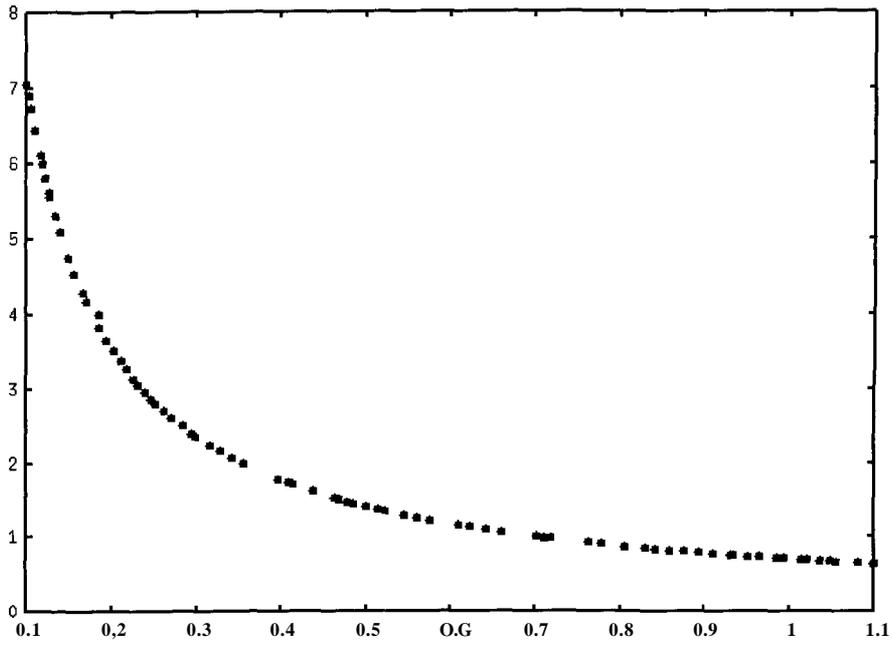


Figura 2.13. Distribución de frente final de una de las ejecuciones en la función MOGAS con el algoritmo NSGA-II.

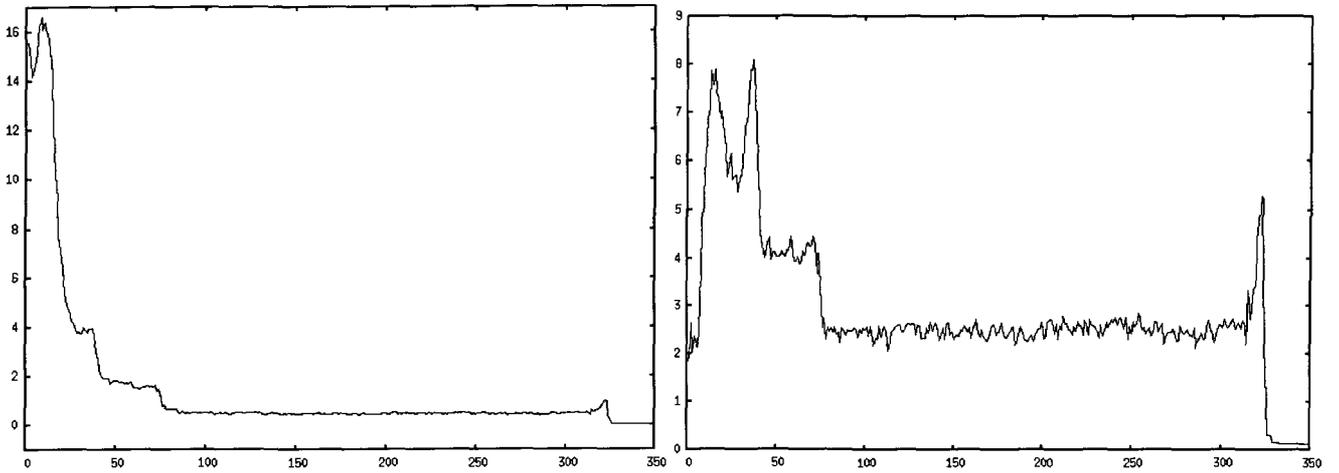


Figura 2.14. Evolución del promedio y desviación típica de la métrica MI^* para el test MOGAS, de 0 a 350 generaciones.

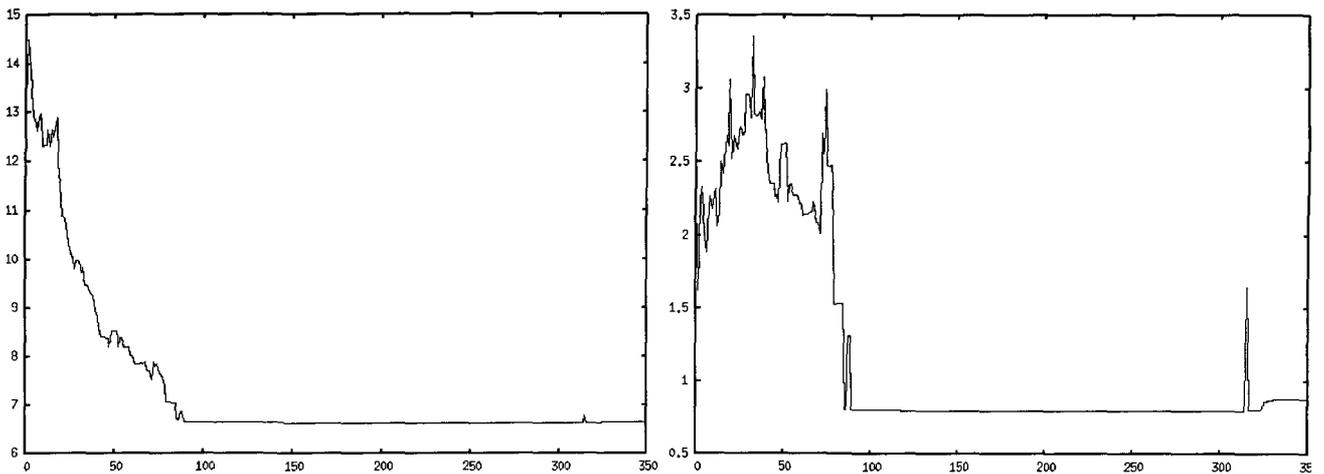


Figura 2.15. Evolución del promedio y desviación típica de la métrica $M3^*$ para el test MOGAS, de 0 a 350 generaciones.

Bibliografía

- [2.1] T.W. Athan, P.Y. Papalambros, "A note on weighted criteria methods for compromise solutions in multi-objective optimization", *Engineering Optimization*, 27 (1996) pp. 155-176.
- [2.2] T. Back. '*Lectures on Evolution Strategies: Algorithms, Theory and Applications*'. Ingenet Lecture Series (1997).
- [2.3] E. Benini, A. Toffolo, A. Lazareto, "Evolutionary Algorithms for multiobjective design optimization of combined-cycle power plants". *Evolutionary Methods for Design, Optimization and Control*. Eds: Gianakoglou, Tsahalis, Periaux, Papailiou, Fogarty. (EUROGEN 2001. Greece). CIMNE pp. 337-342
- [2.4] V. Bhaskar, K. Gupta, K. Ray, "Multiobjective Optimisation of an industrial wiped film poly(ethylene terephthalate) reactor: some further insights", *J. Computers and Chemical Engineering*, Elsevier, 25 (2001), pp. 391-407.
- [2.5] J.T. Buchanan, "A Naïve Approach for Solving MCDM Problems: The GUESS Method", *Journal of the Operational Research Society*, 48 (1997) pp. 202-206.
- [2.6] A. Chames, W.W. Cooper, R.O. Ferguson, "Optimal estimation of executive compensation by linear programming", *Manage. Sci.*, 1 (1955) pp. 138-151.
- [2.7] C.A. Coello Coello, "A Short Tutorial on Evolutionary Multiobjective Optimization", *Lecture Notes in Computer Science*, Evolutionary Multi-criterion Optimization 2001, pp. 21-40.
- [2.8] C.A. Coello Coello, "Multiobjective Optimization using a Micro-Genetic Algorithm" *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, San Francisco, Morgan Kaufmann.
- [2.9] C.A. Coello Coello, "A comprehensive survey of evolutionary-based multiobjective optimization." *Knowledge and Information Systems* 1-3 (1999) pp. 269-308.
- [2.10] C.A. Coello Coello, D. Van Veldhuizen, G. Lamont, "*Evolutionary Algorithms for solving multi-objective problems*". Kluwer Academic Publishers - GENA Series (2002).
- [2.11] D. Come, N. Jerram, J. Knowles, M. Oates, "PESA-11: Region-based Selection in Evolutionary Multiobjective Optimization". In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, San Francisco, Morgan Kaufmann.
- [2.12] D. Come, J. Knowles, M. Oates, "The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization", *Sixth International Conference on Parallel Problem Solving from Nature (PPSN-VI)*, Paris (2000), pp. 839-848.
- [2.13] I. Das, J.E. Dermis, "Normal-boundary Intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems", *SIAM Journal of Optimization*, 8 (1998) pp. 631-657.
- [2.14] K. Deb, *Multiobjective Optimization using Evolutionary Algorithms*. John Wiley & Sons - Series in Systems and Optimization- (2001).
- [2.15] K. Deb, "Multi-Objective Genetic Algorithms: Problem Difficulties and Constructions of Test Problems", *Evolutionary Computation* 7-3 (1999) pp. 205-230. MIT Press.
- [2.16] K. Deb. '*Evolutionary Algorithms for Multi-Criterion Optimisation in Engineering Design*'. *Evolutionary Algorithms in Engineering and Computer Science (EUROGEN 1999)* John Wiley & Sons.

- [2.17] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multiobjective Optimization: NSGA-II", *Sixth International Conference on Parallel Problem Solving from Nature (PPSN-VI)*, París (2000), pp. 849-858.
- [2.18] K. Deb, T. Goel, "Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence", pp. 67-81, in *Lecture Notes in Computer Science, Evolutionary Multi-Criterion Optimization 2001*, Springer.
- [2.19] K. Deb, S. Jain, "Running performance metrics for evolutionary multi-objective optimization, *Technical Report 2002004*, Indian Institute of Technology Kanpur, Mayo 2002.
- [2.20] K. Deb, M. Mohán, S. Mishra, "Towards a quick computation of well-spread Pareto-Optimal Solutions", *Lecture Notes in Computer Science, Evolutionary Multi-Criterion Optimization 2003*, Springer, pp. 222-236.
- [2.21] K. Deb, A. Pratap, S. Agrawal, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation* 6 (2), 182-197, 2002.
- [2.22] C. Fonseca, P. Fleming. On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. In H. M. Voigt, W. Ebeling, I. Redchenberg and H. P. Schwefel (Eds.). *Parallel Problem Solving from Nature - PPSNIV* (1996). Lecture Notes in Computer Science, pp. 584-593. Berlin, Germany. Springer-Verlag.
- [2.23] C. Fonseca, P. Fleming. "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms I: A Unified Formulation". In *IEEE Transactions on Systems, Man and Cybernetics*, 28-1, (1998) pp. 26-37.
- [2.24] C. Fonseca, P. Fleming, Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms II: Application Example. In *IEEE Transactions on Systems, Man and Cybernetics*, 28-1, (1998) pp. 38-47.
- [2.25] C. Fonseca, P. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* 3-1 (1995), pp. 1-16.
- [2.26] C. Fonseca, P. Fleming, An overview of evolutionary algorithms in multiobjective optimization. *Technical Report*, Department of Automatic Control and Systems Engineering, University of Sheffield, UK (1994).
- [2.27] C. Fonseca, P. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization." In S. Forrest (Ed.). *Proceedings of the Fifth International Conference on Genetic Algorithms* (San Mateo, California, 1993), pp. 416-423. University of Illinois at Urbana-Champaign: Morgan Kaufmann Publishers.
- [2.28] M. Galante. 'Genetic Algorithms as an approach to optimise real-world trusses' *International Journal for Numerical Methods in Engineering*, J. Wiley, vol 39, pp 361-382 (1996).
- [2.29] A.M. Geoffrion, J.S. Dyer, A. Feinberg, "An Interactive approach for multicriterion optimization with an application to the operation of an academic department", *Management Science*, 19 (1972) pp. 357-368.
- [2.30] Goldberg D.E. '*Genetic algorithms for search, optimisation, and machine learning*' Reading, MA: Addison Wesley (1989).
- [2.31] Goldberg D. E. and K. Deb 'A Comparative Analysis of Selection Schemes Used in Genetic Algorithms'. In *IFOGA* (1990) pp 69-93. Ed. G. Rawlings. Morgan Kaufmann Publishers
- [2.32] D. Greiner, G. Winter, JM. Emperador. "Optimising Frame Structures by different strategies

- of genetic algorithms". *Finite Elements in Analysis and Design*, Elsevier 2001; 37(5):381-402.
- [2.33] P. Hajela, C.Y. Lin, "Genetic search strategies in multicriterion optimal design", *Structural Optimization*, 4 (1992) 99-107.
- [2.34] G. Harik. 'Finding multiple solutions in problems of bounded difficulty'. *Illigal Report N° 94002(1994)*.
- [2.35] J.H. HoUand. 'Algoritmos Genéticos'. En "*Investigación y Ciencia*" (Septiembre 1992) pp 38-45.
- [2.36] J. Hom, "The Nature of Niching: Genetic Algorithms and the Evolution of Optimal, Cooperative Populations", *zD Thesis*, University of Illinois at Urbana Champaign, EEUU (1997).
- [2.37] J. Hom, N. Nafpliotis, "Multiobjective Optimization using the Niche Pareto Genetic Algorithm", *Technical Report Illigal 93005*, University of Illinois at Urbana Champaign, EEUU (1993).
- [2.38] C.L. Hwang, A.S.M. Masud, "*Múltiple Objective Decision Making - Methods and Applications: A State-of-the-Art Survey*", Springer-Verlag (1979).
- [2.39] A. Jaszkiewicz, R. Slowinski, "The 'Light Beam Search' Approach - An Overview of Methodology and Applications." *European Journal of Operational Research*, 113 (1999) pp. 300-314.
- [2.40] J. Jiménez, D. Greiner, J. Abderramán, P. Cuesta y G. Winter. 'Un análisis comparativo de algoritmos multiobjetivo: NSGA-II y MRSMO', *Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados*, AEB-02. Mérida, España.
- [2.41] J. Knowles, D. Come, One metrics for computing non-dominated set. *Proceedings of the World Congress on Computational Intelligence*, (2002) pp. 711-716.
- [2.42] J. Knowles, D. Come, "Approximating the Nondominated Front Using the Pareto Archive Evolution Strategy", *Evolutionary Computation*, 8-2 (2000) pp. 149-172.
- [2.43] F. Kursawe, "A variant of evolution strategies for vector optimization", *Lecture Notes in Computer Science 496*, Parallel Problem Solving from Nature, pp. 193-197. Springer (1991).
- [2.44] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization", *Evolutionary Computation*, 10-3, (2002) pp. 263-282.
- [2.45] M. Laumarms, G. Rudolph, H.P. Schwefel, "A spatial predator-prey approach to multiobjective optimization: a preliminary study", *Parallel Problem Solving from Nature, Lecture Notes in Computer Science*, (1998) pp. 241-249.
- [2.46] R.T. Marler, J.S. Arora, "Survey of multi-objective optimization methods for engineering", *Structural and Multidisciplinary Optimization*, 26 (2004) pp. 369-395.
- [2.47] W. Mahfoud. 'Niching methods for Genetic Algorithms' *Illigal Report n° 95001* (1995).
- [2.48] A. Messac, A. Ismail-Yahaya, C.A. Mattson, "The normalized normal constraint method for generating the Pareto Frontier", *Structural and Multidisciplinary Optimization*, 25 (2003) pp. 86-98.
- [2.49] A. Messac, "Physical Programming: effective optimization for computational design", *AIAA Journal*, 34 (1996) pp. 149-158.
- [2.50] K. Miettinen, "Some Methods for Nonlinear Multi-objective Optimization", *Lecture Notes in Computer Science*, Evolutionary Multi-criterion Optimization 2001, pp. 1-20.
- [2.51] K. Miettinen, "Nonlinear Multi-objective Optimization", Kluwer Academic Publishers, 1999.
- [2.52] H. Nakayama, "Aspiration level approach to Interactive multi-objective programming and its applications". In: *Advances in Multicriteria Analysis*. Kluwer Academic Publishers (1995) pp. 147-

174.

- [2.53] J.F. Nash, Equilibrium points in n-person games. *Proc. Nat. Acad. Sel U.S.A*, 36 (1950) 46-49.
- [2.54] J.F. Nash, Noncooperative games. *Annals of Mathematics*, 54 (1951) 289.
- [2.55] S. Obayashi, S. Takahashi, Y. Takeguchi, Niching and elitist models for MOGAs. *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, Amsterdam (1998), pp. 250-259.
- [2.56] V. Pareto, "*Cours D' Economie Politique* ", Volume I y II, F. Rouge, Lausanne, 1896.
- [2.57] J. Periaux, M. Sefrioui, "Nash Genetic Algorithms: Examples and Applications", In- *Innovative Tools for Scientific Computation in Aeronautical Engineering*, Ed: J. Periaux, P. Joly, O. Pironneau, E. Oñate. CIMNE, (2001) pp. 391-404.
- [2.58] R. Purshouse, P. Fleming, "Why use Elitism and Sharing in a Multiobjective Genetic Algorithm?", *Genetic and Evolutionary Computation Conference GECCO-2002*, pp. 520-527, New York, Morgan Kaufmann Publishers.
- [2.59] O. Roudenko, "Application des Algorithmes Evolutionnaires aux Problemes d'Optimisation Multi-Objectif avec Contraintes", *PhD Thesis*, Marzo 2004, École Polytechnique.
- [2.60] G. Sánchez Carpena, "Diseño y Evaluación de algoritmos evolutivos multiobjetivo en optimización y modelización difusa". *Tesis Doctoral*, 2002, Universidad de Murcia, Departamento de Ingeniería de la Información y las Comunicaciones.
- [2.61] D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pp. 93-100. (1985) Lawrence Erlbaum.
- [2.62] M. Sefrioui, J. Periaux, "Nash Genetic Algorithms: Examples and Applications", *Proceedings of the Congress on Evolutionary Computation 2000*, IEEE Press, pp. 509-516.
- [2.63] N. Srinivas, K. Deb, 'Multiobjective optimisation using non-dominated sorting in genetic algorithms' *Evolutionary Computation* 2-3 (1995) pp. 221-248.
- [2.64] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms. *Technical report*, Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, India. (1993).
- [2.65] R.E. Steuer, "*Múltiple Criteria Optimization: Theory, Computation and Applications*", John Wiley & Sons, (1986).
- [2.66] D. Van Veldhuizen, G. Lamont, "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art", *Evolutionary Computation*, 8-2 (2000) pp. 125-147.
- [2.67] D. Van Veldhuizen and Gary B. Lamont. "Multiobjective Evolutionary Algorithm Test Suites", *Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 351-357, San Antonio, Texas,. ACM . (1999).
- [2.68] D. Van Veldhuizen, G. Lamont, "Multiobjective Evolutionary Algorithm Research: A History and Analysis". Technical Report TR-98-03, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio (1998).
- [2.69] D. Whitley, S. Rana, R. Heckendorn 'Representation Issues in Neighbourhood Search and Evolutionary Algorithms' *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science* (EUROGEN 1997). John Wiley & Sons.

- [2.70] D. Whitley 'The GENITOR Algorithm and Selection Pressure : Why Rank-Based Allocation of Reproductive Triáis is Best' *Proceedings of the Third International Conference on Genetic Algorithms*. pp 116-121 (1989)
- [2.71] A.P. Wierzbicki, "A Mathematical Basis for Satisficing Decisión Making", *Mathematical Modelling*, 3 (1982) pp. 391-405.
- [2.72] G. Winter, "Recent Advances and Trends in Multiobjective Evolutionary Algorithms for Optimization. Applications in Dependability", In: *Proceedings of the IV Congress on Reliability: Dependability*, Eds: Galván, Winter, Cuesta and Aguasca. September 2002, Spain.
- [2.73] M. Zeleny, "*Linear Multiobjective Programming*", Springer (1974) pp. 197-220.
- [2.74] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications", *phD Tesis*, Institut fur Technische Informatik und Kommunikationsnetze, Swiss Federal Institute of Technology, Zürich (1999).
- [2.75] E. Zitzler, K. Deb, L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", *Evolutionary Computation*, 8-2 (2000) pp. 173-195.
- [2.76] E. Zitzler, M. Laumarms, L. Thiele "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization", *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, John Wiley & Sons. (2001).
- [2.77] E. Zitzler, M. Laumarms, L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", *TIK-Report 103*, (May 2001) Swiss Federal Institute of Technology (ETH) Zurich.
- [2.78] E. Zitzler, M. Laumarms, L. Thiele, C. Fonseca, V. Grunert da Fonseca, "Why quality assessment of multiobjective optimizers is difficult", *Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann (2002) pp. 666-673.
- [2.79] E. Zitzler, L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the SPEA. In *IEEE Transactions on Evolutionary Computation*, 3-4 (1999) pp. 257-271.
- [2.80] E. Zitzler, L. Thiele. "An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach". *TIK-Report 43*, Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology, Switzerland (1998).
- [2.81] E. Zitzler, L. Thiele. "Multiobjective Optimization using evolutionary algorithms - a comparative case study". *Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, Springer (1998) pp. 292-301.

Capítulo 3. Optimización Multicriterio Estructural

3.1 *Estado del arte en Ingeniería Estructural-Civil de Algoritmos Evolutivos y de Optimización Multiobjetivo*

3.1.1 Optimización Estructural con Algoritmos Evolutivos

Las posibilidades que ofrecen los algoritmos evolutivos (AEs) se traducen en un gran número de aplicaciones en ingeniería estructural y civil, sobretodo en años recientes, destacando las referidas a continuación:

La primera aplicación de los algoritmos genéticos en optimización estructural fue la realizada por D. Goldberg y M.P. Samtani (1986) [3.64], en una estructura de nudos articulados clásica de 10 barras para el problema del mínimo peso restringido con variables continuas.

P. Hajela y C.J. Shih (1990) [3.75] aplican los algoritmos genéticos a varios problemas estructurales, caso de la respuesta dinámica de una estructura de diez barras articuladas y de un emparrillado de dos vigas. Analizan aspectos como codificación, operadores de selección, cruce, mutación y tamaño de población.

W.M. Jenkins (1991) [3.92] presenta una metodología de optimización estructural mediante el Algoritmo Genético Simple (AGS), introduciendo aspectos de sus operadores para el acople con el cálculo estructural y resolviendo una estructura articulada de techado, una estructura articulada de tres barras y una sección de pared delgada. También en este año se publica otra referencia precursora, de G.Svedet al. (1991) [3.202].

S. Rajeev y C.S. Krishnamoorthy (1992) [3.173], optimizan estructuras articuladas con variables discretas, presentando detalladamente tablas descriptivas con el proceso de optimización y aplicándolo a una torre de transmisión de 160 barras.

H. Adeli y N.T. Cheng (1993) [3.3] emplean algoritmos genéticos para el diseño óptimo de estructuras de barras articuladas espaciales, sujetas a restricciones de esfuerzos, desplazamientos y de fabricación (disponibilidad de perfiles), que son tratadas mediante una función de penalización cuadrática. Se tratan tres ejemplos con tamaños desde 12 barras hasta 72 barras.

D.E. Grierson y W.H. Pak (1993) [3.68] afrontan el diseño de estructuras de barras considerando el dimensionado discreto, geométrico y topológico de las mismas. Esto es aplicado a dos casos test. Es un trabajo pionero en la optimización de *estructuras de nudos rígidos* mediante algoritmos genéticos.

M. Galante (1993) [3.58][3.59] optimiza mediante algoritmos genéticos varias estructuras de barras articuladas: una estructura de 10 barras con restricción en tensiones y desplazamientos, una estructura-puente de 25 barras y una estructura pórtico de 49 barras que se optimiza también mediante perfiles reales. Realiza esta optimización en dos etapas, siendo este método precursor del renacimiento, introducido con posterioridad [3.57] y mejorando resultados anteriores de otros autores.

J. Sakamoto y J. Oda (1993) [3.184] presentan una técnica híbrida de optimización mediante algoritmos genéticos para la configuración de estructuras de barras de nudos articulados, y combinada con un criterio de optimalidad para la determinación de la sección de las barras.

H. Adeli, N.T. Cheng (1994) [3.2] presentan un algoritmo genético hibridado que integra el método primal-dual como parte de la función de penalización. De ese modo se eliminan las dificultades de determinación de los coeficientes de la función de penalización.

V.K. Koumoussis y P.G. Georgiou (1994) [3.110] emplean un algoritmo genético en discretización de techados mediante barras articuladas de acero, determinando un intervalo de parámetros evolutivos (tamaño poblacional, cruce y mutación) cuyos resultados resultan aceptables.

H. Adeli y S. Kumar (1995) [3.4] consideran la paralelización de la optimización llevada a cabo por el algoritmo genético mediante la distribución en un cluster de estaciones de trabajo conectados en red local. Se efectúa un balanceo dinámico de carga dada la heterogeneidad de los ordenadores en un entorno multiusuario. Se aplica a dos estructuras de barras articuladas sometidas a cargas verticales y horizontales determinadas por la normativa norteamericana, siendo la mayor de ambas de 848 barras.

S.D. Rajan (1995) [3.171] aborda el diseño óptimo de estructuras de barras de nudos articulados espaciales considerando simultáneamente el dimensionado, forma y topología. Se codifica con variables continuas (posiciones nodales) y discretas (conectividades y condiciones de contorno). También hace uso de una base de datos donde se almacenan los valores de las funciones objetivo ya calculadas para contrastar los nuevos cromosomas y evitar su recalcado. Los ejemplos analizados incluyen dos estructuras de 6 y 14 nodos respectivamente.

S-J. Wu y P-T. Chow (1995) [3.217] utilizan algoritmos genéticos de estado-estacionario (steady-state) para la optimización discreta de estructuras de barras de nudos articulados, mostrándose ventajosos respecto del algoritmo genético generacional clásico.

T-Y. Chen y C-J. Chen (1997) [3.20] presentan un estudio donde incorporan elitismo, varios tipos de cruce (de uno a cuatro puntos) así como búsqueda local final, aplicándolo tanto a estructuras de barras articuladas como a diseño topológico con elementos finitos (77 discretizaciones) para el diseño óptimo de mínimo peso.

T. Furukawa y G. Yagawa (1997) [3.56] presentan un método de identificación de conjunto de parámetros de ecuaciones constitutivas del material mediante un algoritmo genético de codificación real, que comparado con uno binario estándar ofrece mejores resultados. Se aplica a dos casos de identificación del modelo viscoplastico de Chaboche.

M-W. Huang y J.S. Arora (1997) [3.87] comparan distintos métodos aplicables a problemas con variables mixtas (continuas y discretas), como son el método de ramificación y acotamiento (branch & bound, basado en programación lineal), cristalización simulada, algoritmos genéticos, redondeo dinámico al alza (dynamic rounding-up) y método de búsqueda por vecindad (neighbourhood search method). Se aplica a quince problemas distintos, entre ellos varios de estructuras de nudos articulados de hasta 200 barras.

W.M. Jenkins (1997) [3.91] aborda la optimización de *estructuras de nudos rígidos* sugiriendo un control de los operadores de mutación y cruce de forma adaptativa para mejorar el rendimiento del algoritmo genético.

X. Liu et al. (1997) [3.126] proponen una optimización mediante cristalización simulada (simulated annealing) y programación lineal secuencial, aplicado al diseño óptimo de localización de accionadores en estructuras de barras de nudos articulados (hasta 90 barras).

S. Rajeev y C.S. Krishnamoorthy (1997) [3.172] aplican en optimización de estructuras de barras articuladas - topológica, de configuración y dimensionamiento de las secciones- dos nuevos

operadores: el método de dos fases (similar al operador renacimiento propuesto por Galante [3.57]), y un cromosoma de longitud variable. Se tratan varios casos test, considerando un caso tridimensional de torre de antena de microondas.

J.P. Yang y C.K. Soh (1997) [3.223] efectúan un estudio comparativo en la optimización de estructuras de la selección por torneo enfrentada a la selección proporcional simple, mostrando las ventajas de la primera frente a la segunda.

C.V. Camp et al. (1998) [3.17] realizan un módulo de optimización mediante algoritmos genéticos en un programa de análisis de elementos finitos, proponiendo diversas funciones objetivo y funciones de penalización. Además realizan un estudio comparativo aplicado a estructuras planas (tanto de nudos articulados como de *nudos rígidos*) donde se comparan los algoritmos genéticos con un método continuo clásico de optimización basado en el criterio de optimalidad.

S.Y. Chen y S.D. Rajan (1998) [3.22] realizan un exhaustivo estudio acerca de la optimización de *estructuras de barras de nudos rígidos* mediante algoritmos genéticos, considerando los efectos de tipo de cruce, tamaño de la población, número de generaciones, representación, etc.

P. Hajela et al. (1998) [3.73] utilizan los algoritmos genéticos para determinar la configuración óptima de emparrillados bi- y tri-dimensionales con restricciones de desplazamientos, tensiones y pandeo. Sugieren un algoritmo de dos niveles, donde en el primer nivel se impone la restricción de estabilidad de la estructura y en el segundo nivel las restricciones de esfuerzos y desplazamientos. Se aplica a un emparrillado bidimensional y a la estructura de cola de un helicóptero.

J.P. Leite y B.H.V. Topping (1998) [3.118] realizan una propuesta sobre el operador de cruce de un punto, analizando el punto de cruce óptimo, además de analizar los operadores para un balance óptimo exploración / explotación que redunde en un mejor rendimiento del algoritmo genético. Esto lo aplican a varios problemas estructurales, como una viga soldada, una estructura de nudos articulados de diez barras y secciones en I pretensadas.

P.B. Nair et al. (1998) [3.152] desarrollan una propuesta que combina modelos aproximados con una optimización mediante algoritmos genéticos, consiguiendo reducir en un 97% el número de evaluaciones de la función objetivo exactas respecto del problema original en un problema de estructuras articuladas de 10 barras.

M.P. Saka (1998) [3.183] optimiza el diseño de emparrillados mediante algoritmos genéticos considerando los efectos de alabeo y cortante y con restricciones de esfuerzos y desplazamientos.

C.K. Soh y J. Yang (1998) [3.195] optimizan el diseño de estructuras articuladas de barras para puentes en dos etapas. En una primera etapa se optimiza simultáneamente la geometría y dimensionamiento de las variables desde un patrón inicial. En la segunda etapa las variables topológicas se optimizan bajo consideraciones de ciertos patrones predefinidos y posteriormente las variables geométricas y de dimensionamiento bajo la nueva topología se redimensionan. Los resultados mejoran diseños obtenidos con técnicas numéricas de optimización tradicionales.

B.H.V. Topping y J.P.B. Leite (1998) [3.206] realizan un análisis de distintos algoritmos genéticos en paralelo, aplicándolos a problemas complejos de ingeniería estructural con funciones objetivo de coste medio-alto, como el diseño de un puente suspendido por cable. La paralelización consigue reducción del tiempo de cálculo.

S. Botello et al. (1999) [3.14] presentan un estudio donde hibridizan los algoritmos genéticos con la cristalización simulada creando el algoritmo de búsqueda general estocástica (GSSA) y lo aplican al

problema del mínimo peso restringido de estructuras de nudos articulados en cinco problemas, de hasta 2440 barras. Además efectúan su paralelización para cuatro procesadores.

M.R. Ghasemi et al. (1999) [3.62] presentan una optimización mediante algoritmos genéticos considerando tanto variables discretas como continuas, con varios casos test de ejemplo de estructuras de barras articuladas bidimensionales de tamaño entre 4 y 940 barras. También muestra las ventajas del operador renacimiento en ellas.

A.A. Groenwold et al. (1999) [3.69] presentan un método híbrido de optimización discreta de estructuras de barras articuladas, donde emplean el método de aproximación esférica (SAM) desarrollado por los autores, para la obtención del óptimo continuo global y a partir de él utilizan un algoritmo genético que realiza la búsqueda en el dominio discreto: algoritmo genético regional. Presentan cuatro casos test de distintas estructuras de hasta 200 barras de tamaño.

I.C. Yeh (1999) [3.224] emplea un algoritmo genético hibridizado para la optimización discreta de estructuras articuladas de barras, mejorando los resultados del algoritmo genético puro.

En S.Y. Chen y S.D. Rajan (2000) [3.21] presentan un estudio de la optimización mediante algoritmos genéticos aplicado al diseño automatizado de estructuras de barras sometidas tanto a cargas estáticas como dinámicas y bajo las restricciones de la normativa norteamericana. Se considera el dimensionamiento de las barras, así como la topología de la estructura en el diseño.

F. Erbatur et al. (2000) [3.47] optimizan estructuras de barras articuladas y *rígidas* de hasta 112 barras mediante algoritmos genéticos, presentando casos de optimización continua, así como optimización real.

En O. Hasancebi y F. Erbatur (2000) [3.78], encontramos un estudio comparativo entre operadores de cruce (un punto, dos puntos, uniforme, un cruce mixto y cruce entre variables directamente) aplicados al problema de minimización de peso restringido en estructuras articuladas y aplicado a tres casos test de 10, 25 y 72 barras respectivamente.

En M.S. Hayalioglu (2000) [3.80], optimiza *estructuras de nudos rígidos* de cuatro y diez plantas, considerando estado elasto-plástico no lineal, mediante algoritmos genéticos.

M. J. Jakiela (2000) [3.88] utilizan los algoritmos genéticos para la resolución del problema de minimización de la masa estructural de una viga en voladizo cargada en su extremo, optimizando la topología de la misma mediante discretización continua de su sección longitudinal y realizando un estudio considerando el número de huecos interiores a la misma.

C.L. Karr et al. (2000) [3.96] optimizan el diseño y rendimiento de hidrociclones (empleados para la separación de materiales mediante altas velocidades de rotación) mediante algoritmos genéticos y lógica difusa.

En [3.129] B. Liu et al. (2000) emplean algoritmos genéticos adaptados a cálculos combinatorios con permutaciones para el diseño de placas compuestas laminadas (considerando diferentes ángulos de inclinación).

K. Matous et al. (2000) [3.140] presentan una variedad de problemas de mecánica computacional resueltos mediante distintos tipos de algoritmos genéticos, comprendiendo el diseño de una viga de hormigón reforzada con acero, compuestos laminados pretensados y el diseño de una celda unitaria que representa la microestructura de un medio compuesto de fibras aleatoriamente dispuestas.

En [3.169] Ponterosso et al. (2000) presentan un algoritmo genético heurístico para la optimización de cargas de colapso en arcos de mampostería que mejora los resultados del algoritmo genético estándar.

R. Spallino y G. Thierauf (2000) [3.197], emplean estrategias evolutivas para el diseño óptimo de compuestos laminados sujetos a pandeo térmico con restricciones de tensión y contigüidad de capas, comparando dos métodos de tratamiento de las penalizaciones: el descarte de soluciones no factibles y una función de penalización.

J. Sziveri et al. (2000) [3.203] presentan una mejora del método de generación de subdominios (SGM) mediante la inserción de un nuevo procedimiento no convexo de bisección y un nuevo módulo de optimización basado en algoritmos genéticos. Este nuevo método mejorado (ESGM) permite la descomposición y refinamiento de mallas de elementos finitos, presentándose su aplicación a los casos de un gancho de maquinaria y una sección del muro de una presa.

H. Chou y J. Ghaboussi (2001) [3.28] resuelven el problema inverso de detección de daños en estructuras de barras articuladas (existencia, posición y extensión) a través de la medición de ciertos desplazamientos nodales y la comparación de los mismos con los cálculos teóricos provenientes del análisis por elementos finitos, de forma satisfactoria mediante algoritmos genéticos con codificación redundante.

K. Deb y S. Gulati (2001) [3.42] realizan optimización topológica, de diseño y dimensionamiento con variables continuas de estructuras de barras articuladas bi- y tri-dimensionales mediante algoritmos genéticos de variable real. Se consideran restricciones de esfuerzo, desplazamiento nodal y pandeo sobre casos test que varían entre 11 y 25 barras.

W. Gutkowski et al. (2001) [3.71] utilizan estrategias evolutivas para resolver el problema del mínimo peso restringido mediante una mutación controlada, en la que se consideran los esfuerzos estructurales para su mejor direccionamiento. En un caso test estándar, mejoran considerablemente los resultados proporcionados por la estrategia evolutiva convencional.

M.N.S. Hadi y Y. Arfiadi (2001) [3.72] optimizan el diseño de pavimentos de hormigón rígidos, con el objeto de obtener diseños seguros y económicos, y basándose en un modelo mediante elementos finitos de los mismos (AUSTROADS).

E.S. Kameshki y M.P. Saka (2001) [3.94] emplean un algoritmo genético para la optimización del problema de mínimo peso restringido aplicado al diseño de estructuras de nudos semirígidos en uniones pilar-viga y aplicado a dos casos test: un edificio de 10 pisos y un vano, y otro de tres pisos y dos vanos.

H. Kim y H. Adeli (2001) [3.103] presentan una optimización de suelos compuestos mediante un algoritmo genético de codificación real, considerando las imposiciones de la normativa norteamericana e incluidas en la función objetivo: el coste del hormigón, de los refuerzos de acero y de cortante. Muestran que un ahorro de coste es posible.

e s. Krishnamoorthy (2001) [3.112] presenta un amplio espectro de aplicaciones de diseño estructural óptimo mediante algoritmos genéticos, tales como estructuras de barras metálicas, torres, puentes, vigas de hormigón reforzado, láminas o planificación de la distribución de edificios.

En [3.128] G.R. Liu y S.C. Chen (2001) resuelven un problema inverso con micro-algoritmos genéticos donde mediante la respuesta temporal armónica de la placa ante una excitación, logran

detectar la localización, área y grado de fisura en placas sandwich modelizadas mediante elementos finitos.

J.C. Miles et al. (2001) [3.147] presentan un diseñador de edificios de oficinas multi-planta mediante algoritmos genéticos, que considera como parámetros de diseño ponderables según las preferencias del diseñador: la longitud de vano entre columnas, el coste (altura del edificio, porcentaje de área neta aprovechable y peso por unidad de área) y el impacto medioambiental (localización, altura libre, tipo de ventilación).

P. Nanakom y K. Meesomklin (2001) [3.153] presentan una fianción de penalización con coeficientes dinámicos durante la evolución, aplicándolo al problema de mínimo peso en estructuras con nudos articulados de 6 y 10 barras y una *estructura de nudos rígidos* de 8 pisos.

M. Papadrakakis et al. (2001) [3.165] presentan un estudio de algoritmos genéticos y estrategias evolutivas aplicadas a la optimización de estructuras de barras. Asimismo aplican técnicas de descomposición de dominio a las mismas para paralelizar el proceso de resolución. También se propone la hibridización de la optimización mediante técnicas de búsqueda local centradas tras la convergencia del optimizador evolutivo, además de la sustitución de parte de las evaluaciones de la función objetivo mediante aproximaciones evaluadas mediante redes neuronales.

I. Rajendran y S. Vijayarangan (2001) [3.174] diseñan óptimamente muelles compuestos en laminas, mediante algoritmos genéticos, y de aplicación a la suspensión de automóviles, consiguiendo una reducción del 75% de peso en un caso test ensayado al considerar como variables el espesor y ancho de lámina.

K.C. Sarma y H. Adeli (2001) [3.189] presentan un algoritmo genético paralelizado en dos niveles para la optimización de estructuras de barras metálicas bajo las restricciones legales americanas. Se realiza combinando la plataforma MPI (Message Passing Interface) con OpenMP-API y se presentan resultados de rendimiento paralelo.

G. Soremekun et al. (2001) [3.196] introducen dos tipos de selección, la selección elitista múltiple y la selección elitista variable, que comparan con un elitismo simple y aplican al diseño combinatorio óptimo de compuestos laminados.

P. Gurfil y N.J. Kasdin (2002) [3.70] emplean algoritmos genéticos con técnicas de nichos permitiendo la coexistencia de múltiples mínimos, para la caracterización de familias de órbitas en el problema espacial elíptico restringido de tres cuerpos (incorporando excentricidad).

W.M. Jenkins (2002) [3.90] introduce una codificación decimal en el cromosoma utilizando algoritmos genéticos con elitismo, para la optimización de estructura articuladas de diez y seiscientos cuarenta barras, considerando restricciones tensionales y de desplazamientos nodales.

J-N. Lagaros et al. (2002) [3.116] realizan un estudio comparativo entre algoritmos genéticos y estrategias evolutivas, hibridizando además con una búsqueda local mediante programación secuencial cuadrática. Lo aplican a varios casos test de *estructuras de nudos rígidos* (la mayor de ellas con 10 pisos y 3 vanos, aunque variables agrupadas totalizando un total de 22).

H. Kawamura et al.(2002) [3.99] presentan un nuevo método de optimización mediante algoritmos genéticos del problema de mínimo peso de estructuras de barras articuladas, donde la codificación de la estructura no se realiza mediante combinación de barras, sino a través de la modelización de combinaciones triangulares. Esto es aplicado a varios casos test.

Q.S. Li et al. (2002) [3.122] emplean un algoritmo genético por niveles (multilevel GA) para la optimización del diseño de control estructural (que actúa sucesivamente en el número de actuadores, el posicionamiento óptimo de los mismos y, control óptimo y análisis de respuesta). Se ensayan dos supuestos, uno bajo carga sísmica y otro bajo carga de viento.

En [3.130] G.R. Liu et al. (2002) emplean algoritmos genéticos hibridizados con mínimos cuadrados para la determinación de las propiedades de compuestos laminados como problema inverso, aplicándolo a laminados carbono-sílice y vidrio-epoxi.

J.L. Marcelin (2002) [3.136] y (2001) [3.137] emplea algoritmos genéticos con reducción del número de evaluaciones de la función objetivo por aproximación, empleando dos métodos, una red neuronal y el método de Ritz. Se aplica a la optimización del refuerzo de placas, considerando la posición y ángulo de los refuerzos.

P. Miltiadou y L.C. Wrobel (2002) [3.148] resuelven un problema inverso mediante algoritmos genéticos y su acoplamiento con elementos de contorno (BEM): la identificación de parámetros de identificación en curvas de polarización en sistemas de protección catódica, minimizando la diferencia entre la predicción del modelo y el dato experimental de superficies de potencial.

E. Sandgren y T.M. Cameron (2002) [3.187], presentan una metodología para el diseño óptimo robusto de estructuras de barras mediante la consideración de la variabilidad de las variables que intervienen en el diseño: módulo de elasticidad del material, carga (dirección y módulo), empleando dos métodos de optimización: algoritmos genéticos y programación no-lineal. Como casos de aplicación emplean una estructura de 10 barras y otra de hasta 200 barras modelizando el bastidor de un vehículo.

F. Tin-Loi y N.S. Que (2002) [3.204] realizan una comparativa de algoritmos genéticos y programación no-lineal, aplicada a problemas de identificación de parámetros de modelos cohesivos de fractura.

En [3.219] Y.G. Xu y G.R. Liu (2002) hacen uso de un micro-algoritmo genético combinado con un optimizador local para la detección de defectos en estructuras laminadas mediante el uso de campos de ondas elásticas dispersas.

Y. Yang y C.K. Soh (2002) [3.222] resuelven la optimización topológica y de tipos de perfiles discretos, minimizando el peso restringido, mediante programación genética, con una flexibilidad mayor puesto que permite una independencia mayor de la geometría inicial que los algoritmos genéticos. Se aplica a un caso test de estructura articulada, cuyo diseño óptimo final es de 29 barras.

N. Ali et al. (2003) [3.6] realizan el acoplamiento de un algoritmo genético autoadaptativo con el software comercial ANSYS, para la optimización topológica de estructuras tridimensionales de nudos articulados (3, 25 y 42 barras) considerando también el pandeo para la minimización del peso total de la estructura.

En O. Hrstka et al. (2003) [3.86] abordan un problema de configuración óptima de una viga de hormigón armado, así como la configuración de una celda unitaria periódica en materiales compuestos. Ello lo realizan comparando varios algoritmos evolutivos de diseño reciente.

K. Jarmai et al.(2003) [3.89] optimizan el diseño de secciones en I soldadas minimizando su volumen, y considerando como restricciones el pandeo lateral por torsión, además del pandeo local (alas y alma). Se comparan los algoritmos genéticos, con otros tres métodos de optimización (método de gradiente, método de búsqueda ortogonal y técnica de evolución diferencial).

R.J. Jiang et al. (2003) [3.93] abordan el problema inverso de identificación de varias masas en movimiento sobre vigas continuas de varios vanos mediante algoritmos genéticos y usando medidas de aceleración.

A. Kaveh y V. Kalatjari (2003) [3.97] y (2002) [3.98] proponen la optimización discreta de estructuras articuladas, tanto de sección como topológica, mediante el acoplamiento de algoritmos genéticos y un resolvidor estructural basado en el método de fuerzas. Asimismo proponen una contracción de la longitud de los cromosomas en la optimización e introducen conceptos de teoría de grafos para la concepción de la población inicial. Se aplica a varios casos test de entre 42 (plana) y 25 barras (tridimensional).

D.K. Liu et al. (2003) [3.132], realizan la optimización de posicionamiento de tres actuadores de control en un edificio de 16 plantas con datos de 18 terremotos diferentes mediante algoritmos genéticos.

En [3.134] T.A. Lovell (2003) presenta una comparativa de algoritmos genéticos frente a técnicas de filtrado de correcciones diferenciales empleados para el análisis de la dinámica orbital de sistemas ligados de satélites, proporcionando ambos métodos resultados de similar calidad.

M. Papadrakakis et al. (2003) [3.163] analizan el diseño óptimo de estructuras de barras articuladas desde el punto de vista de su paralelización, considerando tanto el aspecto de paralelización del optimizador (estrategias evolutivas y algoritmos genéticos), como de la descomposición de dominio del resolvidor (método FETI). Como caso de aplicación mayor evalúan una estructura tridimensional diseñada para cubrir un área de 20.000 m².

B. Pichler et al. (2003) [3.166] presentan un método de identificación de parámetros de modelos en ingeniería geotécnica mediante la simulación del modelo de elementos finitos mediante una red neuronal, de la que hace uso un algoritmo genético para optimizar los parámetros del modelo de tal modo que se aproxime lo más posible a los datos experimentales obtenidos con un coste computacional asequible. Finalmente lo aplican satisfactoriamente a dos casos reales en ingeniería geotécnica.

En M.P. Saka (2003) [3.182], presenta una metodología para la optimización mediante algoritmos genéticos de pórticos inclinados a dos aguas permitiendo el refuerzo de la unión viga superior / pilar mediante cartelas en esquina.

S.S.S. Sakla y E. Elbeltagi (2003) [3.185] optimizan el diseño con mínimo peso y cumpliendo todas las restricciones exigidas en su norma nacional, de la configuración de correas con distancias homogéneas contemplando el caso de depósitos de nieve (cargas no uniformes) mediante algoritmos genéticos.

En E.M.R. Fairbairn (2004) [3.48], se plantea el diseño óptimo de estructuras de hormigón en masa, aplicado al muro de una presa. La optimización se realiza mediante un modelo termo-químico-mecánico de elementos finitos acoplado con algoritmos genéticos, considerando el coste de construcción como función objetivo.

X-T. Feng y C. Yang (2004) [3.53] presentan una metodología para la determinación de un modelo constitutivo no lineal del comportamiento estructural basado en la relación tensión-deformación, hibridando programación genética (localización de la ecuación del modelo) y algoritmos genéticos (localización de los coeficientes de la ecuación) y basado en el contraste con datos experimentales. Esto es aplicado a compuestos laminados, logrando aproximaciones válidas.

A.C.C. Lemonge y H.J.C. Barbosa (2004) [3.120] presentan un nuevo método para introducir restricciones sin dependencia de parámetros y dinámico con la evolución del algoritmo genético para funciones de penalización. Lo testean con casos test matemáticos y de estructuras de barras articuladas, tanto bidimensionales como tridimensionales, obteniendo resultados muy satisfactorios.

3.1.2 Optimización Estructural Multiobjetivo

En lo referente a la optimización multiobjetivo en el campo de la ingeniería estructural y civil, se realizará la distinción entre métodos de computación evolutiva y aquellos que no lo son, a los que se hará referencia como clásicos. Entre ellos se pueden citar las siguientes referencias:

3.1.2.1 Optimización Clásica

Entre las referencias de estados del arte acerca de la optimización multiobjetivo clásica aplicada a la ingeniería estructural-mecánica pueden destacarse entre otras: Stadler (1984) [3.198], Duckstein (1984) [3.46], Koski (1993) [3.107] y Coello (1996) [3.32].

La primera aplicación estructural multicriterio data de 1968 por Krokosky [3.113]. Se parte de una jerarquización de las preferencias cuantitativamente dadas por el diseñador, construyéndose a partir de ellas una matriz de preferencias, a partir de la cual se realiza una búsqueda aleatoria. Esta técnica, además de poco eficiente computacionalmente, era poco práctica pues requería exponer todos los criterios cuantitativamente, algo no siempre posible. De hecho, gran parte de las investigaciones pioneras realizadas en optimización multiobjetivo se centraron en diseño estructural, como puede verse en los trabajos de Baier (1977) [3.9], Leitmann (1977) [3.119], Stadler (1977) [3.199] (1978) [3.200], Koski (1979) (1980) [3.108] o Carmichael (1980) [3.109], tal como se cita en Marler y Arora (2004) [3.138].

Osyczka (1985) elabora un estudio del estado del arte mediante un conjunto de métodos clásicos en [3.160], de aplicación en problemas de ingeniería estructural.

S.S. Rao (1987) [3.176] realiza un estudio de estado del arte de los métodos que permiten originar una frontera de Pareto, relacionándolo con la teoría de juegos. Se aplica a dos casos estructurales sencillos.

Entre los trabajos más recientes se pueden encontrar los siguientes:

Dos casos test estructurales son resueltos mediante técnicas difusas de programación por metas multiobjetivo en Rao et al (1992) [3.177].

En S.S. Rao et al. (1992) [3.178] se presentan dos aproximaciones, el seccionamiento alfa y la formulación lambda, para la optimización multicriterio de sistemas difusos en ingeniería, aplicados a dos casos estructurales de barras articuladas (3 y 25 barras) para minimización de peso y deflexión en un punto, obteniéndose soluciones de compromiso.

R.V. Grandhi et al. (1993) [3.65] presentan un método multiobjetivo de optimización basado en técnicas compuestas generalizadas de escalado. El diseño de compromiso es seleccionado mediante criterios de decisión basados en fiabilidad. Se aplica a varios casos estructurales en aeronáutica, considerando esfuerzos, desplazamientos y frecuencia.

Lounis & Cohn [3.133] (1993) plantean la resolución de un problema bicriterio a través de la conversión de uno de los objetivos en restricción acotada y su posterior resolución mediante un algoritmo lagrangiano proyectado; ello se aplica a una estructura de puente de hormigón pretensado.

Una estrategia propuesta por Matsumoto et al. (1993) [3.141] de optimización clásica que considera una prioridad entre los objetivos a minimizar es aplicada al diseño estructural de motocicletas considerando la maximación de la rigidez torsional, la reducción del nivel de respuesta dinámica y la minimización del peso de dicha estructura. El método propuesto se basa en el lexicográfico, y la optimización se realiza a partir de la sucesiva superposición de optimizaciones monocriterio (no se obtienen frentes multicriterio).

Cohn & Lounis [3.36] (1994), presentan una optimización del diseño de puentes de hormigón, mediante sucesivas aplicaciones de programación no-lineal y técnicas de tamizado (sieve-search) en tres niveles: componentes, configuración estructural y finalmente el sistema en conjunto.

C.J. Shih y T.K. Lai (1994) [3.192] presentan un estudio comparativo de dos métodos diferentes basados en la combinación de pesos (pesado por pertenencia y pesado por objetivos) aplicados a la optimización de sistemas estructurales diñasos.

D. Lee (1997) [3.117] emplea el método de restricción (constraint method), para transformar un problema multiobjetivo en uno monoobjetivo y optimizar el diseño de vehículos marinos considerando dos funciones objetivo: coste de construcción y coste de operación, además de restricciones legales, medioambientales y técnicas. Se incorpora además una base de datos donde se almacenan consideraciones heurísticas del diseño de este tipo de embarcaciones.

En Y-P. Yang et al. (1996) [3.220] y (1997) [3.221] se emplea programación por metas (goal programming) y programación por compromiso (compromise programming), con el objeto de optimizar el diseño del montaje de la suspensión de discos duros considerando tanto el diseño activo como el pasivo, maximizando las frecuencias naturales del sistema y acoplando el optimizador a un programa de análisis de elementos finitos.

En (1998) [3.102], Khot propone el diseño óptimo simultáneo de estructura y un sistema de control basado en un regulador cuadrático lineal.

X.S. Li et al. (1999) presentan en [3.121] un algoritmo multiobjetivo que es extensión directa del método de los centroides de programación no-lineal. Éste se aplica a estructuras de gran tamaño.

Un análisis de sensibilidad, así como optimizaciones mediante programación lineal secuencial y programación cuadrática aplicadas a estructuras de nudos articulados esbeltas son presentados en X.Liu et al. (1999) [3.124].

En P. Tiwari et al. (1999) [3.205] encontramos una modelización de los costes de construcción mediante parámetros de diseño ingenieriles, además de los tradicionales de capital y mano de obra. Se emplea un algoritmo multiobjetivo que permite escoger entre varias funciones de coste de construcción y se aplica todo ello a resolver la función de producción de un habitáculo modular de 38 m³.

T.Y. Chen y C.C. Shieh (2000) [3.23] emplean teoría de decisión borrosa para optimizar topológicamente el diseño estructural modelizado mediante elementos finitos (emplean dos casos: triangulares y cuadrados) considerando dos criterios: minimizar la concordancia (compliance) de la estructura y maximizar el valor propio fundamental de la misma (para obtener una topología con buenas propiedades tanto frente a las cargas estáticas como al comportamiento dinámico). Como caso

test emplean una placa cargada puntualmente en su dirección perpendicular y con 640 elementos. El frente de Pareto se obtiene a partir de superponer sucesivas ejecuciones.

S. Hernández (1997) [3.82] y (2000) [3.81] ofrece un nuevo enfoque de optimización global además de un análisis de sensibilidad para optimización multiobjetivo, para optimización estructural.

S. Min et al. (2000) [3.149] emplean un método de pesos para optimizar la topología de distintas estructuras discretizadas mediante elementos finitos paralelepípedos. Los objetivos considerados son la minimización de la concordancia de la estructura y la maximización del valor propio medio de la misma. Lo aplican a varios casos estructurales bidimensionales y tridimensionales.

J.E. Kim y N. Sarigul-Klijn en (2001) [3.105] emplean el programa CONMIN [3.208] para el diseño multiobjetivo óptimo elastodinámico de alabes rotores considerando mínimo peso y máxima resistencia del material, con posterior minimización de la mínima vibración en una segunda etapa.

Q. Li et al. (2001) [3.123] realizan la optimización topológica de elementos estructurales modelizados mediante elementos finitos considerando simultáneamente la minimización del máximo esfuerzo y la maximización de la rigidez y un proceso de optimización progresivo con flujo de material de las zonas menos óptimas a las más óptimas. El método empleado es el ESO [3.218], y se aplica a casos test de hasta 24 x 40 elementos.

W.H. Zhang et al. (2001) [3.228] presentan un método de los pesos ponderados mejorado al que introducen una formulación multifrontera (MBR) empleando métodos de programación monotónica de aproximación convexa. Se aplica a varios ejemplos test de placas modelizadas mediante elementos finitos con minimización simultánea de peso y máxima tensión, o de desplazamiento y máxima tensión.

W. Zhang y H. Yang (2001) [3.229] presentan un estudio del método de los pesos aplicado a varios casos estructurales: estructura de nudos articulados de diez barras y dos casos bidimensionales, permitiendo la minimización del peso estructural y del desplazamiento de un nodo. Determinan un intervalo de pesos en el cual la solución óptima no varía si las restricciones se hacen activas, permitiendo una adecuada selección de combinación de pesos que determinen varios puntos del frente.

En [3.135] S. Marburg (2002), presenta un completo estado del arte acerca de optimización estructural-acústica, comprendiendo aplicaciones tanto académicas como de casos reales industriales, así como aspectos de optimización multiobjetivo.

K.C. Sarma y H. Adeli (2002) [3.188] presentan una metodología de optimización de estructuras orientada al coste total de la estructura a lo largo de todo su ciclo de vida, por lo que proponen considerar como variables a minimizar: perfiles comercialmente asequibles de menor coste, perfiles comercialmente asequibles de menor peso, número de tipos de perfiles distintos y perfiles comercialmente asequibles de menor perímetro de sección. La optimización la realizan mediante una definición difusa de las funciones objetivo, combinándolo con un procedimiento max-min; todo ello encaminado a reducir la función a optimizar a una monoobjetivo.

G.P. Steven et al. (2002) [3.201] hacen uso del método de los pesos para optimizar topológicamente mediante el método estructural evolutivo (ESO) [3.218] maximizando la rigidez y minimizando la máxima tensión en elementos estructurales bidimensionales mediante elementos finitos.

W.H. Zhang (2003) [3.226] presenta una mejora del método de programación por compromiso, al que complementa con una formulación multifrontera. Se aplica después satisfactoriamente a la

optimización de varios problemas estructurales, tanto de estructuras de barras articuladas, como de placas modelizadas con elementos finitos.

W.H. Zhang (2003) [3.227] presenta un nuevo método de análisis de sensibilidad del frente de óptimos de Pareto aplicado a varios casos test, matemáticos y estructurales (estructura de barras articuladas tridimensional de 25 barras).

3.1.2.2 Optimización Evolutiva

La primera aplicación multiobjetivo mediante algoritmos genéticos, parece ser de P. Hajela y C.Y. Lin (1992) [3.74] realizan una minimización simultánea de peso, esfuerzos y desplazamientos nodales, en estructura de barras de nudos articulados, mediante el uso de un algoritmo genético cuya función objetivo es compuesta por ponderación de pesos de las tres funciones a minimizar (aproximación ponderada min-max), con el objetivo de generar un frente de Pareto en una única ejecución. Se aplica a estructuras desde 10 hasta 72 barras articuladas.

S.S. Rao (1993) [3.175] emplea un algoritmo genético en combinación con teoría de juegos cooperativa, para la optimización multiobjetivo de estructuras de barras de nudos articulados (una estructura de dos aguas y otra de seis aguas) considerando la minimización del peso, la minimización de la deflexión y maximización de la frecuencia fundamental de vibración.

A.D. Belegundu et al. (1994) [3.11] utilizan un algoritmo genético multiobjetivo donde los individuos dominados y los no viables son eliminados y sustituidos por otros generados aleatoriamente. Con esta estrategia optimizan el diseño de alabes de turbomáquinas considerando como funciones objetivo la minimización de la amplitud resonante torsional y de las tensiones térmicas residuales. También es aplicada esta estrategia a estructuras compuestas laminadas.

A.K. Dhingra y B.H. Lee (1994) [3.44][3.45] utilizan algoritmos genéticos mediante un modelo de negociación basado en una función de utilidad, para la optimización mono- y multi-objetivo de estructuras de barras de nudos articulados (una estructura de veinticinco barras y otra de dos aguas) considerando la minimización del peso, la minimización de la deflexión y maximización de la frecuencia fundamental de vibración. Los resultados se comparan satisfactoriamente con el método 'branch & bound' (aplicado a cada criterio por separado).

E. Sandgren (1994) [3.186] emplea un algoritmo genético acoplado con programación por metas (goal programming) para optimizar estructuras articuladas de barras planas y mecanismos planos, considerando en distintos ejemplos test (estructuras articuladas de tres a diez barras) varios objetivos a optimizar (se consideran entre dos a nueve objetivos).

A.K. Dhingra y B.H. Lee (1995) [3.43] emplean una aproximación cooperativa basada en la teoría de juegos acoplada con algoritmos genéticos para la optimización combinada de control y de estructuras espaciales flexibles, considerando como criterios, el peso estructural, la energía del sistema, el índice de estabilidad y el amortiguamiento del controlador activo. El problema de optimización, de variables discretas y continuas es resuelto satisfactoriamente mejorando los resultados proporcionados por técnicas de gradiente clásicas.

J.S. Gero y S.J. Louis (1995) [3.61] optimizan el diseño de sección de una viga considerando como objetivos la minimización del perímetro y la maximización del momento de inercia de la misma.

Utilizan un algoritmo genético con criterio de selección basado en la no-dominancia de Pareto y elitismo.

A. Osyczka y S. Kundu (1995) [3.162] presentan un método de optimización multicriterio a partir de un algoritmo genético, donde la integración de los múltiples objetivos en una única función permite la optimización multicriterio. Los distintos valores no-dominados obtenidos se van almacenando formando un archivo de soluciones no-dominadas (elitismo).

M. Galante (1996) [3.57] realiza la optimización de estructuras de barras articuladas bi- y tri-dimensionales incluyendo el efecto de pandeo. Introduciendo el operador renacimiento que permite aumentar la eficacia y precisión del diseño óptimo final en el problema de peso mínimo restringido. Igualmente propone la optimización multicriterio considerando como criterio adicional al del mínimo peso restringido, la minimización del número de distintos tipos de perfiles utilizados, realizándola aplicando un criterio de pesos ponderados. Propone la aplicación de un algoritmo genético empleando el criterio de Pareto como tarea futura a realizar.

S. Kundu (1996) [3.114] utiliza un algoritmo genético con programación por compromiso (compromise programming) para el diseño multicriterio de un sistema de control estructural de aislamiento de vibraciones sísmicas. Se mantiene un conjunto de soluciones no-dominadas en cada generación (elitismo).

D.R. Wallace et al. (1996) [3.211] emplean un algoritmo genético con una función no lineal de agregación para resolver problemas de diseño multiobjetivo, entre ellos el diseño de una estructura de barras de nudos articulados considerando como objetivos: tensiones, pandeo, coste y relaciones de diámetro de barras.

F.Y. Cheng y D. Li (1997) [3.25] y (1998) [3.24] presentan un algoritmo genético multiobjetivo hibridizado con un método de función de penalización con lógica-difusa, aplicado al diseño estructural y de control integrado de estructuras de barras de nudos articulados (hasta 72 barras). Emplean elitismo en el algoritmo multiobjetivo, obteniendo resultados que mejoran los de los métodos clásicos.

En [3.225] C.H. Yeh et al. (1997) proponen una incorporación de elitismo al MOGA, para la minimización del coste del sistema y la capacidad de parada del nivel de líneas divisorias de cauce para aguas de tormenta y aplicado a un cauce en Pazam, sur de Taiwan.

W.A. Crossley et al. (1998) [3.39] emplean un algoritmo genético 'de dos ramas' con selección por torneo para problemas de diseño multiobjetivo (criterios: minimización del peso y del desplazamiento vertical) y lo aplican a una estructura de nudos articulados de 10 barras comparando los resultados con métodos monoobjetivo de programación lineal de forma satisfactoria.

X. Liu et al. (1998) [3.125] emplean una función de agregación mediante pesos con un algoritmo genético para optimizar la disposición y localización de un sistema de control integrado de una estructura articulada de cuarenta y cinco barras con cuatro actuadores, considerando restricciones de peso de la estructura, estabilidad y valores propios. Los objetivos a optimizar son el costo, la robustez y la controlabilidad modal del sistema.

R.J. Balling et al. (1999) [3.10] realizan la planificación urbana y de transporte mediante algoritmos genéticos considerando como objetivos el tiempo de viaje, el coste y transbordos. Introducen también una función objetivo alternativa en optimización multicriterio: la función objetivo MaxiMin, con la que se consiguen resultados muy satisfactorios.

C.W. Feng et al. (1999) [3.52] abordan el problema de planificación de proyectos de construcción desde la óptica del compromiso tiempo-coste, tomando como objetivos la minimización de la duración del proyecto y el coste del mismo. El desempeño del algoritmo genético con ordenación no-dominada es muy satisfactorio.

Y.J. Kim y J. Ghaboussi (1999) [3.104] optimizan un controlador de estructura mediante un algoritmo genético con aproximación agregativa (la función objetivo total es producto de las distintas funciones objetivos). Los objetivos perseguidos son relativos a la aceleración de pico, desplazamientos de pico y aceleraciones y desplazamientos RMS (root-mean-square). Los resultados son comparados con los obtenidos con un control muestral óptimo estándar, mejorando su desempeño.

S. Narayanan y S. Azarm (1999) [3.154] emplean un MOGA (multiobjective genetic algorithm) modificado, al que se ha incorporado una norma como criterio de convergencia del algoritmo, así como detección de la uniformidad de dispersión a lo largo del frente; y donde además únicamente en las soluciones no dominadas son evaluadas las constantes. Se compara con el MOGA estándar satisfactoriamente aplicándolo al diseño de una estructura de barras articuladas (minimizando volumen y esfuerzo máximo) y al diseño de una plataforma vibratoria (minimizando coste y frecuencia fundamental de la plataforma).

F.Y. Cheng et al. (2000) [3.26] proponen un algoritmo evolutivo multiobjetivo basado en el NPGA, para el diseño sísmico de estructuras, evaluando el tipo de estructura (acero, mixta u hormigón) teniendo en consideración el peso, coste estructural y la energía sísmica.

En C.A. Coello y A.D. Christiansen (2000) [3.33] se aplica una minimización simultánea de peso, esfuerzos y desplazamientos nodales, en estructura de barras de nudos articulados, mediante el uso sucesivo de algoritmo genético cuya función objetivo es compuesta por ponderación de pesos de las tres funciones a minimizar (algoritmo genético con aproximación ponderada min-max), aplicado en estructuras de barras articuladas de tamaño 25 y 200. Se compara con otros algoritmos multiobjetivo como: MOGA, VEGA, NSGA y otras aproximaciones clásicas, resultando la propuesta de los autores la más cercana al frente ideal.

V. Coverstone-Carroll et al. (2000) [3.38] hibridizan el algoritmo NSGA con un algoritmo de gradiente para la optimización de trayectorias interplanetarias, considerando conjuntamente la minimización del tiempo de viaje y la maximización de la masa transferida en el mismo. Se aplica a dos rutas, Tierra-Marte y Tierra-Mercurio.

A. Kurapati y S. Azarm (2000) [3.115] optimizan el diseño de una caja de cambios considerando la minimización del peso de la caja reductora y simultáneamente el esfuerzo en uno de los ejes. El algoritmo optimizador empleado es el MOGA acoplado con un sistema inmune artificial como coordinador.

Una recopilación del estado del arte hasta el año 2000 de los algoritmos evolutivos multiobjetivo se puede encontrar en Coello [3.31].

I. Ono et al. (2000) [3.158] optimiza el diseño de lentes simultáneamente minimizando la distorsión y maximizando la resolución de una lente de tres elementos, mediante algoritmos evolutivos.

T. Furukawa (2001) [3.55] propone un método multiobjetivo evolutivo (MCEA) que incorpora elitismo y lo aplica a la identificación de parámetros de modelos de manera más eficaz que los

métodos monoobjetivo con los que lo compara, aplicándolo finalmente a un modelo viscoplástico de materiales (relación tensión-deformación) basado en el modelo de Chaboche.

S. Khajepour y D. Grierson (2001) [3.101] emplean un algoritmo genético con vector de aproximación objetivo, para el diseño óptimo conceptual de edificios de oficinas rectangulares, considerando como criterios la minimización del coste de inversión, minimización del coste operativo y maximización de los ingresos recibidos de explotación del edificio.

A. Osyczka y S. Krenich (2001) [3.161] realizan el diseño óptimo de una viga con sección variable, considerando la minimización del volumen minimizando el desplazamiento de la viga bajo una carga determinada, todo ello mediante un algoritmo genético con criterio de no-dominancia de Pareto.

J. Wu y S. Azarm (2001) [3.216] emplean un algoritmo genético con no-dominancia de Pareto para el diseño óptimo de una plataforma vibratoria considerando la maximización de la frecuencia fundamental de la viga y la minimización del costo de la misma. Utilizan una irnovadora técnica de manejo de las restricciones, que comparan con una clásica función de penalización de manera satisfactoria.

D.W. Fanjoy y W.A. Crossley (2002) [3.49][3.50] presentan diseño topológico de secciones planas mediante el acoplamiento de algoritmos genéticos y elementos finitos, considerando flexión, torsión y cargas combinadas. Destacan la importancia de elección adecuada del cruce y conectividades estructurales. Se efectúa una optimización multiobjetivo sobre las mismas.

Una estrategia evolutiva (evolution strategy) multiobjetivo, Papadrakakis (2002) [3.164] es empleada para el diseño de estructuras de barras con cargas estáticas y dinámicas, partiendo de ciertos acelerogramas tipo y sus soluciones son comparadas frente a los métodos de diseño convencionales dados por los códigos sísmicos.

R.F. Coelho et al. (2003) [3.29] presentan un método multiobjetivo que trata de tener en cuenta las preferencias del diseñador a priori, integrando en un algoritmo genético monocriterio el sistema de designación de preferencias basado en PROMETHEE [3.15]. Se aplica a problemas test e industriales, como el diseño de una válvula de purga del vehículo espacial europeo Ariane5, aunque los resultados reflejan que la eficacia de este método es menor que la de los métodos a posteriori evolutivos (NSGAI, SPEA2, etc.) en distribución a lo largo del frente y cercanía al mismo.

En [3.35] (2003) Coello propone la aplicación de un algoritmo evolutivo multiobjetivo micro-GA aplicado al diseño de una viga en voladizo con carga puntual en su extremo para minimizar su deflexión y peso simultáneamente.

3.2 Definición del problema estructural

3.2.1 Introducción.

Para resolver el problema de optimización estructural es preciso obtener la información necesaria sobre las condiciones de trabajo de la estructura abordada, en el caso tratado, en un pórtico plano metálico. Esto se efectuará mediante un programa de cálculo matricial.

La posibilidad de establecer el modelo de una estructura mediante barras es la condición para que ésta pueda ser analizada mediante el cálculo matricial de estructuras. Éste puede ser considerado el método más extendido actualmente para el cálculo de estructuras, gracias al desarrollo de los ordenadores, que permiten la simulación de ecuaciones y la obtención de soluciones en un espacio de tiempo relativamente breve.

En cualquier problema estructural existen tres tipos de condiciones que se han de satisfacer, y de las que se dispone para la resolución del mismo. Son las ecuaciones de equilibrio -dentro de la estructura y con las cargas externas-, ecuaciones de compatibilidad -entre los elementos de la estructura y con los elementos de contorno- y la ley de comportamiento. Para el cálculo de estructuras hiperestáticas, se han de emplear conjuntamente estas condiciones para su resolución, mediante el método de equilibrio, también llamado de los desplazamientos o de rigidez.

En el método de rigidez se aplican sucesivamente ley de comportamiento, ley de compatibilidad y ley de equilibrio para la resolución de los desplazamientos de los nudos. A partir de las leyes de compatibilidad y comportamiento, a posteriori se obtienen deformaciones y esfuerzos. Los métodos de rigidez, dado que originan gran número de ecuaciones, no fueron usados mientras que el cálculo fue manual. Para el cálculo mediante ordenador, el método de rigidez resulta el más adecuado al ser el más sistematizable y mecanizable, no teniendo que realizarse elecciones dependientes de la estructura particular de estudio y sí definirse desde un punto de vista más general. Particularmente, el método directo de rigidez [3.207][3.213][3.139] es el universalmente empleado para el cálculo de estructuras de barras por su sistematicidad y por ser conceptualmente sencillo [3.51]. Este es el método que ha sido implementado para obtener la información necesaria en la resolución de la función objetivo de masa restringida en el algoritmo genético.

3.2.2 Etapas de Cálculo del Método Directo de la Rigidez

Las etapas esenciales en el análisis de una estructura por el método directo de rigidez se resumen a continuación:

- Previamente, se debe seleccionar un modelo matemático apropiado que permita describir el comportamiento de la estructura, así como definir con detalle las propiedades mecánicas de los materiales y el carácter de la deformación. En nuestro caso se tratarán solamente problemas de equilibrio estático de estructuras con pequeños desplazamientos y comportamiento elástico lineal de los materiales (considerándose a posteriori efecto de pandeo).
- Discretizar la estructura en un conjunto de elementos de barras.
- Para cada elemento, calcular su matriz de rigidez K y el vector de fuerzas nodales equivalentes \bar{f} .

- Ensamblar las matrices de rigidez elementales en la matriz de rigidez de toda la estructura, global; asimismo ensamblar los vectores de fuerzas nodales equivalentes en el vector de fuerzas de toda la estructura f (o matriz F caso de existir varios casos de carga distintos).
- Tras imponer las condiciones de contorno, calcular los desplazamientos nodales resolviendo el sistema de ecuaciones de la discretización $K U = F$.
- Calcular para cada barra otras magnitudes de interés, como deformaciones, tensiones o reacciones de la misma.

3.2.3 La Matriz de Rigidez. Características.

Aplicando el Cálculo Matricial se puede disponer de un sistema de ecuaciones que relacione las cargas en nudos con los desplazamientos en los nudos. Puesto que en los pórticos planos cada nudo posee tres grados de libertad (desplazamiento horizontal, desplazamiento vertical y giro), el número total de ecuaciones del sistema será de $3n$ donde n es el número total de nudos de la estructura (variable que depende de la discretización que se adopte).

La matriz de rigidez relaciona por tanto las fuerzas aplicadas, según los diferentes grados de libertad, con los desplazamientos de esos mismos grados de libertad :

$$\begin{matrix}
 \text{índice} & | & & & \bullet K_m & \Lambda & \backslash \\
 & & A_{21} & A_{22} & K_{2m} & & \\
 & & & & & &
 \end{matrix} \quad (3.1)$$

La matriz de rigidez del sistema es independiente de las ligaduras externas, es decir, de la clase y disposición de los apoyos, que afectarán a los resultados en tanto en cuanto coartan los movimientos de los nudos asociados a éstos.

Las características de la matriz de rigidez se describen a continuación:

- *Singularidad:* Ello implica que aquellas estructuras que carezcan de apoyos, serán cinemáticamente inestables.
- *Simetría:* K es una matriz simétrica, es decir $K_{ij} = K_{ji}$. La fuerza producida en el grado de libertad j al mover una unidad el i , manteniendo a cero todos los demás, es igual que la que se produciría en i al desplazar j en las mismas condiciones. Ello es consecuencia directa del teorema de la reciprocidad o de Maxwell-Betti.
- *En banda:* Es posible concentrar sobre una banda en torno a la diagonal principal de la matriz todos los elementos no nulos mediante una adecuada numeración de la misma, debido a que todos los elementos correspondientes a los grados de libertad no conectados entre sí serán nulos (nudos no conectados por alguna barra). Esto permite el almacenamiento en memoria sólo de dicha banda central (ancho de banda).
- *Definida positiva:* Físicamente esto implica que la estructura es un conjunto indeformable, sin mecanismos internos y unida a su entorno por apoyos suficientes para impedir movimientos como sólido rígido de la estructura. Matemáticamente conlleva que el menor autovalor de la misma será mayor que cero.

- *Diagonalmente dominante*: Esta propiedad no se cumple estrictamente, aunque sí suele ser así. Todos los elementos de la diagonal principal son positivos e iguales o mayores a la suma del resto de los elementos de su fila o columna. Junto a la propiedad de definida positiva permite el buen condicionamiento del sistema y por consiguiente su resolución de modo sencillo.

3.2.4 La Matriz de Rigidez. Ensamblaje y resolución.

La determinación de la matriz de rigidez de la estructura implica el ensamblaje de las matrices de rigidez elementales (de cada barra). La matriz de rigidez elemental en ejes locales para el caso de pórtico plano con barras de sección constante, es la mostrada en la ecuación (3.2).

$$\begin{matrix}
 \begin{matrix} EA \\ L \end{matrix} & 0 & 0 & \begin{matrix} EA \\ L \end{matrix} & 0 & 0 \\
 0 & \begin{matrix} UEI \\ L^3 \end{matrix} & \begin{matrix} 6EI \\ L^2 \end{matrix} & 0 & \begin{matrix} UEI \\ L^3 \end{matrix} & \begin{matrix} 6EI \\ L^2 \end{matrix} \\
 0 & \begin{matrix} 6EI \\ L^2 \end{matrix} & \begin{matrix} AEI \\ L \end{matrix} & 0 & \begin{matrix} 6EI \\ L^2 \end{matrix} & \begin{matrix} 2EI \\ L \end{matrix} \\
 \begin{matrix} EA \\ L \end{matrix} & 0 & 0 & \begin{matrix} EA \\ L \end{matrix} & 0 & 0 \\
 0 & \begin{matrix} UEI \\ L^3 \end{matrix} & \begin{matrix} 6EI \\ L^2 \end{matrix} & 0 & \begin{matrix} UEI \\ L^3 \end{matrix} & \begin{matrix} 6EI \\ L^2 \end{matrix} \\
 0 & \begin{matrix} 6EI \\ L^2 \end{matrix} & \begin{matrix} 2EI \\ L \end{matrix} & 0 & \begin{matrix} 6EI \\ L^2 \end{matrix} & \begin{matrix} 4EI \\ L \end{matrix}
 \end{matrix} \tag{3.2}$$

donde los coeficientes c_i , son dependientes de las condiciones de unión de la barra. Así pues, dichos coeficientes adoptan los siguientes valores en función de esas condiciones de unión:

Barra biarticulada : $c_1 = c_2 = c_3 = c_4 = c_5 = c_6 = c_7 = c_8 = c_9 = c_{10} = 0.0$

Barra biempotrada : $c_1 = c_2 = c_3 = c_4 = c_5 = c_6 = c_7 = c_8 = c_9 = c_{10} = 1.0$

Barra articulada en primer nodo :

$$\begin{aligned}
 c_1 = c_3 = c_8 &= 0.25 \\
 c_4 = c_9 &= 0.5 \\
 c_{10} &= 0.75 \\
 c_2 = c_5 = c_6 = c_7 &= 0.0
 \end{aligned}$$

Barra articulada en segundo nodo :

$$\begin{aligned}
 c_1 = c_3 = c_8 &= 0.25 \\
 c_2 = c_6 &= 0.5 \\
 c_5 &= 0.75 \\
 c_4 = c_7 = c_9 = c_{10} &= 0.0
 \end{aligned}$$

y aplicando la transformación de coordenadas, obtenemos los términos de la matriz elemental de rigidez en coordenadas globales:

Se señala a continuación el sistema de almacenamiento de la matriz de rigidez. Con el fin de aprovechar sus propiedades de simetría y banda, las características del almacenamiento son las siguientes:

- Se han eliminado las coacciones de apoyo, empleándose por tanto la matriz de rigidez efectiva. Esta matriz efectiva es de menores dimensiones que la total, siendo de (número de grados de libertad efectivos x número de grados de libertad efectivos) fi-ente a (número de grados de libertad totales x número de grados de libertad totales).
- Con el fin de almacenar un número más pequeño de términos de la matriz de rigidez únicamente se almacenan los pertenecientes a la banda de la matriz efectiva. Los elementos quedan ordenados en líneas paralelas a la diagonal principal, requiriéndose por consiguiente un número de términos que será de número de grados de libertad restringidos x semiancho de banda. Se almacena pues en memoria los coeficientes situados por encima de la diagonal principal, incluyendo esta diagonal y teniendo como ancho el semiancho de banda. Se almacena de forma que cada diagonal de la semibanda ocupa una columna de la matriz K de rigidez. Así pues, la primera columna de la misma quedará constituida por la diagonal principal, y así sucesivamente.

Se destaca igualmente que el semiancho de banda de la matriz efectiva no es necesariamente el mismo que el obtenido a partir de la matriz de rigidez total, haciéndose necesario modificar el algoritmo de cálculo del mismo [3.5].

3.2.6 Optimización del Ancho de Banda. Renumeración nodal.

Se deben destacar múltiples ventajas obtenidas al optimizar el ancho de banda.

- Reducción de *memoria* requerida para el almacenamiento de la matriz de rigidez.
- Reducción del *tiempo de operación* en la resolución del sistema de ecuaciones.
- Reducción de los *errores o imprecisiones* por truncamiento en la resolución del sistema al verse igualmente reducido el número de operaciones necesarias para llevarlo a cabo.

Por tanto, la minimización del ancho de banda podrá ser efectuada mediante la minimización de la diferencia de numeración nodal b , factor que lo determina implícitamente. Puesto que en la práctica conviene otorgar libertad de asignación de ordenación nodal al diseñador, será conveniente el disponer de un algoritmo que permita optimizar el ancho de banda independientemente de la ordenación nodal adoptada. Por tanto, se implementa un algoritmo de reordenación nodal, cuyos efectos serán a nivel de cálculo interno exclusivamente, puesto que tanto la lectura de datos de la estructura tratada como los resultados finales, se corresponderán con la numeración nodal dada por el proyectista.

Cuthill y McKee propusieron en 1969 un método de reordenación denominado algoritmo de reordenación de Cuthill Mc-Kee [3.40]. Aquí se implementa el algoritmo inverso, que permite una mejora en términos de almacenamiento y complejidad computacional, reportada por Liu y Sherman [3.131], además de una frecuente reducción en la envolvente, reportada por George y Liu [3.60].

Se debe destacar especialmente que el ahorro en el tiempo de cálculo es objetivo primordial para una eficacia aceptable de la optimización, puesto que el algoritmo genético requiere miles de evaluaciones funcionales (es decir, miles de resoluciones de sistemas lineales de ecuaciones). Así pues este ahorro obtenido es aprovechable en todas y cada una de ellas. Basta sin embargo un único cálculo de la

reordenación nodal para obtener tan sustancioso beneficio en todas las evaluaciones funcionales, tiempo despreciable de computación frente al ahorro obtenido.

3.2.7 Resolución Matemática del Sistema de Ecuaciones

La resolución del sistema de ecuaciones estructural, debe ser abordada desde el punto de vista de resolución de un sistema algebraico de ecuaciones:

$$K U = F \quad (3.3)$$

donde K es la matriz de rigidez efectiva, F es la matriz de casos de carga almacenados por columnas y U la matriz solución de desplazamientos para cada uno de los casos de carga evaluados. Éstos serán posteriormente combinados entre sí con los correspondientes coeficientes para determinar las hipótesis de carga.

En los primeros años 60 los métodos iterativos fueron populares para la resolución de este sistema, limitado por la capacidad de las computadoras de la época, alejada respecto de lo que suponemos medio en la actualidad. Su ventaja reside en requerir menor memoria, al bastar con almacenar la ecuación que se trata en cada momento ; por ello su programación es más simple con una memoria limitada. Sus inconvenientes residen en la finalización de la convergencia y la proporcionalidad del tiempo de cálculo en el caso de existir varios casos de carga, como el que nos ocupa. Los métodos directos permiten afrontar esto con un incremento de tiempo muy reducido. Citemos una frase de Y. Saad [3.181]: 'Los métodos directos también han sido preferidos para cierto tipo de problemas reconocidos de poseer matrices muy bien condicionadas, como por ejemplo en aplicaciones de ingeniería estructural'.

Se adoptará pues un método directo, que, una vez expuestas las características de la matriz de rigidez con anterioridad, las tenga en cuenta a la hora de la resolución del sistema, valiéndonos de ellas para incrementar la eficacia de la misma.

Es posible aprovechar las características de simetría y en banda acopladas al método de eliminación de Gauss, así pues, se reduce el número de operaciones requeridas a la mitad caso de ser simétrica la matriz, pasando de ser n^3 a ser n^2 , respectivamente.

Además, si el ancho de banda es ab , el espacio necesario para almacenar la matriz se reduce en la proporción ab/n , siendo el número de operaciones en la triangulación, fase más larga del proceso, reducido en la proporción (ab/n) . El número de operaciones pasa a ser por tanto del orden de $n^2(ab)$, con la consiguiente reducción de tiempo de operación y errores de truncamiento. Esta posibilidad de aprovechamiento de las características de simetría y en banda se basa en las siguientes propiedades del proceso de triangulación del método de Gauss [3.150]:

Al ser la matriz de rigidez simétrica, tras restar una fila i de las siguientes, la parte no triangulada (filas $i+1$ a la n) mantiene su simetría, mientras que la parte ya triangulada (filas 1 a la i) está formada por i filas que se conservan e i columnas de ceros que no es necesario conservar.

Además, por tener nulos la matriz de rigidez los elementos salvo los situados en la banda alrededor de la diagonal principal, los elementos nulos fuera de esta banda durante la triangulación siguen permaneciendo nulos.

El algoritmo de resolución descrito implementado en lenguaje C se expone a continuación, pudiéndose observar cómo dentro del mismo bucle de resolución son tratados sucesivamente todos los casos de

carga posibles, evitando así sucesivas resoluciones de sistemas con el consiguiente costo computacional adicional.

```

for(i=0;i<(nec-1);i++) /* Triangulación */
{
  for(j=1;j<sb;j++)
  {
    nfi = i+ j ;
    if ( (nfi <= nec) && ( *(*(ktl+i)+j) != 0.0)
    {
      /* f : factor por el que se multiplica
      la fila i antes de restarla */
      f= *(*(ktl+i)+j) / *(*(ktl+i));
      k=0;
      for(m = j; m<sb; m++)
      {
        *(*(ktl+nfi)+k) -= f* *(*(ktl+i)+m);
        k++;
      }
      for(nn=0;nn<ncas+2;nn++)
      {
        *(*(v+nfi)+nn) -= f* *(*(v+i)+nn);
      }
    }
  }
}
for(i=(nec-1); i>=0; i--) /* Sustitución hacia atrás
calculo de cada colimna de incógnitas */
{
  for(nn=0;nn<ncas+2;nn++)
  {
    s=0;
    for(j=1;j<sb;j++)
    {
      nfi = i+ j ;
      if(nfi <= nec) s += *(*(ktl+i)+j) * *(*(v+nfi)+nn);
    }
    *(*(u+i)+nn)= ( *(*(v+i)+nn) -s) / *(*(ktl+i));
  }
}

```

Resolución lineal de sistema de ecuaciones

3.2.8 Determinación de esfuerzos y tensiones a lo largo de todas las barras

La resolución de la ecuación matricial efectiva, nos otorga los desplazamientos nodales en ejes generales. A partir de ellos, el análisis de esfuerzos en las barras se realiza del siguiente modo:

1. Mediante los desplazamientos obtenidos del cálculo matricial pasados a coordenadas locales, se determinan los esfuerzos en extremos de barras considerando el tipo de enlace. Estos esfuerzos se interpolan linealmente a lo largo de la barra.
2. Se le suman a los esfuerzos del apartado 1, los obtenidos mediante resistencia de materiales a partir de las cargas sobre la barra considerando su tipo de enlace.

Esta simia nos proporciona los esfuerzos (axiles, cortantes y flectores) a lo largo de todo el elemento, en tantas secciones como se desee.

Una vez obtenidos los esfuerzos para cada una de las secciones intermedias en que se discretiza cada uno de los elementos o barras de la estructura a analizar, la determinación de las tensiones en ellas es inmediata.

Se considerarán dos puntos en cada sección de cálculo, los más desfavorables cara a las tensiones. Éstos serán el punto inferior del ala en su intersección con el plano medio longitudinal del alma (1) y la intersección de ese mismo plano con el plano tangente inferior del ala (2). Estos dos puntos quedan representados en la figura 3.1 mediante sendas circunferencias color bermellón, supuestos los perfiles a aplicar en forma de H (series de perfiles en norma EA-95: IPN, IPE, HEB, HEA, HEM):



v[^]

Figura 3.1. Sección de perfil HEB con puntos de control de tensión.

La sección de un pórtico plano estará sometida a un esfuerzo axil N , un esfuerzo cortante V y un momento flector M . La relación entre la altura del alma del perfil h_l y su altura total h será h_l/h . Las tensiones normal G^{\wedge} y tangencial a^{\wedge} originadas se evalúan mediante las siguientes ecuaciones:

$$\begin{matrix} N \\ \text{xxn} \end{matrix} \quad A \quad (3.4)$$

$$\begin{matrix} M \\ \text{xxn} \end{matrix} \quad W \quad (3.5)$$

$$\frac{V}{A_{alma}}$$

siendo A el área de la sección, W el módulo de rigidez y A_{alma} el área del alma de la sección :

Punto 1 :

$$\sigma_{x,x} = \frac{M}{W} + \frac{V}{A} \tag{3-7}$$

Puesto que en el punto 1 la tensión tangencial es nula, sólo existirá tensión normal.

Punto 2 :

$$\sigma_{x,x} = \frac{M}{W} + \frac{V}{A} \tag{3.8}$$

$$\sigma_{x,x} = \frac{M}{W} + \frac{V}{A} \tag{3.9}$$

En el punto 2 debe reducirse en el factor h_l/h la tensión debida al momento flector, puesto que W, modulo de rigidez se corresponde con el punto inferior del perfil y debe reducir su valor en la citada proporción.

Para cada una de las secciones se evalúan ambos puntos y para cada barra se almacena el mayor valor entre ambos y entre todas las secciones de la misma (se consideran 20 secciones por barra), que será el utilizado a la hora de proceder a la evaluación de las posibles penalizaciones por sobretensión en el cálculo a resistencia.

3.2.9 Efecto de Pandeo

Dado que se persigue un método de optimización en el cálculo de un pórtico plano genérico, la estructura final obtenida es previsible que se adapte con gran afinidad a los valores límite de las restricciones impuestas. Por tanto es de vital importancia que el modelo seguido en la resolución de la misma se adapte y acerque con suficiente precisión a los requerimientos reales para posibilitar una construcción factible sin colapsar, como podría ocurrir caso de no considerar el efecto de pandeo.

Con el fin de obtener las longitudes eficaces de los pilares de una estructura se requiere el análisis de su pandeo global. Este análisis es complicado, aún disponiendo de ordenadores personales [3.8], por lo que las normativas recurren a procedimientos aproximados.

El método de Julián y Lawrence (1959) es el recogido entre otras por la normativa española NBE-EA-95, la normativa francesa de construcción metálica CM-66 o por el Boston Building Code.

La formulación de esta teoría conduce a una ecuación trascendente resoluble mediante tanteos o numéricamente, o bien a un nomograma equivalente en función de las rigideces relativas de las barras. Éste variará en función de que el pórtico sea traslacional o intraslacional.

La norma española EA-95 plantea la siguiente ecuación para determinación de las rigideces relativas de estructuras compuestas de vigas y pilares, con las que se obtiene la p mediante la sustitución de las

mismas en ecuaciones equivalentes a los nomogramas antes citados, y que dependen de si el pórtico es o no traslacional (función de los vínculos del mismo).

Éstas se muestran a continuación. Para cada uno de los nudos de la barra tratada se determina según (3.10):

$$k = \frac{L_{izq}^3 + L_{dra}^3}{L_{izq}^3 + L_{dra}^3 + 3L_{si}^3} \quad (3.10)$$

siendo k el grado de empotramiento del nudo del pilar tratado, izq y dra los subíndices izquierdo y derecho correspondientes a la viga izquierda y derecha respecto del nudo, así como el subíndice si correspondiente al pilar inferior o superior según corresponda al pilar tratado. La ausencia de subíndice se corresponde con el propio pilar tratado.

Si la unión de un pilar al extremo considerado no es rígida será $k=0$, y $k=1$ si empotra en la cimentación.

En el caso de pandeo intraslacional, si la estructura es de nudos rígidos se aplica la ecuación (3.11):

$$B = \frac{3 - \sqrt{6(k + A_1) + 0.84A_1 - k}}{3 - (k + k^2) + 0.2A_1 - k^2} \quad (3.11)$$

En el caso de pandeo traslacional, se aplica la ecuación (3.12):

$$G = \frac{1.6 + 2.4(A_1 + A_2) + 1.1A_1 - k}{(A_1 + A_2) + 5.5A_1 - k} \quad (3.12)$$

todo ello, queda ampliado en la norma española EA-95.

Como generalización de esta determinación para el caso de pilares, se ha programado el cálculo de las P mediante la ecuación aproximativa (3.13), que puede valer como orientación, para determinar los grados de empotramiento de las barras y a partir de ellos, las P :

$$k = \frac{\sum_{j=1}^{NB} h_j^2 \sin^2 \alpha_j}{\sum_{j=1}^{NB} MB_j} \quad (3.13)$$

donde NB es el número de barras confluyentes en el nudo de cálculo y el ángulo α_j es el formado por la barra confluente y la que se está efectuando el cálculo de su coeficiente p . Para el caso de vigas y pilares perpendiculares, que es el que contempla la norma, equivale a la ecuación contenida en la misma. En el caso de que la viga tenga una inclinación distinta a la perpendicular, provoca

disminución de su término de rigidez relativa en el numerador (así, en el caso extremo de ser paralela, es decir, un pilar, no se considera en el numerador, tal como se establece en la norma).

También es posible la inclusión directa de valores del coeficiente (3 estimados por el proyectista para cada una de las barras caso de estimarse conveniente).

Una vez obtenida las p de cada una de las barras, se procede al cálculo de su esbeltez mecánica según (3.14):

$$\dot{A} = -i^3 I \quad (3.14)$$

Con ella se obtiene el coeficiente w a partir de las tablas existentes en la norma EA-95 (que han sido integradas en el programa). Este coeficiente w es ponderador de las tensiones normales y es equivalente a la reducción de la tensión máxima admisible experimentada por el acero con el incremento de la carga (la tensión admisible no se altera pero se aumenta la carga aplicada a través de este coeficiente en análoga proporción).

La comprobación a pandeo de los elementos se efectuará únicamente en el 40% de la sección central de las mismas a instancia de lo dictado en la norma española EA-95, siendo igualmente evaluada en los dos puntos antes descritos. Las comprobaciones son las siguientes, análogas a las efectuadas a resistencia:

$$w \quad (3.15)$$

$$\frac{M_{xx}}{W} \quad (3.16)$$

$$\frac{V_{xy}}{A_{alma}} \quad (3.17)$$

Punto 1 :

$$\sigma_{tot} = \sigma_{xx} + \sigma_{xxm} \quad (3.18)$$

Puesto que en el punto 1 la tensión tangencial es nula, sólo existirá tensión normal.

Punto 2:

$$\frac{m_{xx}}{I} + \sigma_{xxm} \quad (3.19)$$

$$(\sigma_{tot} = 4J + \sigma_{xxm}) \quad (3.20)$$

La tensión máxima así obtenida por barra es comparada con la máxima a resistencia y de ambas se escoge la mayor, que será la empleada en la evaluación de la restricción con la tensión última seleccionada por el proyectista.

3.3 Optimización del problema estructural

La optimización de pórticos metálicos de nudos rígidos en régimen elástico destaca en importancia por el gran número de construcciones que responden a esta tipología estructural, tanto en ingeniería industrial, civil o en arquitectura.

Se destaca como propiedad de los algoritmos genéticos su carácter de optimizadores globales, su robustez y capacidad de búsqueda en pos del mínimo global del problema, no deteniéndose necesariamente en los mínimos locales de los mismos, abundantes con frecuencia en el tipo de problema tratado aquí y garantizando cuando menos una alta calidad media de la población final.

La resolución del problema planteado, el cual es la optimización estructural multicriterio de pórticos planos metálicos mediante algoritmos evolutivos, requiere, como todo problema a abordar con esta técnica, la concreta y correcta definición acorde al mismo de: Las funciones objetivo a minimizar (en optimización multiobjetivo) y la codificación genotípica en el cromosoma (individuo o candidato a constituir la solución óptima).

- *Funciones Objetivo:* Las funciones objetivo engloban la información necesaria por el algoritmo genético para efectuar la búsqueda del óptimo individuo. Mide la adaptación de cada individuo al entorno, que no es otra que la aptitud de la solución descrita por el mismo. Por consiguiente, cada una de las funciones objetivo debe incluir todos aquellos aspectos requeridos a la solución deseada, debiendo ser su definición elaborada con cautela.
- *Codificación en cromosoma:* En el cromosoma se debe incluir la información que distinga una solución de otra, es decir, aquella información variable entre soluciones, que permita conjuntamente con el resto de parámetros fijos que definen el problema, su reconstrucción para evaluar su adaptación al medio. Además, esta información debe estar estructurada de modo tal que el intercambio entre dos cromosomas mediante el operador de cruce o bien la modificación sutil de la misma mediante mutación, produzca otro individuo de forma coherente. La población filial originada deberá ser por tanto consistente con la definición del problema y evaluable por la función objetivo.

3.4 Codificación de Soluciones Candidatas

En el problema que nos ocupa, la información que constituye y define el mismo se puede resumir en los siguientes puntos: Coordenadas nodales de la estructura; Conectividades entre barras; Características geométricas de la sección de la barra (área, momento de inercia, módulo de rigidez, radio de giro, área del alma, etc); Características de material/es de las barras; Vínculos de apoyos / asentamientos de ciertos nodos; Casos de carga de la estructura; Hipótesis de carga de la estructura; Peso de la estructura; Número de perfiles diferentes utilizados; Límites máximos tensionales, de esbeltez o desplazamiento de ciertos puntos estructurales admisibles.

El problema de optimización a resolver de modo genérico será el de localizar la estructura/s de mínimo peso restringido (de tal modo que todos los límites máximos de restricciones tensionales, de desplazamiento y pandeo impuestos no se vean sobrepasados, en cuyo caso la estructura no sería viable) conjuntamente con el mínimo número de tipos de perfiles diferentes. Considerando fijos el

resto de parámetros excepto las características de sección de las barras, el problema se ceñirá pues a encontrar aquellas geometrías de barra idóneas para una topología y cargas fijadas de la misma.

La codificación genética de cada uno de los individuos o soluciones candidatas, contendrá la información necesaria para la definición de las características geométricas citadas de cada una de las barras, que es aquella información variable que distingue una solución de otra.

Si se pretende asignar perfiles reales de la norma española EA-95, caso real de aplicación, bastará con elegir el conjunto de entre el que se quiere elegir, ordenarlo en función de alguno de los criterios de la sección (área, momento de inercia, etc.) y codificar en el cromosoma por cada una de las barras, el orden del perfil en la base de datos como número natural (optimización discreta). Así, por ejemplo, para una estructura de 8 barras (cada una supuesta una variable independiente), si cada una de ellas puede ser elegida por 16 perfiles distintos, el cromosoma contendrá : $16 = 2^4 \Rightarrow 4 \text{ bits/barra} * 8 \text{ barras} = 32 \text{ bits / estructura}$.

3.5 Funciones Objetivo

En nuestro caso, las funciones objetivo a minimizar son dos, el peso de la estructura sometido a una serie de restricciones tensionales, de desplazamientos y esbeltez, denominándolo por tanto peso restringido; y el número de distintos perfiles usados en la misma:

1. La *minimización del peso estructural restringido*
2. La *minimización del número de perfiles distintos*

3.5.1 Función Objetivo 1: Minimización del peso estructural restringido

La **función objetivo 1 (minimización del peso estructural restringido)** responde a la premisa de minimizar el coste de adquisición de materia prima de la estructura diseñada, puesto que en el caso de estructura metálica, este coste está en correlación directa con el peso total de la misma. Esta minimización de peso tiene como límite aquellas condiciones que hagan cumplir al pórtico diseñado su función: el servir de soporte a las cargas objeto de diseño sin colapsar o deformarse excesivamente. Esta función objetivo constituye un problema condicionado, pues estará sometida a una serie de restricciones, que para las solicitaciones de las múltiples hipótesis de carga previsible, serán las tensiones admisibles, los desplazamientos límite en ciertos puntos de la estructura, como nudos o eventualmente puntos centrales de vanos de las vigas, o limitaciones de esbeltez por el efecto del pandeo.

La existencia de restricciones puede ser abordada de diversos modos en los algoritmos evolutivos [3.30][3.144][3.145]:

- *Método de función penalty*. Este método transforma un problema sometido a restricciones (condicionado) en otro sin ellas (no condicionado) incorporándolas en caso de violarse, como un costo añadido a la función objetivo (en este caso el peso). Se podría evaluar un individuo, y dado el caso que violara una o más restricciones, descartarlo puesto que al ser la solución representada por él no viable, su función objetivo no tiene lugar. Sin embargo, puede que se originen gran número de individuos que incumplan alguna de las restricciones y resulta de mayor utilidad aprovechar la

información aportada por ellos incorporando a su peso esta infracción degradándolo; además de ese modo el tiempo de cálculo invertido en evaluarlos es también aprovechado. Este método de función penalty es el más comúnmente usado en optimización evolutiva para incorporar restricciones [3.180][3.64][3.193]. Su principal inconveniente es la necesidad de afinar las constantes que ponderan el incremento de la función objetivo que supone la violación de la restricción. Las técnicas empleadas en su aplicación son múltiples y variadas, siendo este método objeto de constantes y recientes aportaciones [3.146].

- Aplicación de *algoritmos reparadores* especiales para corregir cualquier solución inviable generada de ese modo. Este algoritmo debe ser específicamente diseñado para la aplicación a la que va destinado y suelen ser costosos computacionalmente; para ciertos problemas el proceso de corregir una solución puede ser tan costoso como el resolver el problema original.
- Uso de *especiales mapeados, representaciones o decodificadores* del problema que garanticen o permitan incrementar la probabilidad de generar una solución válida o el uso de operadores específicos que garanticen esta validez. Sin embargo, estos decodificadores suelen ser costosos en tiempo de ejecución, no todas las constantes son implementables de este modo y el algoritmo aplicable debe ser hecho a medida para el problema abordado.
- Separación de *restricciones y objetivos* (mediante optimización multicriterio entre otros [3.34][3.85]).
- Métodos *híbridos*, empleando entre otros, lógica difusa, sistemas irmiunes, algoritmos culturales u optimización de colonias de hormigas.

En la resolución del problema de optimización multicriterio estructural que concierne a este trabajo se ha escogido el método de la función penalty, siendo el frecuentemente empleado por investigadores en la presente materia, como Rajeev y Krishnamoorthy [3.173], Goldberg y Samtani [3.64] o Hasancebi y Erbatur [3.79], siguiendo asimismo el método empleado por M. Galante [3.57]. La penalización en peso a la barra en caso de superarse los límites de restricción será en proporción de 6% de peso total de la estructura por un 10% de violación, tal como se recomienda en [3.64] y criterio adoptado igualmente en [3.57].

Las restricciones consideradas de manera genérica se detallan a continuación:

- Tensión de comparación menor a la tensión límite, que depende del material de la barra (frecuente valor adoptado para estructuras de acero es su límite elástico, de 2600 kgp/cm² en el acero S275 JR). La tensión de comparación considerada sigue el criterio de Von Mises, considerando tanto los esfuerzos axiales, cortantes y flectores. Para cada barra:

$$(y_{i,j})_{i,j} - (J_i)^2 < Q \quad (3.21)$$

- Esbeltez lambda a compresión menor a 200 (límite impuesto por la normativa española EA-95). El criterio empleado es el de la normativa española, basado en el de Julián y Lawrence. Para cada una de las barras donde se considere el efecto de pandeo:

$$A - 4 \lambda^2 < 0 \quad (3.22)$$

- Desplazamientos límites (en cada uno de los tres grados de libertad posibles) determinados en ciertos puntos, nudos o eventualmente centros de vanos de vigas:

$$u - u_0 < 0 \quad (3.23)$$

Con todo ello la función objetivo masa restringida resulta ser la dada por la ecuación (3.24).

$$F_{\text{objetivo}} = \sum_{i=1}^{N_{\text{barras}}} \rho_i A_i - P r h \left[i+k- \sum_{j=1}^{N_{\text{violac}}} j \cdot \text{viol}_{ij} \right] \quad (3.24)$$

donde :

A_i = área de sección de barra i

ρ_i = densidad de barra i

l_i = longitud de barra i

k = constante que regula la equivalencia entre restricción y masa, y que para la relación adoptada es de 0.6, siguiendo el criterio citado en el párrafo anterior (6% de incremento en la masa por un 10% de violación)

viol_{ij} = para cada una de las restricciones violadas, es la relación entre el valor que supone violación de la restricción (sea tensión, desplazamiento o esbeltez) y su límite de referencia.

3.5.2 Función Objetivo 2: Minimización del número de tipos de perfiles distintos

La **función objetivo 2 (minimización del número de tipos de perfiles distintos)** supone un condicionamiento de orden constructivo, especialmente en estructuras de gran número de perfiles, materializándose en la necesidad de ayudar a un mejor control de calidad durante la ejecución de la obra. Este factor ha sido también recientemente relacionado con el coste del ciclo de vida de la estructura en Sarma y Adeli [3.188], así como en Liu et al. [3.127].

El cálculo de esta función objetivo ha sido implementado mediante comparaciones sucesivas de los tipos de perfiles existentes en una estructura determinada. Se presenta a continuación el pseudo-algoritmo implementado, que puede ser también consultada en Greiner et al. [3.66]:

```

ntipperf = 0;
for(i=0;i<nbarr;i++){
    fbuc = FALSO;
    for(j2=(i+1);j2<nbarr && !fbuc j2++){
        if(*(perfil+j2) == *(perfil+i)){
            ntipperf++;
            fbuc = VERDAD;
        }
    }
}
ntipperf = nbarr - ntipperf;

```

Algoritmo de cálculo del número de perfiles distintos

donde las variables empleadas son las siguientes:

$n_{tipperf}$: valor de la función objetivo a evaluar, es decir, el número de tipos de perfiles distintos

n_{barr} : número de barras (variables diferentes) codificadas en el algoritmo genético.

$perfil$: puntero que contiene el tipo de perfil asociado a cada barra de la solución tratada en ese momento.

El algoritmo desarrollado para el cálculo de esta segunda función objetivo, requiere en *su caso más desfavorable* (todos los perfiles de diferente tipo) un total de: $(n-1) + (n-2) + (n-3) + \dots + 2 + 1 = n(n-1)/2$ comparaciones, siendo n el número de barras distintas codificadas en el algoritmo genético; y para el *caso más favorable* (todos los perfiles del mismo tipo) requiere un total de: $(n-1)$ comparaciones. Cuantos menos tipos de perfiles distintos haya en la estructura, el cálculo será más cercano a este segundo valor y será más rápido.

3.6 Acoplamiento Evaluador Estructural- Algoritmo Genético

Para alcanzar unos tiempos de cálculo reducidos en lo posible, se debe tratar de agilizar el proceso evaluador desarrollado por el programa de cálculo matricial, ya que será el que proporcione la información necesaria para evaluar el entorno de cada uno de los individuos por el algoritmo genético.

Así pues, aquellas operaciones que basta realizar una única vez por el matricial, y que son válidas para todas las estructuras independientemente de que sus perfiles varíen, se realizarán sólo una vez; con ello se ahorrarán todas esas operaciones, cada una de las miles de evaluaciones necesarias de efectuar.

Por ello, se divide el programa matricial en dos bloques:

- *Primer bloque* : Operaciones iniciales que se efectúan una sola vez para todas las evaluaciones. Se incluyen las siguientes :
 1. Lectura de ficheros conteniendo los datos que definen el problema a optimizar. Esta lectura es análoga a la que puede realizar un matricial convencional.
 2. Cálculo de longitudes de barras y ángulos globales entre barras.
 3. Cálculo de la reenumeración optimizada mediante el algoritmo de Cuthill - McKee inverso.
 4. Cálculo del vector que almacena las ecuaciones activas o efectivas de la estructura.
 5. Cálculo del semiancho de banda de la estructura.
 6. Cálculo del vector global de fuerzas para cada caso de carga, comprendiendo el ensamblaje del mismo, (se excluyen los casos de carga de peso y asentamientos por depender de cada estructura particular).
- *Segundo bloque* : Este bloque es el que se calcula para cada una de las evaluaciones efectuadas por el algoritmo genético, ahorrándose todas las operaciones citadas con anterioridad en el primer bloque. Será calculado miles de veces y en él se efectúan el resto de operaciones necesarias del programa matricial, que esquemáticamente se detallan a continuación :
 1. Cálculo de la matriz de rigidez efectiva de la estructura total.
 2. Consideración del peso y asentamientos como casos especiales de carga.
 3. Resolución del sistema $K \cdot U = F$ y obtención de los desplazamientos nodales.
 4. Cálculo de los esfuerzos de la estructura.

5. Cálculo de esbelteces de cada barra, así como de desplazamientos y tensiones para cada una de las hipótesis de carga.

En función de esta información disponible proporcionada por el programa de cálculo matricial mediante el método directo de rigidez, se podrá efectuar la evaluación de la función objetivo para cada uno de los individuos, que medirá su adaptabilidad al entorno y será instrumento para alcanzar la optimización de la estructura solicitada.

3.7 Otras consideraciones de Optimización

3.7.1 Introducción

El lenguaje de programación considerado para implementar y desarrollar los códigos de optimización estructural mediante los algoritmos genéticos es el lenguaje C++. Este lenguaje de alto nivel posee gran flexibilidad, potencia y ante todo, una gran portabilidad que permite a un mismo código poder ser ejecutado en gran número de plataformas. La gran mayoría de la programación de algoritmos genéticos está realizada en este lenguaje. Además se debe señalar la posesión de ciertas características de lenguaje de bajo nivel que permiten realizar operaciones normalmente sólo posibles en lenguaje ensamblador o máquina, como son las operaciones a nivel de bits.

3.7.2 Consideraciones respecto a la optimización de tiempo de cálculo

La función objetivo del algoritmo genético deberá ser evaluada miles de veces, por lo que toda optimización encaminada a mejorarla en tiempo de ejecución repercutirá notablemente en el tiempo final de cálculo de optimización del algoritmo genético. Esta mejora en eficiencia temporal podrá ser llevada a cabo de dos modos:

- Optimizando el algoritmo de la función objetivo: la resolución matricial del método directo de la rigidez en sí mismo (por ejemplo, incorporando directamente las expresiones de la matriz de rigidez elemental en coordenadas globales en lugar de realizar la transformación de coordenadas mediante la multiplicación de matrices in situ). Ha sido descrito en los apartados anteriores del capítulo 3.
- Optimizando la programación de ese algoritmo, empleando adecuadamente las posibilidades que ofrece el lenguaje de programación C/C++.

Este último aspecto se trata a continuación de forma más detallada.

3.7.3 Costes computacionales

La optimización de programación consiste generalmente en un balance entre tiempo de ejecución y tamaño de código o memoria ocupada. Cuando prima una mejora en uno de los factores normalmente el otro es penalizado.

Cuando el objetivo reside en obtener un código lo más optimizado posible de cara a la velocidad, se debe contar con una referencia de coste en tiempo de ejecución que permita decidir cómo es más efectiva la programación. Con la finalidad de proporcionar una orientación relativa del coste computacional en términos de tiempo, se muestran en la tabla 3.1 ciertas operaciones frecuentes en programación referidas al lenguaje C/C++ [3.209]:

Tabla 3.1. Coste relativo de algunas operaciones frecuentes en lenguaje C/C++

<i>Operación</i>	<i>Coste relativo</i>
Escritura o lectura de ficheros, incluyendo las funciones printf y scanf	1000
Asignación o liberación dinámica de memoria	800
Funciones trigonométricas	500
Operación en coma flotante (real)	100
División de entero	30
Multiplicación de entero	20
Llamada a una función	10
Indexación de array simple	6
Desplazamientos de bits	5
Suma o resta	5
Derreferencia a puntero	2
AND, OR o NOT a nivel de bits	1
AND, OR o NOT a nivel lógico	1

Dentro de las estrategias de optimización se encuentra por tanto el uso de *operaciones menos costosas* que otras equivalentes en lo posible, de acuerdo con la tabla 3.1.

Como se puede observar, aquellas operaciones asociadas a la manipulación directa de bits u operaciones de bajo nivel son las más económicas. En lo posible, durante la programación de los algoritmos genéticos se deberá aprovechar esta característica y emplear la representación interna del ordenador en forma de bits para acelerar el tiempo de ejecución (así, una variable `unsigned long int` en una arquitectura computacional convencional es interpretada como una `cadena_binaria` de 32 bits).

Entre *otras estrategias de optimización* se pueden citar las siguientes :

- Uso de punteros en lugar de arrays.
- El uso de funciones `inline`, aún a costa del incremento en tamaño de código. Ello elimina el sobrecosto asociado a una llamada funcional sustituyendo ésta por el código de la misma cuando se compila; de gran utilidad sobretodo si existen llamadas funcionales en el interior de bucles con gran número de iteraciones.
- El uso del tipo de almacenamiento `register`. Estas variables en C/C++ son aplicables sobretodo a las declaraciones de tipo entero y se almacenan dentro de los registros de la unidad central de proceso (CPU) del ordenador. Allí es donde se realizan las operaciones aritméticas y lógicas, transfiriéndose normalmente la información desde la memoria de la computadora a estos registros. El tipo de

almacenamiento `register` permite el ahorro temporal de esa transferencia, pudiendo ser considerablemente significativo.

-Operaciones en bucles, como :

- * Ordenación de bucles anidados exterior a interior, de menor a mayor número de iteraciones; consiguiendo reducir el número de inicializaciones y condiciones de terminación del mismo.
- * Eliminación de código invariante en el interior de bucles; con ello se evita la repetición de una sentencia en el interior de un bucle innecesariamente si es posible su ejecución de modo extemo.
- * Despliegue de bucles: puede ser beneficioso en ocasiones el despliegue de un bucle, como en el caso de un bucle extemo constante, que puede ser incorporado al intemo, o bien combinación de varios bucles en uno si tienen los mismos límites de ejecución.
- * Eliminación de tests redundantes en el interior de los mismos.

Además se deben tener en cuenta igualmente aquellas *opciones del compilador* utilizado para la obtención del código ejecutable que estén encaminadas a la obtención de código más optimizado. Su uso permite que parte de las opciones citadas con anterioridad sean incorporadas al código final aún en el caso de no ser tenidas en consideración en el código fíente. En el caso particular de emplear el sistema operativo UNIX / Linux, y el compilador `gnu`, están englobadas en las opciones `-O` del mismo.

3.7.4 Generación de números pseudoaleatorios

Los algoritmos genéticos hacen uso de operadores estocásticos y dependen así del algoritmo generador de números pseudoaleatorios intemo implementado en el programa. Se denomina generador de números pseudoaleatorios, atendiendo a que toda secuencia numérica generada por un ordenador es en realidad determinística y dependiente del primer número empleado para generarla denominado semilla. Esta semilla generalmente se toma del contador temporal intemo del ordenador (fimción `time (time_t)` en C/C++).

El lenguaje C/C++ posee dentro de su librería estándar de fínciones un generador de números pseudoaleatorios entero en un rango entre 0 y 65535 (16 bits de longitud). Sin embargo, en la operatividad de algoritmos genéticos resulta de mucha mayor utilidad el uso de `vm` generador pseudoaleatorio de 32 bits de longitud (tamaño `unsigned long int`), con lo que con un número aleatorio generado en realidad generamos 32 bits aleatorios de una tacada (entero en rango entre 0 y 4294967295).

Para ello se ha tomado de [3.215] un generador pseudoaleatorio de 32 bits propuesto por Marsaglia del tipo `multiply-with-carry (MWC)`. El algoritmo en C/C++ implementado se presenta a continuación:

```
unsigned long int irán;  
unsigned long int jran;  
  
unsigned long int random32()  
{  
    irán = 30903* (iran&65535) + (iran»16) ;  
    jran = 18000* (jran&:65535) + {jran»16) ;  
  
    return ( (iran«16) + jran) ;  
}
```

Algoritmo generador de números pseudoaleatorios de 32 bits MWC (!\larsagiia).

Una de las características de un generador pseudoaleatorio es su ciclo o periodo de generación, que es aquel registrado antes de una repetición en periodicidad de la secuencia generada. El generador MWC empleado posee un periodo de generación antes de la repetición del ciclo de 2^m números generados (notablemente mayor al de la función de la librería estándar, que es de 2^m) y una rapidez de generación considerablemente mayor a los generadores comúnmente empleados para tal fin [3.215]. Analicemos en tiempo relativo en términos dados por la tabla 3.1, cuánto requiere la generación de un número aleatorio mediante este algoritmo:

1 llamada funcional	= 10
2 multiplicaciones enteras	= $2 * 20$
3 sumas	= $3 * 5$
3 desplazamientos de bits	= $3 * 5$
2 AND a nivel de bits	= $2 * 1$
Total	= 82

, que resulta ser menor en coste por ejemplo, al de una multiplicación real que requiere 100.

Es posible la realización de tests empíricos que traten de contrastar la aleatoriedad de una secuencia de números desde el punto de vista estadístico, mostrando la confianza que se puede tener en el generador pseudoaleatorio que genera la secuencia. Se han analizado secuencias numéricas para contrastar la hipótesis de que esos números procedan de múltiples ensayos independientes de un experimento. Para ello se empleará como contraste de ajuste a una distribución uniforme, el test X^2 de no correlación o también llamado contraste de Pearson. Se han realizado análisis de secuencias unidimensionales, bidimensionales y tridimensionales de este generador [3.67], obteniéndose que como conclusión final, se puede afirmar que no ha podido rechazarse la hipótesis de aleatoriedad uniforme de las secuencias numéricas generadas por el MWC para los sucesos analizados.

3.8 Casos Test Estructurales

El problema de optimización estructural monocriterio de minimización de la masa restringida mediante algoritmos genéticos ha sido validado exitosamente con otros métodos tradicionales de optimización en [3.67]. En él se han comparado los algoritmos genéticos en optimización monocriterio continua con el método de direcciones eficientes mejorado con criterio de agotamiento mediante dos casos test publicados en [3.84]. Estos casos test abordan el problema monocriterio de optimización continua de masa restringida sin tener en cuenta el efecto de pandeo y sin tener en cuenta el peso propio de las barras de la estructura. En [3.67] se comparan estos resultados con los obtenidos por la optimización global continua mediante algoritmos genéticos, introduciendo además el peso propio y el efecto de pandeo, considerado el caso traslacional. Se aborda también el problema monocriterio de optimización de masa restringida mediante perfiles discretos (optimización discreta), además de introducir el problema multicriterio que se afronta en profundidad en este trabajo. Por ello, esos dos casos test estructurales serán los empleados en la presente tesis. Serán denominados caso X (estructura de 4 barras) y caso Y (estructura de 55 barras).

3.8.1 Caso X

El caso test X se representa gráficamente en la figura 3.2, donde las cotas están expresadas en metros.



Figura 3.2. Caso Test Estructural X.

Las barras de la 1 a la 4 tienen unas longitudes de 10, 20, 6 y 6 metros respectivamente. Las cargas uniformes a que están sometidas las barras son de 15 T/m, 10 T/m, 0 T/m y 0.2 T/m respectivamente. Además existe una restricción de flecha máxima en la barra nº 2 de un trescientosavo de su longitud, que equivale a 6.6 cm. Como problema de dominio discreto, los perfiles de elección disponibles son del tipo IPE recogido en la norma española. Asimismo, se ha tenido en consideración el efecto de pandeo, así como el peso propio de la estructura. La tensión máxima admisible considerada es de 2400 kg/cm^2 . La densidad y módulo de elasticidad E son las del acero: 7.85 T/mV 2100 T/cm²

En optimización multiobjetivo son especialmente valorados los problemas de referencia con soluciones óptimas conocidas, de cara a evaluar correctamente la eficacia de los métodos empleados en su resolución. Se destaca de este caso particular, que el frente óptimo es conocido, permitiendo comparar éste con los resultados obtenidos. En este problema, el frente óptimo es determinable

mediante valoración exhaustiva del espacio de búsqueda completo ($2^{16}=65536$ soluciones). Este frente de Pareto óptimo se representa en la figura 3.3, donde el eje de abcisas representa el peso de la estructura, y el de ordenadas el número de distintos perfiles empleados.

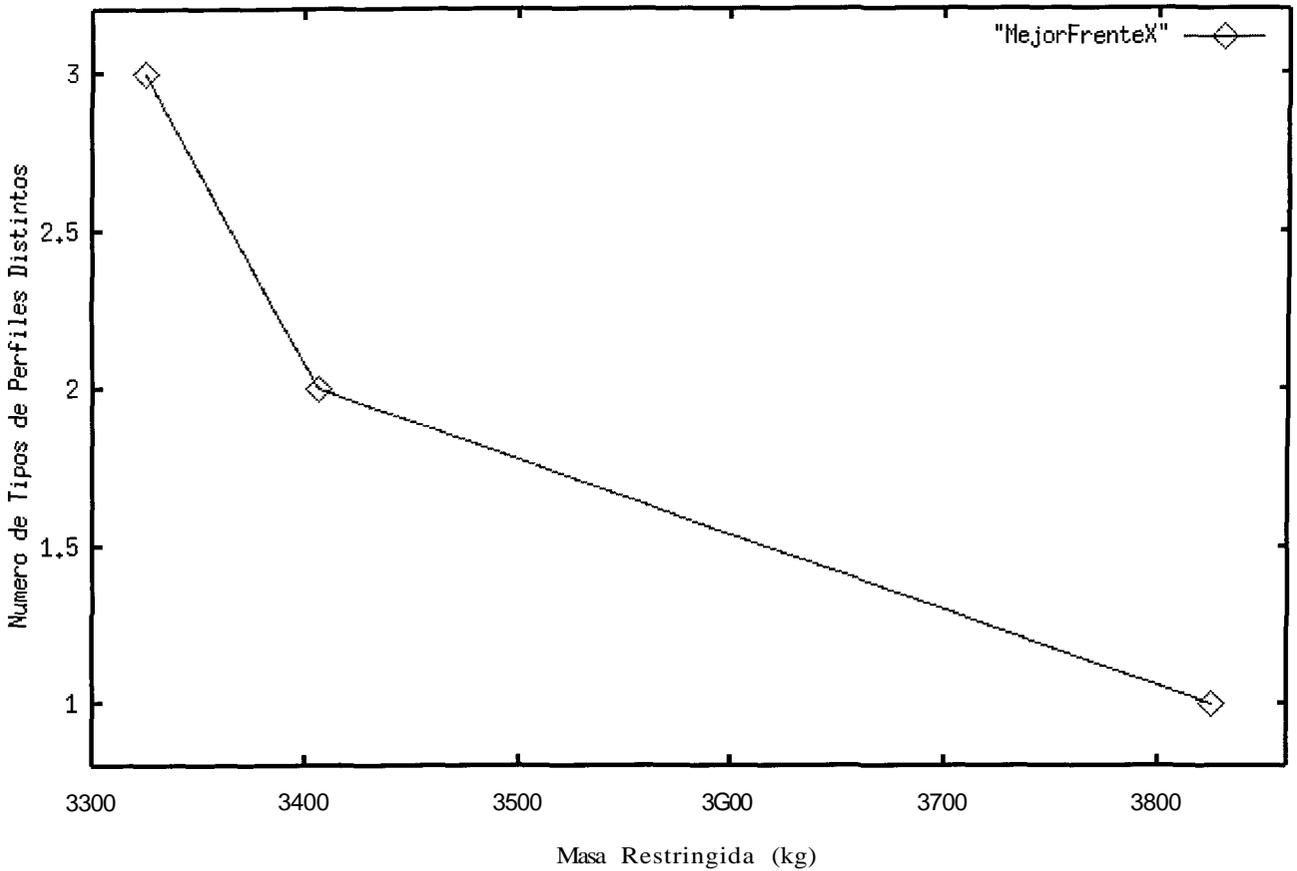


Figura 3.3. Frente de Pareto Óptimo del Caso Test X.

En la tabla 3.2 se representan las soluciones integrantes del frente de Pareto detalladas, exponiendo el número de tipos de perfiles distintos, la masa restringida (kg), la masa (kg) y la componente de restricción (kg). Se desglosa cada barra según su tipo de perfil y se indica el valor de la tensión máxima de cada barra (kg/cm^2) y del desplazamiento de control.

Tabla 3.2. Frente de Pareto Óptimo Detallado del Caso Test X.

Número de tipos de perfiles distintos (F1)	3	2	1
Masa Restringida (kg) (F2)	3324.318	3405.330	3824.520
Restricción (kg)	0.0	0.0	0.0
Masa (kg)	3324.318	3405.330	3824.520
Barra n° 1	IPE-330	IPE-330	IPE-500
Barra n° 2	IPE-500	IPE-500	IPE-500
Barra n° 3	IPE-450	IPE-500	IPE-500
Barra n° 4	IPE-500	IPE-500	IPE-500
Tensión Barra n° 1 (kg/cm^2)	2356.56	2241.35	1157.62
Tensión Barra n° 2 (kg/cm^2)	1817.68	1891.93	1957.97
Tensión Barra n° 3 (kg/cm^2)	1353.23	1180.14	944.60
Tensión Barra n° 4 (kg/cm^2)	1677.99	1647.09	1620.13
Desplazamiento de Control (cm.)	6,611	6.390	6.196

3.8.2 Caso Y

El caso test Y se representa gráficamente en la figura 3.4.

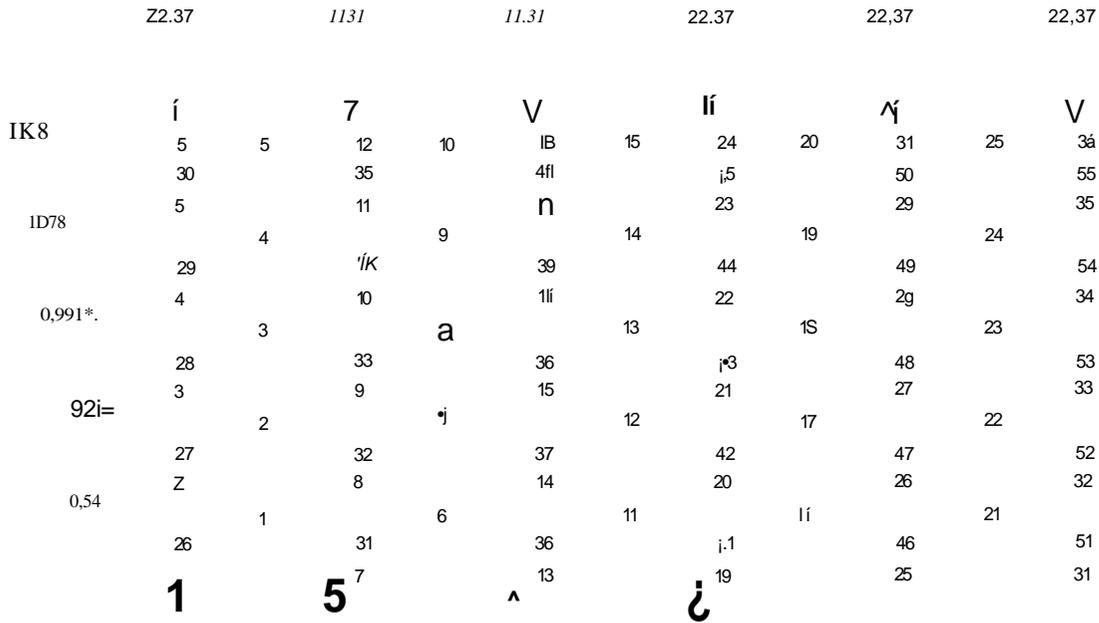


Figura 3.4. Caso Test Estructural Y.

En ella se incluye la numeración de las barras, la de los nodos, así como las cargas puntuales expresadas en toneladas. Para los nudos 1, 2, 3, 4, 5, 6 estas cargas puntuales son respectivamente de 840, 924, 994, 1078 y 1148 kpg; y para los nudos 6, 12, 18, 24, 30 y 36 de 22370 kpg. Además en todas las vigas (barras de la 1 a la 25) existe una carga uniformemente repartida de 4076 kpg/m. La luz de todas las vigas es de 5.6 m y la altura de todos los pilares de 2.80 m. Se consideran perfiles HEB para los pilares e IPE para las vigas, siendo las tensiones admisibles de 2.0 T/cm² y 2.2 T/cm² para vigas y pilares (barras de la 26 a la 55), respectivamente. Se tiene en cuenta asimismo una restricción de desplazamiento en el punto medio de todas las vigas de im trescientosavo de la longitud (1.86 cm). La densidad y módulo de elasticidad E son las del acero: 7.85T/m³ y 2100T/cm²

Si bien en el caso test anterior la restricción que imponía la solución óptima era la del desplazamiento máximo, en este caso test son las tensiones las que tienen mayor relevancia.

En este caso test Y, el espacio de búsqueda es de $2^{220} \sim 1.7 \cdot 10^{66}$. Si intentáramos evaluar el fi-ente óptimo mediante valoración exhaustiva del espacio de búsqueda completo como en el caso test X, ello nos llevaría en un ordenador con procesador Itanium 1.5Ghz un total aproximado de 410⁶⁶ años. Así pues, el frente de Pareto óptimo considerado es el mejor frente obtenido a partir de todas las ejecuciones realizadas durante el presente trabajo. Dicho frente, considerado de referencia, se representa en la figura 3.5, donde el eje de abcisas representa el peso de la estructura, y el de ordenadas el número de distintos perfiles empleados.

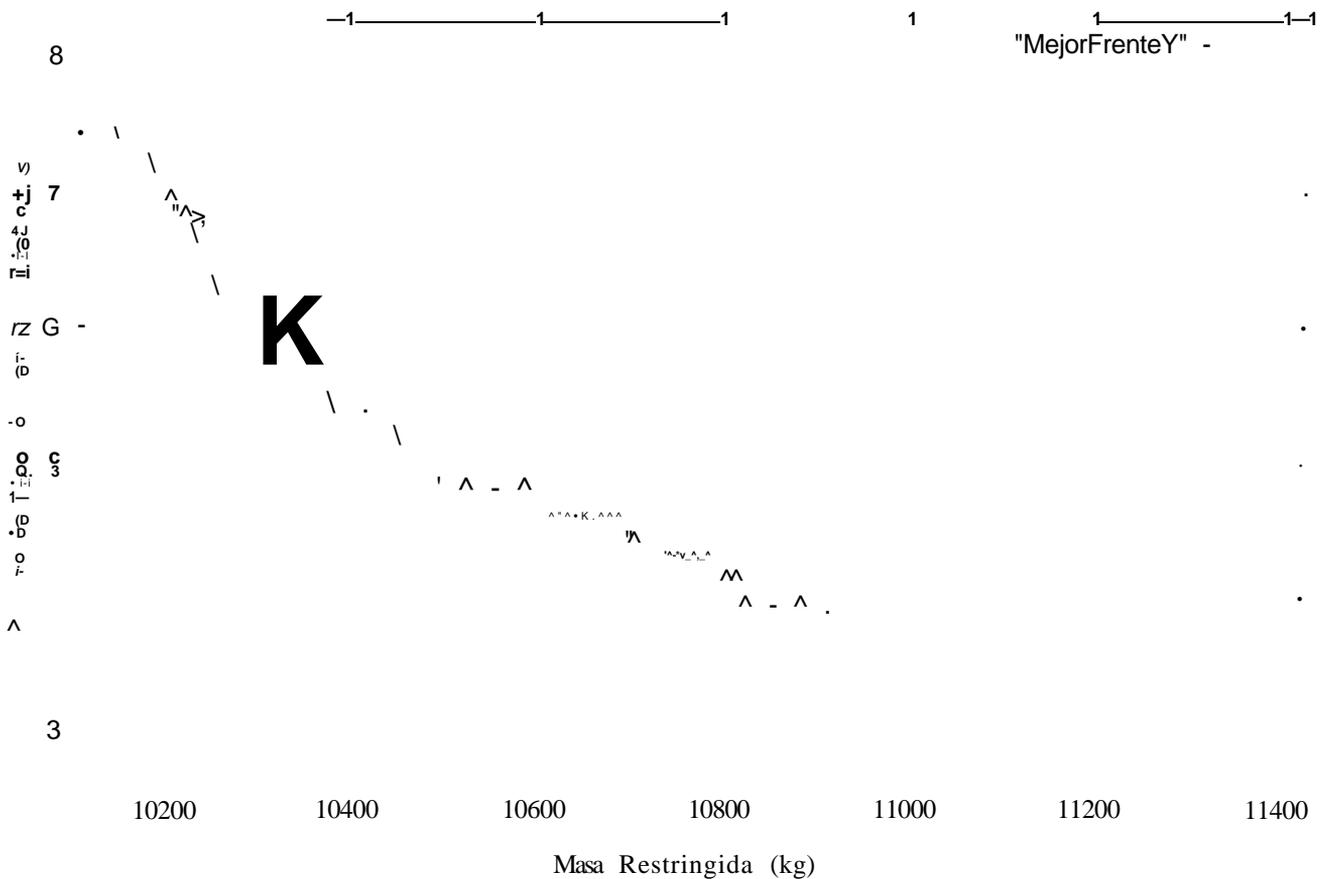


Figura 3.5. Fronte de Pareto Óptimo del Caso Test Y.

En las tablas 3.3 y 3.4 se representan las soluciones integrantes del frente de Pareto detalladas, exponiendo el número de tipos de perfiles distintos, la masa restringida (kg), la masa (kg) y la componente de restricción (kg). Se desglosa cada barra según su tipo de perfil y se indica el valor de la tensión máxima de cada barra (kpg/cm). Igualmente se destacan en *itálica* los valores de las tensiones que superan los límites impuestos y que implican un valor de restricción (kg) no nulo. Para cada uno de todos los números de tipos de perfiles distintos se han obtenido soluciones que cumplen todas las restricciones, o sea, con valor de restricciones nulas. Por coherencia se muestra en la tabla 3.3 aquellas soluciones con función objetivo menor de entre las halladas, independientemente del valor de su restricción.

Tabla 3.4. Frente de Pareto Óptimo del Caso Test Y. Tensiones Detalladas (kpg/cm[^])

Número de tipos de perfiles distintos (F1)	8	7	6	5	4	3
Masa Restringida (kg) (F2)	10130.04	10212.07	10318.95	10517.65	10865.28	11394.45
Restricción (kg)	2.75	5.65	0.00	0.00	213.55	272.134
Masa (kg)	10127.29	10206.41	10318.95	10517.65	10651.73	11122.32
Tensión Barra n° 1	1939.3	1939.2	1919.5	1920.2	2236.8	2244.9
Tensión Barra n° 2	1869.2	1869.1	1860.8	1863.2	2166.5	2198.4
Tensión Barra n° 3	1876.3	1875.8	1893.3	1860.4	2198.1	2163.4
Tensión Barra n° 4	2188.7	2196.2	2189.7	2150.4	2101.9	2073.8
Tensión Barra n° 5	1800.3	1815.0	1804.4	1808.4	2193.9	2186.5
Tensión Barra n° 6	2065.1	2065.1	2074.8	2074.7	2113.9	2121.0
Tensión Barra n° 7	2047.2	2047.2	2070.5	2068.7	2131.6	2132.4
Tensión Barra n° 8	2009.8	2010.0	2015.7	1994.1	2077.7	2090.6
Tensión Barra n° 9	2026.7	2019.0	2018.7	2002.8	2008.7	2057.9
Tensión Barran" 10	2165.4	2160.1	2140.6	2114.8	2064.1	2076.6
Tensión Barra n° 11	2129.1	2129.1	2131.4	2131.6	2125.6	2129.7
Tensión Barra n° 12	2073.8	2073.8	2087.6	2087.6	2124.3	2125.5
Tensión Barra n° 13	2047.6	2047.6	2043.7	2052.5	2070.7	2079.7
Tensión Barra n° 14	2036.9	2034.9	2051.0	2054.6	2018.2	2048.4
Tensión Barra n° 15	2049.3	2048.0	2064.5	2059.6	2060.2	2059.7
Tensión Barra n° 16	2166.0	2165.9	2150.3	2150.1	2147.3	2140.6
Tensión Barra n° 17	2053.8	2053.7	2065.2	2064.2	2125.3	2149.2
Tensión Barra n° 18	2057.9	2057.7	2051.1	2062.2	2063.7	2180.0
Tensión Barra n° 19	2021.2	2015.5	2023.1	2036.4	1973.9	2054.4
Tensión Barra n° 20	2071.5	2070.6	2077.1	2079.2	2100.4	2107.9
Tensión Barra n° 21	2024.9	2024.8	1977.9	1977.7	1943.8	1940.8
Tensión Barra n° 22	1940.1	1940.0	1927.8	1927.7	1976.2	1941.8
Tensión Barra n° 23	1999.4	1999.0	1997.0	1986.6	1906.6	1951.6
Tensión Barra n° 24	2000.0	2004.8	1987.4	1992.4	1976.6	1974.5
Tensión Barra n° 25	2118.1	2132.2	2114.1	2138.6	2098.9	2134.9
Tensión Barra n° 26	1906.9	1906.2	1869.2	1873.4	1325.8	1312.5
Tensión Barra n° 27	1849.4	1848.8	1592.1	1596.5	1331.3	1360.1
Tensión Barra n° 28	1994.3	1994.3	1991.0	1923.1	1404.4	1274.4
Tensión Barra n° 29	1972.9	1970.4	1935.7	1698.2	1971.2	1462.9
Tensión Barra n° 30	1854.0	1842.3	1556.3	1552.0	1691.3	1438.7
Tensión Barra n° 31	1848.3	1848.8	1807.9	1803.9	1774.6	1777.7
Tensión Barra n° 32	1830.0	1830.7	1856.0	1851.6	1800.1	1514.8
Tensión Barra n° 33	1745.2	1746.5	1458.9	1433.2	1481.2	1751.7
Tensión Barra n° 34	1610.0	1610.9	1604.4	1604.9	1966.2	1250.1
Tensión Barra n° 35	1779.1	1346.3	1348.7	1070.1	1377.1	898.4
Tensión Barra n° 36	2000.5	2000.9	1987.9	1989.9	1736.7	1754.1
Tensión Barran" 37	1681.5	1682.0	1701.5	1703.5	1739.4	1468.2
Tensión Barra n° 38	1959.8	1960.5	1950.9	1965.5	1395.0	1714.1
Tensión Barra n° 39	1900.4	1917.6	1914.8	1507.9	1935.5	1239.5
Tensión Barra n° 40	1626.0	1215.6	1236.0	953.1	1299.9	807.8
Tensión Barran" 41	1754.1	1754.3	1734.2	1734.3	1743.5	1755.4
Tensión Barra n° 42	1710.2	1710.5	1725.1	1725.1	1739.7	1461.2
Tensión Barra n° 43	1990.0	1990.5	1986.2	1996.0	1396.4	1713.3
Tensión Barra n° 44	1944.3	1960.6	1962.5	1546.4	1929.3	1246.9
Tensión Barra n° 45	1650.7	1234.2	1257.3	973.7	1280.0	809.6
Tensión Barra n° 46	1770.7	1771.4	1754.8	1757.3	1761.7	1788.0
Tensión Barra n° 47	1716.1	1717.0	1727.6	1730.9	1729.7	1489.4
Tensión Barra n° 48	1991.7	1993.1	1978.0	1999.8	1393.0	1223.0
Tensión Barra n° 49	1953.0	1973.4	1971.6	1550.4	1972.8	1301.6
Tensión Barra n° 50	1692.8	1281.4	1251.5	985.0	1307.7	789.4
Tensión Barra n° 51	1983.1	1982.4	1733.2	1732.6	1482.2	1480.0
Tensión Barra n° 52	1882.4	1881.9	1832.1	1831.0	1574.2	1592.9
Tensión Barra n° 53	1754.8	1754.3	1755.8	1756.2	1620.7	1999.2
Tensión Barra n° 54	1888.9	1888.6	1892.8	1903.1	1204.6	1720.7
Tensión Barra rf 55	1791.9	1776.6	1817.3	1785.9	1825.4	1498.2

Bibliografía

- [3.1] R. Abascal García. '*Cálculo Matricial de Estructuras*'. Apuntes E.T.S.I.I.
- [3.2] H. Adeli, N.T. Cheng, "Augmented Lagrangian Genetic Algorithms for Structural Optimization", *Journal of Aerospace Engineering*, 7-1 (1994) pp 104-118.
- [3.3] H. Adeli, N.T. Cheng, "Integrated genetic algorithm for optimization of space structures", *Journal of Aerospace Engineering*, 6-4 (1993) pp 315-328.
- [3.4] H. Adeli, S. Kumar, "Distributed genetic algorithm for structural optimization", *Journal of Aerospace Engineering*, 8-3 (1995) 156-163.
- [3.5] E. Alarcón Álvarez, R. Álvarez Cabal, M.S. Gómez Lera. '*Cálculo Matricial de Estructuras*' Ed. Reverte (1986).
- [3.6] N. Ali, K. Behdinan, Z. Fawaz, "Applicability and viability of a GA based finite element analysis architecture for structural design optimization", *Computers & Structures* 81 (2003) 2259-2271.
- [3.7] P. Almeida, G. Winter, G. Montero, R. Montenegro. 'Contribución numérica en la resolución directa de sistemas de ecuaciones asociados al método de elementos finitos con utilización de teoría de grafos' I Congreso de Métodos Numéricos en Ingeniería SEMNI1990.
- [3.8] R. Arguelles Álvarez, R. Arguelles Bustillo. *Análisis de Estructuras: Teoría, problemas y programas*'. E.T.S.I. Montes Fundación Conde del Valle de Salazar (1996) Madrid.
- [3.9] H. Baier, "Über Algorithmen zur Ermittlung und Charakterisierung Pareto-optimaler Lösungen bei Entwurfsaufgaben Elastischer Tragwerke", *Z Angew. Math. Mech.* 57 (1977) pp. 318-320.
- [3.10] R.J. Balling, J.T. Taber, M.R. Brown, K. Day, "Multiobjective Urban Planning using a Genetic Algorithm", *ASCE Journal of Urban Planning and Development* 125-2 (1999) pp. 86-99.
- [3.11] A.D. Belegundu, D.V. Murthy, R.R. Salagame, E.W. Constants, "Multiobjective Optimization of Laminated Ceramic Composites using Genetic Algorithms", *Fifth AIAA/USAF/NASA Symposium on Multidisciplinary Analysis and Optimization*, (1994) pp. 1015-1022.
- [3.12] J.R. Beveridge, K. Balasubramaniam, D. Whitley, "Matching horizon features using a Messy Genetic Algorithm", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 499-516.
- [3.13] H-G. Beyer, "Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 239-267.
- [3.14] S. Botello, J.L. Marroquin, E. Oñate, J.V. Horebeek, "Solving structural optimization problems with genetic algorithms and simulated annealing", *International Journal for Numerical Methods in Engineering*, 45 (1999) 1069-1084.
- [3.15] J.-P. Brans, B. Mareschal, "How to select and how to rank projects: the PROMETHEE method for MCDM", *European Journal of Operations Research*, 24 (1986) pp. 228-238.
- [3.16] S.A. Bums, *Recent Advances in Optimal Structural Design*, Institute of American Society of ASCE-SEI (2002).
- [3.17] C.V. Camp, S. Pezeshk, G. Cao, "Optimized Design of Two-Dimensional Structures using a Genetic Algorithm", *ASCE Journal of Structural Engineering*, 124-5 (1998).
- [3.18] E. Cantú-Paz, D.E. Goldberg, "Efficient parallel genetic algorithms: theory and practice", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 221-238.
- [3.19] D.G. Carmichael, "Computation of Pareto óptima in structural design", *International Journal*

of *Numerical Methods in Engineering*, 15 (1980) pp. 925-952.

[3.20] T-Y. Chen, C-J. Chen, "Improvements of simple genetic algorithm in structural design", *International Journal for Numerical Methods in Engineering*, 40 (1997) 1323-1334.

[3.21] S.Y. Chen, S.D. Rajan, "A robust genetic algorithm for structural optimization", *Structural Engineering and Mechanics*, 10-4 (2000) 313-336.

[3.22] S.Y. Chen, S.D. Rajan, "Improving the efficiency of genetic algorithms for frame designs", *Engineering Optimization*, 30-3,4 (1998) 281-307.

[3.23] T.Y. Chen, C.C. Shieh, "Fuzzy multiobjective topology optimization" , *Computers & Structures* 78 (2000) 459-466.

[3.24] F.Y. Cheng y D. Li, "Genetic Algorithm Development for Multi-Objective Optimization of Structures", *AIAA Journal*, 36-5 (1998) 1105-1112.

[3.25] F.Y. Cheng, D. Li, "Multiobjective optimization design with Pareto genetic algorithm", *Journal of Structural Engineering*, 123-9(1997) 1252-1261.

[3.26] F.Y. Cheng, D. Li, J. Ger, "Multiobjective optimization of Seismic Structures", *Advanced Technology in Structural Engineering*, (2000), Section 6, chapter 4.

[3.27] F.Y. Cheng, Z.Q. Wang, "Development of seismic structural optimization and future trends", *Structural Engineering and Mechanics* (1999) 101-111.

[3.28] J-H. Chou, J. Ghaboussi, "Genetic algorithm in structural damage detection" , *Computers & Structures* 79 (2001) 1335-1353.

[3.29] R.F. Coelho, H. Bersini, P. Bouillard, "Parametrical mechanical design with constraints and preferences: application to a purge valve", *Comput. Methods Appl. Mech Engrg*, 192 (2003) 4355-4378.

[3.30] C.A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art", *Comput. Methods Appl. Mech Engrg*, 191 (2002) 1245-1287.

[3.31] C.A. Coello Coello, "An updated survey of GA-based multiobjective optimization *XQc\miquQs\ACMComputingSurveys*, 32-2 (2000) 109-143.

[3.32] C.A. Coello Coello, "An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design", *PhD Thesis*, Tulane University, LA, U.S.A. (1996).

[3.33] C.A. Coello Coello, A.D. Christiansen, "Multiobjective optimization of trusses using genetic algorithms", *Computers & Structures* 75 (2000) 647-660.

[3.34] C.A. Coello Coello, E. Mezura-Montes. Handling constraints in genetic algorithms using dominance-based tournaments. In *Proceedings of the Fiñh International Conference on Adaptive Computing Design and Manufacture (ACDM 2002)*, vol. 5, University of Exeter, Devon, UK, Parme IC (ed.). Springer: Berlin, April 2002; 273-284.

[3.35] C.A Coello Coello, G.T. PuUdo, A.H. Aguirre, "Multiobjective evolutionary algorithms for structural optimization", *Computational Fluid and SolidMechanics* (2003) 2244-2248.

[3.36] M.Z. Cohn, Z. Lounis, "Optimal Design of Structural Concrete Bridge Systems", *Journal of Structural Engineering*, 120-9 (1994) 2653-2674.

[3.37] T.H. Gormen, CE. Leiserson, R.L. Rivest. *'Introduction to Algorithms'*. The MIT Press. Me GrawHill. (1990).

[3.38] V. Coverstone-Carroll, J.W. Hartmann, W.J. Masón, "Optimal multi-objective low-thrust

- spacecraft trajectories", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 387-402.
- [3.39] W.A. Crossley, A.M. Cook, D.W. Fanjoy, V.B. Venkayya, "Using Two-Branch Tournament Genetic Algorithm for Multiobjective Optimization", *AISS / ASME / ASCE / AHS / ASC Structures, Structural Dynamics & Materials Conference*, Vol 2, pp. 1752-1762 AIAA, USA (1998).
- [3.40] E. Cuthill, J. McKee, 'Reducing the bandwidth of sparse symmetric matrices'. *Proceedings Association Computing Machinery, National Conference* New York. (1969) pp. 157-172.
- [3.41] K. Deb, "An efficient constraint handling method for genetic algorithms" , *Comput. Methods Appl. Mech Engrg*, 186 (2000) 311-338.
- [3.42] K. Deb, S. Gulati, "Design of truss-structures for minimum weight using genetic algorithms", *Finite Elements in Analysis and Design*, 37-5 (2001) 447-465.
- [3.43] A.K. Dhingra, B.H. Lee, "Multiobjective Design of actively controlled structures using a hybrid optimization method", *International Journal for Numerical Methods in Engineering*, 38-20 (1995) pp. 3383-3401.
- [3.44] A.K. Dhingra, B.H. Lee, "A Genetic Algorithm Approach to Single and Multiobjective Structural Optimization with Discrete-Continuous Variables", *International Journal for Numerical Methods in Engineering*, 37 (1994) pp. 4059-4080.
- [3.45] A.K. Dhingra, B.H. Lee, "Optimal placement of actuators in actively controlled structures", *Engineering Optimization*, 23-2 (1994) pp. 99-118.
- [3.46] L. Duckstein, "Multiobjective Optimization in structural design: The model cholee problem", in: *New Directions in Optimum Structural Design*, John Wiley & Sons (1984) pp. 459-481.
- [3.47] F. Erbatir, O. Hasancebi, I. Tütüncü, H. Kihc, "Optimal design of planar and space structures with genetic algorithms" , *Computers & Structures* 75 (2000) 209-224.
- [3.48] E.M.R. Fairbairn, M. Silvano, R. Toledo Filho, J. Alves, N. Ebecken, "Optimization of mass concrete construction using genetic algorithms" , *Computers & Structures* 82 (2004) 281-299.
- [3.49] D.W. Fanjoy, W.A. Crossley, "Topology design of planar cross-sections with a genetic algorithm: Part 1 - Overcoming the obstacles", *Engineering Optimization*, 34-1 (2002) 30-49.
- [3.50] D.W. Fanjoy, W.A. Crossley, "Topology design of planar cross-sections with a genetic algorithm: Part 2 - Bending, torsion and combined loading applications", *Engineering Optimization*, 34-1(2002)49-64.
- [3.51] C.A. Felippa, "A historical outline of matrix structural analysis: a play in three acts", *Computers & Structures* 19-U (2001) 1313-1324.
- [3.52] C.W Feng, L. Liu, S.A. Bums, "Using Genetic Algorithms to Solve Construction Time-Cost Trade-Off Problems", *Journal of Computing in Civil Engineering*, 10-3 (1999) pp. 184-189.
- [3.53] X-T. Feng, C. Yang, "Coupling recognition of the structure and parameters of non-linear constitutive material models using hybrid evolutionary algorithms", *International Journal for Numerical Methods in Engineering*, 59 (2004) 1227-1250.
- [3.54] D.M. Frangopol (Ed.), "Optimal performance of Civil infrastructure systems", Proceedings of International workshop on Optimal Performance of Civil Infrastructure Systems, ASCE, 1998 222 p.
- [3.55] T. Furukawa, "Parameter identification with weightless regularization", *International Journal for Numerical Methods in Engineering*, 52 (2001) 219-238.
- [3.56] T. Furukawa, G. Yagawa, "Inelastic constitutive parameter Identification using an evolutionary algorithm with continuous individuals", *International Journal for Numerical Methods*

in *Engineering*, 40 (1997) 1071-1090.

[3.57] M. Galante, "Genetic Algorithms as an approach to optimize real-world trusses", *International Journal Numerical Methods Engineering*, 39 (1996) pp. 361-382.

[3.58] M. Galante, "Un algoritmo genético simple para la optimización de estructuras planas articuladas", *Métodos numéricos para cálculo y diseño en ingeniería*, 9-2 (1993) pp 179-200.

[3.59] M. Galante, M. Cerrolaza, "Un algoritmo genético para optimizar estructuras esbeltas reales", *II Congreso de Métodos Numéricos en Ingeniería*, La Coruña, (1993).

[3.60] A. George, J. W. Liu, *Computer Solution for Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[3.61] J.S. Gero, S.J. Louis, "Improving Pareto Optimal Designs using Genetic Algorithms", *Microcomputers in Civil Engineering*, 10-4 (1995) pp. 241-249.

[3.62] M.R. Ghasemi, E. Hinton, R.D. Wood, "Optimization of trusses using genetic algorithms for discrete and continuous variables", *Engineering Computations*, 16-2,3 (1999) 272-301.

[3.63] L. Gil, A. Andreu, "Shape and cross-section optimization of a truss structure", *Computers & Structures* 79 (2001) 681-689.

[3.64] D.E. Goldberg, M.P. Samtani, "Engineering Optimization via genetic algorithm". Proceedings Ninth Conference on Electronic Computation, ASCE, New York, NY, (1986), pp. 471-482.

[3.65] R.V. Grandhi, G. Bharatram, V.B. Venkayya, "Multiobjective Optimization of large-scale structures", *Journal*, 31-7 (1993) 1329-1337.

[3.66] D. Greiner, G. Winter, J.M. Emperador, B. Galván. 'A Comparative Analysis of Controlled Elitism in the NSGA-II applied to Frame Optimization'. In *Solid Mechanics and its applications* n° 116, IUTAM Symposium on Evolutionary Methods in Mechanics, Kluwer Academic Publishers, (2004) pp. 101-110.

[3.67] D. Greiner, G. Winter, J.M. Emperador, "Aplicación de los Algoritmos Genéticos en Problemas de Optimización de Estructuras de Pórticos Planos y Nudos Rígidos", Proyecto Fin de Carrera, ETSI Industriales, ULPGC (2000).

[3.68] D.E. Grierson, W.H. Pak, "Optimal sizing, geometrical and topological design using a genetic algorithm", *Structural Optimization*, 6-3 (1993) 151-159.

[3.69] A.A. Groenwold, N. Stander, J.A. Snyman, "A regional genetic algorithm for the discrete optimal design of truss structures", *International Journal for Numerical Methods in Engineering*, 44 (1999) 749-766.

[3.70] P. Gurfil, N.J. Kasdin, "Niching genetic algorithms-based characterization of geocentric orbits in the 3D elliptic restricted three-body problem", *Comput. Methods Appl. Mech Engrg*, 191 (2002) 5683-5706.

[3.71] W. Gutkowski, Z. Iwanow, J. Bauer, "Controlled mutation in evolutionary structural optimization", *Structural and Multidisciplinary Optimization*, 21-5 (2001) 355-360.

[3.72] M.N.S. Hadi, Y. Arfiadi, "Optimum rigid pavement design by genetic algorithms", *Computers & Structures* 79 (2001) 1617-1624.

[3.73] P. Hajela, E. Lee, H. Cho, "Genetic Algorithms in Topologic Design of Grillage Structures", *Computer-Aided Civil and Infrastructure Engineering*, Vol 13, pp. 13-22 (1998).

[3.74] P. Hajela, C.Y. Lin, "Genetic search strategies in multicriterion optimal design", *Structural Optimization*, 4 (1992) 99-107.

- [3.75] P. Hajela, C.J. Shih, "Multiobjective optimum design in mixed integer and discrete design variable problems", *AIAA Journal*, 28-4 (1990) pp. 670-675.
- [3.76] G.R. Harik, D.E. Goldberg, "Linkage learning through probabilistic expression", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 295-310.
- [3.77] S.D. Harris, L. Elliott, D.B. Ingham, M. Pourkashanian, C.W. Wilson, "The optimisation of reaction rate parameters for chemical kinetic modelling of combustion using genetic algorithms", *Comput. Methods Appl. Mech Engrg*, 190 (2000) 1065-1090.
- [3.78] O. Hasancebi, F. Erbatur, "Evaluation of crossover techniques in genetic algorithm based optimum structural design", *Computers & Structures* 78 (2000) 435-448.
- [3.79] O. Hasancebi, F. Erbatur, "Constraint handling in genetic algorithm integrated structural optimization". *Acta Mechanica*, 139-1-4, (2000) pp. 15-31.
- [3.80] M.S. Hayalioglu, "Optimum design of geometrically non-linear elastic-plastic steel frames via genetic algorithm", *Computers & Structures* 77 (2000) 527-538.
- [3.81] S. Hernández, "Post-optimal procedures for structural optimization", *Advances in Engineering Software*, 31-6 (2000) 401-409.
- [3.82] S. Hernández, "Post-optimal procedures for structural optimization", *Computer Aided Optimum Design of Structures V*, (1997) 85-100.
- [3.83] S. Hernández, "Del diseño convencional al diseño óptimo. Posibilidades y variantes. Parte I. Análisis de sensibilidad y optimización local y global". *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, Vol 9-1,(1993) pp. 91 -110.
- [3.84] S. Hernández. 'Métodos de diseño óptimo de estructuras'. Colección Seinor. Colegio de Ingenieros de Caminos, Canales y Puertos (Madrid). (1990).
- [3.85] A. Hernández Aguirre, S. Botello, C. Coello, G. Lizárraga, E. Mezura-Montes, "Handling constraints using multiobjective optimization concepts", *International Journal for Numerical Methods in Engineering*, 59 (2004) pp. 1989-2017.
- [3.86] O. Hrstka, A. Kucerová, M. Leps, J. Zeman, "A competitive comparison of different types of evolutionary algorithms", *Computers & Structures* 81 (2003) 1979-1990.
- [3.87] M-W. Huang, J.S. Arora, "Optimal design with discrete variables: some numerical experiments", *International Journal for Numerical Methods in Engineering*, 40 (1997) 165-188.
- [3.88] M. J. Jakiela, C. Chapman, J. Duda, A. Adewuya, K. Saitou, "Continuum structural topology design with genetic algorithms", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 339-356.
- [3.89] K. Jarmai, J.A. Snyman, J. Farkas, G. Gondos, "Optimal design of a welded I-section frame using four conceptually different optimization algorithms", *Structural and Multidisciplinary Optimization*, 25-1 (2003) 54-61.
- [3.90] W.M. Jenkins, "A decimal-coded evolutionary algorithm for constrained optimization" , *Computers & Structures* 80 (2002) 471-480.
- [3.91] W.M. Jenkins, "On the Application of Natural Algorithms to Structural Design Optimisation", *Engineering Structures*, 19-4 (1997) pp 302-208.
- [3.92] W.M. Jenkins, "Towards structural optimization via the genetic algorithm" , *Computers & Structures* 40 (1991) 1321-1327.
- [3.93] R.J. Jiang, F.T.K. Au, Y.K. Cheung, "Identification of masses moving on multi-span beams based on a genetic algorithm", *Computers & Structures* 81 (2003) 2137-2148.

- [3.94] E.S. Kameshki, M.P. Saka, "Optimum design of nonlinear steel frames with semi-rigid connections using a genetic algorithm", *Computers & Structures* 79 (2001) 1593-1604.
- [3.95] H. Kargupta, S. Bandyopadhyay, "A perspective on the fomidation and evolution of the linkage learning genetic algorithms", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 269-294.
- [3.96] C.L. Karr, D.A. Stanley, B. McWhorter, "Optimization of hydroclone operation using a geno-fuzzy algorithm", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 517-530.
- [3.97] A. Kaveh, V. Kalatjari, "Topology optimization of trusses using genetic algorithm, forcé method and graph theory", *International Journal for Numerical Methods in Engineering*, 58 (2003) 771-791.
- [3.98] A. Kaveh, V. Kalatjari, "Genetic algorithm for discrete-sizing optimal design of trusses using the forcé method", *International Journal for Numerical Methods in Engineering*, 55 (2002) 55-72.
- [3.99] H. Kawamura, H. Ohmori, N. Kito, "Truss topology optimization by a modified genetic algorithm", *Structural and Multidisciplinary Optimization*, 23-6 (2002) 467-472.
- [3.100] M. Keser, S. Stupp, "Genetic algorithms in computational materials science and engineering: simulation and design of self-assembling materials", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 373-385.
- [3.101] S. Khajepour, D. Grierson, "Conceptual Design using Adaptive Computing", *Proceedings of the Genetic and Evolutionary Computation Conference* (2001), pp. 62-67.
- [3.102] N.S. Khot, "Multicriteria optimization for design of structures with active control", *Journal of Aerospace Engineering*, 11-2 (1998) 45-51.
- [3.103] H. Kim, H. Adeli, "Discrete cost optimization of composite floors using a floating-point genetic algorithm", *Engineering Optimization*, 33-4 (2001) 485-501.
- [3.104] Y.J. Kim, J. Ghaboussi, "A New Method of Reduced Order Feedback Control using Genetic Algorithms", *Earthquake Engineering and Structural Dynamics*, 28-2 (1999) pp. 193-212.
- [3.105] J.E. Kim, N. Sarigul-Klijn, "Elastic-dynamic rotor blade design with multiobjective optimization", *JZ4^ Journal*, 39-9 (2001) 1652-1661.
- [3.106] D.E. Knuth. 'El Arte de programar ordenadores' VII volúmenes. (1980).
- [3.107] J. Koski, "Multicriteria optimization in structural design: state of the art", in: *Proceedings of the 7P^ Design Automation Conferences ASME*, (1993) pp. 621-629.
- [3.108] J. Koski, "Truss Optimization with Vector Criterion" Report n° 6, Tampere University of Technology, Tampere, Finland (1979).
- [3.109] J. Koski, "Truss Optimization with Vector Criterion. Examples" Report n° 7, Tampere University of Technology, Tampere, Finland (1980).
- [3.110] V.K. Koumouis, P.G. Georgiou, "Genetic Algorithms in Discrete Optimization of Steel Truss Roofs", *Journal of Computing in Civil Engineering*, 8-3 pp. 309-325.
- [3.111] J.R. Koza, F.H. Bennett III, D. Andre, M.A. Keane, "Synthesis of topology and sizing of analog electrical circuits by means of genetic programming", *Comput. Methods Appl. Mech Engrg*, 186(2000)459-482.
- [3.112] C.S. Krishnamoorthy, "Structural optimization in practice: Potential applications of genetic algorithms", *Structural Engineering and Mechantes*, 11-2 (2001) 151-170.
- [3.113] E.M. Krokosky, "The ideal multifunctional constructural material", *Journal of the Structural División ASCE* 94 (1968) 958-981.

- [3.114] S. Kundu, "A Multicriteria Genetic Algorithm to Solve Optimization Problems in Structural Engineering Design", *Information Processing in Civil and Structural Engineering Design*, (1996), Civil-Comp Press, pp. 225-233.
- [3.115] A. Kurapati, S. Azarm, "Immune Network Simulation with Multiobjective Genetic Algorithms for Multidisciplinary Design Optimization", *Engineering Optimization*, 33 (2000) pp. 245-260.
- [3.116] N. Lagaros, M. Papadrakakis, G. Kokossalakis, "Structural optimization using evolutionary algorithms", *Computers & Structures* 80 (2002) 571-589.
- [3.117] D. Lee, "Multiobjective design of a marine vehicle with aid of design knowledge" , *International Journal for Numerical Methods in Engineering*, 40 (1997) 2665-2677.
- [3.118] J.P. Leite, B.H.V. Topping, "Improved Genetic Operators for Structural Engineering Optimization", *Advances in Engineering Software*, 29-7/9 (1998) 529-562.
- [3.119] G. Leitmann, "Some problems of scalar and vector-valued optimization in linear viscoelasticity", *Journal Optim. Theory Appl.* 23 (1977) pp. 93-99.
- [3.120] A.C.C. Lemonge, H.J.C. Barbosa, "An adaptive penalty scheme for genetic algorithms in structural optimization", *International Journal for Numerical Methods in Engineering*, 59 (2004) 703-736.
- [3.121] X.S. Li, G.D. Cheng, D.Y. Changsong, "A central path-following algorithm for multi-objective optimization of large scale structures", *Proceedings of Structural and Mechanical systems*, Xidian University Press, (1999) 453-460.
- [3.122] Q.S. Li, D.K. Liu, A.Y.T. Leung, N. Zhang, Q.Z. Luo, "A multilevel genetic algorithm for the optimum design of structural control systems", *International Journal for Numerical Methods in Engineering*, 55 (2002) 817-834.
- [3.123] Q. Li, G.P. Steven, Y.M. Xie, "Evolutionary thickness design with stiffness maximization and stress minimization criteria", *International Journal for Numerical Methods in Engineering*, 52 (2001) 979-995.
- [3.124] X. Liu, D.W. Begg, "Second-order sensitivity of Smart Structures", *Journal of Aerospace Engineering*, 12-1 (1999) 15-22.
- [3.125] X. Liu, D.W. Begg, R.J. Fishwick, "Genetic approach to optimal topology/control design of adaptive structures", *International Journal for Numerical Methods in Engineering*, 41 (1998) pp. 815-830.
- [3.126] X. Liu, D.W. Begg, D.R. Matravers, "Optimal Topology/Actuator Placement Design of Structures using GA", *Journal of Aerospace Engineering*, 10-3 (1997) 119-125.
- [3.127] M. Liu, S.A. Bums, Y.K. Wen, "Optimal seismic design of steel frame buildings based on life cycle cost considerations", *Earthquake Engng. Struct.* 32-9 (2003) 1313-1332.
- [3.128] G.R. Liu, S.C. Chen, "Flaw detection in sandwich plates based on time-harmonic response using genetic algorithm", *Comput. Methods Appl. Mech Engrg*, 190 (2001) 5505-5514.
- [3.129] B. Liu, R.T. Hañka, M.A. Akgün, A. Todoroki, "Permutation genetic algorithm for stacking sequence design of composite laminates", *Comput Methods Appl. Mech Engrg*, 186 (2000) 357-372.
- [3.130] G.R. Liu, X. Han, K.Y. Lam, "A combined genetic algorithm and nonlinear least squares method for material characterization using elastic waves", *Comput. Methods Appl. Mech Engrg*, 191 (2002) 1909-1921.

- [3.131] W.-H. Liu and A. H. Sherman. Comparative analysis of the Cuthill-McKee and the reverse CuthillMcKee ordering algorithms for sparse matrices. *SIAM Journal on Numerical Analysis*, 13(2):198-213, Apr. 1976.
- [3.132] D.K. Liu, Y.L. Yang, Q.S. Li, "Optimum positioning of actuators in tall buildings using genetic algorithm", *Computers & Structures* 81 (2003) 2823-2827.
- [3.133] Z. Lounis, M.Z. Cohn, "Multiobjective Optimization of Prestressed Concrete Structures", *Journal of Structural Engineering*, 119-3 (1993) 794-808.
- [3.134] T.A. Lovell, "Application of genetic algorithms to state estimation of tethered systems", *Comput. Methods Appl. Mech Engrg*, 192 (2003) 1799-1819.
- [3.135] S. Marburg, "Developments in structural-acoustic optimization for passive noise control", *Archives of Computational Methods in Engineering*, 9-4 (2002) 291-370.
- [3.136] J.L. Marcelin, "Genetic optimization of stiffened plates without the FE mesh support" , *International Journal for Numerical Methods in Engineering*, 54 (2002) 685-694.
- [3.137] J.L. Marcelin, "Genetic optimization of stiffened plates and shells", *International Journal for Numerical Methods in Engineering*, 51 (2001) 1079-1088.
- [3.138] R.T. Marler, J.S. Arora, "Survey of multi-objective optimization methods for engineering", *Structural and Multidisciplinary Optimization*, 26 (2004) pp. 369-395.
- [3.139] H.C. Martin, G.C. Carey, "Introduction to finite element analysis: theory and application", New York: McGraw Hill; 1973.
- [3.140] K. Matous, M. Leps, J. Zeman, M. Sejnoha, "Applying genetic algorithms to selected topics commonly encountered in engineering practice", *Comput. Methods Appl. Mech Engrg*, 190 (2000) 1629-1650.
- [3.141] M. Matsumoto, J. Abe, M. Yoshimura, "A multiobjective optimization strategy with priority ranking of the design objectives", *Journal of Mechanical Design*, 115-4 (1993) 784-792.
- [3.142] D.S. McCorkle, K.M. Bryden, C.G. Carmichael, "A new methodology for evolutionary optimization of energy systems", *Comput. Methods Appl. Mech Engrg*, 192 (2003) 5021-5036.
- [3.143] M. Metcalf 'FORTRAN optimization'. Academic Press (1985).
- [3.144] Z. Michalewicz, 'Genetic Algorithms + Data Structures: Evolution Programs', Springer Verlag, (1994).
- [3.145] Z. Michalewicz, M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation* 1996; 4(1): 1-32.
- [3.146] K. Miettinen, M. Makela, J. Toivanen, "Numerical comparison of some penalty-based constraint handling techniques in genetic algorithms", *Journal of Global Optimization*, 27-4 (2003) pp. 427-446.
- [3.147] J.C. Miles, G.M. Sisk, C.J. Moore, "The conceptual design of commercial buildings using a genetic algorithm", *Computers & Structures* 79 (2001) 1583-1592.
- [3.148] P. Miltiadou, L.C. Wrobel, "A BEM-based genetic algorithm for Identification of polarization curves in cathodic protection systems", *International Journal for Numerical Methods in Engineering*, 54 (2002) 159-174.
- [3.149] S. Min, S. Nishiwaki, N. Kikuchi, "Unified topology design of static and vibrating structures using multiobjective optimization", *Computers & Structures* 75 (2000) 93-116.
- [3.150] F. Moran Cabré. 'Análisis Matricial de estructuras en ordenadores personales compatibles'.

Editorial Rueda. (1990).

- [3.151] L. Moya Ferrer. 'Análisis Matricial de estructuras de barras' Ediciones UPC (1995).Barcelona.
- [3.152] P.B. Nair, A.J. Keane, R.P. Shimpi, "Combining Approximation Concepts with Genetic Algorithm-Based Optimization", *AISS / ASME / ASCE / AHS / ASC Structures, Structural Dynamics & Materials Conference*, Vol 2, pp. 1741-1751 AIAA, USA (1998).
- [3.153] P. Nanakom, K. Meesomklin, "An adaptive penalty function in genetic algorithms for structural design optimization", *Computers & Structures* 79 (2001) 2527-2539.
- [3.154] S. Narayanan, S. Azarm, "On Improving Multiobjective Genetic Algorithms for Design Optimization", *Structural Optimization*, 18 (1999) pp. 146-155.
- [3.155] E. Nimwegen, J. Crutchfield, "Optimizing epochal evolutionary search: population-size independent theory", *Comput. Methods Appl Mech Engrg*, 186 (2000) 171-194.
- [3.156] Normativa española básica de edificación EA-95.
- [3.157] Normativa española básica de edificación AE-88.
- [3.158] I. Ono, S. Kobayashi, K. Yoshida, "Optimal lens design by real-coded genetic algorithms using UNDX", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 483-497.
- [3.159] E. Oñate. 'Cálculo de estructuras por el método de elementos finitos', CIMNE, (1992).
- [3.160] A. Osyczka, "Multicriteria optimization for engineering design", in: *Design Optimization*, Academic Press (1985) 193-227.
- [3.161] A. Osyczka, S. Krenich, "Evolutionary Algorithms for Multicriteria Optimization with Selecting a Representative Subset of Pareto Optimal Solutions", *First International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science* 1993, (2001) pp. 141-153.
- [3.162] A. Osyczka, S. Kundu, "New Method to solve generalized multicriteria optimization problem using the simple genetic algorithm", *Structural Optimization*, 10-2 (1995) pp. 94-99.
- [3.163] M. Papadrakakis, N. Lagaros, Y. Fragakis, "Parallel computational strategies for structural optimization", *International Journal for Numerical Methods in Engineering*, 58 (2003) 1347-1380.
- [3.164] M. Papadrakakis, N. Lagaros, V. Plevris, "Multi-objective optimization of skeletal structures under static and seismic loading conditions", *Engineering Optimization*, 34-6 (2002) 645-669.
- [3.165] M. Papadrakakis, N. Lagaros, Y. Tsompanakis, V. Plevris, "Large scale structural optimization: Computational methods and optimization algorithms", *Archives of computational methods in engineering*, 8-3 (2001) 239-301.
- [3.166] B. Pichler, R. Lackner, H.A. Mang, "Back analysis of model parameters in geotechnical engineering by means of soft computing", *International Journal for Numerical Methods in Engineering*, 57 (2003) 1943-1978.
- [3.167] R. Picón. 'Resistencia de materiales' E.T.S.I.I. U.L.P.G.C. (1989)
- [3.168] C. Poloni, A. Giurgevich, L. Onesti, V. Pediroda, "Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 403-420.
- [3.169] P. Ponterosso, R. J. Fishwick, D. St. J. Fox, X.L. Liu, D.W. Begg, "Masonry arch collapse loads and mechanisms by heuristically ceded genetic algorithm", *Comput. Methods Appl. Mech Engrg*, 190 (2000) 1233-1243.

- [3.170] H.N. Press, B.P. Flannery, S.A. Teukolsly, W.T. Vetterling. 'Numérica! Recipes in C. Cambridge University Press. (1989).
- [3.171] S.D. Rajan, "Sizing, shape and topology design optimization of trusses using genetic algorithms", *ASCE Journal of Structural Engineering*, 121-10 (1995) 1480-1487.
- [3.172] S. Rajeev, C.S. Krishnamoorthy, "Genetic algorithms-based methodologies for design optimization of trusses", *Journal of Structural Engineering-ASCE*, 123-3 (1997) 350-358.
- [3.173] S. Rajeev, C.S. Krishnamoorthy, "Discrete Optimization of Structures using Genetic Algorithms". *Journal of Structural Engineering*, v.1 18, n° 5, (1992) pp 1233-1250.
- [3.174] I. Rajendran, S. Vijayarangan, "Optimal design of a composite leaf spring using genetic algorithms", *Computers & Structures* 79 (2001) 1121-1129.
- [3.175] S.S. Rao, "Genetic Algorithmic Approach for Multiobjective Optimization of Structures", *ASME Annual Winter Meeting, Structures and Controls Optimization*, vol. AD-38, pp. 29-38 (1993).
- [3.176] S.S. Rao, "Game theory approach for multiobjective structural optimization", *Computers & Structures* 25 (1987) 119-127.
- [3.177] S.S. Rao, K. Sundararaju, B.G. Prakash, C. Balakrishna, "Multiobjective fuzzy optimization techniques for engineering design", *Computers & Structures* 42 (1992) 37-44.
- [3.178] S.S. Rao, K. Sundararaju, B.G. Prakash, C. Balakrishna, "Fuzzy goal programming approach for structural optimization", *AIAA Journal*, 30-5 (1992) 1425-1432.
- [3.179] I. Rechenberg, "Case studies in evolutionary experimentation and computation", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 125-140.
- [3.180] J.T. Richardson, M.R. Palmer, G. Liepins, M. Hilliard. Some guidelines for genetic algorithms with penalty functions. In *Proceedings of the Third International Conference on Genetic Algorithms (ICGA-89)*, San Mateo, CA, Schaffer JD (ed.). George Masón University, Morgan Kaufmann: Los Altos, CA, June 1989; 191-197.
- [3.181] Y. Saad. 'Preconditioned Krylov Subspace Methods'. In "Algorithms for large scale systems. State of the art and applications in Science and Engineering". Kluwer. Editors G. Winter and E. Spedicato. (1997)
- [3.182] M.P. Saka, "Optimum design of pitched roof steel frames with haunched rafters by genetic algorithm", *Computers & Structures* 81 (2003) 1967-1978.
- [3.183] M.P. Saka, "Optimum Design of Grillage Systems using Genetic Algorithms", *Computer Aided Civil and Infrastructure Engineering*, Vol 13, pp. 297-302 (1998).
- [3.184] J. Sakamoto, J. Oda, "Technique for determination of optimal truss layout using genetic algorithm", *Nippon Kikai Gakkai Ronbunshu, a Hen v* 59-562 (1993), pp. 1568-1573.
- [3.185] S.S.S. Sakla, E. Elbeltagi, "Design of steel roofs subjected to drifted snow using genetic optimization", *Computers & Structures* 81 (2003) 339-348.
- [3.186] E. Sandgren, "Multicriteria Design Optimization by Goal Programming", *Advances in Design Optimization*, (1994) pp 225-265.
- [3.187] E. Sandgren, T.M. Cameron, "Robust design optimization of structures through consideration of variation", *Computers & Structures* 80 (2002) 1605-1613.
- [3.188] K.C. Sarma, H. Adeü, "Life-cycle cost optimization of steel structures", *International Journal for Numerical Methods in Engineering*, 55 (2002) 1451-1462.
- [3.189] K.C. Sarma, H. Adeli, "Bilevel parallel genetic algorithms for optimization of large steel

- structures", *Computer-Aided civil and infrastructure engineering*, 16-5 (2001) 295-304.
- [3.190] Herbert Schildt. 'Lenguaje C. Programación Avanzada'. Me Graw Hill. (1988).
- [3.191] M. Schütz, H.P. Schwefel, "Evolutionary approaches to solve three challenging engineering tasks", *Comput. Methods Appl MechEngrg*, 186 (2000) 141-170.
- [3.192] C.J. Shih, T.K. Lai, "Fuzzy weighting optimization with several objective functions in structural design", *Computers & Structures* 52 (1994) 917-924.
- [3.193] A.E. Smith, D.W. Coit. Constraint handling techniques—penalty functions. In: *Handbook of Evolutionary Computation*, Báck T, Fogel DB, Michalewicz Z (eds). Chapter C 5.2. Oxford University Press and Institute of Physics Publishing: Bristol, UK, 1997.
- [3.194] R.E. Smith, B.A. Dike, R.K. Mehra, B. Ravichandran, A. El-Fallah, "Classifier systems in combat: two-sided learning of maneuvers for advanced fighter aircraft", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 421-437.
- [3.195] C.K. Soh, J. Yang, "Optimal Layout of Bridge Trusses by Genetic Algorithms", *Computer-Aided Civil and Infrastructure Engineering*, Vol 13, pp. 247-254 (1998).
- [3.196] G. Soremekun, Z. Gürdal, R.T. Hañka, L.T. Watson, "Composite lamínate design optimization by genetic algorithm with generalized elitist selection", *Computers & Structures* 79 (2001) 131-143.
- [3.197] R. Spallino, G. Thierauf, "Thermal buckling optimization of composite laminates by evolution strategies", *Computers & Structures* 78 (2000) 691-697.
- [3.198] W. Stadler, "Multicriteria optimization in mechanics (a survey)", *Applied Mechantes Review*, 37-3 (1984) pp. 277-286.
- [3.199] W. Stadler, "Natural Structural shapes of shallow arches", *J. Appl. Mech.* 44 (1977) pp. 291-298.
- [3.200] W. Stadler, "Natural Structural shapes (the static case)", *Q. J. Mech. Appl. Math*, 31 (1978) pp. 169-217.
- [3.201] G.P. Steven, Q. Li, Y.M. Xie, "Multicriteria optimization that minimizes máximum stress and maximizes stiffness", *Computers & Structures* 80 (2002) 2433-2448.
- [3.202] G. Sved, L.J. Schmid, A.R. Simpson, "Mínimum Weight Structures designed by Genetic Algorithms", *Computational Mechanics*, Eds: Cheung, Lee & Leung, University of Hong-Kong (1991) pp. 317-322.
- [3.203] J. Sziveri, C.F. Scale, B.H.V. Topping, "An enhanced parallel sub-domain generation method for mesh partitioning in parallel finite element analysis", *International Journal for Numerical Methods in Engineering*, 47 (2000) pp. 1773-1800.
- [3.204] F. Tin-Loi, N.S. Que, "Identification of cohesive crack fracture parameters by evolutionary search", *Comput. Methods Appl MechEngrg*, 191 (2002) 5741-5760.
- [3.205] P. Tiwari, K. Parikh, J. Parikh, "Structural design considerations in house builders model: Optimization approach", *Journal of Infrastructure Systems*, 5-3 (1999) 102-110.
- [3.206] B.H.V. Topping, J.P.B. Leite, "Parallel genetic models for structural optimization", *Engineering Optimization*, 31-1 (1998) 65-99.
- [3.207] M.J. Tumer, H.C. Martin, R.C. Weikel, "Further development and applications of the stiffness method AGARD structures and materials panel", París, France, July 1962. In: *AGARDograph 72: Matrix Methods of Structural Analysis*, Pergamon Press, Oxford 1964. pp. 203

266.

- [3.208] G.N. Vanderplaats, "CONMIN - A FORTRAN program for constrained function minimization. User's Manual", NASA Technical Memorandum TM X-62,282.
- [3.209] P. Van der Linden. 'Expert C Programming. Deep C Secrets'. Prentice Hall (1994).
- [3.210] M. Vose, J.E. Rowe, "Random heuristic search: applications to Gas and functions of unitation", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 195-220.
- [3.211] D.R. Wallace, M.J. Jakiela, W. Flowers, "Design Search under Probabilistic Specifications using Genetic Algorithms", *Computer-Aided Design*, 28-5 pp. 405-421 (1996).
- [3.212] D. S. Weile, E. Michielssen, "The use of domain decomposition genetic algorithms exploiting model reduction for the design of frequency selective surfaces", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 439-458.
- [3.213] E.L. Wilson, "Automation of the finite element method - a historical view", *Finite Element in Analysis and Design*, 13 (1993) pp. 91-104.
- [3.214] G. Winter, C. Conde. *Métodos y algoritmos básicos del álgebra numérica* Reverte. Barcelona (1990).
- [3.215] G. Winter, M. Galán, P. Cuesta and D. Greiner, 'Genetic Algorithms : A Stochastic Improvement Technique. Tools, Skills, Pitfalls and Examples' In '*Genetic Algorithms in Engineering and Computer Science*', John Wiley & Sons, pp. 217-249. (1995).
- [3.216] J. Wu, S. Azarm, "On a new Constraint Handling Technique for Multi-Objective Genetic Algorithms", *Proceedings of the Genetic and Evolutionary Computation Conference* (2001), pp. 47-50.
- [3.217] S-J. Wu, P-T. Chow, "Steady-state genetic algorithms for discrete optimization of trusses", *Computers & Structures* 56 (1995) 979-991.
- [3.218] Y.M. Xie, G.P. Steven, "A simple evolutionary procedure for structural optimization", *Computers & Structures* 49 (1993) 885-896.
- [3.219] Y.G. Xu, G.R. Liu, "Detection of flaws in composites from scattered elastic-wave field using an improved nGA and a local optimizer", *Comput Methods Appl. Mech Engrg*, 191 (2002) 3929-3946.
- [3.220] Y-P. Yang, Y-A. Chen, "Multiobjective optimization of hard disk suspension assemblies: Part II - integrated structure and control design", *Computers & Structures* 59 (1996) 771-782.
- [3.221] Y-P. Yang, C-C. Kuo, "Passive and active design of hard disk suspension assemblies using multiobjective optimization techniques", *Comput. Methods Appl. Mech Engrg*, 145 (1997) 147-166.
- [3.222] Y. Yang, C.K. Soh, "Automated optimum design of structures using genetic programming" , *Computers & Structures* 80 (2002) 1537-1546.
- [3.223] J.P. Yang, C.K. Soh, "Structural optimization by genetic algorithms with tournament selection", *Journal of Computing in Civil Engineering*, 11-3 (1997) 195-200.
- [3.224] I.C. Yeh, "Hybrid Genetic Algorithms for Optimization of Truss Structures", *Microcomputers in Civil Engineering*, 14-3 (1999) pp. 199-206.
- [3.225] C-H. Yeh, J.W. Labadie, "Multiobjective Watershed-Level Planning of Storm-Water Detention Systems", *Journal of Water Resources Planning and Management*, 123-6 (1997) 336-343.
- [3.226] W.H. Zhang, "A compromise programming method using multibounds formulation and dual approach for multicriteria structural optimization", *International Journal for Numerical Methods in*

Engineering, 58 (2003) 661-678.

[3.227] W.H. Zhang, "On the Pareto optimum sensitivity analysis in multicriteria optimization" , *International Journal for Numerical Methods in Engineering*, 58 (2003) 955-977.

[3.228] W.H. Zhang, M. Domaszewski, C. Fleury, "An improving weighting method with multibounds formulation and convex programming for multicriteria structural optimization", *International Journal for Numerical Methods in Engineering*, 52 (2001) 882-902.

[3.229] W. Zhang, H. Yang, "A study of the weighting method for a certain type of multicriteria optimization problem", *Computers & Structures* 79 (2001) 2741-2749.

Capítulo 4. Contribuciones

4.1 Introducción

Se exponen en este capítulo las contribuciones originales de este trabajo.

Las secciones 4.2 y 4.3 se engloban dentro de las aportaciones en algoritmos evolutivos. En la sección 4.2 se presenta una aportación dentro del diseño de algoritmos evolutivos: una adaptación del operador truncamiento encargado de mantener la diversidad a lo largo del frente en el SPEA2 para problemas bifuncionales, con mejora de su eficiencia.

En la sección 4.3 se proponen dos nuevos algoritmos evolutivos multiobjetivo caracterizados por el mantenimiento y promoción de la diversidad poblacional: los algoritmos DENSEA y DENSEA-NSH. Estos algoritmos son especialmente aplicables a problemas de espacio discreto y fi-ente no-dominado de reducido tamaño, y por tanto, de aplicación a las características del problema multicriterio estructural que deseamos resolver (optimización de la masa restringida y del número de distintos tipos de perfiles).

Las siguientes secciones, de la 4.4 a la 4.11 se enmarcan dentro de las aportaciones específicas en la resolución del problema estructural multicriterio de optimización.

Los algoritmos evolutivos han permitido la resolución de este problema multicriterio, realizando una optimización discreta, global y multiobjetivo. Se efectúa en estas secciones un análisis exhaustivo sobre los dos casos test expuestos en el capítulo anterior, considerando: distintas codificaciones (empleando tanto código binario como código Gray; sección 4.4), características (mostrando empíricamente las ventajas del uso del elitismo frente a algoritmos no elitistas; sección 4.5), probabilidades de mutación (se consideran cuatro probabilidades de mutación para cada uno de los casos test ensayados; sección 4.6), tamaños poblacionales (se consideran tres tamaños poblacionales: 50, 100 y 200 individuos) y algoritmos más eficientes (algoritmos: NSGA, NSGAI, SPEA2, NSGAI con elitismo controlado, DENSEA y DENSEA-NSH; sección 4.7). Todo ello con el objetivo de determinar un conjunto de características eficientes en el problema estructural abordado.

Se ha efectuado un análisis comparativo con los dos casos test, y trece algoritmos diferentes, en sus versiones generacional, y además en una versión de cada uno de ellos en modo de estado estacionario o steady-state (NSGA, NSGAI, dos versiones de NSGAI + elitismo controlado, SPEA2, algoritmo propio y algoritmo propio con operador de compartición de aptitud independiente de parámetros). Este estudio ha incluido además, dos codificaciones distintas: código binario frente a código gray, tres tamaños distintos de población (50, 100 y 200 individuos), así como cuatro probabilidades diferentes de mutación (para el caso test de la estructura de cuatro barras: 0.8%, 1.5%, 3% y 6%, para el caso test de la estructura de 55 barras: 0.4%, 0.8%, 1.5% y 3%). Debido al carácter estocástico de los algoritmos evolutivos, y de cara a efectuar una comparativa significativa de los resultados de cada uno de los supuestos considerados, para cada una de las posibles combinaciones, se han efectuado 30 ejecuciones independientes. Todo ello totaliza un total de 2 casos test x 13 algoritmos x 2 codificaciones x 4 probabilidades de mutación x 3 tamaños de población x 30 ejecuciones = 18.720 optimizaciones con algoritmos evolutivos. Teniendo en cuenta que en el caso test menor, cada ejecución implica 20.000 evaluaciones estructurales, y en el caso test mayor, cada ejecución implica 200.000 evaluaciones

estructurales, ello totaliza en conjunto $18.720 \times 110.000 = 2059,2$ millones de evaluaciones estructurales efectuadas.

Además, se ha realizado un análisis comparativo de la multiobjetivización del problema monocriterio de minimización de la masa restringida, empleando como función objetivo adicional el número de distintos tipos de perfiles (sección 4.8).

También se ha estudiado el tratamiento poblacional (generacional frente a estado estacionario; sección 4.9), además de una comparativa específica entre los algoritmos DENSEA y DENSEA-NSH (sección 4.10).

Finalmente se propone el operador renacimiento autoadaptativo para optimización estructural, acompañado de un análisis de sus parámetros, contrastándolo con el operador renacimiento original, además de compararlo con otros algoritmos similares, como el microalgoritmo genético (sección 4.11). Para finalizar la sección y previo a la bibliografía se encuentra un *Anexo* conteniendo aquellas tablas y figuras correspondientes a las secciones anteriores que no se han considerado fundamentales para el desarrollo de la exposición del presente trabajo y que se han preferido agrupar al final de este cuarto capítulo (si bien se ha mantenido la numeración de forma que su disposición es consecutiva de acuerdo a la subsección a la que pertenecen).

4.2 Un operador truncamiento para problemas bifuncionales en el algoritmo SPEA2

4.2.1 Introducción

El énfasis en la eficiencia de los algoritmos multiobjetivo es una de las áreas de gran interés reciente, tal como se señala en Coello [4.9], destacándose entre otros los aspectos del número de comparaciones efectuadas en la ordenación de la población, modos de mantener la diversidad, o procedimientos para reducir el coste computacional de funciones objetivo costosas. En los algoritmos evolutivos multicriterio o multiobjetivo uno de los dos factores principales de adecuado desempeño, conjuntamente con la aproximación hacia el frente óptimo de Pareto, es mantener la diversidad en el entorno de esta frontera, es decir, el reparto homogéneo de soluciones en ella. Dentro de este contexto, se propone en esta sección un desarrollo eficiente del operador truncamiento en el SPEA2, en términos de optimización temporal, aplicado específicamente a problemas multicriterio bifuncionales (con dos funciones objetivo).

4.2.2 Una Propuesta Eficiente del Operador Truncamiento en el SPEA2

La inclusión del operador elitismo en los algoritmos multicriterio ha supuesto que el operador cuya misión es mantener la diversidad a lo largo del frente de soluciones no-dominadas sea un operador 'eliminador'. Éste descarta soluciones candidatas de entre las soluciones no-dominadas obtenidas en cada fase del algoritmo. Esta eliminación pretende dejar aquellas soluciones más homogéneamente repartidas, de acuerdo al criterio empleado por el mismo. El operador truncamiento lleva a cabo esta tarea en el SPEA2 [4.53] [4.54].

En el SPEA2, al igual que en su antecesor, el SPEA, se utiliza una población auxiliar (tamaño $N/2$) donde se almacenan las soluciones no-dominadas obtenidas hasta el momento por el algoritmo. En cada nueva generación la nueva población (tamaño $N/2$) obtenida a partir de las operaciones genéticas de selección, cruce y mutación se combina con esta población externa; de esta población combinada (tamaño N) se extraen las soluciones no-dominadas que son las que conforman la nueva población auxiliar o externa. Si el número de soluciones no-dominadas es menor a $N/2$, se completa esta población auxiliar con los mejores restantes individuos dominados (en términos de no-dominancia). Si el número de soluciones es mayor a $N/2$, entonces se deben eliminar tantas soluciones no-dominadas como excedan de $N/2$, mediante el operador truncamiento.

Cada iteración de eliminación de un individuo i se define de tal modo que el individuo i eliminado cumple que $f(i) < f(j)$ para todo $j \in P$, con:

$$f(i) < f(j) \quad \forall j \in P \quad \text{y} \quad \forall k \in P: (f(k) < f(i) \implies (j = k) \vee \exists t \in P: (f(t) < f(k) \wedge t \neq i))$$

siendo $d(i)$ la distancia de i a su A -ésimo vecino más cercano en P , donde P es la población. De ese modo, el individuo que posee mínima distancia a otro individuo de la población es elegido en cada iteración; si existen varios individuos con menor distancia igual, se tiene en cuenta la segunda menor distancia, y así sucesivamente.

Así pues, siendo L el número de soluciones no dominadas, obtenidas de integrar la población auxiliar o exterior con la población descendiente y cuyo máximo valor será A , el procedimiento es el siguiente:

Se calculan las distancias de cada individuo al resto de individuos $O(Z)$; se ordenan las listas de distancias $O(L \log L)$; la elección de un individuo para eliminación puede ser efectuada en $O(L)$ y la actualización de la lista de distancias tras esa eliminación puede ser realizada en $O(Z)$.

Tal como queda expuesto en [4.53], la complejidad de cálculo de este operador es de $O(Z^A)$ en el caso más desfavorable, si bien la complejidad media es menor, de $O(L \log L)$, determinada por la ordenación de lista de distancias, pues los individuos suelen diferir en la segunda o tercera distancia a lo sumo, y de ese modo es la ordenación de distancias la que gobierna la complejidad del proceso.

En el caso que el problema a optimizar sea bifuncional (con dos funciones que simultáneamente deseamos optimizar, ya sea minimizando o maximizando), es posible la implementación particular de un operador truncamiento que aproveche las ventajas que un espacio bifuncional reducido puede proporcionar. En ese caso, se propone en esta sección, aprovechar la linealidad de la distribución del frente de soluciones no-dominadas obtenido, puesto que en ese caso las soluciones quedan distribuidas topológicamente una a continuación de otra formando una línea. Si se tiene en cuenta esta circunstancia en la operatividad del operador truncamiento, se obtiene un procedimiento más eficiente y de menor tiempo de computación.

Para ello, si ordenamos el conjunto de soluciones no-dominadas candidatas según una de las dos funciones a optimizar, (es igual cual), se obtiene la secuencia de soluciones adyacentes entre sí. De ese

Tal como se expresa en [4.48], determinada por una métrica en el dominio tratado, e.g. la métrica euclídea.

modo, el coste del cálculo de distancias se reduce, puesto que únicamente son necesarias de calcular $L-1$ distancias, pudiéndose deducir el resto de ellas mediante los vínculos de las soluciones entre si y simples sumas de las anteriores. Así, este cálculo es reducido a una operación de ordenación de $O(Z \log Z)$, más un cálculo de distancias de $O(Z-1)$. Igualmente, la ordenación de distancias queda reducida a $O((Z-1) \log(Z-1))$.

Además se cuenta con la ventaja añadida de poseer un control sobre el cálculo del número de distancias que queremos evaluar. Las dos menores distancias de cada individuo están determinadas por las distancias a los individuos contiguos en la línea topológica determinada. Las siguientes dos las determina la suma de éstas con los dos siguientes puntos contiguos y así sucesivamente. Si bien en el caso general es necesario el cálculo de todas las distancias al resto de puntos para la ordenación de las mismas y establecer el criterio de descarte, en este caso es conocido a priori cuáles son las menores distancias por pares. Por tanto, se puede reducir el cálculo enormemente, pues tal como se señala en [4.53], los individuos suelen diferir a lo sumo en la segunda o tercera distancia. Así, si tenemos en cuenta las dos primeras distancias, el cálculo de operaciones para elección de un individuo es de $O(2X)$ y si tenemos en cuenta las cuatro primeras distancias es de $O(4i)$.

Por consiguiente, queda reducido globalmente el coste operacional del operador truncamiento, teniendo en cuenta las consideraciones citadas a $O(2Z \log Z)$.

4.2.3 Resultados Experimentales. Casos Test

Se emplean dos funciones test muy consideradas en optimización multicriterio, para las cuales se ha empleado una codificación paramétrica de 16 bits por variable, una población de 80 individuos tanto para la población auxiliar como la de cálculo, una probabilidad de mutación de 0.03 y de cruce de 1.0. La primera de ellas, de las empleadas por Van Veldhuizen como caso test de frente no convexo. La segunda función, como ejemplo de función con frente discontinuo, empleada por Coello Coello para testear su microalgoritmo genético.

T.MOP2 de Van Veldhuizen: (Frente de Pareto no convexo) [4.43]:

$$\begin{aligned} \min f_1(x) &= 1 - e^{-\sum_{i=1}^{n_1} x_i^2} \\ \min f_2(x) &= 1 - e^{-\sum_{i=1}^{n_2} x_i^2} \end{aligned} \quad (4.1)$$

$-2 < x_i^2 < 2$

MCGA2: (Frente de Pareto discontinuo) [4.10]:

$$\min_{x} f(x) = \sum_{i=1}^n \frac{1}{10 - e^{-2A_i(x_i + X_{i+1})}}$$

$$\min W = 2^{10.8} + 5 \cdot \sin\{X_i\}$$

$$-5 < x_j, X_2, X_3 < 5$$
(4.2)

Se muestra a continuación (figuras 4.1 a 4.4) la evolución media de convergencia de dichas funciones sobre un total de 30 ejecuciones diferentes. Para ello se hace uso de dos métricas; la primera, representativa de la aproximación al frente de Pareto óptimo, MI*; y la segunda, representativa de la dispersión total del mismo, M3*. Ambas están referenciadas en Zitzler [4.55] y se corresponden con el espacio funcional (se consideraron las dos menores distancias de cada individuo en el operador truncamiento). El número total de generaciones considerado es de 375, para un total de 30.000 evaluaciones funcionales.

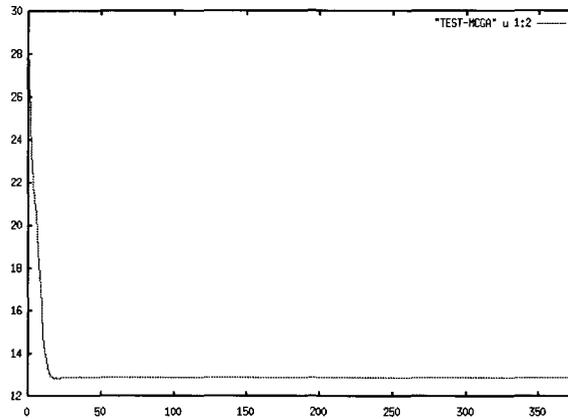


Figura 4.1. Métrica M3 de Función T. MCGA2, promedio de 30 ejecuciones.

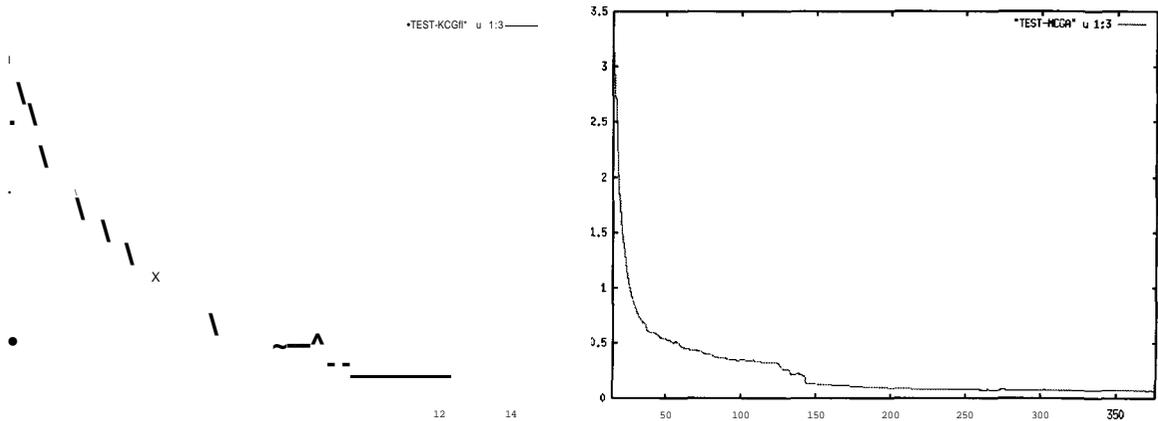


Figura 4.2. Métrica MI de Función T. MCGA2 de generación 0 a 15, y de generación 15 a 375, promedio de 30 ejecuciones.

Capítulo 4. Contribuciones

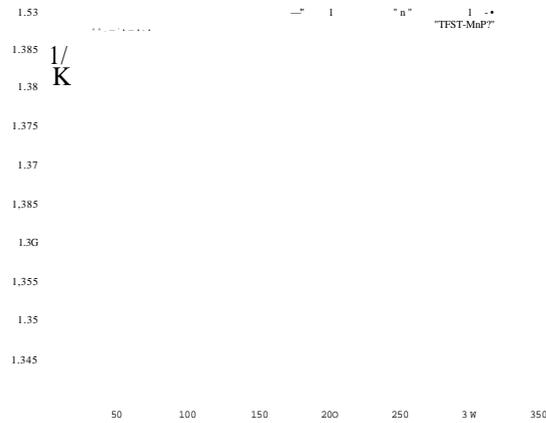


Figura 4.3. Métrica M3 de Función T. MOP2, promedio de 30 ejecuciones.

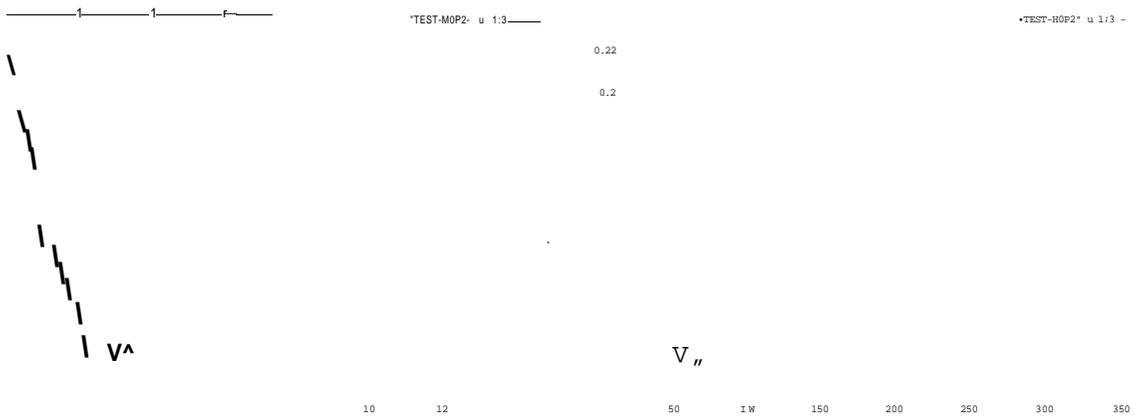
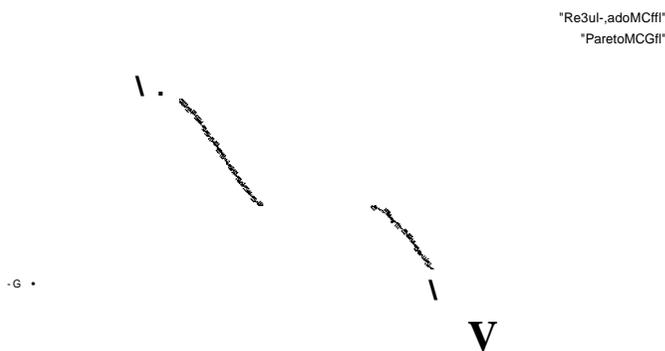


Figura 4.4. Métrica MI de Función T. MOP2 de generación 0 a 15, promedio de 30 ejecuciones.

Se observa cómo el algoritmo evoluciona correctamente hacia el frente de Pareto. También se representa el frente final de una de las ejecuciones para cada función con el óptimo (figuras 4.5 y 4.6).



-15

Figura 4.5. Frente de Pareto Óptimo y solución obtenida por el SPEA2 en la Función T. MCGA2 tras 375 generaciones.

Capítulo 4. Contribuciones

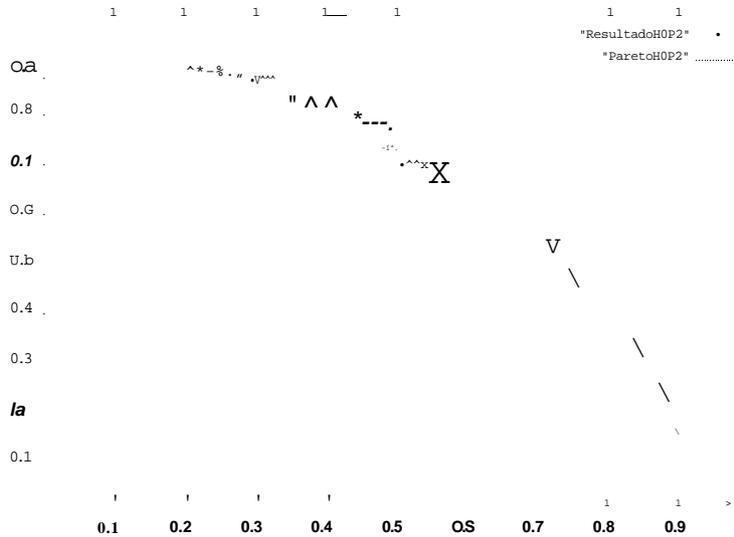


Figura 4.6. Frente de Pareto Óptimo y solución obtenida por el SPEA2 en la Función T. MOP2 tras 375 generaciones.

En estas figuras 4.5 y 4.6 se puede apreciar cómo se alcanza el frente de Pareto óptimo, además de distribuirse homogéneamente las soluciones candidatas finales a lo largo del mismo, pudiéndose observar el correcto comportamiento del operador truncamiento. Todo ello, se puede considerar como muestra del correcto desempeño del algoritmo para la consecución final de soluciones óptimas.

Contrastan los resultados anteriores (alcance y cobertura completa del frente óptimo de forma homogénea) con los obtenidos si se suprime el operador truncamiento del algoritmo SPEA2. En las figuras 4.7 y 4.8 se observan ejecuciones con el mismo numero de evaluaciones de la función objetivo en las que el operador truncamiento se ha anulado. En ellas se aprecia cómo la cobertura del frente deja de ser completa; las soluciones no se distribuyen homogéneamente, y el frente final de soluciones no-dominadas no alcanza el frente óptimo de Pareto.

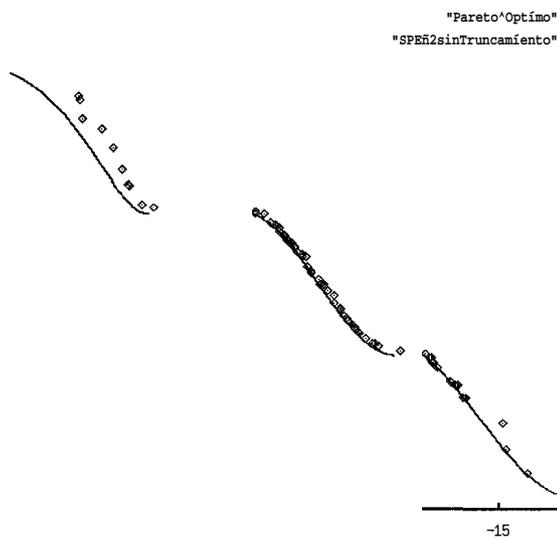


Figura 4.7. Frente de Pareto Óptimo frente a una solución obtenida por el SPEA2 sin operador truncamiento en la Función T. MCGA2 tras 375 generaciones.

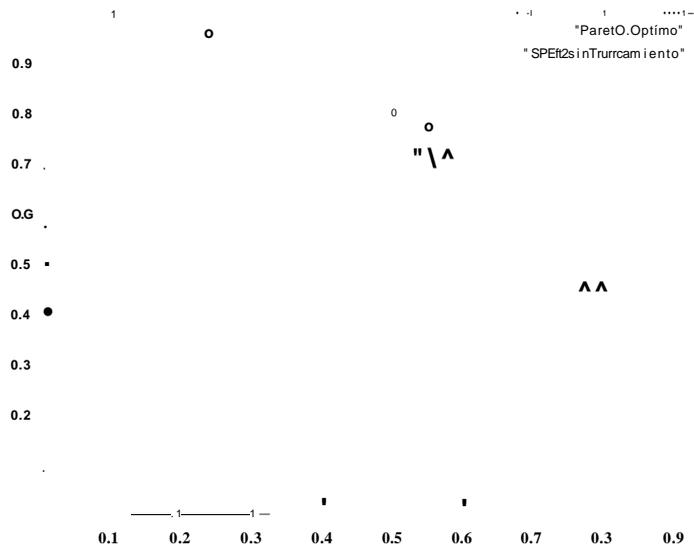


Figura 4.8. Frente de Pareto Óptimo frente a una solución obtenida por el SPEA2 sin operador truncamiento en la Función T. MOP2 tras 375 generaciones.

4.2.4 Conclusiones

En el contexto de búsqueda de soluciones eficientes [4.9] aplicadas en algoritmos evolutivos multicriterio, se ha propuesto en la presente sección una adaptación del operador de truncamiento del SPEA2 en problemas biñfccionales y orientado a reducir su tiempo de cálculo. Configurado el SPEA2 con dicho operador, se han testeados dos ñincciones test clásicas de los algoritmos evolutivos multiobjetivo, mostrando la validez de los resultados obtenidos.

4.3 Algoritmos DENSEA y DENSEA-NSH

4.3.1 Introducción

El estado del arte de los algoritmos evolutivos ha sido comentado en el capítulo anterior (apartado 2.2.3). Entre los algoritmos mostrados más eficaces en la variedad de aplicaciones de diferentes campos, destacan el SPEA2 y el NSGAI. Las características de algoritmos elitistas, regidos por el criterio de no-dominancia de Pareto y con operadores que distribuyan los individuos de la población homogéneamente a lo largo de todo el frente son comunes en ellos.

La inclusión del operador elitismo en los algoritmos multicriterio ha conllevado que el operador encargado del mantenimiento de la diversidad a lo largo del frente se convierta en un operador 'eliminador', y se encargue de ir descartando soluciones candidatas de entre las soluciones no-dominadas obtenidas en cada fase del algoritmo, para dejar aquellas más homogéneamente repartidas de acuerdo al criterio empleado por el mismo. El diseño de estos operadores (operador de 'clustering' en el SPEA, operador truncamiento en el SPEA2, distancia de 'crowding' en el NSGAI, etc.) supone que ante una mayor cantidad de soluciones generadas pertenecientes al mejor frente no-dominado (soluciones no dominadas) que las requeridas para alcanzar el tamaño poblacional deseado, se descartan aquellas más próximas entre sí, conservando en la siguiente generación las que dan una distribución más uniforme en todo el frente.

Esta distribución uniforme sobre un espacio amplio de soluciones numerosas a lo largo de todo un frente parece ser el problema dominante en cuanto se aumenta el número de objetivos a optimizar. Con un número de objetivos creciente, la dimensionalidad del espacio de objetivos se incrementa y el número de soluciones no-dominadas también lo hace, siendo frecuente el que todas las soluciones integrantes de la población sean no-dominadas en esos casos [4.13]. Disponemos de una superficie de Pareto amplia, con gran número de soluciones en ella y donde todos los miembros de la población pueden estar constituidos por soluciones no-dominadas. El problema es la localización de entre un gran número de soluciones posibles, de aquellas que componen la superficie de Pareto de la forma más homogéneamente distribuida posible.

Sin embargo, existen problemas (como el problema estructural multicriterio que se resuelve), donde el número de soluciones integrantes del frente óptimo de Pareto es de menores dimensiones que el tamaño de la población. Este tipo de problemas suele tener una o más funciones objetivo discretas y de valores reducidos, limitando de esa forma la capacidad de cubrir el espacio ñancional homogéneamente por las soluciones. Un espacio discreto con número limitado y reducido de soluciones influye no sólo en el frente final sino en toda la evolución del algoritmo. Es en ese caso cuando la eficacia de los operadores eliminadores anteriormente citados se ve drásticamente limitada, puesto que se produce una acumulación de soluciones en esos puntos discretos sin posibilidad de discriminación. Así, en el caso del problema multiobjetivo estructural que nos ocupa, la función objetivo número de distintos tipos de perfiles es una función discreta, que oscilará como máximo entre 1 y el número de barras de la estructura o el número de distintos tipos de perfiles considerados en la base de datos que sirve de referencia para la codificación del cromosoma (el menor de estos dos últimos valores). Si nos centramos en los casos test considerados, el caso test X tendrá sus posibles valores entre 1 y 4 (número

de barras de la estructura) y el caso test Y entre 1 y 32 (16 tipos distintos de perfiles IPE y 16 tipos distintos de perfiles HEB; número que es menor que el número total de barras de la estructura que es de 55). Los mejores fi^{entes} de Pareto obtenidos incluso reducen estos valores máximos en ambos casos test, reduciéndolos a 3 en el caso test X y a 8 en el caso test Y. Queda ermiarcado claramente entonces el problema estructural dentro de la tipología descrita en el presente apartado.

En este tipo de problemas de optimización parece necesario enfatizar en la creación y mantenimiento de diversidad en la población, para evitar que el escaso número de soluciones no-dominadas sature la misma, entorpeciendo y enlenteciendo su convergencia y conduciendo a una convergencia prematura. Incidiendo en este aspecto, se presenta a continuación el algoritmo DENSEA (Elimination //on-dominated Sorting Evolutionary y4lgorithm; Algoritmo de ordenación no-dominada con eliminación de duplicados).

4.3.2 Algoritmo DENSEA

Se propone aquí un algoritmo multicriterio que permita considerar la creación y mantenimiento de la diversidad poblacional como parte fundamental de su desarrollo, como medio de mejora ante la dificultad expuesta en el apartado anterior (la reducida cantidad de soluciones no-dominadas del espacio fiancional firente al tamaño de la población).

El esquema del pseudoalgoritmo es el siguiente:

```

crear población inicial (P)
evaluar población inicial (P);
ordenNSGA(P);
mientras (NO condición de terminación)
{
    P1 = seleccionar(P);
    P2 = cruzar(P1);
    P3 = mutar(P2);
    evaluar(P3);
    ordenNSGA(P3);
    P4 = nuevapoblación (P,P3);
    ordenNSGA(P4);
    P = P4 ;
    verificar condición de terminación;
}

```

Este algoritmo DENSEA (Elimination A'on-dominated S'orting jEvolutionary ^Igorithm; Algoritmo de ordenación no-dominada con eliminación de duplicados) es un algoritmo basado en el NSGA (puesto que la selección la realiza por el criterio ordenación según los fi-entes de no-dominancia) al que se ha incorporado un tipo de elitismo, y que se caracteriza por ofi-ecer un mantenimiento de la diversidad poblacional basado en varias características del mismo:

- La eliminación de soluciones duplicadas

- El reemplazamiento de esas soluciones duplicadas

- La selección de reemplazamiento de la población de la siguiente generación.

Tras la creación de la población inicial, se evalúan los distintos individuos según las distintas funciones objetivo. La población obtenida se ordena según el criterio de no-dominancia establecido en el algoritmo NSGA, considerando en caso de pertenecer a la misma ordenación del frente un segundo criterio según el operador de distribución a lo largo del mismo (el criterio de compartición de aptitud adoptado se expone más adelante). Tras esta ordenación (ordenación no dominada más compartición de aptitud) se le asigna a cada individuo una probabilidad de selección en función de una selección lineal por ordenación, empleándose la Roulette Wheel Selection (selección por tiro de ruleta) para la determinación del modo de elección de esas probabilidades individuales. En base a ella, se efectúa la selección de individuos para su cruce y posterior mutación, formándose la población filial (P3 en el esquema del pseudoalgoritmo). Esta población es también ordenada según el criterio de no-dominancia más compartición de aptitud.

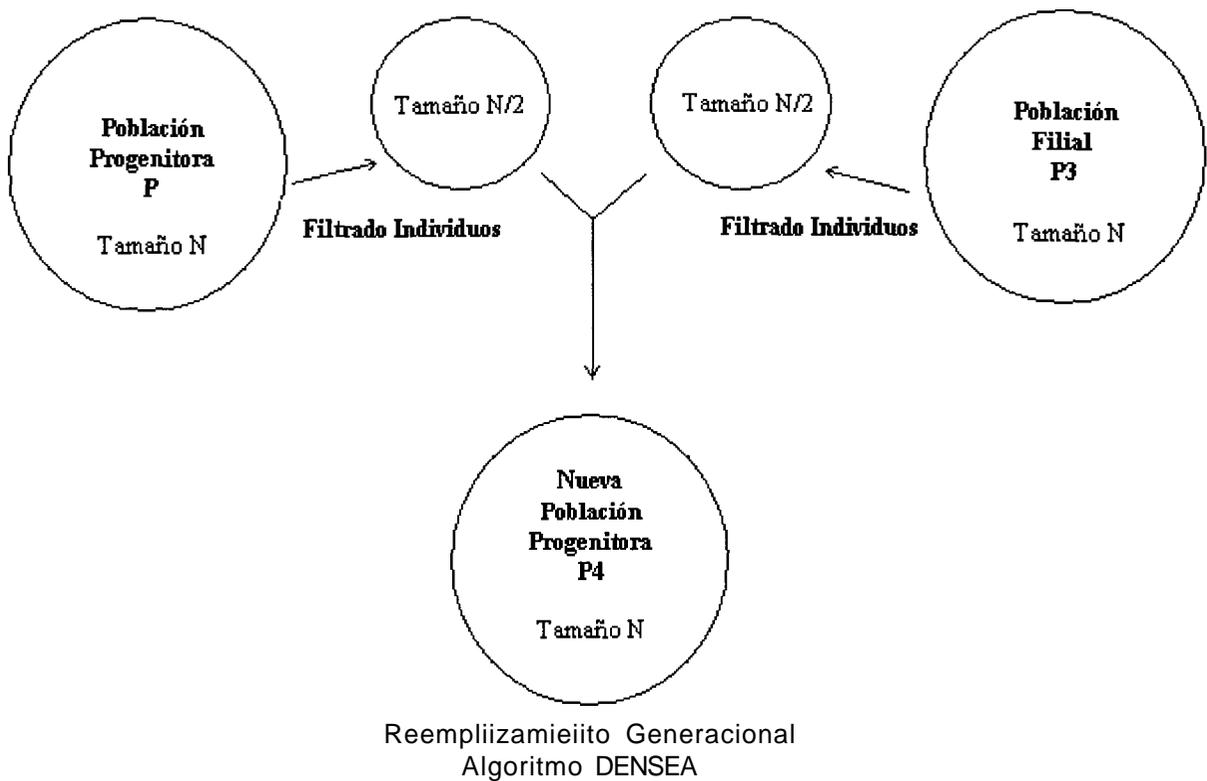


Figura 4.9. Esquema de la selección por reemplazamiento en el algoritmo DENSEA.

El DENSEA incorpora específicamente un operador de eliminación de soluciones duplicadas (el algoritmo multicriterio elimina las posibles soluciones duplicadas acumuladas debido a la reducida cantidad de soluciones no-dominadas del espacio funcional). La sustitución de esa solución duplicada se realiza no por el siguiente individuo de la población en términos de ordenación no-dominada, sino por el individuo de ordenación equivalente de la mitad inferior de la población (que de otro modo no pasaría a formar parte de la siguiente generación) hasta completar el 50% de la población tratada. Así, si se detecta que el tercer individuo es duplicado, éste es sustituido por el individuo $(N/2)+3$, siendo N el tamaño de la población. De este modo se fomenta la inclusión de soluciones diversas que reemplacen a los duplicados. Este proceso de filtrado de soluciones es implementado tanto en la población progenitora (P) como en la recién creada población filial (P3).

A partir de ambas poblaciones (P y P3) se generará la población progenitora de la siguiente generación (P4). Esta población P4 se obtiene a partir de fusionar los individuos obtenidos en el proceso anterior, constituyendo la integración de la mitad de cada una de las poblaciones P y P3 la población siguiente. De esta manera en la selección de los individuos pertenecientes a la población progenitora de la siguiente generación (P), se implementa un elitismo que además garantiza la renovación de un 50% de los individuos en cada generación, fomentando la diversidad poblacional y sin perjuicio de pérdida de información útil de las mejores soluciones (puesto que el número de soluciones no-dominadas es pequeño comparado con el tamaño poblacional típico, por las características del problema). Dicho proceso queda esquematizado en la figura 4.9.

La compartición de aptitud integrante del proceso de ordenación de la población, tras computarse la ordenación de no-dominancia, se ha implementado considerando la distancia fenotípica en el espacio entre variables y no en el espacio funcional, basándose en el esquema descrito en [4.12] para el NSGA. Se efectúa en cada uno de los frentes de no-dominancia obtenidos, para mantener la diversidad a lo largo del mismo. El proceso se describe a continuación. Primeramente, para cada solución i , una distancia euclídea normalizada se evalúa según la ecuación 4.3:

$$d_i = \frac{\sum_{p=1}^P (x_j - x_j^{\wedge})}{P} \quad (4.3)$$

donde P es el número de variables en el problema y x_j y x_j^{\wedge} son los límites superior e inferior de la variable x (en el problema estructural abordado se traduciría en valores de área, inercia, radio de giro, etc.). La compartición de aptitud es evaluada al comparar d_i con un parámetro preespecificado $share$ según la ecuación 4.4:

$$Sh(d_i) = \begin{cases} 1 - \frac{d_i}{share} & \text{si } d_i \leq share \\ 0 & \text{en otro caso.} \end{cases} \quad (4.4)$$

Este proceso es iterado hasta que todos los puntos en el frente son considerados, siendo entonces el contador de subespeciación para la solución i ésima calculado mediante la expresión de la ecuación 4.5:

$$m_i = Y^{Sh(d_i)} \quad (4.5)$$

Este contador es empleado para ordenar las soluciones del mismo frente (cuanto menor es el contador, implica menor número de individuos similares al tratado, por lo que se le debe primar frente a otros con más similitudes). El proceso continúa para cada solución del frente tratado y se extiende a su vez a cada uno de los frentes, permitiendo así la ordenación completa de toda la población.

El parámetro σ^{max} puede ser calculado del modo siguiente [4.12] :

$$0.5$$

$$V_i$$

donde q es el número de soluciones deseados en el frente de Pareto final, y P el número total de parámetros fenotípicos considerados.

4.3.3 Algoritmo DENSEA-NSH

Entre las características comunes de los algoritmos evolutivos multicriterio de segunda generación, destacan el elitismo y la búsqueda de independencia paramétrica en los operadores de mantenimiento de la diversidad a lo largo del frente. Persiguiendo esta independencia paramétrica y aprovechando la circunstancia de que una de las funciones objetivo (el número de diferentes tipos de perfiles) es una función discreta, se ha modificado el operador de mantenimiento de la diversidad en el frente del DENSEA por un operador independiente de parámetros. Este algoritmo modificado es denominado DENSEA-NSH (NSH: No SHaring). El cómputo de la distancia de 'crowding' se realiza considerando únicamente las distancias en términos de la segunda función objetivo (número de distintos tipos de perfiles), con lo que el cálculo de este operador se simplifica sobremanera, además de reducirse en tiempo de computación (concretamente se reduce a simples restas de números enteros). Este operador se basa en la naturaleza discreta del espacio funcional de búsqueda, y por tanto considerará como valor de dispersión entre soluciones a la diferencia del número de distintos perfiles (segunda función objetivo) de las soluciones pertenecientes a su frente en términos de ordenación no-dominada, inmediatamente superior e inferior.

4.4 La Codificación en optimización multiobjetivo estructural: Código Binarlo frente a Código Gray

4.4.1 El Código Gray

La codificación binaria en el problema de optimización estructural abordado, donde las variables son discretas (tipos de perfiles) parece ser la natural. Los números decimales obtenidos de ella serán los que determinen el tipo de perfil a partir de la ordenación en una base de datos correspondiente con las variables dadas por la normativa española de perfiles estandarizados.

Una de las ventajas habitualmente esgrimidas a favor de la representación binaria, es que ésta permite el mayor número de esquemas por unidad de información entre las codificaciones posibles al ser la de menor cardinalidad y que ello resulta muy beneficioso desde el punto de vista teórico de propagación de los bloques de construcción [4.18]. La codificación binaria posibilita igualmente el análisis teórico y la disponibilidad de elegantes operadores genéticos, inspirados en los existentes en la evolución natural.

No obstante, se debe señalar igualmente, que la propiedad de paralelismo implícito de los algoritmos genéticos no es exclusiva de la representación binaria, existiendo para otras cardinalidades alfabéticas de codificación.

Tabla 4.1. Código Binario / Gray para los primeros 16 decimales

Distancia Hamming (Cod. Bin)	Código Binario	Entero equivalente	Código Gray	Distancia Hamming (Cod. Gray)
	0000	0	0000	
1	000i	1	0007	
2	0070	2	0071	
1	0017	3	0010	
3	0100	4	0710	
1	010i	5	0117	
2	01io	6	0101	
1	0117	7	0100	
4	1000	8	7100	
1	1007	9	1107	
2	1070	10	1171	
1	1017	11	1110	
3	1100	12	1010	
1	1107	13	1017	
2	1170	14	1001	
1	1117	15	1000	

Sin embargo, el código binario tradicional adolece de no ser homogéneo con respecto a su equivalente número en representación decimal, empleada esta última para la decodificación. Así por ejemplo.

algunos números enteros consecutivos, caso de los números 7 y 8, tienen representaciones binarias en base dos el 0111 y el 1000, donde ningún alelo coincide entre ambos (lo que es llamado acantilado Hamming o *Hamming Cliff* en el vocabulario de computación evolutiva). Si bien en el espacio fenotípico representan valores similares y cercanos, en el genotípico son completamente opuestos. Es deseable por tanto una representación que mantenga las similitudes existentes tanto en el espacio fenotípico como en el genotípico.

Un *código binario Gray* se define precisamente como una representación con equivalencia biunívoca entre fenotipo y genotipo tal que para cualesquiera dos enteros consecutivos su codificación binaria derive en un único bit o gen de diferencia entre ellos (distancia de Hamming de valor unidad). Se obtiene con él un mapeado del espacio explorado más homogéneo. Permite preservar la estructura vecinal del espacio de búsqueda real discretizado, tal como se señala en Whitley et al. (1996) [4.49].

Veámoslo gráficamente con un ejemplo para las posibles cadenas de cuatro genes en la tabla 4.1.

Estudios experimentales sobre una amplia gama de funciones test usadas en algoritmos genéticos, evidencian que en general se obtienen algoritmos genéticos más eficientes mediante el uso de un código Gray frente a la codificación binaria tradicional (Caruana y Schaffer (1988) [4.6]; Hinterding et al (1989) [4.26]; Mathias y Whitley (1994) [4.33]). Si bien, el teorema NFL (No Free Lunch) de D.H. Wolpert y W.G. MacReady (1995) [4.51] (1997) [4.52] postula que 'todos los algoritmos de búsqueda son equivalentes comparados sobre *todas las posibles funciones discretas*'. Igualmente encontramos en Radcliffe y Surry (1995) [4.37] un teorema similar y de directa aplicación a la comparación de representaciones binaria / gray: 'todos los algoritmos son equivalentes comparados sobre todas las posibles representaciones'. Este caso se refiere por D. Whitley como un claro ejemplo de contradicción entre teoría y práctica en (1999) [4.45], donde se muestra que por ser la codificación influyente en el número de óptimos que el fenotipo genera, se obtienen funciones distintas para un mismo genotipo. Citándolo:

"Due to the No Free Lunch result, since Gray codes induces more óptima than Binary over functions with 2^n óptima, Gray codes induce fewer óptima than Binary over all remaining functions- providing Gray Code with a Free Lunch over a clear and pragmatically defined subset of all possible functions". ("Debido al resultado de No Free Lunch (no hay ventaja), como los códigos Gray inducen más óptimos que los binarios sobre funciones con 2^n óptimos, los códigos Gray inducen menos óptimos que los binarios sobre el resto de funciones, proporcionando al código Gray un Free Lunch (ventaja) sobre un subconjunto claro y pragmáticamente definido de todas las funciones posibles").

El resultado citado se refiere al número de óptimos de la función resultante y ello no necesariamente supone una mayor facilidad de resolución del problema, tal como se indica en U.K. Chakraborty y C.Z. Janikow (2003) [4.7]: Mediante modelización con cadenas de Markov, muestran que puede ocurrir

que una codificación con mayor número de óptimos posea un tiempo de convergencia esperado menor que otra con menor número. En este artículo también se muestra que la eficiencia de la codificación es dependiente de los operadores empleados en la búsqueda.

La codificación Gray es un campo abierto de interés, tal como se puede apreciar en el reciente artículo, de L. Barbulescu et al. (2004) [4.4].

El código Gray no es único, existiendo tantos posibles como combinaciones permitan establecer un mapeado continuo en el espacio genotípico frente al fenotípico. Para transformar una cadena binaria en

una con código Gray, se pueden realizar múltiples posibles transformaciones consistentes en operaciones con matrices.

Se ha optado por tomar de [4.50] la sugerencia que permite una más inmediata transformación. Consiste en transformar la cadena binaria mediante un desplazamiento de un bit a la derecha y efectuar un XOR con esa misma cadena binaria. El cromosoma resultante es el código Gray del original, denominándose Código Gray Estándar Reflejado Binario (K.E. Mathias, D. Whitley (1994) [4.33]). Como ejemplos se muestran varias cadenas transformadas en la tabla 4.2.

Tabla 4.2. Ejemplo Construcción Código Gray

Cadena binaria Cb	0001	0010	0011	0100
Operación XOR con	0001	0010	0011	0100
Resultado : código Gray de Cb	0001	0011	0010	Olio

Esto permite en lenguaje C/C++ un rápido cálculo, siendo el modo optado para la presente implementación computacional. Esta operación basta con realizarla una sola vez y almacenar la equivalencia binario - Gray en un puntero global, que será llamado cuando sea necesario, incrementando aún más la velocidad de ejecución.

La influencia de la codificación en la ordenación de la base de datos y su correspondencia con los tipos de perfiles puede ser vista tanto en la tabla 4.3 como en la figura 4.10 en el caso de los perfiles IPE y en la tabla 4.4 y en la figura 4.11 para el caso de los perfiles HEB. La homogeneidad en la correspondencia entre genotipo y fenotipo es mostrada en términos de dos de las magnitudes geométricas de los perfiles: área y momento de inercia. Figuras análogas a la 4.10 y 4.11 son obtenidas para otras magnitudes geométricas, tal como radio de inercia, área del alma, etc.

Tabla 4.3. Código Binario / Gray considerando codificación de perfiles IPE

Área cm [^]	Código Binario			Entero Equivalente	Cadena	Código Gray		
	Momento de inercia cm [^] * cm ⁴	IPE	Cadena			IPE	Área cm [^]	Momento de inercia cm ⁴
7.64	80.1	80	0000	0	0000	80	7.64	80.1
10.3	171	100	0007	1	0007	100	10.3	171
16.4	541	140	0010	2	0071	120	13.2	318
13.2	318	120	0017	3	0010	140	16.4	541
33.4	2770	220	0100	4	0710	160	20.1	869
28.5	1940	200	0107	5	0117	180	23.9	1320
20.1	869	160	Olio	6	0101	200	28.5	1940
23.9	1320	180	0117	7	0100	220	33.4	2770
116.0	48200	500	1000	8	7100	240	39.1	3890
98.8	33740	450	1007	9	1107	270	45.9	5790
72.7	16270	360	1070	10	1171	300	53.8	8360
84.5	23130	400	1017	11	1110	330	62.6	11770
39.1	3890	240	1700	12	1010	360	72.7	16270
45.9	5790	270	1107	13	1017	400	84.5	23130
62.6	11770	330	1170	14	1001	450	98.8	33740
53.8	8360	300	1117	15	1000	500	116.0	48200

Capítulo 4. Contribuciones

Tabla 4.4. Código Binario / Gray considerando codificación de perfiles HEB

Área cm ²	Código Binario				Código Gray			
	Momento de inercia cm ⁴	HEB	Cadena	Entero Equivalente	Cadena	HEB	Área cm ²	Momento de inercia cm ⁴
26	450	100	0000	0	0000	100	26	450
34	864	120	0007	1	0007	120	34	864
54.3	2490	160	0070	2	0071	140	43	1510
43	1510	140	0017	3	0010	160	54.3	2490
106	11260	240	0100	4	0710	180	65.3	3830
91	8090	220	0107	5	0117	200	78.1	5700
65.3	3830	180	0170	6	0101	220	91	8090
78.1	5700	200	0117	7	0100	240	106	11260
218	79890	450	1000	8	7100	260	118	14920
198	57680	400	1007	9	1107	280	131	19270
171	36660	340	1070	10	1171	300	149	25170
181	43190	360	1017	11	1110	320	161	30820
118	14920	260	1100	12	1010	340	171	36660
131	19270	280	1107	13	1017	360	181	43190
161	30820	320	1170	14	1001	400	198	57680
149	25170	300	1117	15	1000	450	218	79890

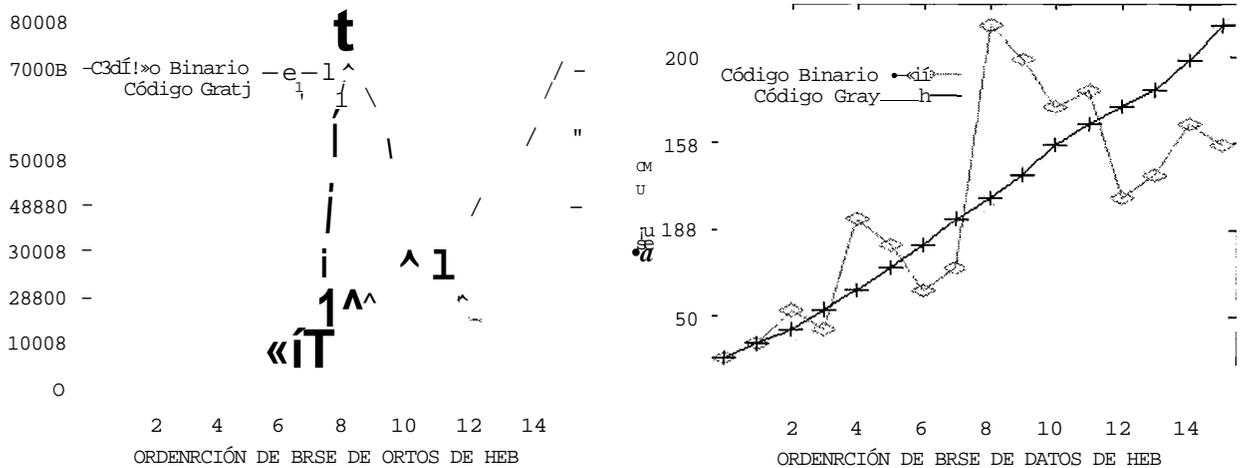


Figura 4.10. Área y Momento de Inercia en la ordenación de la base de datos de perfiles de tipo HEB considerando códigos binarios y Gray.

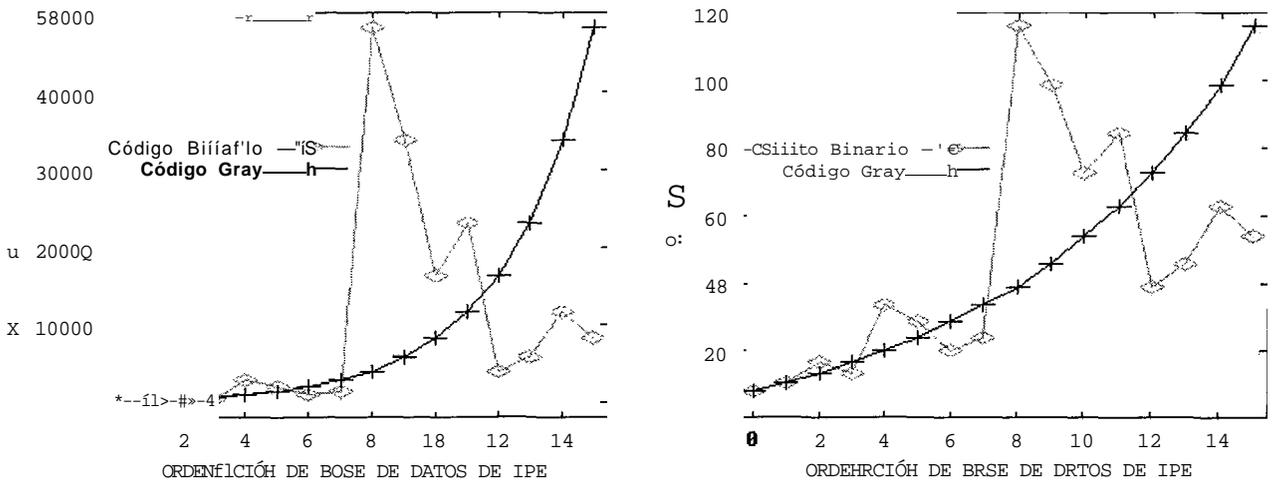


Figura 4.11. Área y Momento de Inercia en la ordenación de la base de datos de perfiles de tipo IPE considerando códigos binarios y Gray.

Si bien el uso del código Gray se ha mostrado beneficioso en optimización monocriterio en el caso del problema de masa restringida [4.25][4.24], aquí se propone el análisis desde el punto de vista de un problema multiobjetivo.

Se ha realizado un análisis exhaustivo comparativo de resultados entre el uso de las codificaciones binaria estándar y el código Gray citado (Código Gray Estándar Reflejado Binario) en los dos casos test estructurales considerados y comparando en cada uno de ellos tres tamaños de población diferentes (50, 100 y 200 individuos), cuatro probabilidades de mutación (entre 0.4% y 6%) y trece algoritmos evolutivos multiobjetivo (NSGA, NSGAI, NSGAI con elitismo controlado y $r=0.0$, NSGAI con elitismo controlado y $r=0.4$, SPEA2, DENSEA, DENSEA-NSH; y sus versiones steady-state: NSGAI, NSGAI con elitismo controlado y $r=0.0$, NSGAI con elitismo controlado y $r=0.4$, SPEA2, DENSEA, DENSEA-NSH). Para cada uno de los casos, se han efectuado 30 ejecuciones independientes para reducir las oscilaciones de resultados inherentes a la característica aleatoria de los algoritmos evolutivos). Los resultados fundamentales obtenidos se muestran en los apartados siguientes, si bien la totalidad puede ser consultada en el *Anexo*.

4.4.2 Caso Test X. Comparativa Código Binario / Gray

En este caso se evaluará para un máximo de 20000 evaluaciones por ejecución, el número de veces que se encuentra el frente óptimo (formado por tres soluciones), siendo evaluados un total de:

- 30 ejecuciones por algoritmo
- X 4 Probabilidades de mutación (0.8%, 1.5%, 3%, 6%)
- X 3 tamaños de población (50, 100, 200)
- X 13 algoritmos

Total: 4680 ejecs

Desglosadas por tamaño poblacional (sobre el número total de ejecuciones por grupo: 1560) se muestran a continuación el número de veces que fue localizado el frente óptimo para este caso test X dentro de las 20.000 evaluaciones consideradas como número máximo para cada caso.

Código <i>Binario</i>		Código <i>Gray</i>	
Población 50:	1045	Población 50:	1078
Población 100:	1119	Población 100:	1107
Población 200:	1182	Población 200:	1155
Total:	3346	Total:	3340

En base al cómputo global, podemos observar resultados similares: 3346 en el caso del código binario frente a 3340 en el caso del código Gray.

Las tablas conteniendo los resultados descompuestos agrupados por algoritmo (tablas de la 4.5 a la 4.17) se pueden encontrar en el *Anexo*. En este caso test, los resultados no muestran ventaja competitiva entre ambas codificaciones, obteniendo incluso el código binario un mayor número de frentes óptimos localizados, si bien la diferencia no parece significativa por su bajo valor.

4.4.3 Caso Test Y. Comparativa Código Binario / Gray

En este caso test Y se considerarán un máximo de 200.000 evaluaciones por ejecución, calculando el valor promedio y la varianza de los dos indicadores de bondad del líente de Pareto (acercamiento al frente y dispersión del frente) a lo largo de toda la ejecución del algoritmo, siendo evaluados un total de:

	30 ejecuciones por algoritmo
X	4 Probabilidades de mutación (0.8%, 1.5%, 3%, 6%)
X	3 tamaños de población (50, 100, 200)
X	13 algoritmos

Total: 4680 ejecs

Las figuras comparativas obtenidas a partir del cálculo de las estadísticas de resultados son las figuras de la 4.12 a la 4.63. La influencia de la codificación es mayor con una mayor longitud del cromosoma, obteniéndose en un mayor espacio de búsqueda los beneficios de uso de la codificación Gray.

Así, puede observarse en la mitad izquierda de las figuras (de la 4.12 a la 4.63), donde se representa el acercamiento al frente, cómo independientemente de la probabilidad de mutación escogida, del tamaño de la población empleada y del algoritmo utilizado, el uso de una codificación Gray posibilita un acercamiento al frente óptimo considerablemente más veloz (pendiente más vertical) y más preciso (valor menor de la métrica) en todas las circunstancias. La varianza de esta métrica también es más baja, presentando mayor robustez con el código Gray.

En cuanto a la amplitud de cobertura del frente, se aprecian los resultados obtenidos en la división derecha de las figuras (de la 4.12 a la 4.63). Se muestra con el fin de clarificar la interpretación de estos resultados, la tabla 4.18 donde se puede observar para cada una de las combinaciones qué

codificación presenta mayor amplitud del frente (G: Gray; B: Binario; ~: no existe ventaja apreciable). En total se obtienen 30 casos donde son similares ambas codificaciones, 61 donde es mejor el código Gray y 65 donde es mejor el código binario. De estas últimas 36 se corresponden con los casos de probabilidad de mutación 3%, que son los que ofrecen peores resultados globales y por tanto, los menos significativos. Sin ellos, los resultados totales serían 61 favorables a Gray, 30 similares y 29 favorables al binario. Considerando la gran ventaja obtenida en la métrica medidora del acercamiento al frente, ello implica que existen 61 casos donde además de un acercamiento más rápido y preciso al frente óptimo, también el código Gray proporciona mayor amplitud de cobertura; 29 donde la amplitud de cobertura es similar y el resto, donde la amplitud de cobertura es menor (aproximadamente entre 0.25 a 0.5 unidades). Esto nos permite concluir que resulta ventajoso también en términos de esta segunda métrica el uso del código Gray.

Tabla 4.18. Resultados Comparativos Globales Caso Test Y Código Gray / Binario de la Amplitud del Frente

Amplitud Frente - Caso Y	50				100				200			
	0.4	0.8	1.5	3.0	0.4	0.8	1.5	3.0	0.4	0.8	1.5	3.0
Gen-NSGA												
Gen-NSGAH	~	G	G	B	G	G	~	B	G	G	—	B
<u>Gen-NSGAIIel.cont.r00</u>	G	G	G	B	G	G	G	B	G	G	—	B
<u>Gen-NSGAIIel.cont.r04</u>	B	—	G	B	G	G	G	B	G	G	—	B
Gen-SPEA2	B	B	G	B	G	G	G	B	G	G	G	B
Gen-DENSEA	B	B	B	B	B	G	G	B	B	—	—	B
Gen-DENSEA-NSH	B	B	G	B	B	B	G	B	B	G	G	B
Std-Stt-NSGAI	~	B	G	B	B	—	—	B	G	G	G	B
<u>Std-Stt-NSGAIel.cont.r00</u>	B	~	G	B	G	G	G	B	G	G	G	B
<u>Std-Stt-NSGAIel.cont.r04</u>	B	B	—	B	~	~	G	B	G	G	G	B
Std-Stt-SPEA2	G	—	G	B	G	G	G	B	G	G	G	B
Std-Stt-DENSEA	B	B	B	B	B	B	B	B	B	—	G	B
Std-Stt-DENSEA-NSH	B	B	G	B	B	B	~	B	G	G	G	B

Se destaca igualmente cómo la importancia de homogeneidad en la correspondencia fenotípica y genotípica aumenta en un problema como el tratado, donde se emplean variables discretas en lugar de una equivalencia binaria - real, puesto que cada una de las variables (perfil real con su área, momento de inercia, etc.) presenta un cambio significativo al incrementarse en tan solo una unidad.

Las ventajas argumentadas que a nivel teórico el código Gray nos podía proporcionar, permitiendo una mayor homogeneidad en la correspondencia entre el espacio genotípico y el espacio fenotípico, y basadas en otros estudios de aplicación en casos de optimización monoobjetivo, quedan corroboradas en este trabajo a través del análisis mostrado de los resultados experimentales obtenidos, en optimización multiobjetivo. De estos resultados se concluye que, el uso de la codificación Gray, permite obtener en promedio menor coste computacional, mayor precisión, obteniendo más frecuentemente frentes más completos y simultáneamente con una mayor robustez en optimización multicriterio de estructuras, acentuándose su efecto ventajoso incrementalmente con el aumento del tamaño del problema. Un resumen de estos resultados puede ser consultado también en [4.21].

A continuación se exponen las figuras de los algoritmos NSGAI y DENSEANSH en sus estrategias generacional y de estado estacionario (el resto puede consultarse en el *Anexo*).

Caso Test Y) BINARIO / GRAY: NSGA-II Mutación 0.4%, 0.8%, 1.5%, 3%

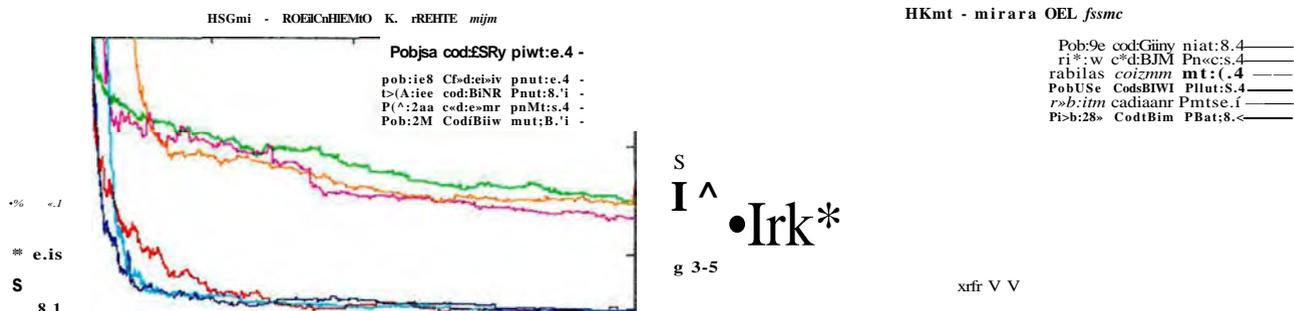


Figura 4.16. Comparativa Código Gray / Binario con NSGAII, Pmut 0.4% y reemplazamiento Generacional en Caso Y

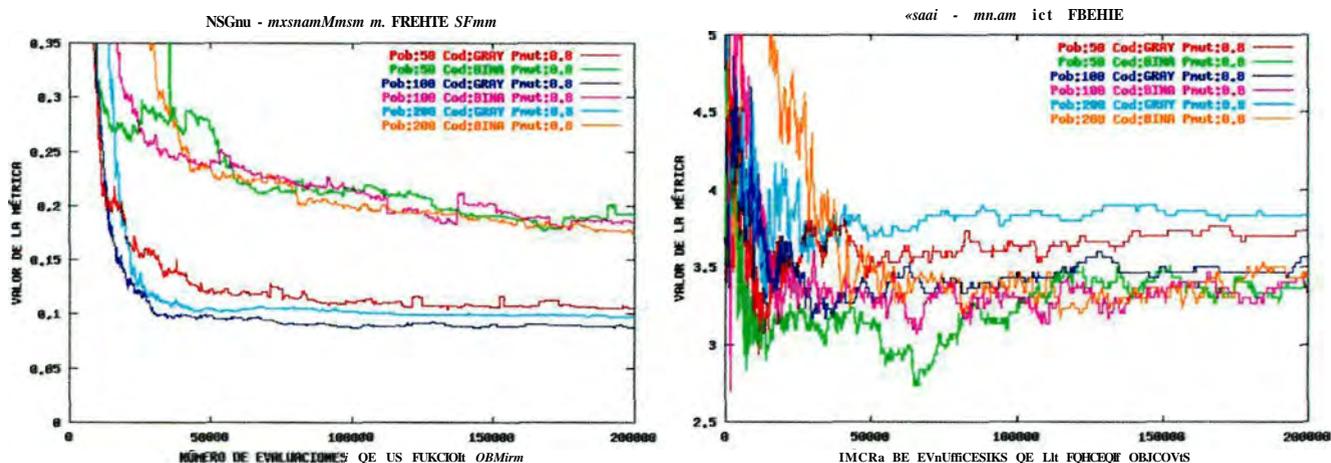


Figura 4.17. Comparativa Código Gray / Binario con NSGAII, Pmut 0.8% y reemplazamiento Generacional en Caso Y

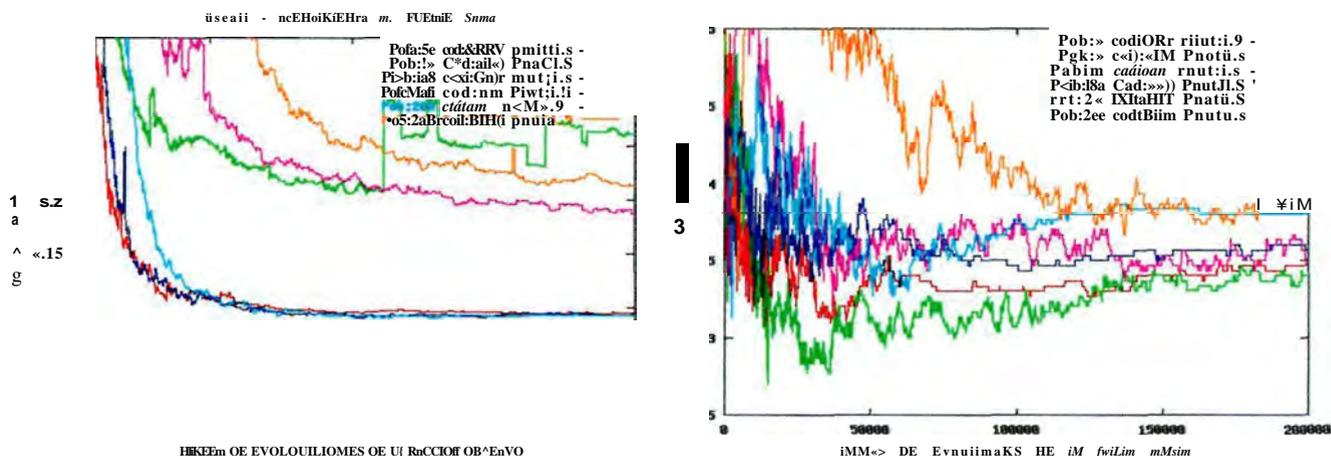


Figura 4.18. Comparativa Código Gray / Binario con NSGAII, Pmut 1.5% y reemplazamiento Generacional en Caso Y

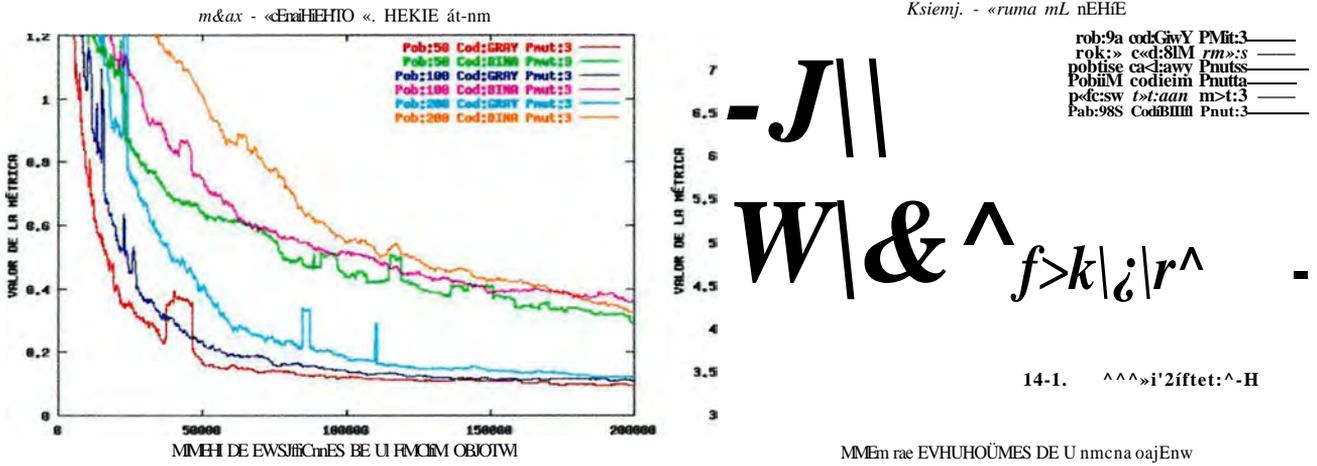


Figura 4.19. Comparativa Código Gray / Binario con NSGAI, Pmut 3.0% y reemplazamiento Generacional en Caso Y

Caso Test Y) BINARIO / GRAY : DENSEA-NSH Mutación 0.4%, 0.8%, 1.5%, 3%

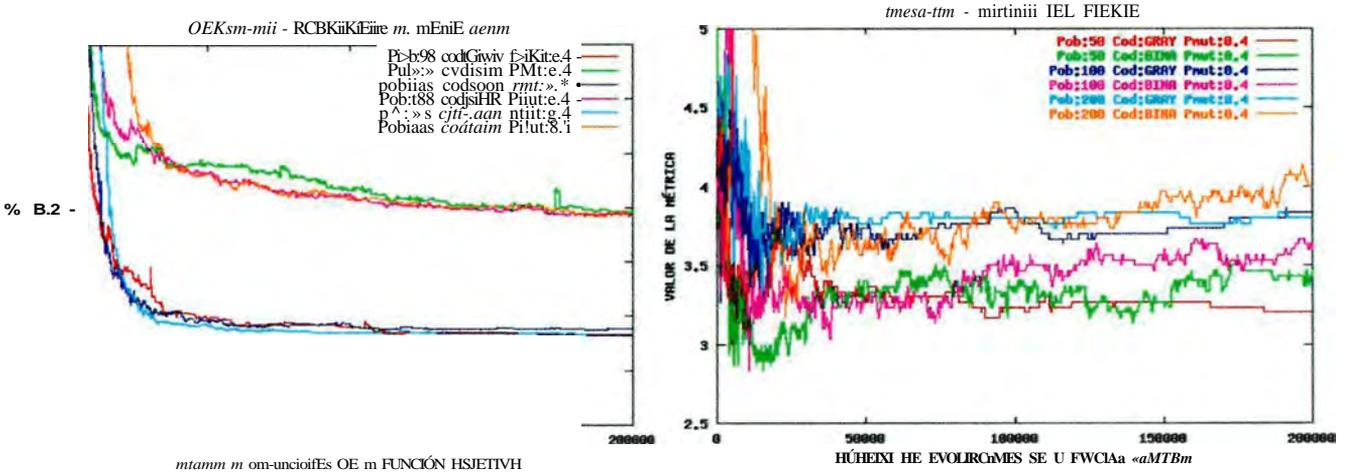


Figura 4.36. Comparativa Código Gray/Binario con DENSEANSH, Pmut 0.4%, reemplazamiento Generacional en Caso Y

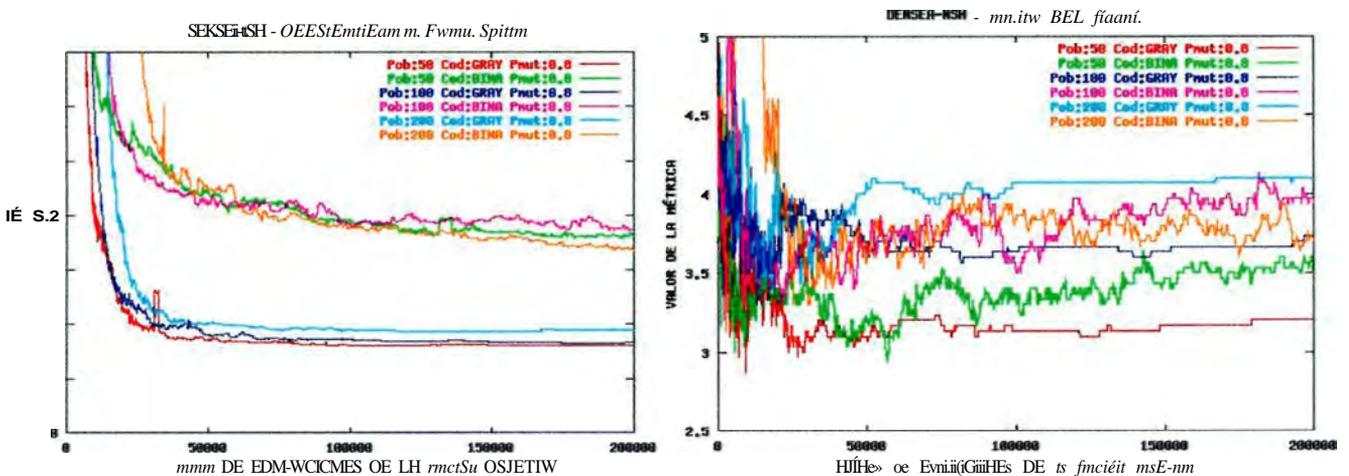


Figura 4.37. Comparativa Código Gray/Binario con DENSEA-NSH, Pmut 0.8%, reemplazamiento Generacional en Caso Y

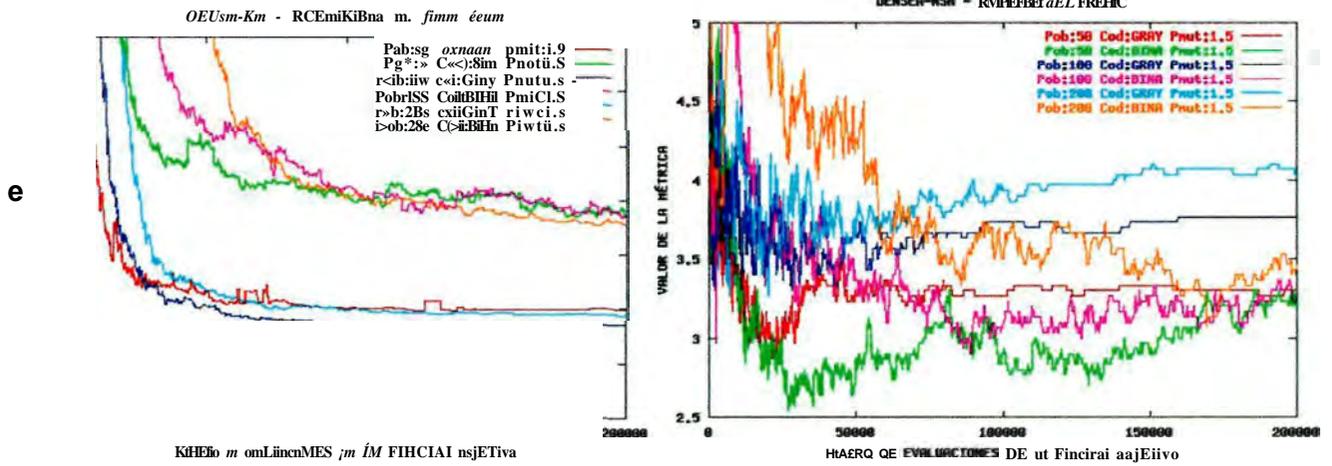


Figura 4.38. Comparativa Código Gray/Binario con DENSEANSH, Pmut 1.5%, reemplazamiento Generacional en Caso Y

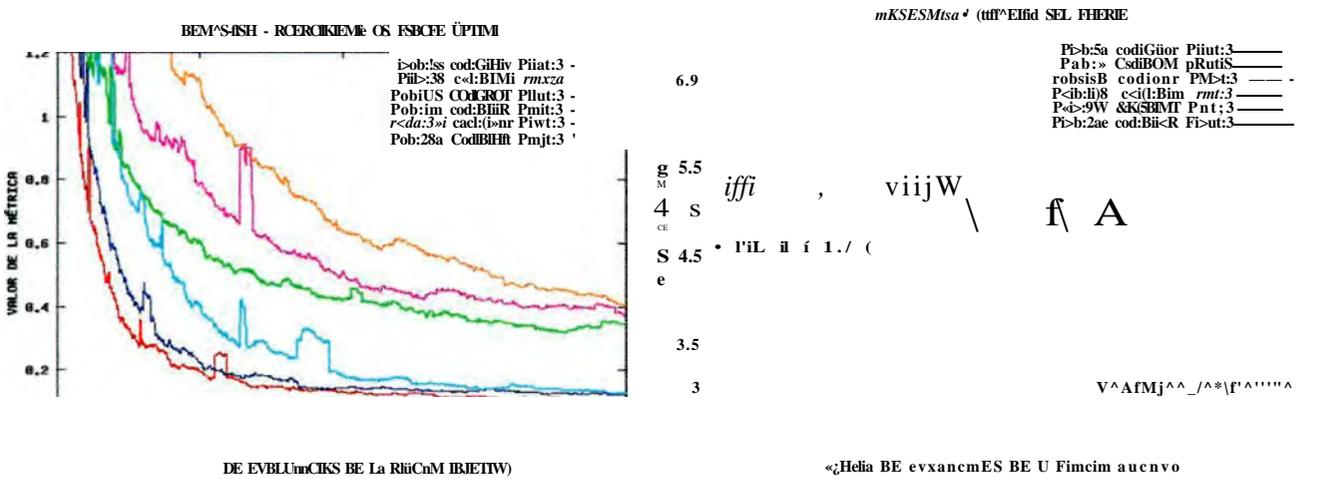


Figura 4.39. Comparativa Código Gray/Binario con DENSEANSH, Pmut 3.0%, reemplazamiento Generacional en Caso Y
Caso Test Y)BINARIO/GRAY: NSGAI SteadyState. Mutación 0.4%, 0.8%, 1.5%, 3%

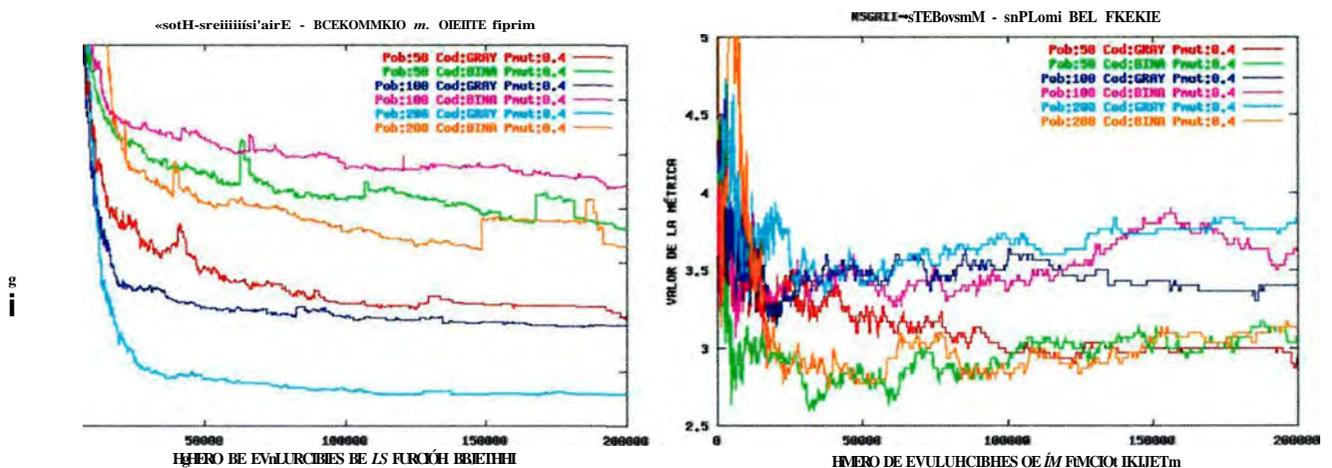


Figura 4.40. Comparativa Código Gray/Binario con NSGAI, Pmut 0.4%, reemplazamiento EstadoEstacionario en Caso Y

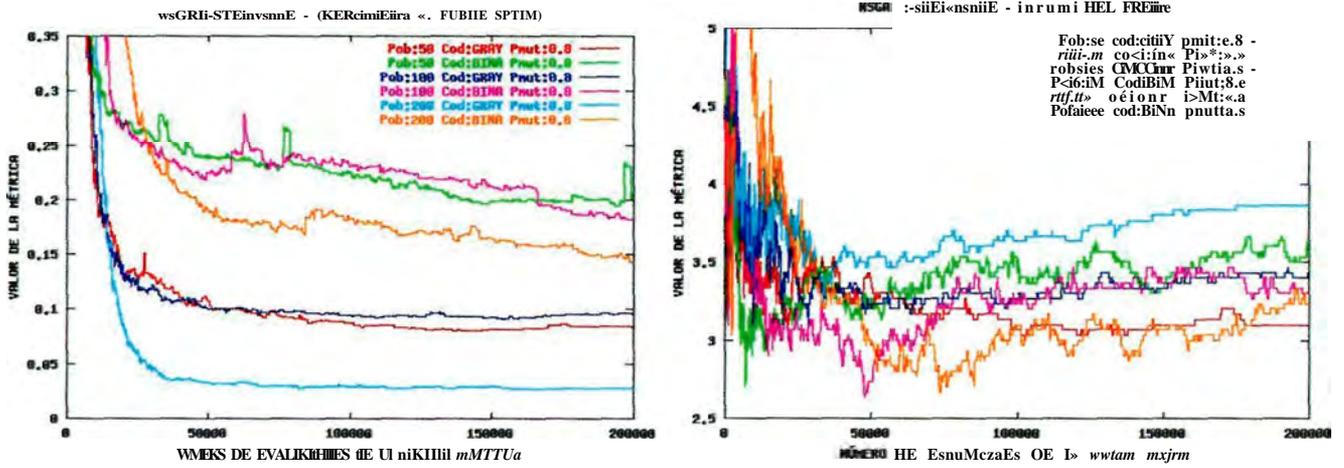


Figura 4.41. Comparativa Código Gray/Binario con NSGAI, Pmut 0.8% y remplazamiento EstadoEstacionario en Caso Y

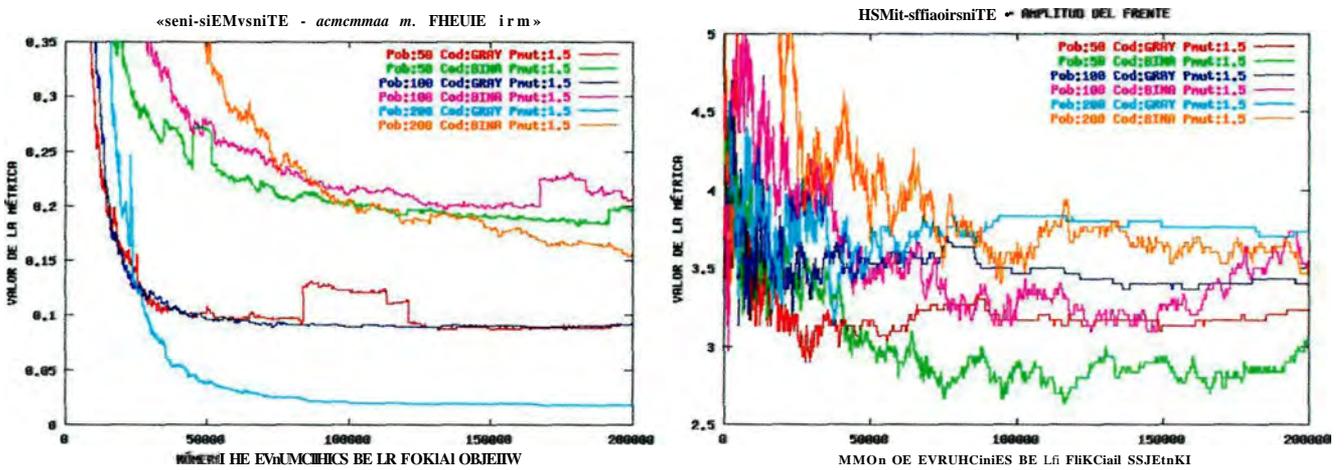


Figura 4.42. Comparativa Código Gray/Binario con NSGAI, Pmut 1.5% y remplazamiento EstadoEstacionario en Caso Y

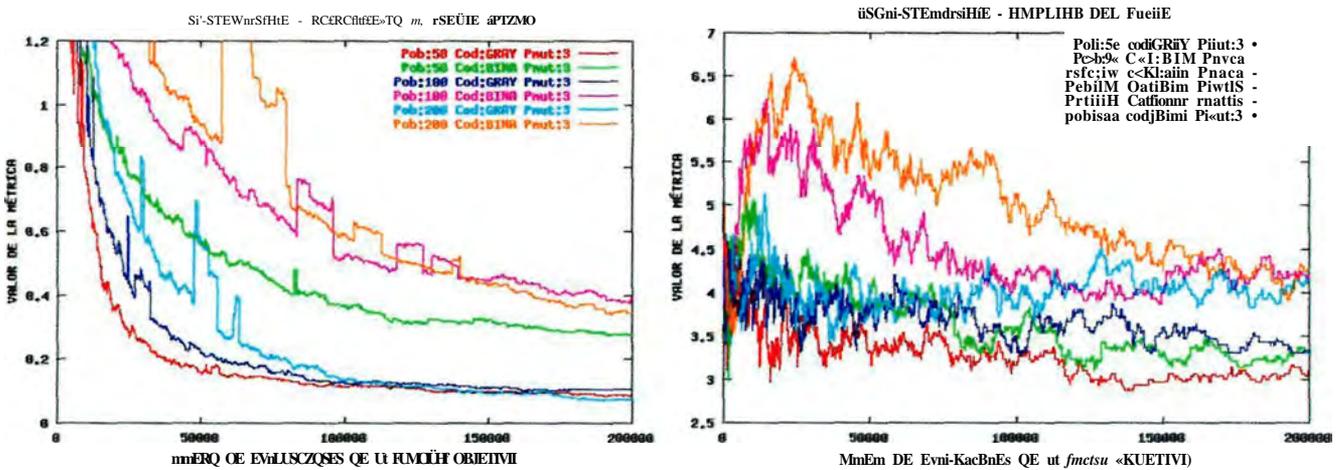
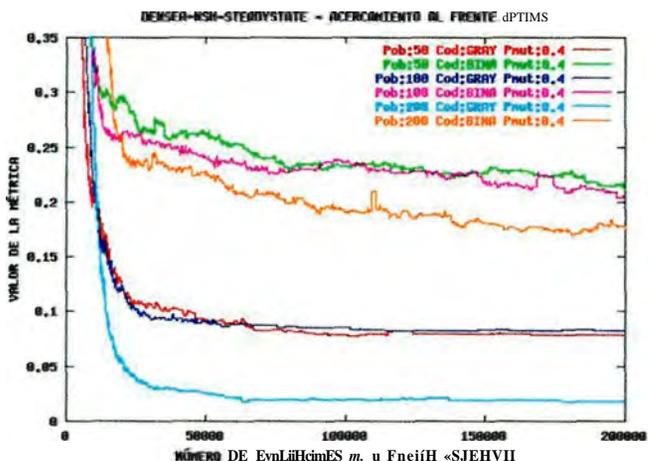


Figura 4.43. Comparativa Código Gray/Binario con NSGAI, Pmut 3.0% y remplazamiento EstadoEstacionario en Caso Y

Caso Test Y)BINARIO/GRAY: DENSEANSH Steady-State Mut 0.4%, 0.8%, 1.5%, 3%



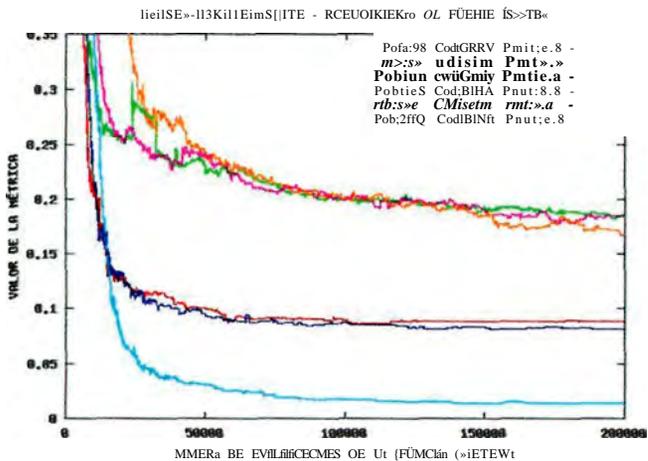
tieRse^«s«-sie«nrsiife - effiojcrffii oe. rsePE

1

Pob:9a cod:Ghrr rmit:».''' _____
 feb:» C«l:BiM nwtji:4 _____
 Polisiae c«l:GinY rmit:».''' _____
 INi«:i8e cad:«iHii Fii«it:a.4 _____
 ra6:iM tai«:aun p«wt:«.4 _____
 Piib:28B cod:Bi«in Piiut:».''' _____

mteia tiE EvnHIGEeMes E : u mmim OBMVWB

Figura 4.60. Comparativa Código Gray / Binario con DENSEA-NSH, Pmut 0.4% y reemplazamiento EstadoEstacionario en Caso Y



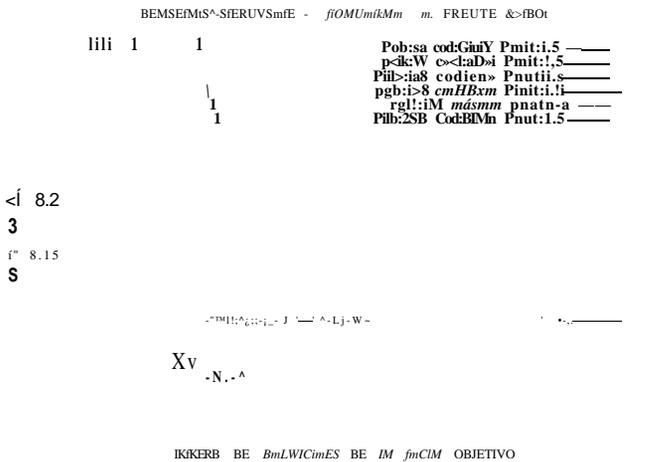
OEKSBMISH-SCCiiInfSnIE - miPLitBI) SEt FritOnE

4.3

pob:se e«GgGrry Pnut:a.B _____
 PiA:M C«d:«im PMit:».''' _____
 Pob:18a c«d:Gnir Fnat:8.8 _____
 Pob:ta8 cod:Bnm Pmit:e.8 _____
 IN*IS«i C«eOBW PnrtiB.I _____
 Pal:2«e ciHiiBlwi p«iut:e.8 _____

HiáEia BE EvnLunncnes BE ts nntcife OBJEfiW

Figura 4.61. Comparativa Código Gray / Binario con DENSEA-NSH, Pmut 0.8% y reemplazamiento EstadoEstacionario en Caso Y



OEKSEIMISH-iiTaiWfSraE - IWPLitIO DEL fUmtr

5

Pof!:» Cod:SGRHY Pnut:1.»' _____
 Pgti:» Clázam Pmi«:1.9 _____
 Piib:i88 cad:Giat pmit:i.s _____
 PobitSa Cod:BiM PnitiU.S _____
 P«>:1»a CMiaarr Pnxtn.s _____
 Pob:2a0 Cod:BiH Pnut:1.5 _____

MMERa BE evsnúioKES CE u pncwiii muejvia

Figura 4.62. Comparativa Código Gray / Binario con DENSEA-NSH, Pmut 1.5% y reemplazamiento EstadoEstacionario en Caso Y

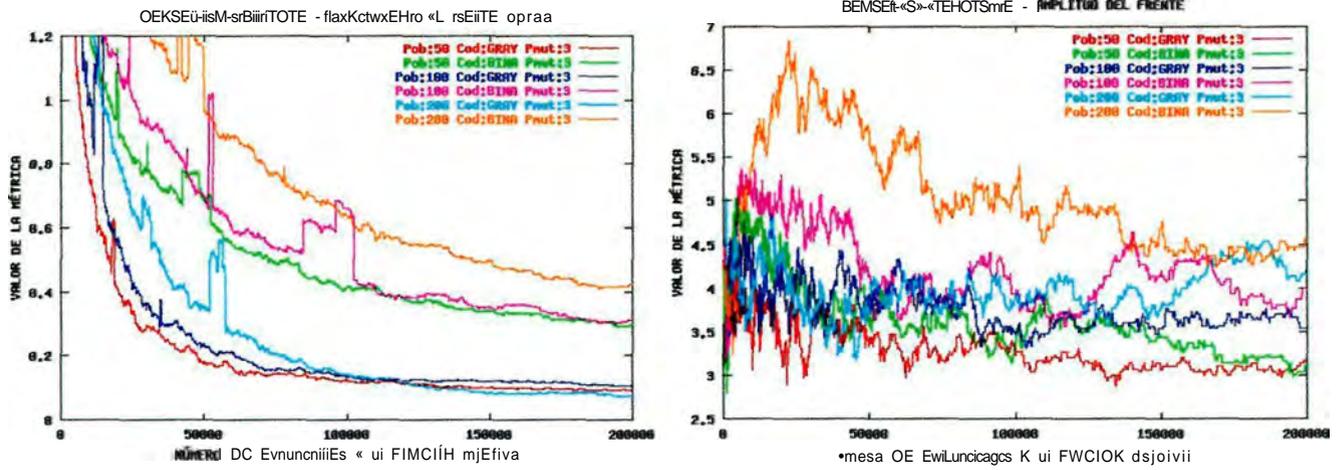


Figura 4.63. Comparativa Código Gray / Binario con DENSEA-NSH, Pmut 3.0% y reemplazamiento EstadoEstacionario en Caso Y

4.5 El operador Elitismo en optimización multiobjetivo estructural

El operador de elitismo ha sido ya citado como factor clave en los algoritmos evolutivos multiobjetivo denominados de segunda generación y cuyo estado del arte ha sido ya tratado en un anterior capítulo. Pretende evitar que durante el trascurso de la evolución, algún individuo óptimo pueda no generar descendencia o ser ésta tal que no mejore al progenitor y así perderse de ese modo su valiosa información genética. Recordemos que el algoritmo genético hace uso de operadores estocásticos y por tanto este hecho sí es posible en la práctica. Mediante elitismo, el mejor (o varios de los mejores) individuos de la población en una generación son directamente incluidos en la población filial, impidiéndose así su extinción y manteniendo la información evolutiva obtenida. (Rudolph (1994) [4.40], teóricamente demuestra mediante modelización en cadenas de Markov, cómo un algoritmo genético canónico al que se le incorpora elitismo, alcanza a medida que el tiempo tiende a infinito la solución óptima. Sin embargo, esto no está garantizado en su ausencia).

Se muestra a continuación una comparativa del rendimiento de un algoritmo evolutivo multiobjetivo de primera generación (NSGA) frente a algoritmos de segunda generación que incorporan elitismo (NSGA-II y SPEA2) para los casos test estructurales tratados y descritos con anterioridad.

A partir de este apartado 4.5, y de acuerdo con las conclusiones obtenidas de la comparativa de codificación entre código binario y código Gray, en la sección 4.4, se considerará la codificación Gray a la hora de establecer el análisis de los resultados, no teniendo en cuenta los resultados de la codificación binaria estándar.

4.5.1 Caso Test X. Comparativa Elitismo / No Elitismo

Las figuras mostradas (4.64 y 4.65) tienen el formato de candelabro ('candlestick'), donde se expresan los extremos máximos y mínimos de la magnitud medida como los puntos extremos de las líneas exteriores a cada columna. En este caso la magnitud medida en el eje de ordenadas es el número de evaluaciones computado para la obtención del frente de Pareto. Cada una está centrada en la media de la distribución y su altura indica el valor de la desviación típica.

El número contenido en cada columna es indicativo del número total de veces sobre 30 que se ha alcanzado el frente de Pareto, compuesto por tres soluciones en este caso test X. Cuanto más abajo esté la columna, indicará menor número medio de evaluaciones necesarias para la obtención del frente óptimo, y cuanto más achatada, menores oscilaciones tendrá la muestra.

Se debe señalar que en el caso del algoritmo NSGA, carente de elitismo, los valores mostrados son para la primera localización del frente óptimo en las ejecuciones realizadas. No obstante, sucede con frecuencia que esta solución óptima es perdida en el final de la ejecución, dada por 20.000 evaluaciones. Esta circunstancia será fácilmente observable en el caso test Y, donde se evidenciará cómo la ausencia de elitismo provoca oscilaciones frecuentes y de alto valor en la evolución del algoritmo hacia el frente.

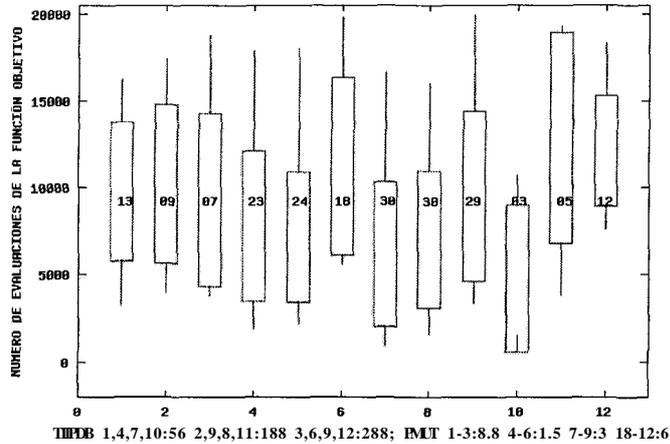


Figura 4.64. Comparativa Elitismo / NoElitismo en Caso X. Algoritmo NSGA

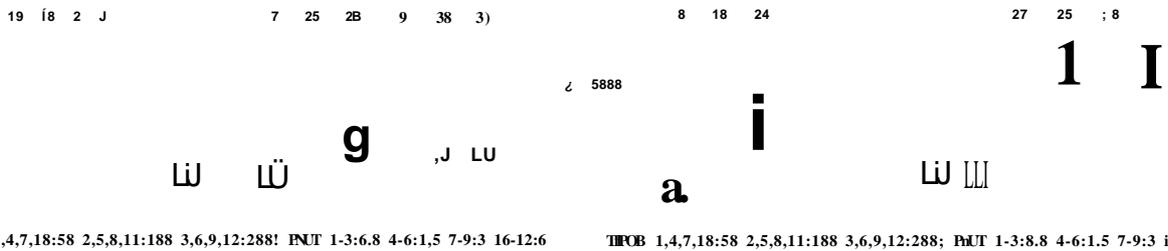


Figura 4.65. Comparativa Elitismo / NoElitismo en Caso X. Algoritmos NSGAII y SPEA2

Analizando las figuras 4.64 y 4.65 podemos observar cómo los algoritmos de segunda generación (NSGAII y SPEA2) son capaces de localizar en un mayor número de veces el frente óptimo, con menor número de evaluaciones y con menores desviaciones típicas, considerando distintos tamaños poblacionales y distintas probabilidades de mutación.

4.5.2 Caso Test Y. Comparativa Elitismo / No Elitismo

En las figuras (4.66 a 4.69) se muestra el análisis comparativo entre los algoritmos NSGA (no elitista) y NSGAII (elitista), para distintos tamaños poblacionales y clasificadas por probabilidad de mutación. En las gráficas a la izquierda se puede observar el acercamiento al frente óptimo, donde independientemente del tamaño poblacional y de la probabilidad de mutación escogidas, la convergencia al frente óptimo es más rápida y más precisa con el NSGAII. Además, la ausencia de elitismo evidencia una pérdida de información constante en la búsqueda, mostrada por el elevado número y amplitud de las oscilaciones apreciadas en la métrica.

En la medida de amplitud del frente, observable en las figuras de la derecha, podemos inferir la menor amplitud alcanzada por el NSGA comparado con sus casos homólogos del NSGAII. También la oscilación es mucho más elevada que en el algoritmo elitista de segunda generación.

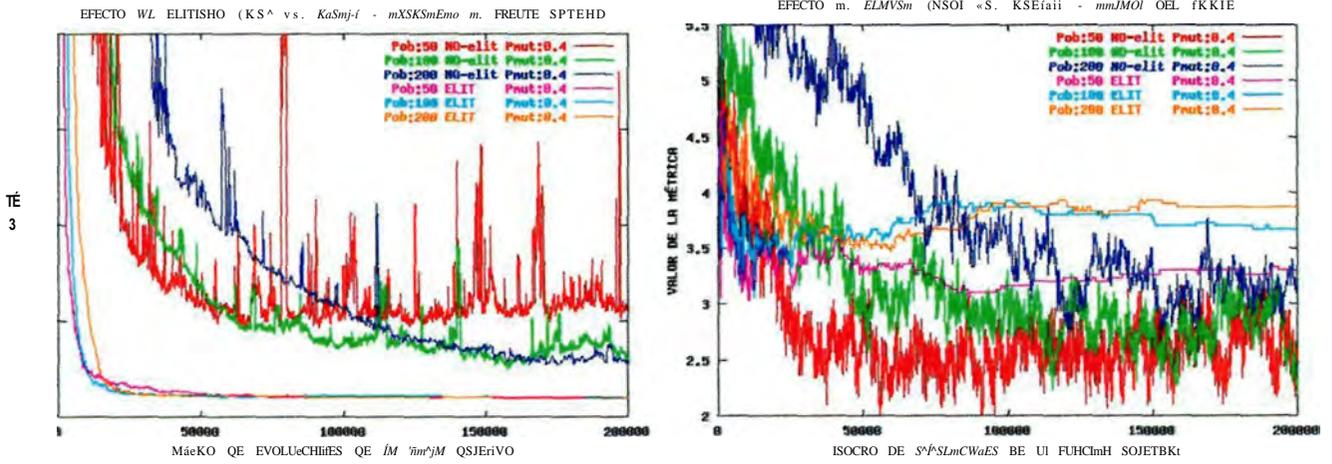


Figura 4.66. Comparativa Elitismo / NoElitismo: NSGA vs. NSGAII en Caso Y. Probabilidad Mutación 0.4%

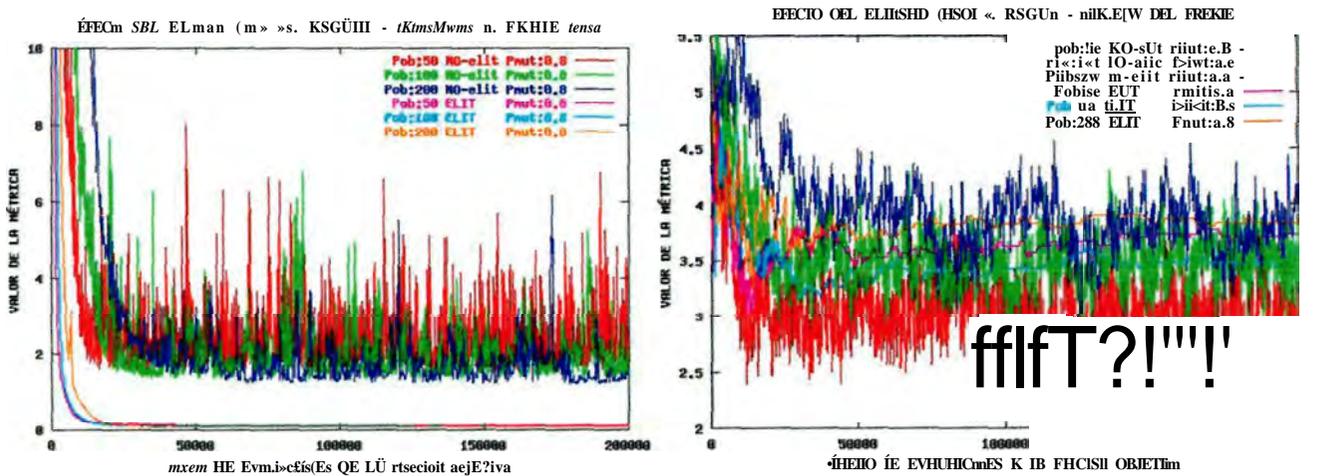


Figura 4.67. Comparativa Elitismo / NoElitismo: NSGA vs. NSGAII en Caso Y. Probabilidad Mutación 0.8%

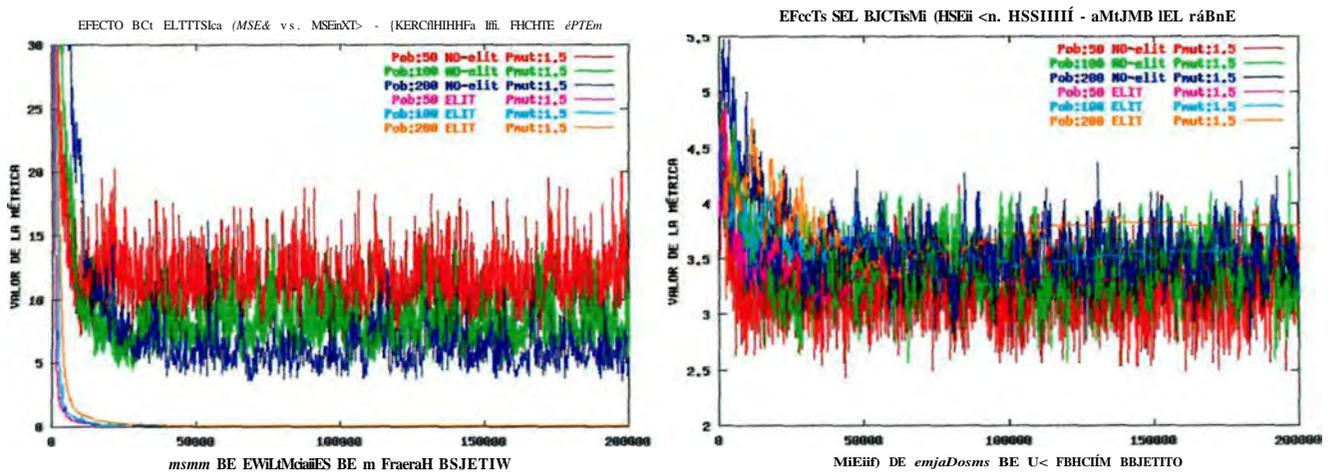


Figura 4.68. Comparativa Elitismo / NoElitismo: NSGA vs. NSGAII en Caso Y. Probabilidad Mutación 1.5%

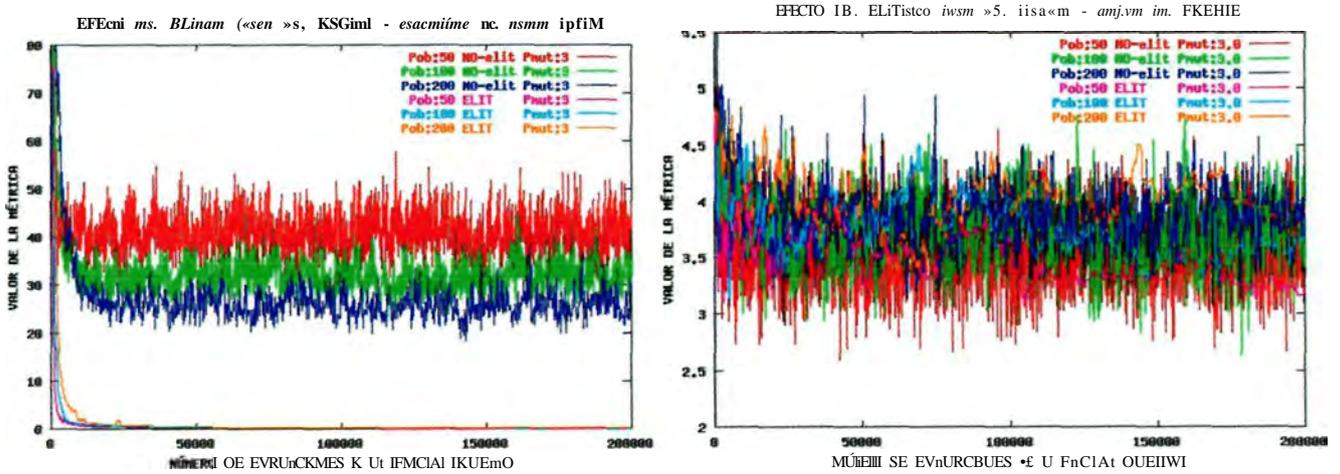


Figura 4.69. Comparativa Elitismo / NoElitismo: NSGA vs. NSGAII en Caso Y. Probabilidad Mutación 3.0%

Es clara por tanto la ventaja que proporciona el elitismo en el comportamiento de los algoritmos evolutivos multicriterio, habiéndose mostrado tanto en términos de acercamiento al frente óptimo, como en términos de amplitud del frente de Pareto alcanzado, durante toda la evolución del algoritmo, y todo ello para ambos casos test tratados: tanto el caso test X, como el caso test Y.

4.6 La Probabilidad de Mutación en optimización multiobjetivo estructural

El equilibrio exploración-explotación que se plantea en la búsqueda de la(s) solución(es) óptima(s), se puede observar desde la óptica de un balance entre la diversidad poblacional (explorativa) y la presión de selección (explotativa). Dentro de los instrumentos que infieren diversidad en la población podemos citar: el tamaño o número de individuos en la población, el tipo de cruce y la probabilidad de mutación. Este último factor será objeto de estudio comparativo en este apartado para los dos casos test tratados, habiéndose ensayado cuatro probabilidades distintas para cada uno de ellos: 0.8%, 1.5%, 3% y 6% para el caso test X y 0.4%, 0.8%, 1.5% y 3% para el caso test Y.

4.6.1 Caso Test X. Comparativa de Probabilidades de Mutación

Tabla 4.19. Comparativa Probabilidades Mutación en Caso X con reemplazamiento generacional.

Caso test X - Generacional	Probabilidad Mutación (%)	50			100			200			1- Total			m
		N	Me	Dt	N	Me	Dt	N	Me	Dt	N	Me	Dt	
SPEA2	0.8%	18	3888	4889	18	4066	4545	24	2158	608	60	3249	3347	
	1.5%	17	5544	5969	23	3752	4709	24	2425	813	64	3730	3830	
	3.0%	23	5513	5981	25	4508	5469	27	2533	975	175	4105	4141	
	6.0%	25	2966	2202	28	2157	1603	29	3317	1597	82	2813	1800	
NSGAI	0.8%	15	4736	4811	18	3867	4409	23	2869	2250	56	3689	3823	
	1.5%	21	4526	5593	26	3542	4296	23	2400	907		3461	3598	
	3.0%	27	3459	4192	25	3512	3935	26	2453	872	1-78	3140	2999	
	6.0%	29	2965	2937	30	2380	1657	30	3186	2376	89	2842	2323	
NSGAI	0.8%	18	5341	5775	17	3770	5578	23	4460	4749	58	4531	5367	
El.cont.r00	1.5%	19	4150	5767	24	3037	3155	26	3830	3044	69	3642	3988	
	3.0%	26	6401	5323	26	3461	3924	29	3206	1000	81	4313	3415	
	6.0%	27	2264	2065	25	2464	933	30	3626	960	82	2823	1319	
El.cont.r04	0.8%	29	5213	4186	26	5519	AllZ	28	"^90(r	3950	83	5203	4303	
	1.5%	30	2725	2305	30	4623	5239	30	4366	3437	90	3904	3660	
	3.0%	19	2767	2679	30	4130	4726	30	3940	2442	89	3621	3282	
	6.0%	30	1861	813	29	4031	3082	30	3646	1532	89	3169	1809	
DENSEA	0.8%	30	2021	1721	30	2756	1844	30	3106	1453	90	2627	1672	
	1.5%	30	1628	1363	30	2306	1626	30	2906	843	90	2280	1277	
	3.0%	30	1533	998	30	2416	1297	30	2786	824	90	2245	1039	
	6.0%	30	1325	620	30	2130	898	30	3360	930	90	in\	816	
DENSEA	0.8%	30	2470	1986	30	2293	1173	30	2560	895	90	2441	1351	
NSH	1.5%	30	1576	900	30	1993	985	30	3200	1138	90	2256	1007	
	3.0%	30	1225	711	30	1720	620	30	2626	631	90	1857	654	
	6.0%	30	1383	671	30	1840	649	30	2986	977	90	2069	765	
Total	0.8%	140	3736	3185	139	3610	2994	158	3338	1986	437	3552	2775	
	1.5%	147	3034	2894	163	3183	2965	163	3233	1580	473	3154	2537	
	3.0%	165	3330	2964	166	3243	3002	172	2940	994	503	3168	2346	
	6.0%	171	2095	1411	172	2496	1323	179	3353	1243	522	2658	1341	

Capítulo 4. Contribuciones

En las tablas 4.19 a 4.21 se expone para cada algoritmo el resultado de treinta ejecuciones independientes desglosado por tamaño poblacional y probabilidad de mutación. Se presenta el número de veces sobre treinta que se ha encontrado el frente óptimo (N), el promedio del número de evaluaciones necesarias en esas ejecuciones N (Me), y su desviación típica (Dt). Se han desglosado igualmente los resultados según el tratamiento poblacional, presentándose primero los resultados de los algoritmos generacionales (4.19), posteriormente los de los algoritmos de estado-estacionario (4.20) y en la tabla 4.21 los resultados globales agrupados.

Tabla 4.20. Comparativa Probabilidades Mutación en Caso X con reemplazamiento de estado estacionario.

Caso test X - SteadyState Probabilidad Mutación (%)	50			100			200			1	Total		
	N	Me	Dt	N	Me	Dt	N	Me	Dt		Me	Dt	
SPEA2	0.8%	13	8687	6105	13	4525	5371	19	3140	4360	45	5142	5278
	1.5%	15	4657	4474	17	4737	5478	21	3963	5297	53	4407	5083
	3.0%	26	5202	5981	26	4180	5100	27	3731	4188	79	4362	5089
	6.0%	28	4561	4805	26	4192	4703	30	2298	1733	84	3638	3747
NSGAI I	0.8%	5	6652	7174	17	4222	4891	24	3401	4095	46	4057	5386
	1.5%	21	8231	6809	21	5908	5879	22	2887	3531	164	5631	5406
	3.0%	25	3775	3328	26	2777	4707	25	3661	3847	76	3396	3960
	6.0%	30	2538	3147	29	3319	3573	28	2924	2620	87	2922	3113
NSGAI I	0.8%	17	4465	5707	14	5385	6772	21	4365	4552	52	4672	5677
ELcontr00	1.5%	21	5720	5563	24	4786	5494	28	3571	3661	73	4588	4906
	3.0%	27	3788	3980	28	3736	4631	30	3017	2407	85	3498	3672
	6.0%	30	3262	3038	30	2839	2581	29	4025	3446	89	3368	3021
NSGAI I el.contr04	0.8%	27	7322	5875	30	4195	4299	24	3747	2639	81	5104	4271
	1.5%	30	4545	2995	30	4139	3560	29	3877	3656	89	4190	3403
	3.0%	30	3080	2914	30	3768	4008	29	3101	1960	189	3318	2960
	6.0%	30	2386	2305	30	2811	2023	30	3129	1325	90	2775	1884
DEN SEA	0.8%	30	2262	2342	30	2467	1888	30	2353	1252	90	2360	1827
	1.5%	30	1622	1330	30	2000	1289	30	2383	929	90	2001	1182
	3.0%	30	1569	900	30	2096	1061	30	2502	1298	90	2055	1086
	6.0%	30	1673	1048	30	2010	865	30	2666	1327	90	2116	1080
DEN SEA	0.8%	30	3144	2648	30	1665	867	30	2853	1704	90	2554	1739
NSH	1.5%	30	1841	1737	30	1709	931	30	2732	1404	90	2094	1357
	3.0%	30	1497	1119	30	1807	1098	30	2691	1144	90	1998	1120
	6.0%	30	1301	687	30	1856	667	30	2729	992	90	1962	782
Total	0.8%	122	4770	3698	134	3401	3012	148	3236	2649	404	3754	3195
	1.5%	147	4102	3195	152	3650	3184	160	3203	2752	459	3639	3064
	3.0%	168	3073	2786	170	3033	3169	171	3090	2184	509	3065	2725
	6.0%	178	2598	2307	175	2804	2199	177	2956	1670	530	2785	2080

Capítulo 4. Contribuciones

Tabla 4.21. Comparativa Probabilidades Mutación en Caso X, Totales.

Caso test X - Acumulados Probabilidad Mutación (%)	50			100			200			Total		
	N	Me	Dt	N	Me	Dt	N	Me	Dt	N	Me	Dt
Gen-Total 0.8%	140	3736	3185	139	3610	2994	158	3338	1986	437	3552	2775
1.5%	147	3034	2894	163	3183	2965	163	3233	1580	473	3154	2537
3.0%	165	3330	2964	166	3243	3002	172	2940	994	503	3168	2346
6.0%	171	2095	1411	172	2496	1323	179	3353	1243	522	2658	1341
StSte-Total 0.8%	<i>ni</i>		3698	134	3401	3012	148	3236	2649	404	3754	3195
1.5%	147	4102	3195	152	3650	3184	160	3203	2752	459	3639	3064
3.0%	168	3073	2786	170	3033	3169	171	3090	2184	509	3065	2725
6.0%	178	2598	2307	175	2804	2199	177	2956	1670	530	2785	2080
Total General 0.8%	262	4217	3424	273	3507	3003	306	3289	2307	841	3649	2977
1.5%	294	3568	3045	315	3409	3071	323	3219	2161	932	3392	2797
3.0%	333	3200	2874	336	3137	3086	343	3015	1587	1012	3116	2537
6.0%	349	2351	1868	347	2651	1765	356	3156	1455	1052	2722	1713

De estos resultados de las tablas 4.19 a 4.21 se puede destacar para este caso test X, que la probabilidad de mutación del 6% es la que proporciona un mayor número de localizaciones del frente óptimo (N), en un menor número de evaluaciones (Me), y con menor desviación típica, habiéndose destacado dichos valores en las tablas con cursiva y negrita además de sombreado. La evolución de resultados además parece mejorar a medida que se aumenta la probabilidad de mutación, tal como se puede observar en los resultados acumulados, donde es progresiva la evolución en aumento de N, y en disminución de Me y Dt partiendo de la probabilidad de 0.8% y hasta la probabilidad del 6%. Esta probabilidad se corresponde con una probabilidad de mutación de $1/n$, siendo n la longitud del cromosoma, que para este caso son 16 bits, siendo su inversa de 6.25%.

Si bien para la población de tamaño 200, la mejor probabilidad de mutación total global es la de 3%, en cambio, si dividimos el comportamiento en tratamiento poblacional generacional y de estado-estacionario, se observa cómo también en el caso de estado estacionario es el 6% la mejor (como en las poblaciones de 50 y 100 individuos), y sólo es el 3% en los casos generacionales. Esto es concordante con el hecho de que la gestión poblacional que ofrece el estado-estacionario presenta una mayor presión de selección que la generacional, y se puede compensar con una mayor diversidad poblacional, es decir, con mayor probabilidad de mutación desde el punto de vista del equilibrio exploración-explotación.

También se puede observar cómo para el tamaño de cromosoma correspondiente a este caso test, los mejores resultados se obtienen con una población de 50 individuos (2351 evaluaciones), puesto que el número de evaluaciones necesarias es menor que 100 individuos (2651 evaluaciones) y que 200 individuos (3015 evaluaciones). La diversidad poblacional con 50 individuos es suficiente y permite una convergencia más rápida que con los otros dos tamaños poblacionales, sólo se aprecia un leve incremento de la desviación típica y un leve descenso de N.

Capítulo 4. Contribuciones

Se muestran a continuación las figuras (4.70 a 4.76) donde se reflejan los datos de las tablas anteriores por algoritmo, agrupadas por tamaño poblacional y probabilidades de mutación, de tal forma que se aprecian agrupadas en tres bloques (tamaños poblacionales de 50, 100 y 200 individuos) cada uno de cuatro barras (probabilidades de mutación de 0.8%, 1.5%, 3% y 6%). Las figuras empleadas tienen el formato de candelabro ('candlestick'), donde se expresan los extremos máximos y mínimos de la magnitud medida como los puntos extremos de las líneas exteriores a cada columna. En este caso la magnitud medida en el eje de ordenadas es el número de evaluaciones computado para la obtención del frente de Pareto. Cada una está centrada en la media de la distribución y su altura indica el valor de la desviación típica. El número contenido en cada columna es indicativo del número total de veces sobre 30 que se ha alcanzado el frente de Pareto, compuesto por tres soluciones en este caso test X. Cuanto más abajo esté la columna, indicará menor número medio de evaluaciones necesarias para la obtención del frente óptimo, y cuanto más achatada, menores oscilaciones tendrá la muestra.

83 89 24 38 er TS' M IB

1

5aaa

T LL T

THOB 1-jHQ 5>8;18e 9-12:286; PMU 1,5,9:8.8 2,6,18:1.5 3,7,11:3 4,8,12:1

Figura 4.70. Comparativa Probabilidades Mutación en Caso X. Reemplazamiento Generacional. Algoritmo NSGA. Gráfica sobre Frente de Pareto

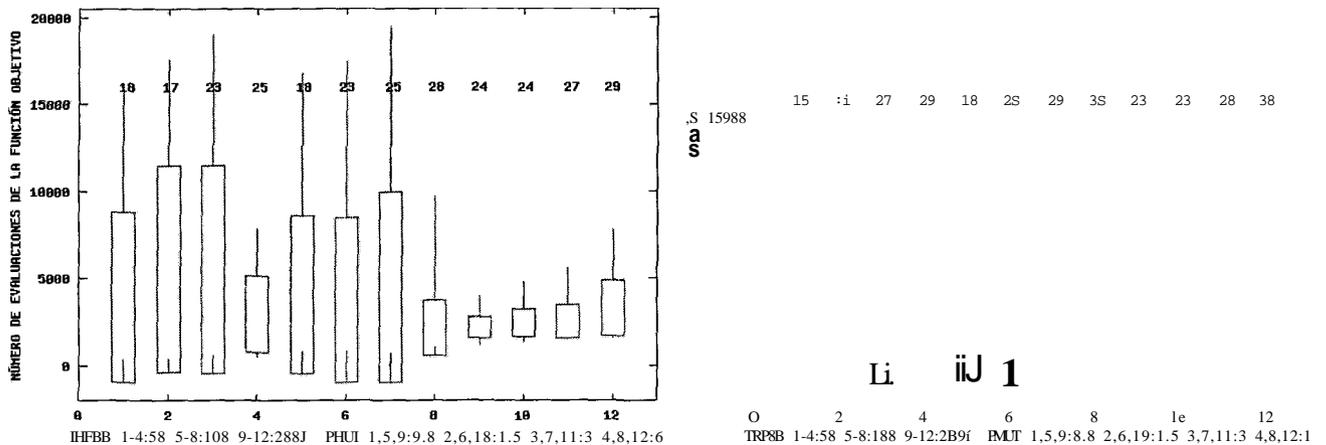


Figura 4.71. Comparativa Probabilidades Mutación en Caso X. Reemplazamiento Generacional. Algoritmos SPEA2 y NSGAII. Gráficas sobre Frente de Pareto

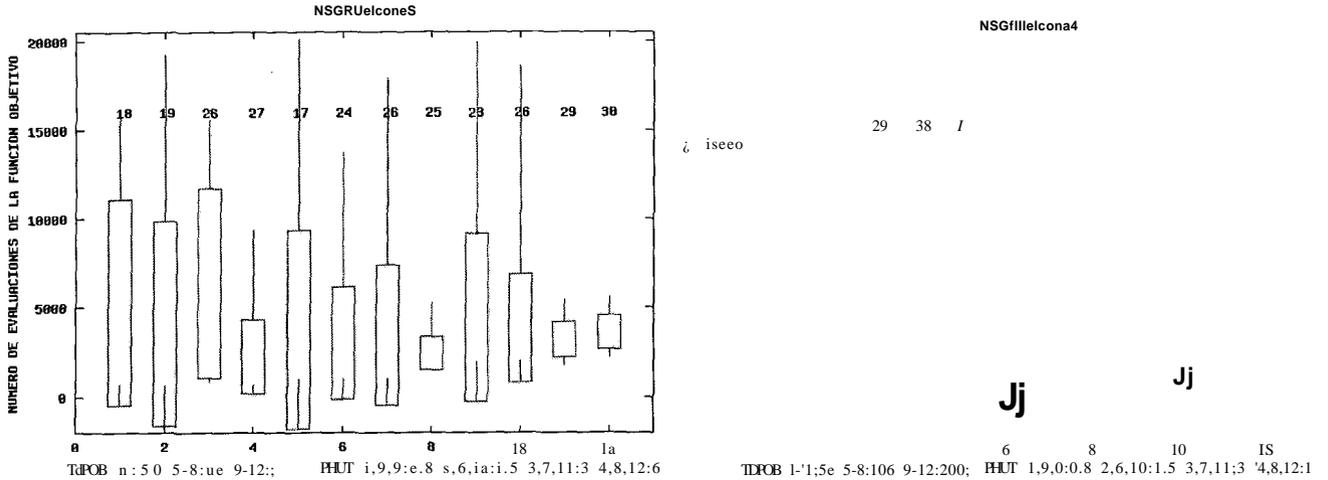


Figura 4.72 Comparativa Probabilidades Mutación en Caso X. Reemplazamiento Generacional. Algoritmos NSGAI con elitismo controlado, $r=0,0$ y $r=OA$. Gráficas sobre Frente de Pareto

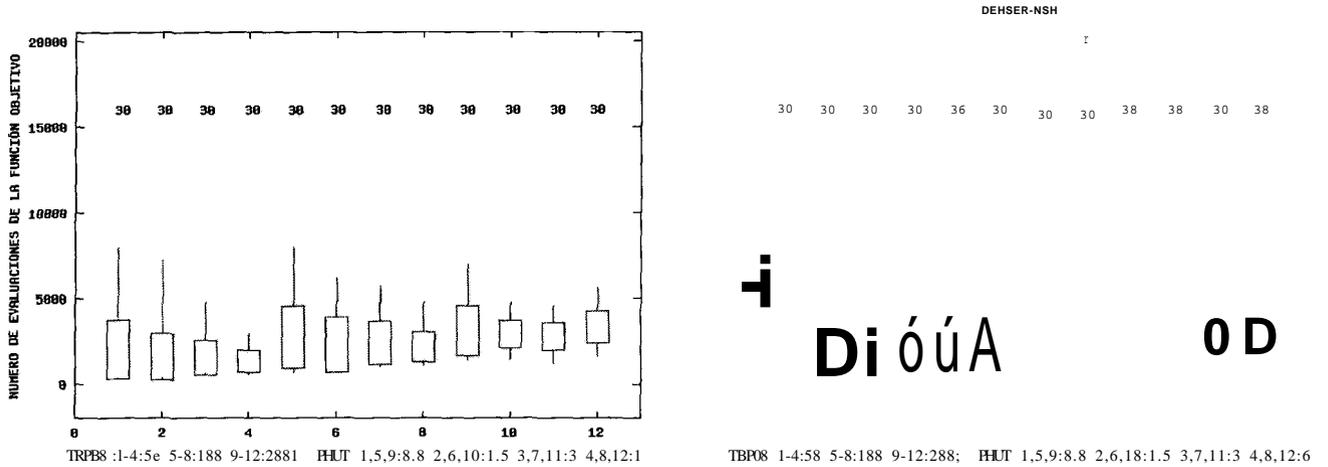


Figura 4.73. Comparativa Probabilidades Mutación en Caso X. Reemplazamiento Generacional. Algoritmos DENSEA y DENSEA-NSH. Gráficas sobre Frente de Pareto

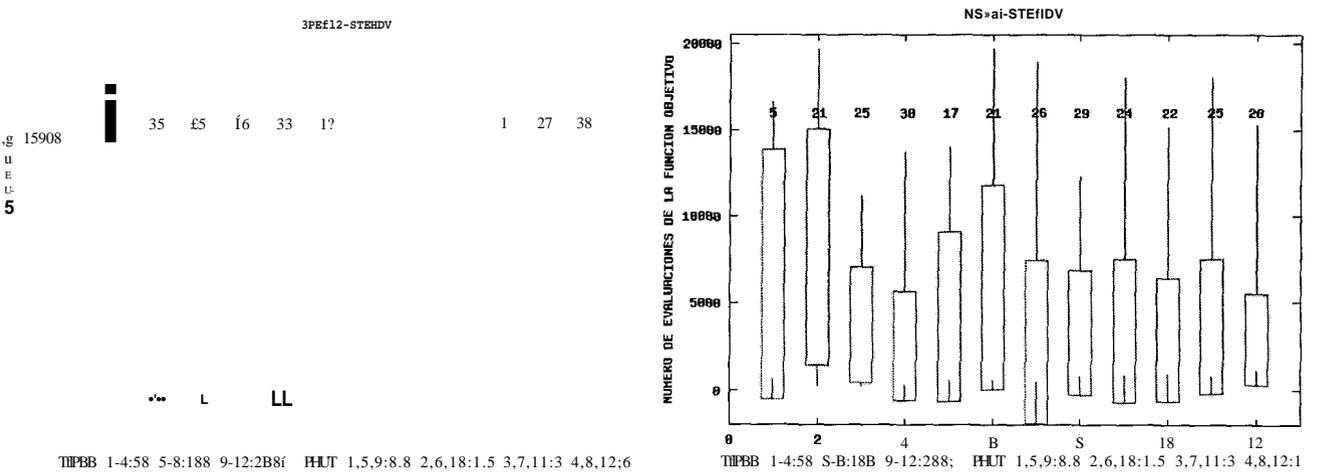


Figura 4.74. Comparativa Probabilidades Mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmos SPEA2 y NSGAI. Gráficas sobre Frente de Pareto

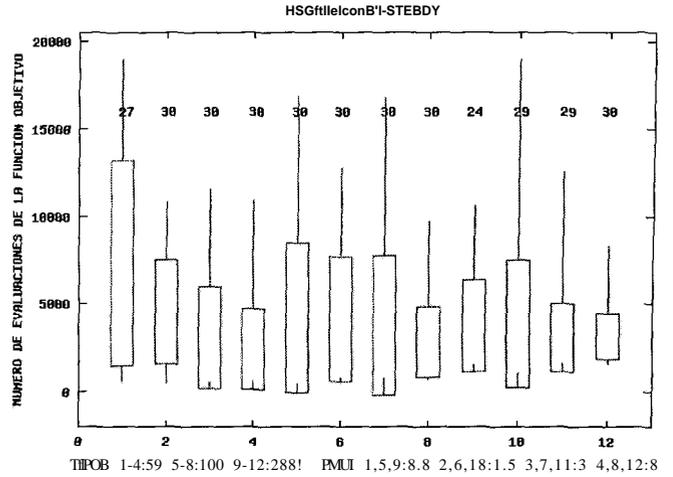
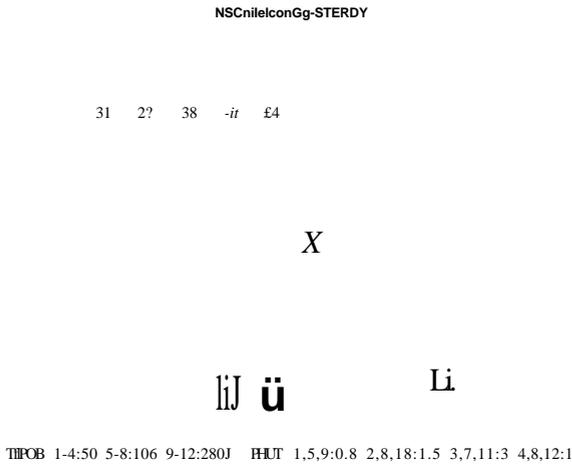


Figura 4.75. Comparativa Probabilidades Mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmos NSGAI con elitismo controlado, $r=0.0$ y $r=0.4$. Gráficas sobre Frente de Pareto

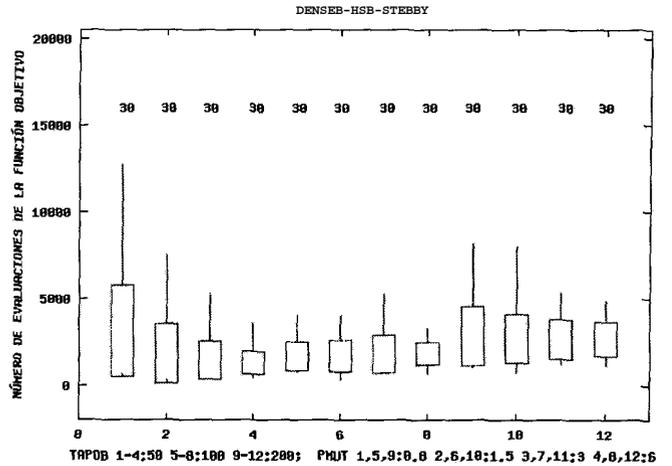
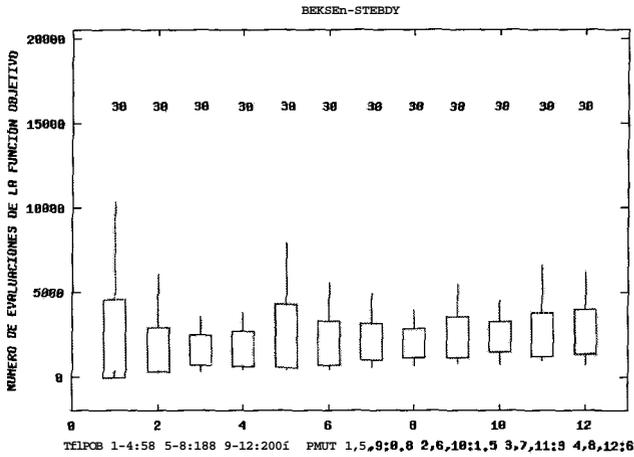


Figura 4.76. Comparativa Probabilidades Mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmos DENSEA y DENSEA-NSH. Gráficas sobre Frente de Pareto

Se muestra en las figuras 4.77 a 4.112 para el promedio de los treinta casos ejecutados, la evolución de la convergencia de las dos métricas: acercamiento al frente (en la mitad de la izquierda) y amplitud del frente (en la mitad de la derecha). Estas figuras se encuentran en el *Anexo*.

4.6.2 Caso Test Y. Comparativa de Probabilidades de Mutación

En la tabla 4.22 se expone para cada algoritmo el resultado de treinta ejecuciones independientes desglosado por tamaño poblacional y probabilidad de mutación. Se han desglosado igualmente los resultados según el tratamiento poblacional, presentándose primero los resultados de los algoritmos generacionales y posteriormente los de los algoritmos de estado-estacionario. En ella se ha realizado un análisis cualitativo comparativo y se evalúa la métrica acercamiento al frente óptimo, habiéndose ponderado los algoritmos de 1° a 4° para cada tamaño poblacional en función de la probabilidad de mutación.

Tabla 4.22. Comparativa Probabilidades de Mutación Caso Y.

Caso test Y	Probab. Mutación (%)	Generaciona				EstadoEstacionario				Total
		50	100	200	Total	50	100	200	Total'	
SPEA2	0.4%	2°	3°	1°	; 2.00	2°	2°	3°	2.33	2.17
	0.8%	1°	r	2°	' 1.33	3°	3°	2°	2.67 J	2.00
	1.5%	3°	2°	3°	2.67	r	1°	r	1.00	1.83
	3.0%	4°	40	40	4.00	40	4°	40	4.00	4.00
NSGAI	0.4%	2°	3°	3°	\ 2.67	40	2°	40	3.33	3.00
	0.8%	4°	1°	2°	[2.33	1°	3°	2°	2.00 i	2.17
	1.5%	1°	2°	1°	\ 1.33	3°	1°	1°	1.67	1.50
	3.0%	3°	40	40	1 3.67	2°	40	3°	3.00	3.33
<u>NSGAIel.contrOO</u>	0.4%	3°	1°	3°	1 2.33	2°	3°	3°	2.67	2.50
	0.8%	T	2°	2°	2.00	1°	2°	2°	1.67	1.83
	1.5%	1°	3°	1°	I 1.67	3°	1°	1°	1.67 ,	1.67
	3.0%	4°	40	40	14.00	40	40	40	4.00	4.00
<u>NSGAIel.contr04</u>	0.4%	3°	r	3°	2.33	2°	2°	3°	2.33	2.33
	0.8%	2°	2°	2°	\ 2.00	3°	1°	1°	1.67	1.83
	1.5%	r	3°	1°	1 1.67	1°	3°	2°	2.00	1.83
	3.0%	40	40	4°	1 4.00	40	4°	40	4.00	4.00
DENSEA	0.4%	3°	2°	r	1 2.00	T	r	2°	1.67	1.83
	0.8%	r	3°	2°	2.00	30	3°	r	2.33	2.17
	1.5%	2°	1°	3°	1 2.00	1°	2°	3°	2.00	2.00
	3.0%	40	40	40	14.00	40	40	40	4.00	4.00
DENSEA-NSH	0.4%	2°	3°	1°	¡2.00	1°	3°	2°	2.00	2.00
	0.8%	r	1°	2°	i 1.67	2°	2°	1°	1.67	1.50
	1.5%	3°	2°	3°	1 2.67	3°	r	3°	2.33	2.50
	3.0%	40	40	40	4.00	40	40	40	4.00	4.00
PromedioAlgoritmo	0.4%	2.50	2.17	2.00	2.22	2.17	2.17	2.83	2.39	2.30
	0.8%	1.83	1.67	2.00	1.83	2.17	2.33	1.5	2.00	1.91
	1.5%	1.83	2.17	2.00	2.00	2.00	1.50	1.83	1.77	1.88
	3.0%	3.83	4.00	4.00	3.94	3.67	4.00	3.83	X8J	3.88

Se puede apreciar cómo con diferencia la mayor probabilidad de mutación de las ensayadas en este caso test Y (el 3%) es en general la peor de todas, conduciendo a un comportamiento de mayor carácter caótico en la convergencia y en la amplitud del frente. Sin embargo, las probabilidades de mutación de 0.8% y 1.5% son las más exitosas, por encima de la probabilidad de 0.4%, que queda en

penúltimo lugar. Si observamos el cómputo total (en negrita y cursiva en la tabla 4.22) vemos cómo el índice es similar en ambas probabilidades de mutación (1.88 y 1.91, respectivamente). En el conjunto de los algoritmos generacionales, es mejor la probabilidad de 0.8% que 1.5%, mientras que en el conjunto de los algoritmos de estado-estacionario, ocurre a la inversa. Esto es concordante con el hecho que la gestión poblacional que ofrece el estado-estacionario presenta una mayor presión de selección que el generacional, y se puede compensar con una mayor diversidad poblacional, es decir, con mayor probabilidad de mutación desde el punto de vista del equilibrio exploración-explotación.

En este segundo caso test Y, el tamaño de población que obtiene los mejores resultados es el de 200 individuos con reemplazamiento de estado-estacionario. Así pues, el mayor tamaño poblacional, también asociado con la mayor diversidad poblacional para este cromosoma con 220 bits, parece resultar beneficioso.

A continuación se representan las gráficas correspondientes a los algoritmos SPEA2 (4.113 a 4.155 y 4.131 a 4.133) y DENSEA-NSH (4.128 a 4.130 y 4.146 a 4.148), tanto en sus versiones generacionales como en sus versiones de estado estacionario. Las figuras del resto de algoritmos se encuentran en el Anexo.

Se muestra el algoritmo Spea2 (generacional) con 3 tamaños de población en las figuras 4.113a 4.115.

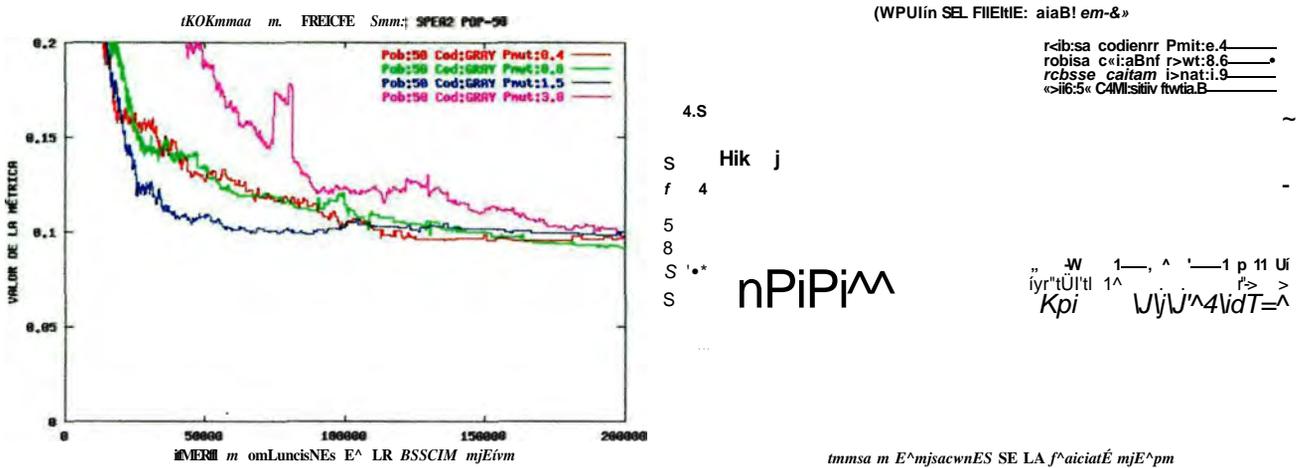


Figura 4.113. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo SPEA2 con tamaño de población 50.

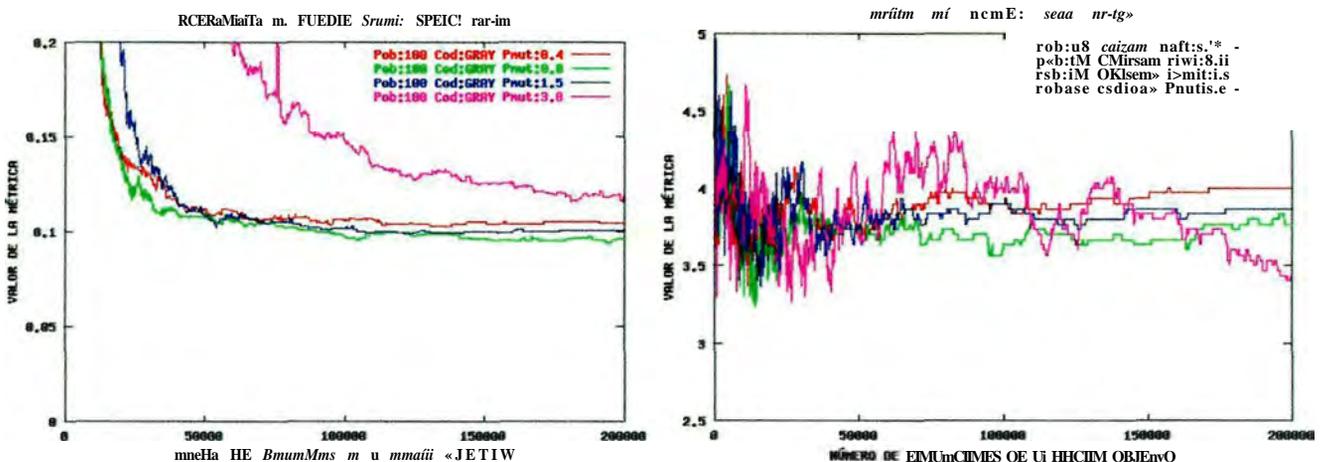


Figura 4.114. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo SPEA2 con tamaño de población 100.

Capítulo 4. Contribuciones

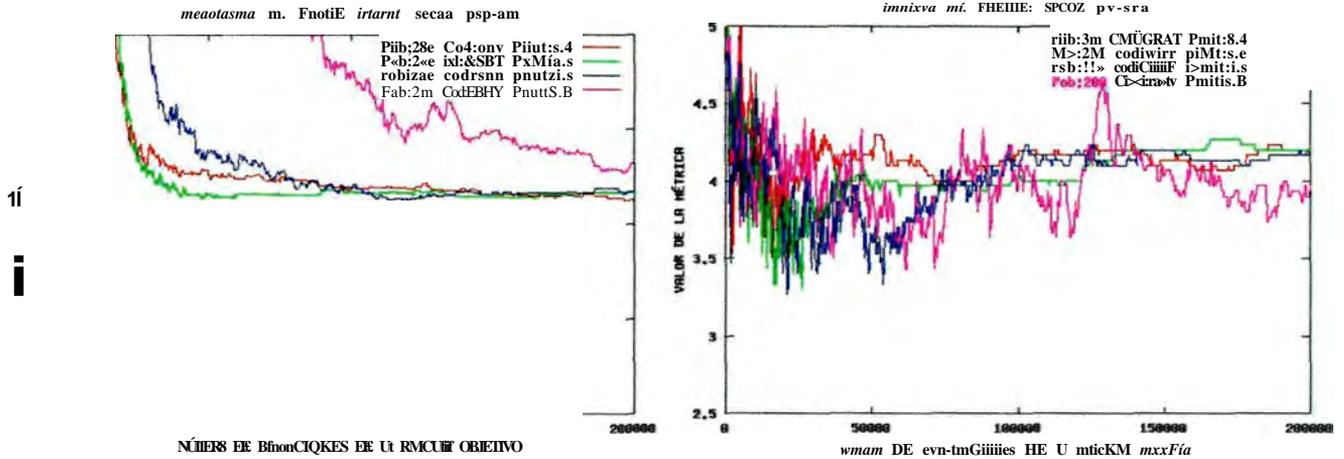


Figura 4.115. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo SPEA2 con tamaño de población 200.

Se muestra en el algoritmo DENSEA-NSH (generacional), con 3 tamaños de población en las figuras 4.128 a 4.130.

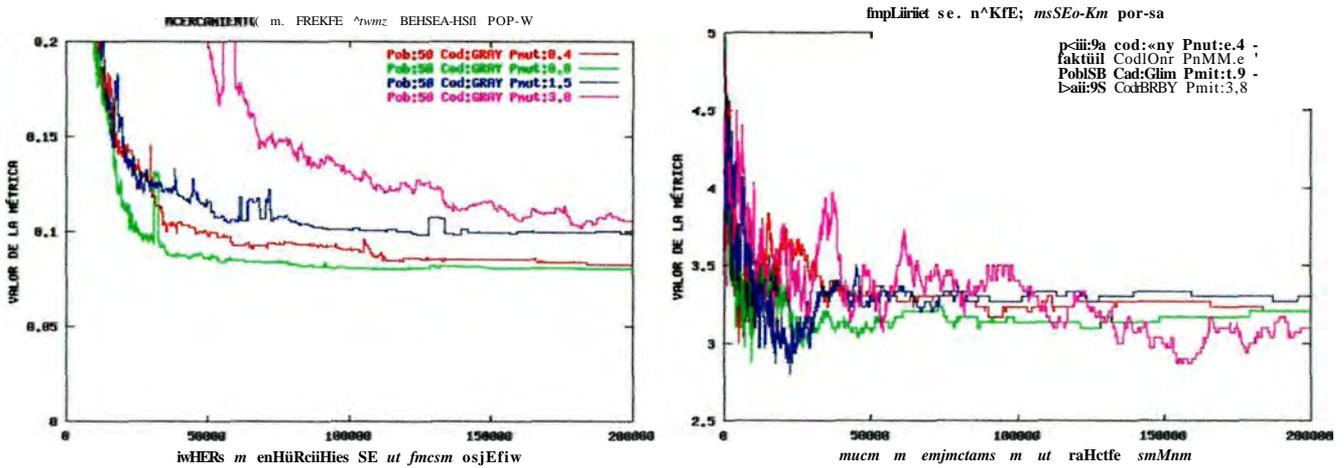


Figura 4.128. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo DENSEA-NSH con tamaño de población 50.

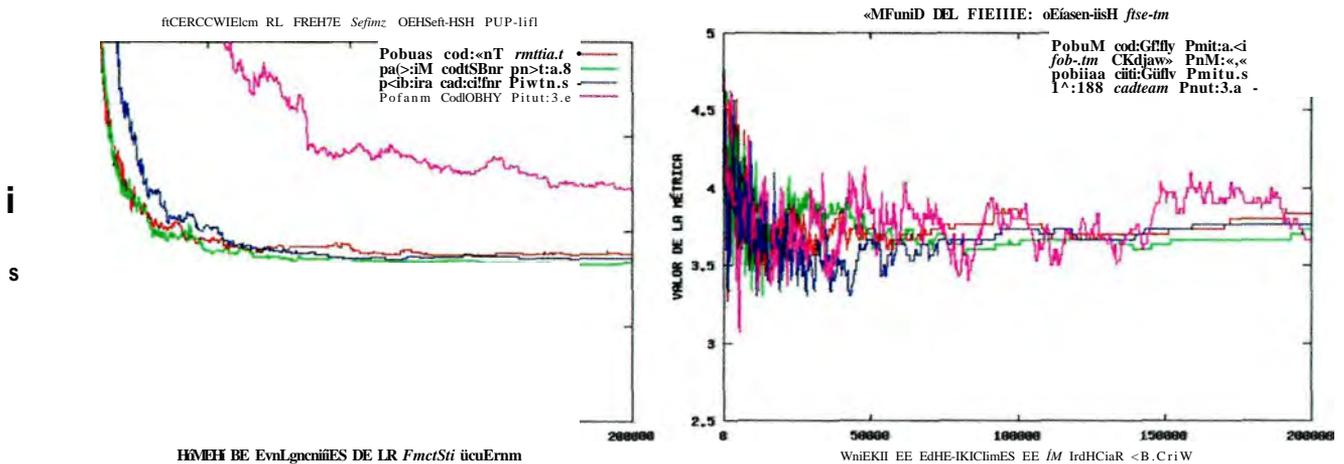


Figura 4.129. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo DENSEA-NSH con tamaño de población 100.

S
•m

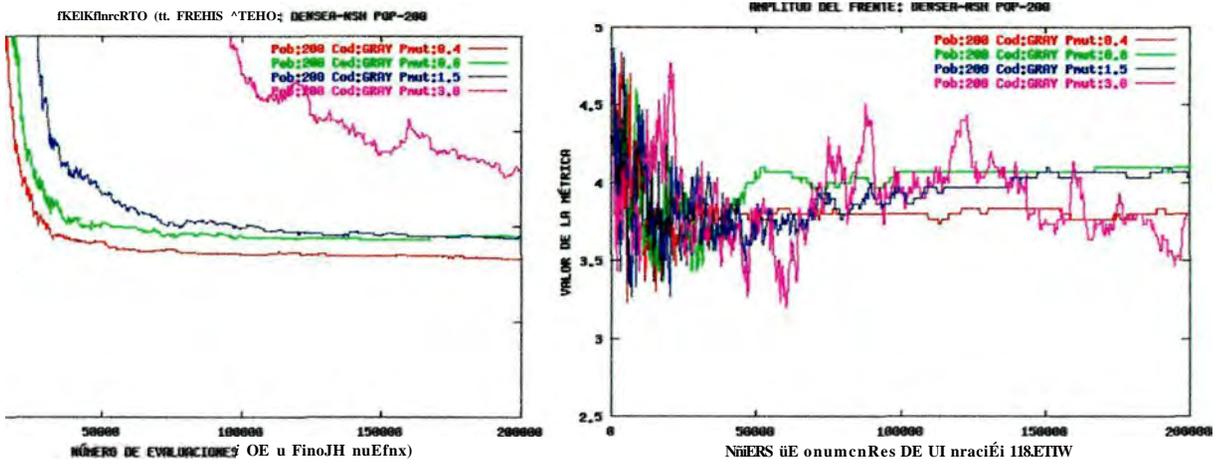


Figura 4.130. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo DENSEA-NSH con tamaño de población 200.

Se muestra en el algoritmo Spea2 (steady-state), con 3 tamaños de población en las figuras 4.131 a 4.133.

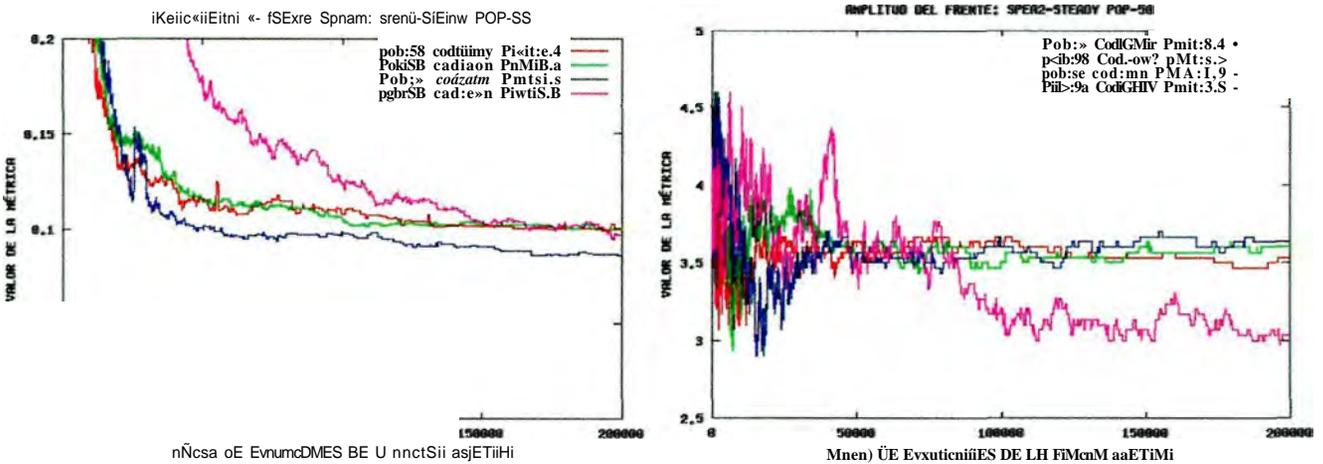


Figura 4.131. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo SPEA2 con tamaño de población 50.

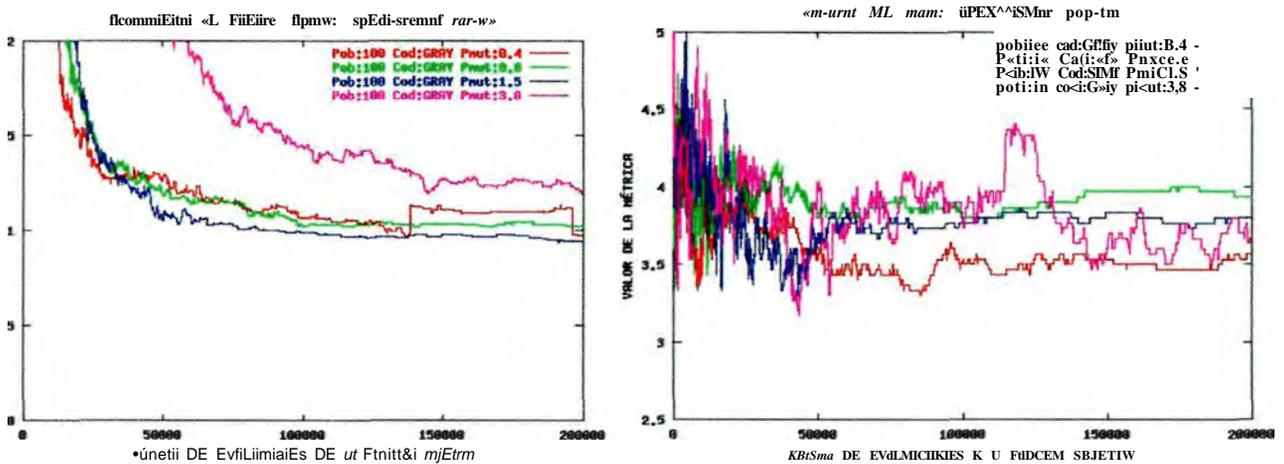
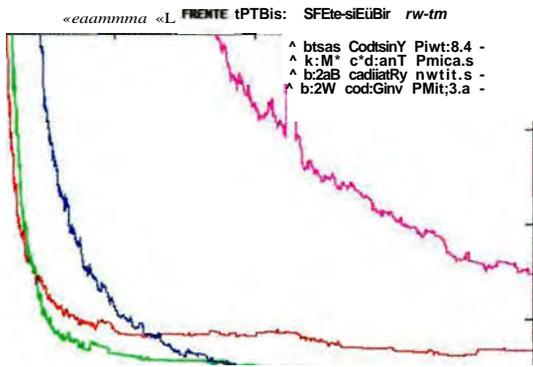


Figura 4.132. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo SPEA2 con tamaño de población 100.



Métrica DE Evolucionar IE LR FWC1&I nejcTim

fmpunai BEL FHEUIE: srena-srEMHf «w-ase

Potisaaa *tafoan* Pnutie.i
 M>:2n CMISOBT PwtM.a
 ral>:2s« caiiainr nwt:i.5
 M>i2M *fismm* *TmAts.a*

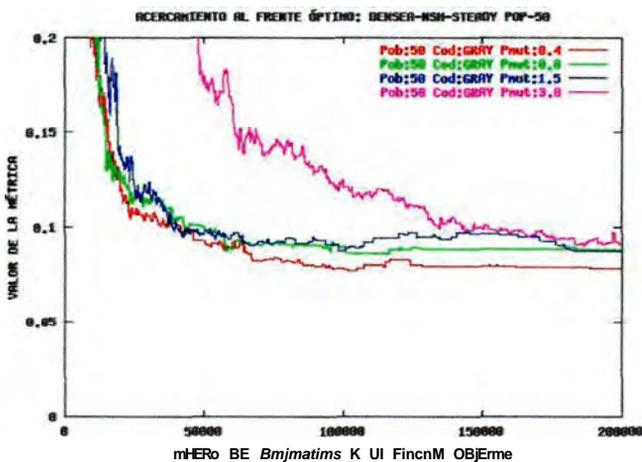
i 4

"V" TptfTL-.-Aft- yj;il/iQff^ii c4

8

Figura 4.133. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo SPEA2 con tamaño de población 200.

Se muestra en el algoritmo DENSEA-NSH (steady-state), con 3 tamaños de población en las figuras 4.146 a 4.148.



m P U t W DEL FREHE: OEUSEI-HISII-SnEaiin fw-9»

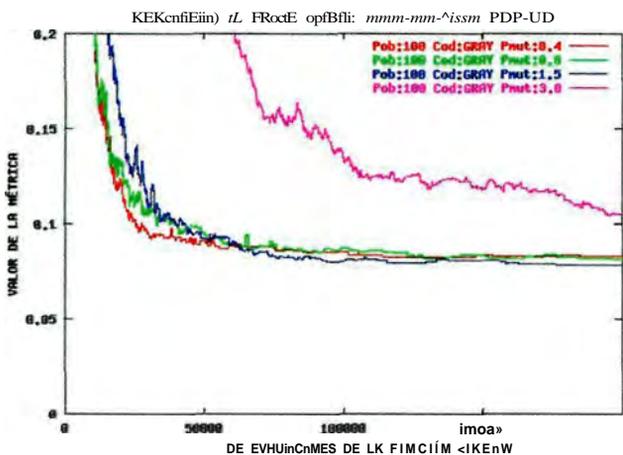
rob:se *coitaiim* mut:>.'t *
 Pob:!» Cod:c«iv rmicl.s -
 !^A:5a Cod:GRBY Pmit:3.9

VALOR DE LA MÉTRICA

-^=t^

Sx~~~~~'~~~~~Sx^

Figura 4.146. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA-NSH con tamaño de población 50.



Mirunn DEL BIEUCE: aaeM isii-ste«OT np-ua

Pob:iBe cod:GRtv Pi<ute.4
 p<b:iaa *fimsa* Pwts.a
 Pob:i8e *daieum* Pint:i.s
 p<b:M8 cod:SHI» Pn>it:3.a

i ^

jHHUjflufurjLJLAft^A^A^A^|~Vnmio^ W^A r 1 w_, f4 JVir~"

i^J^Y^~t=c_i^i^_::

WME m DE E<t)UICI(K(ES DE UI rMGIDH WJETIVE

Figura 4.147. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA-NSH con tamaño de población 100.

I
3
8 ..1

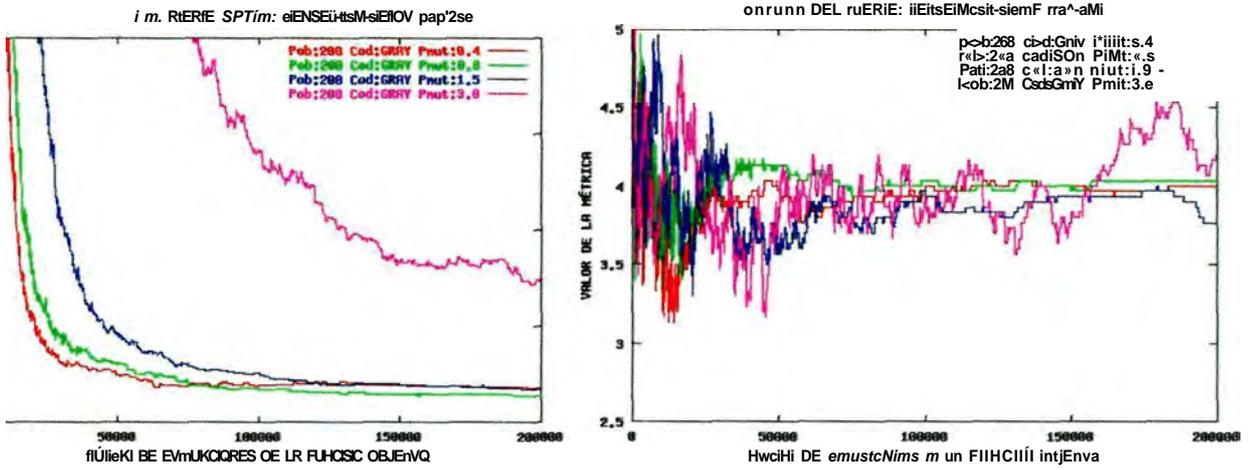


Figura 4.148. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA-NSH con tamaño de población 200.

Si bien en el caso test X, la probabilidad de mutación que ofrecía mejores resultados era la cercana al $1/n$, en este caso test Y, esa probabilidad para una longitud de cromosoma de 220 bits se correspondería con 0.45%. Tal como se desprende de los resultados son más beneficiosas probabilidades de mutación mayores.

Así pues, para el caso X, con longitud de cromosoma de 16 bits (4 barras x 4 bits/barra) la probabilidad de mutación más adecuada se sitúa entre el 3 y el 6%, mientras que para el caso Y, con longitud de cromosoma de 220 bits (55 barras y 4 bits/barra) la probabilidad de mutación más adecuada oscila entre el 0.8 y el 1.5%. Si representamos gráficamente esta circunstancia, acompañado de la gráfica de probabilidad de mutación dictada por la inversa del número de bits del cromosoma, obtenemos la figura 4.149 representada en escala logarítmica.

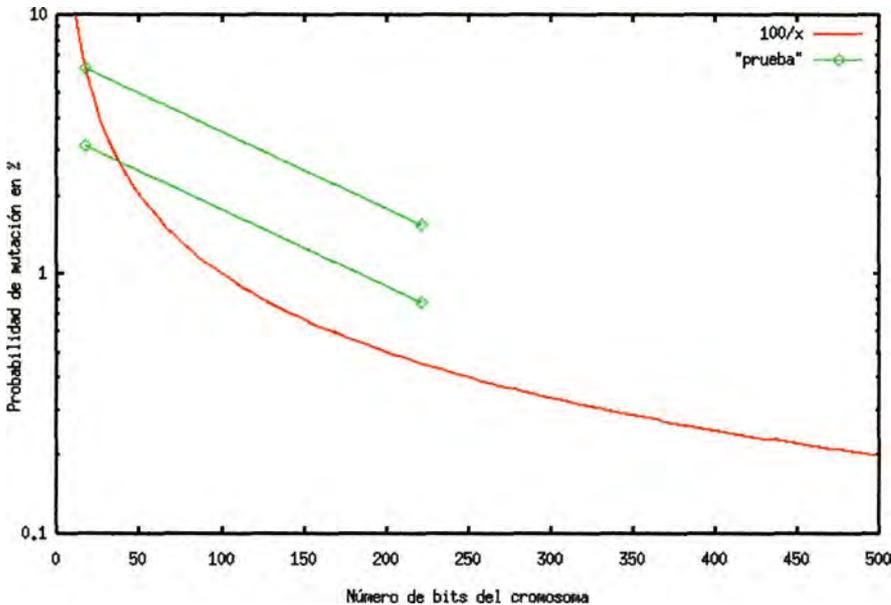


Figura 4.149. Probabilidades de mutación más adecuadas obtenidas empíricamente

Basándonos en los resultados obtenidos a partir de los dos casos test ensayados y considerando una

función del tipo: $a + \frac{b}{x+b}$, se obtiene a partir de los intervalos de probabilidades de mutación observados experimentalmente como más apropiados, las dos funciones representadas en las ecuaciones 4.7 y 4.8:

$$\text{Como límite superior: } 1.1150212 + \frac{100}{x+3.4742771} \quad (4.7)$$

$$\text{Como límite inferior: } 0.3649852 + \frac{100}{x+20.2316896} \quad (4.8)$$

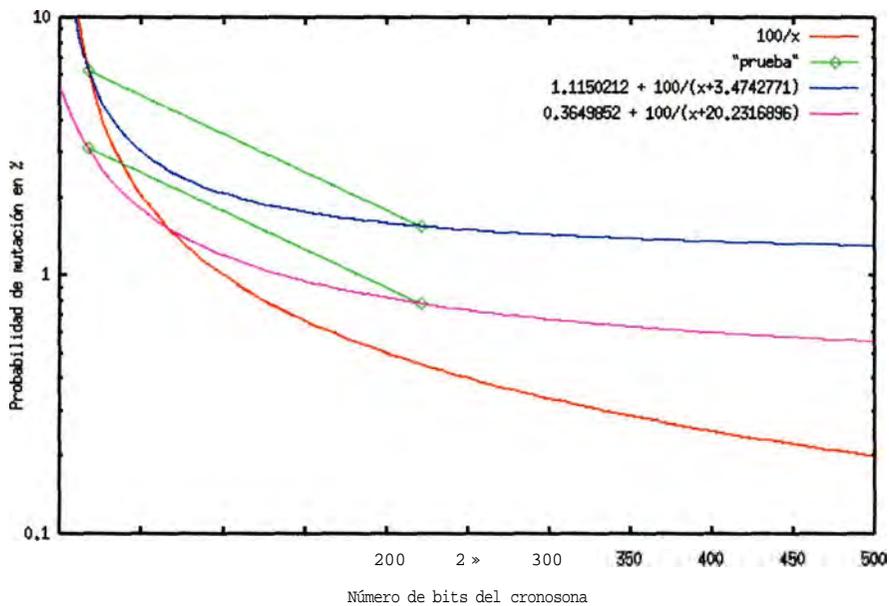


Figura 4.150. Aproximación de probabilidades de mutación.

Estas funciones propuestas quedan representadas en la figura 4.150. Un estudio posterior con mayor número de estructuras que permita mayor número de puntos permitirá realizar una regresión por mínimos cuadrados que nos indique una aproximación más certera a una ley para establecer una probabilidad de mutación adecuada en el problema multiobjetivo a resolver.

4.7 Comparativa entre Algoritmos Evolutivos Multiobjetivo en Optimización Estructural

Se comparan a continuación los algoritmos implementados. Se consideran los algoritmos de segunda generación NSGAI y SPEA2. Además, el algoritmo NSGAI con elitismo controlado, con dos valores del parámetro r que regula el número de soluciones de los sucesivos frentes con el fin de independizar los dos efectos del elitismo controlado. Se emplea un primer valor de $r=0.0$, que proporciona un nulo efecto sobre la distribución de frentes sucesivos, no restringiendo el número de soluciones de cada uno de ellos, pero sí suavizando la presión de selección del algoritmo al eliminar la selección por torneo en la elección de la población intermedia previa al cruce. Un segundo valor de $r=0.4$ permite además fijar la distribución del número de soluciones máximas posibles en los sucesivos frentes. Se completan estos algoritmos con los algoritmos DENSEA y DENSEA-NSH, donde se ha incorporado eliminación de duplicados. Estos seis casos se han implementado también en su versión de estado estacionario (steady-state).

Para posibilitar una interpretación de las gráficas de resultados más clara, se han ordenado cualitativamente los cuatro mejores algoritmos para cada uno de los casos considerados -3 tamaños poblacionales y 4 probabilidades de mutación distintas totalizan 12 combinaciones por algoritmo- y se han representado en las tablas 4.23 a 4.28 para el caso test X y en las tablas 4.29 a 4.34 para el caso test Y. La tasa de mutación que ofrece peores resultados (la menor tasa, 0.8% en el caso X y la mayor tasa, 3% en el caso Y) ha sido eliminada de las estadísticas que sí consideran las otras tres probabilidades. Se ha contabilizado el número de apariciones del algoritmo en esta ordenación de los cuatro mejores (N) y el puesto medio resultante de ellas (F_u). Las dos primeras tablas diferencian los casos generacionales y de estado-estacionario, mostrándose los resultados acumulados en las tablas 4.25 y 4.28 para el caso test X y en las tablas 4.31 y 4.34 para el caso test Y.

4.7.1 Caso Test X. Comparativa entre Algoritmos

Respecto de la métrica acercamiento al frente, la tabla 4.23 muestra la ordenación de algoritmos con reemplazamiento generacional, la tabla 4.24 se corresponde con el reemplazamiento de estado estacionario, mientras que la 4.25 ofrece los resultados acumulados por algoritmo.

Tabla 4.23. Comparativa entre Algoritmos en Caso X. Reemplazamiento Generacional. Acercamiento al frente

Caso X Generacional	50				100				200				Pu	N	
	0.8	1.5	3.0	6.0	0.8	1.5	3.0	6.0	0.8	1.5	3.0	6.0			
Gen-SPEA2								4°					$\hat{\mu}_h$: 4.00	1	
Gen-NSGAI			4°	4°	3°	40	40	3°					40	3.83	6
Gen-NSGAIelcontr00	4°	4°			40				40	40	40	3°	3.75	4	
Gen-NSGAIel.cont.r04	3°	2°	3°	3°		3°	3°		3°	3°	3°		2.86	7	
Gen-DENSEA	2°	3°	2°	1°	2°	2°	2°	2°	2°	2°	2°	2°	2.00	9	
Gen- DENSEA-NSH	1°	1°	1°	2°	1°	r	1°	r	r	1°	1°	1°	1.11	9	

Capítulo 4. Contribuciones

Tabla 4.24. Comparativa entre Algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Acercamiento al frente

Caso X SteadyState	50				100				200				Pu	N
Probab. Mutación (%)	0.8	1.5	3.0	6.0	0.8	1.5	3.0	6.0	0.8	1.5	3.0	6.0	^ κ	
St-St-SPEA2												40	4.00	1
St-St-NSGAI			4°		40				3°				4.00	1
St-St-NSGAIel.cont.r00	4°	40		40		40	40	40	40		3°		3.86	7
St-St-NSGAIel.cont.r04	3°	3°	3°	3°	3°	3°	3°	3°		3°	40	3°	3.11	9
St-St-DENSEA	r	1°	1°	2°	2°	2°	1°	2°	r	1°	1°	2°	1.44	9
St-St-DENSEA-NSH	2°	2°	2°	r	1°	r	2°	1°	2°	2°	2°	1°	1^6	9

Tabla 4.25. Comparativa entre Algoritmos en Caso X. Resultados Globales. Acercamiento al frente

Caso test X	Pu-Gen	N-Gen	Pu-StSt	N-StSt	PuX	NX
SPEA2	>>.00	1	4.00	1	4.00	2
NSGAI	3.83	6	4.00	1	3.85	7
NSGAIel.cont.r00	3.75	4	3.86	7	3.82	11
NSGAIel.cont.r04	2.86	7	3.11	9	3.00	16
DENSEA	2.00	9	1.44	9	1.72	18
DENSEA-NSH	1.11	9	1.56	9	1.34	18

Respecto de la métrica amplitud del frente, la tabla 4.26 muestra la ordenación de algoritmos con reemplazamiento generacional, la tabla 4.27 se corresponde con el reemplazamiento de estado estacionario, mientras que la 4.28 ofrece los resultados acumulados por algoritmo.

Amplitud del frente:

Tabla 4.26. Comparativa entre Algoritmos en Caso X. Reemplazamiento Generacional. Amplitud del frente

Caso X Generacional	50				i(10)				2(0)				Pu	N
Probab. Mutación (%)	0.8	1.5	3.0	6.0	0.8	1.5	3.0	6.0	0.8	1.5	3.0	6.0		
Gen-SPEA2	40							3°	40				3.00	1
Gen-NSGAI		40	40	40	3°	3°	40			40			3.83	6
Gen-NSGAIel.cont.r00											3°	3°	3.00	2
Gen-NSGAIel.cont.r04	3°	3°	3°	2°	40	40	3°	40	3°	3°	40	40	3.33	9
Gen-DENSEA	2°	2°	2°	1°	2°	T	2°	2°	2°	1°	2°	2°	1.78	9
Gen- DENSEA-NSH	1°	r	1°	3°	1°	1°	r	1°	1°	2°	1°	1°	1.33	9

Tabla 4.27. Comparativa entre Algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Amplitud del frente

Caso X SteadyState	50				i(0)				2(0)				Pu	N	
Probab. Mutación (%)	0.8	1.5	3.0	6.0	0.8	1.5	3.0	6.0	0.8	1.5	3.0	6.0			
Gen-SPEA2					40								1°	1.00	1
Gen-NSGAI		40	40	3°		40			3°				3.75	4	
Gen-NSGAIel.cont.r00	40						40	40		3°	r		3.00	4	
Gen-NSGAIel.cont.r04	3°	3°	3°	40	3°	3°	3°	3°	40	40	40	40	3.44	9	
Gen-DENSEA	r	1°	1°	2°	2°	2°	1°	2°	r	P	2°	3°	1.67	9	
Gen- DENSEA-NSH	2°	2°	2°	1°	1°	r	T	1°	2°	2°	3°	2°	W^A	9	

Tabla 4.28. Comparativa entre Algoritmos en Caso X. Resultados Globales. Amplitud del frente

Caso test X	Pu-Gen	N-Gen	Pu-StSt	N-StSt	PuY	NY
SPEA2	K3	1	1	1	2.00	2
NSGAII	3.83	6	3.75	4	3.80	10
NSGAIIel.contr.00	3	2	3	4	3.00	6
NSGAIIel.contr.04	3.33	9	3.44	9	3.38	18
DENSEA	1.78	9	1.67	9	1.73	18
DENSEA-NSH	1.33	9	1.78	9	1.56	18

Se muestran a continuación las figuras de convergencia correspondientes a la probabilidad de mutación del 6% tanto para la estrategia generacional como para la de estado estacionario, estando el resto de figuras en el *Anexo*.

Caso X) Diferentes algoritmos: Probabilidad Mutación 6%; Estrategia Generacional (4.160 a 4.162)

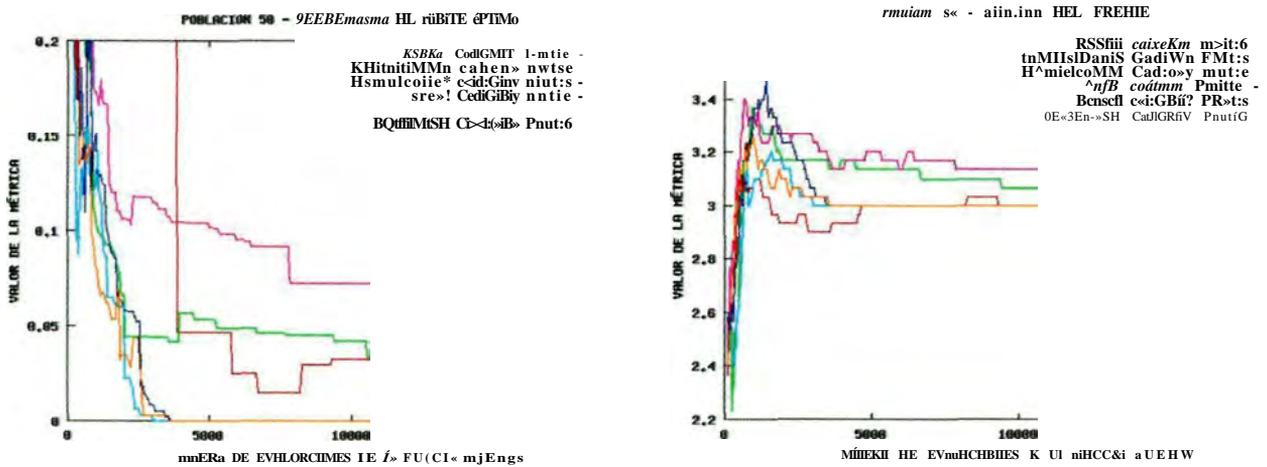


Figura 4.160. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 6.0% y tamaño de población 50.

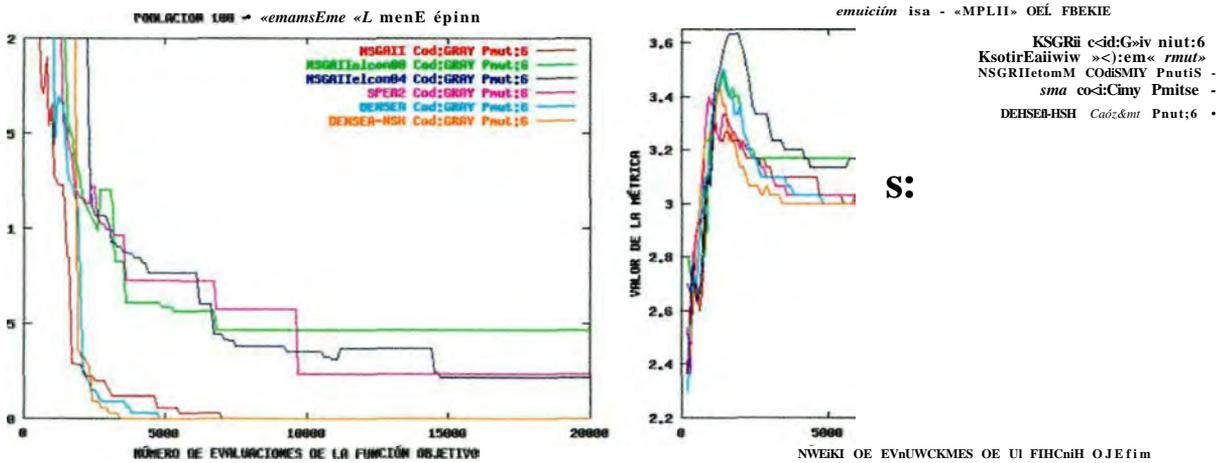


Figura 4.161. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 6.0% y tamaño de población 100.

Capítulo 4. Contribuciones

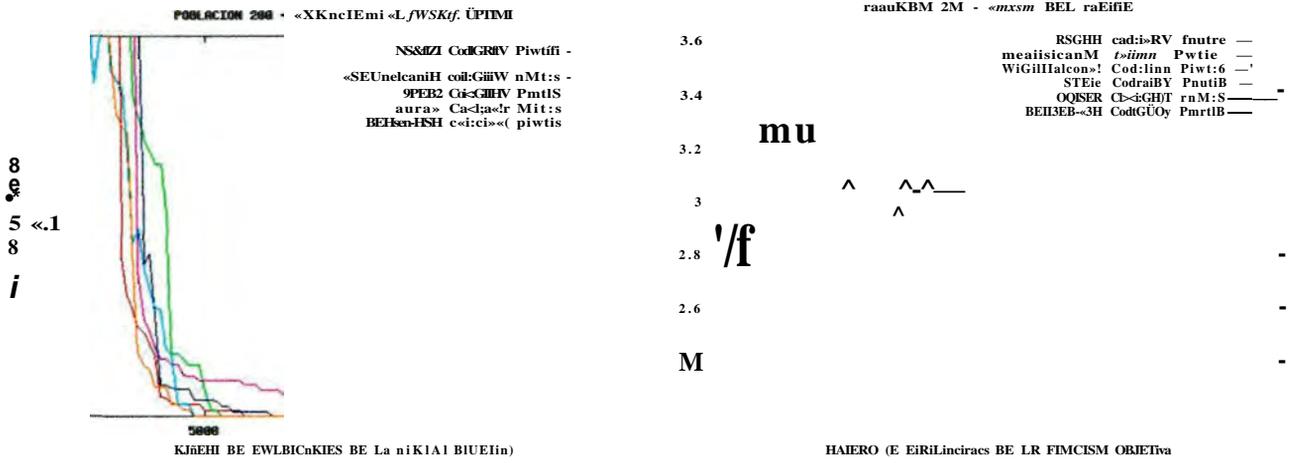


Figura 4.162. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 6.0% y tamaño de población 200.

Caso X) Diferentes algoritmos: Probabilidad Mutación 6%; Estrategia Steady-State (4.172 a 4.174)

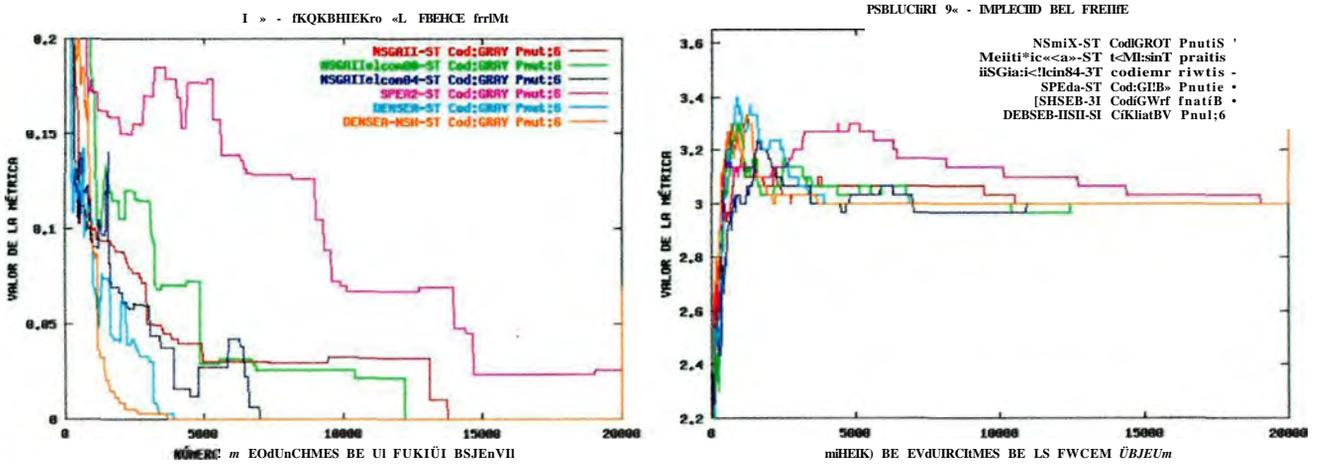


Figura 4.172. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 6.0% y tamaño de población 50.

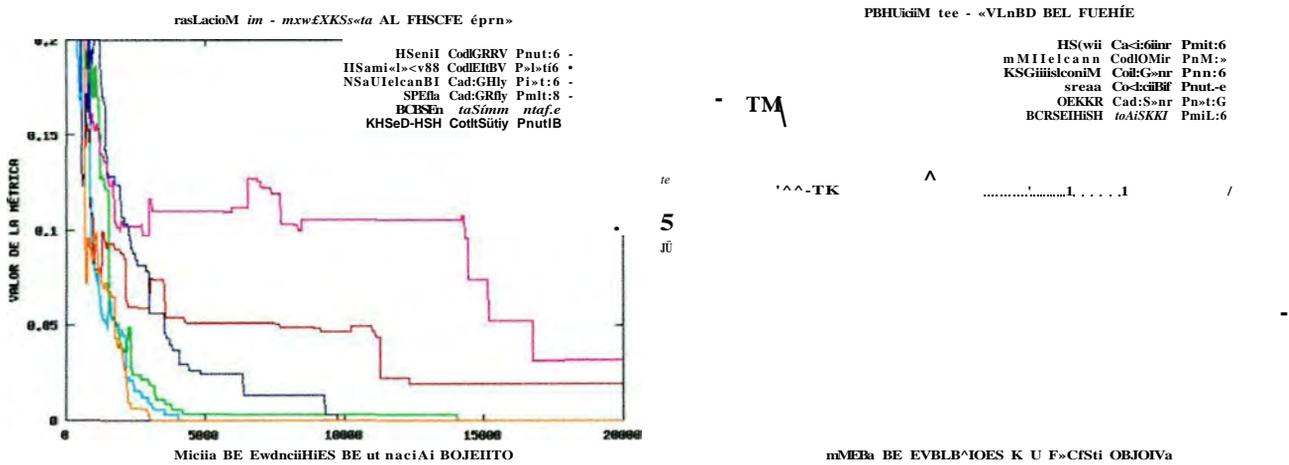


Figura 4.173. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 6.0% y tamaño de población 100.

Capítulo 4. Contribuciones

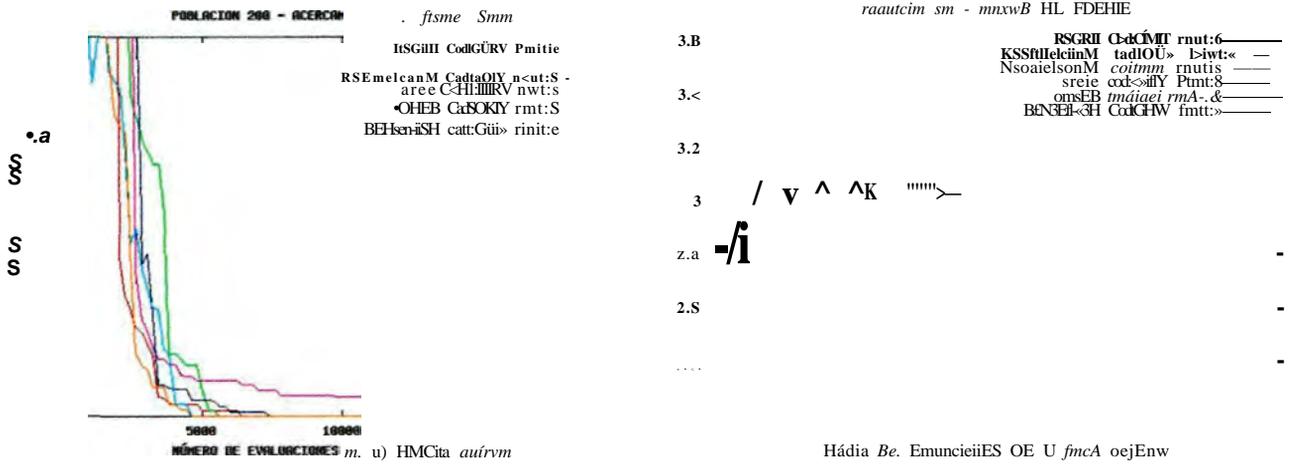


Figura 4.162. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 6.0% y tamaño de población 200.

Caso X) Diferentes algoritmos: Probabilidad Mutación 6%; Estrategia Steady-State (4.172 a 4.174)

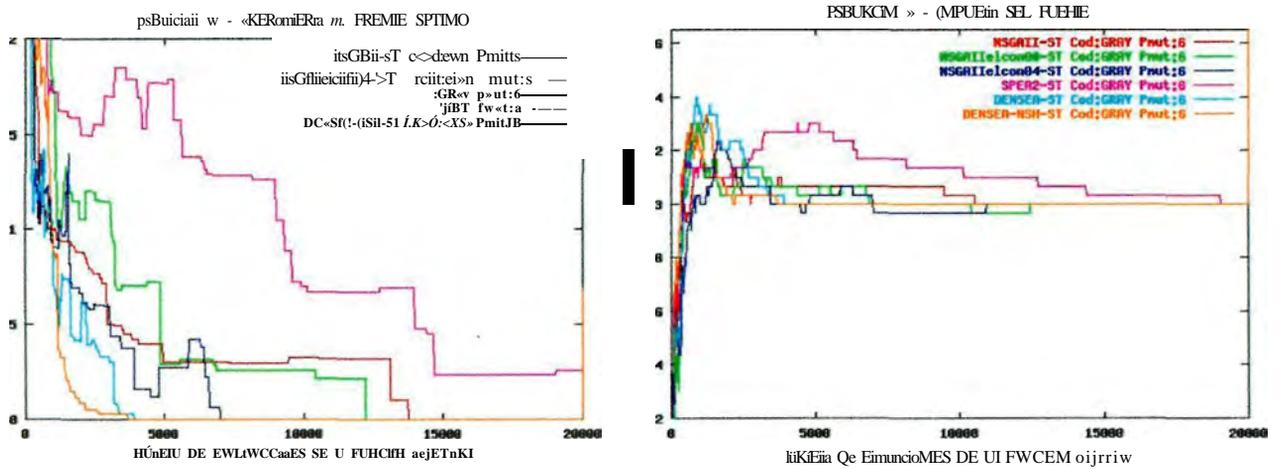


Figura 4.172. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Estado Estacionario. Probabilidad de mutación 6.0% y tamaño de población 50.

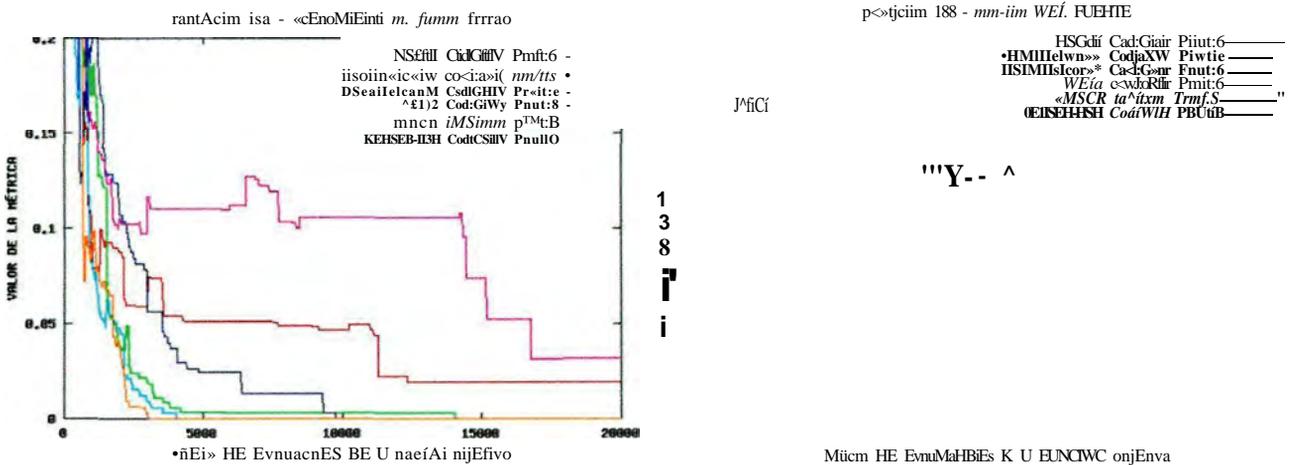


Figura 4.173. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Estado Estacionario. Probabilidad de mutación 6.0% y tamaño de población 100.

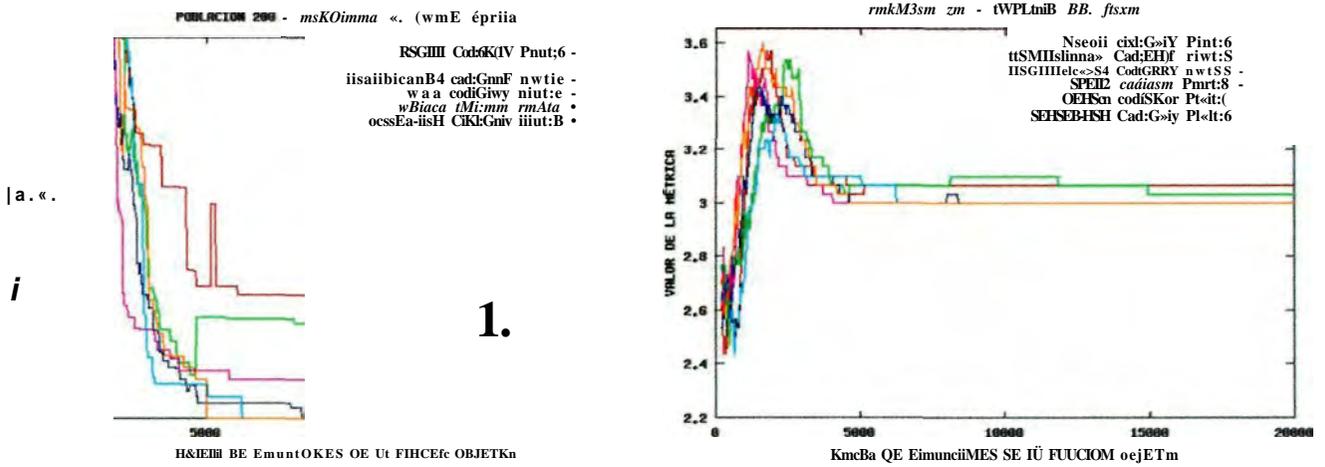


Figura 4.174. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 6.0% y tamaño de población 200.

Cuantitativamente se pueden comparar los algoritmos también en este caso X a partir de las figuras 4.71 a 4.76 mostradas en la sección 4.6, donde se aprecia claramente el mejor comportamiento de los algoritmos DENSEA y DENSEA-NSH frente al resto, al requerir menor número de evaluaciones (barras más bajas), con menor desviación típica (barras más estrechas) y localizar en mayor número de ejecuciones sobre las 30 ensayadas el frente óptimo.

4.7.2 Caso Test Y. Comparativa entre Algoritmos

Respecto de la métrica acercamiento al frente la tabla 4.29 muestra la ordenación de algoritmos con reemplazamiento generacional, la tabla 4.30 se corresponde con el reemplazamiento de estado estacionario, mientras que la 4.31 ofrece los resultados acumulados por algoritmo.

Acercamiento al frente:

Tabla 4.29. Comparativa entre Algoritmos en Caso Y. Reemplazamiento Generacional. Acercamiento al frente

Caso test Y Generac.	50				100				2{10				Pu	N
Probab. Mutación (%)	0.4	0.8	1.5	3.0	0.4	0.8	1.5	3.0	0.4	0.8	1.5	3.0		
Gen-SPEA2				2°				2°				T		
Gen-NSGAI			40	1°	T	40	1°	40	40	T	1°	3.33	6	
Gen-NSGAIel.contr.OO	40	40	2°	40	1°	40					4°	3.17	6	
Gen-NSGAIel.coit.r04	3°	3°	3°	40		3°		3°	3°			3.14	7	
Gen-DENSEA	2°	2°	1°		2°	3°	1°	3°	1°	1°	1°	40	1.55	9
Gen-DENSEA-NSH	1°	1°		3°	3°	r	T	40	T	2°	3°	3°	1.87	8 i

Tabla 4.30. Comparativa entre Algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Acercamiento al frente

Casotest Y SteadState	50				1{10				2{10				Pu	N
Probab. Mutación (%)	0.4	0.8	1.5	3.0	0.4	0.8	1.5	3.0	0.4	0.8	1.5	3.0		
St-St-SPEA2			3°					40				40	3.00	1
St-St-NSGAI		T		1°	40			r			3°	3°	3.00	3
St-St-NSGAIel.contr.OO	40	1°			40	3°		r	3°	40	40		3.28	7
St-St-NSGAIel.cont.r04	3°		1°	40	3°	3°	40		40	3°			3.00	7
St-St-DENSEA	2°	40	2°	2°	1°	r	r	2°	1°	2°	r	r	1.66	9
St-St-DENSEA-NSH	r	3°	40	3°	2°	2°	2°	3°	T	r	r	2°	2.11	9

Tabla 4.31. Comparativa entre Algoritmos en Caso Y. Resultados Globales. Acercamiento al frente

Caso test Y	Pu-Gen	N-Gen	Pu-StSt	N-StSt	PuY	NY
SPEA2	W		3.00	1	3.00	1
NSGAI	3.33	6	3.00	3	3.22	9
<u>NSGAIel.cont.rOO</u>	3.17	6	3.28	7	3.23	13
<u>NSGAIel.cont.r04</u>	3.14	7	3.00	7 4	3.07	14
DENSEA	1.55	9	1.66	9 1	1.60	18
DENSEA-NSH	1.87	8	2.11	9 J	2.00	17

Observando las tablas 4.29 a 4.31, podemos inferir claramente, que tanto para el caso de reemplazamiento generacional, como para el caso de estado estacionario, los algoritmos con mayor número de apariciones (N), así como con el índice menor, indicativo de mayor calidad en el acercamiento al frente también en este caso test Y, son los algoritmos DENSEA y DENSEA-NSH.

Respecto de la métrica amplitud del frente la tabla 4.29 muestra la ordenación de algoritmos con reemplazamiento generacional, la tabla 4.30 se corresponde con el reemplazamiento de estado estacionario, mientras que la 4.31 ofrece los resultados acumulados por algoritmo.

Amplitud del frente:

Tabla 4.32. Comparativa entre Algoritmos en Caso Y. Reemplazamiento Generacional. Amplitud del frente

Caso test Y Generac.	50				100				200				Pu	N
Probab. Mutación (%)	0.4	0.8	1.5	3.0	0.4	0.8	1.5	3.0	0.4	0.8	1.5	3.0		
Gen-SPEA2	1°	3°	2°	3°	2°	2°	2°		1°	1°	r	40	1.67	9
Gen-NSGAI		1°	3°		40				3°				2.75	4
<u>Gen-NSGAIel.cont.rOO</u>			4°	1°			1°	r	4°		40	2°	3.25	4
<u>Gen-NSGAIel.cont.r04</u>	2°	2°	1°	2°	1°	r	3°	2°	2°	3°		1°	1.87	8
Gen-DENSEA	4°	4°		40		3°		40		40	3°	3°	3.6	5
Gen-DENSEA-NSH	3°				3°	40	40	3°		2°	2°		3.00	6 j

Tabla 4.33. Comparativa entre Algoritmos en Caso Y. Reemplazamiento Estado Estacionario. Amplitud del frente

Caso test Y SteadState	50				i(0)				2(0)				Pu	N
Probab. Mutación (%)	0.4	0.8	1.5	3.0	0.4	0.8	1.5	3.0	0.4	0.8	1.5	3.0		
St-St-SPEA2	1°	r	r		2°	r	1°		2°	2°	r		1.33	9
St-St-NSGAH										40		40	4.00	1
<u>St-St-NSGAIel.cont.rOO</u>			2°	1°	1°	40	40	2°			2°	1°	2.60	5
<u>St-St-NSGAIel.cont.r04</u>	2°	2°	40	2°	3°	2°	2°	r	r	3°	3°	2°	2.44	9
St-St-DENSEA	40	40		3°			3°	3°	40		40		3.80	5
St-St-DENSEA-NSH	3°	3°	3°	40	40	3°		40	3°	1°		30	6	7

Tabla 4.34. Comparativa entre Algoritmos en Caso Y. Resultados Globales. Amplitud del frente

Caso test Y	Pu-Gen	N-Gen	Pu-StSt	N-StSt	PuY	NY
SPEA2	1.67	9	1.33	9	1.50	18
NSGAI	2.75	4	4	1	3.00	5
<u>NSGAIel.cont.rOO</u>	3.25	4	2.6	^	2.89	9
<u>NSGAIel.cont.r04</u>	1.87	8	2.44	9 i	2.18	17
DENSEA	3.6	5	3.8	5	3.70	10
DENSEA-NSH	i 3	6	2.86	7 1	2.92	13

De las tablas 4.32 a 4.34, se observa que en términos de amplitud del frente, el mejor algoritmo, tanto considerando el número de apariciones (N), como su calidad (PuY), es el SPEA2, si bien este algoritmo tenía un pésimo desempeño relativo en la métrica de acercamiento al frente. Le sigue el algoritmo NSGAI con elitismo confrolado y $r=0.4$, con sólo una aparición menos y menor índice de calidad, si bien considerando el acercamiento al frente es mucho más robusto que el algoritmo SPEA2 y con mejores resultados (ver tabla 4.31). Tras él en ftmcción del número de apariciones (N) y en calidad (PuY) vendrían los algoritmos DENSEA y DENSEA-NSH, que en este caso test Y presentan los mejores resultados en aproximación al frente óptimo, si bien a costa de una disminución relativa de la amplitud del frente. Esta disminución está asociada a la menor localización de las soluciones con menor número de perfiles distintos, que son las soluciones con mayor dificultad de localización, si bien estas soluciones exfremas no suelen ser las más preferibles en el diseño.

Se muestran a continuación las figuras de convergencia correspondientes a la probabilidad de mutación del 0.4% y 0.8% para los tres tamaños poblacionales ensayados, tanto para la esfrategia generacional como para la de estado estacionario, estando el resto de figuras en el *Anexo*.

Caso Y) Diferentes algoritmos: Probabilidad Mutación 0.4%; Esfrategia Generacional (4.175 a 4.177)

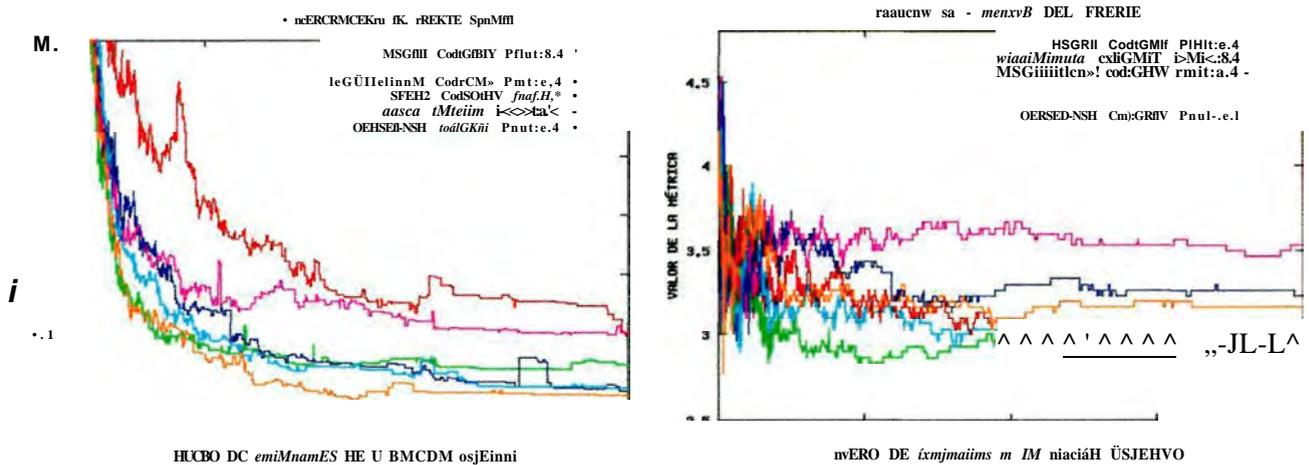


Figura 4.175. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 0.4% y tamaño de población 50.

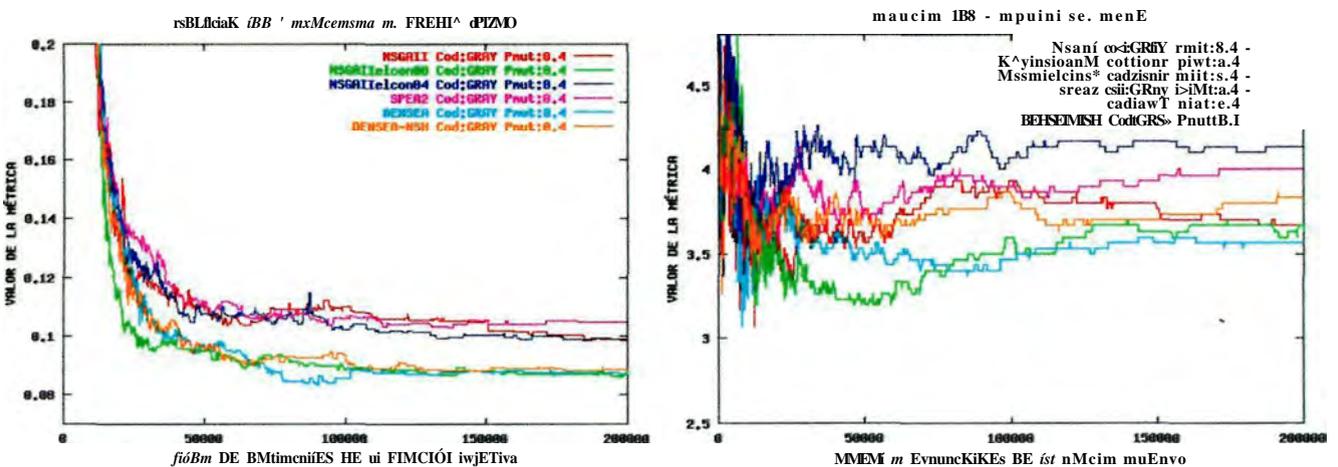


Figura 4.176. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 0.4% y tamaño de población 100.

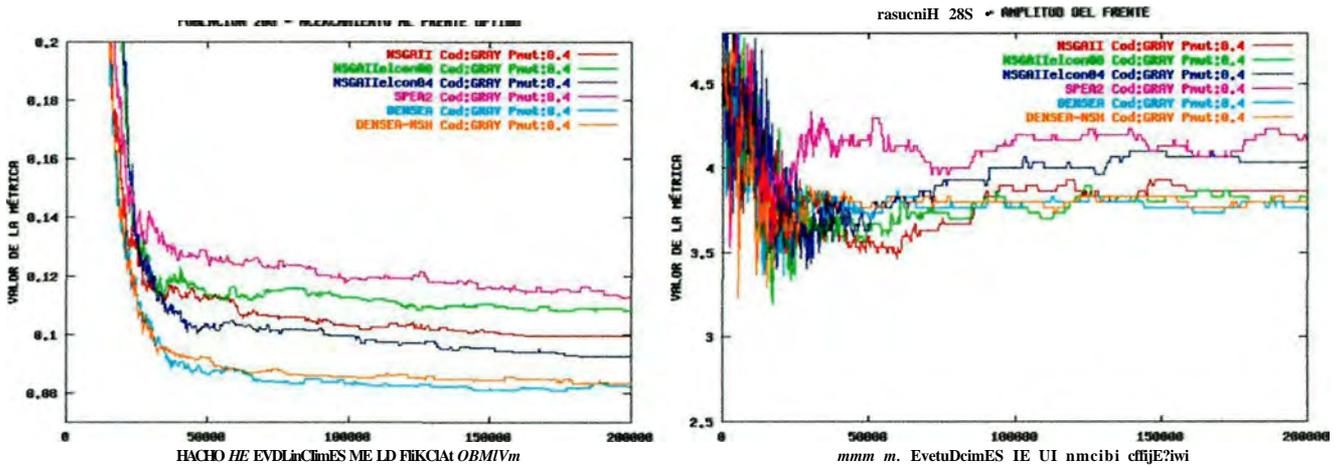


Figura 4.177. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 0.4% y tamaño de población 200.

Caso Y) Diferentes algoritmos: Probabilidad Mutación 0.8%; Estrategia Generacional (4.178 a 4.180)

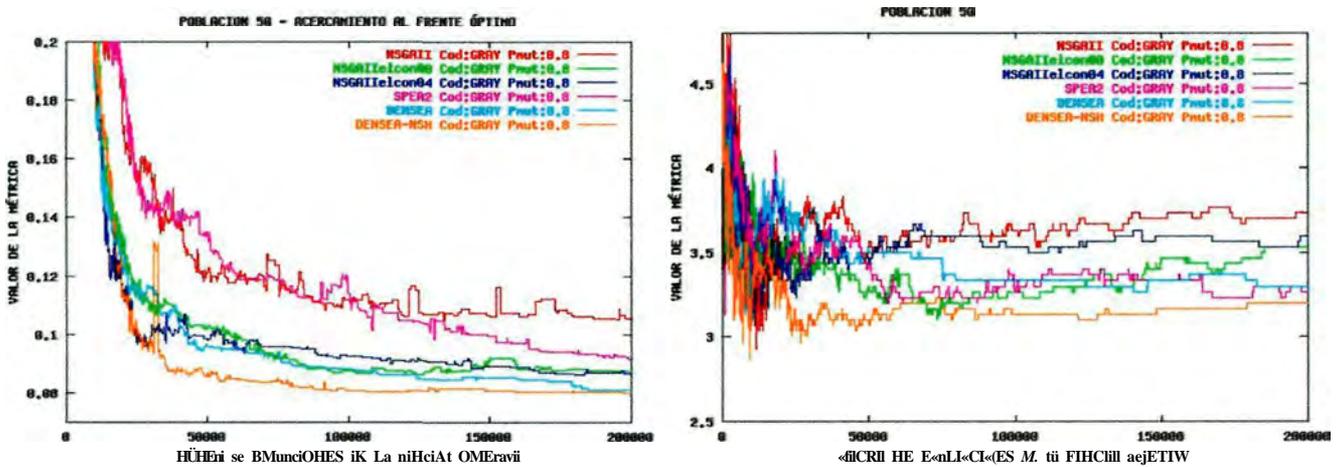


Figura 4.178. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 0.8% y tamaño de población 50.

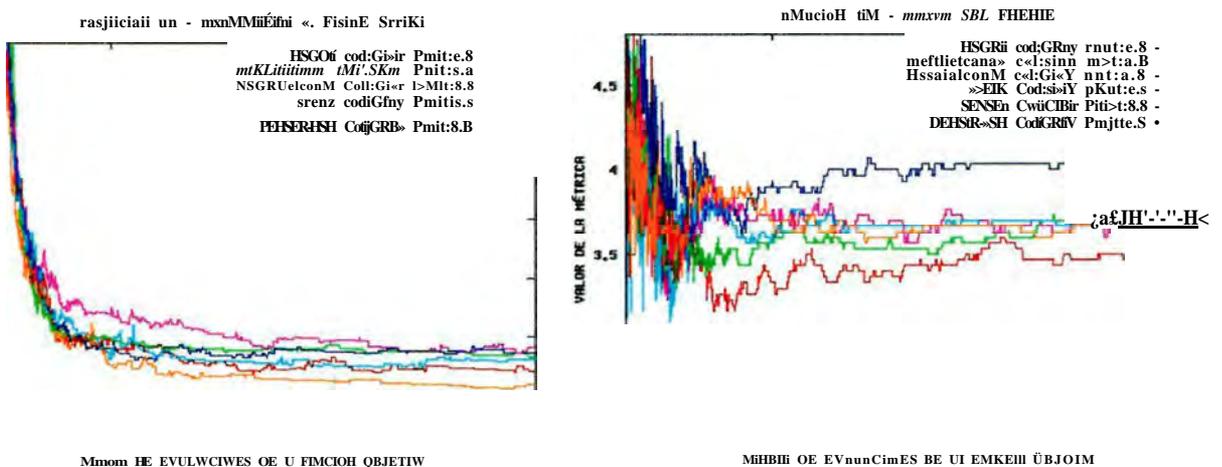


Figura 4.179. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 0.8% y tamaño de población 100.

Capítulo 4. Contribuciones

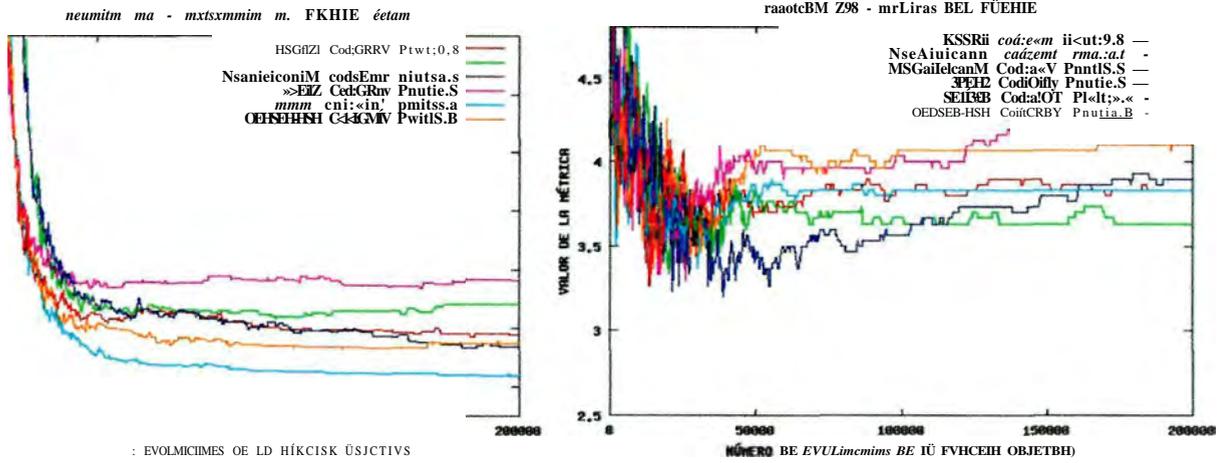


Figura 4.180. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 0.8% y tamaño de población 200.

Caso Y) Diferentes algoritmos: Probabilidad Mutación 0.4%; Estrategia Steady-State (4.187 a 4.189)

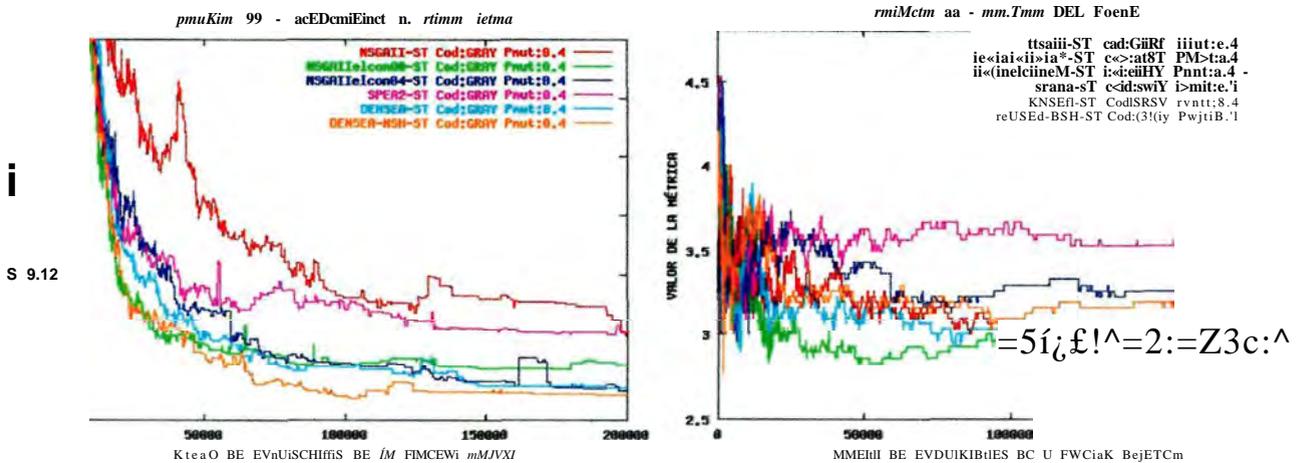


Figura 4.187. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 0.4% y tamaño de población 50.

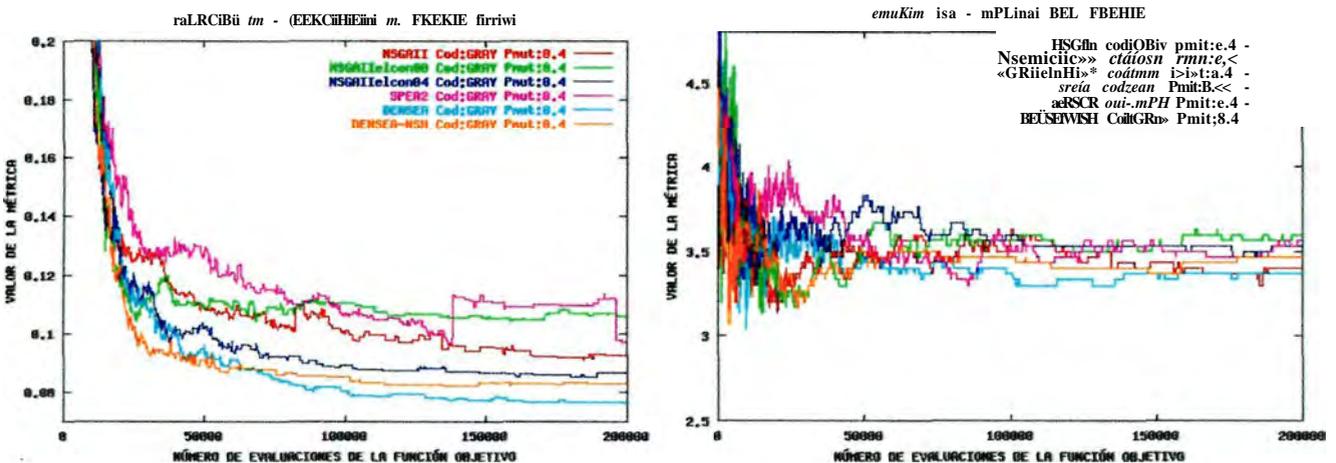


Figura 4.188. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 0.4% y tamaño de población 100.

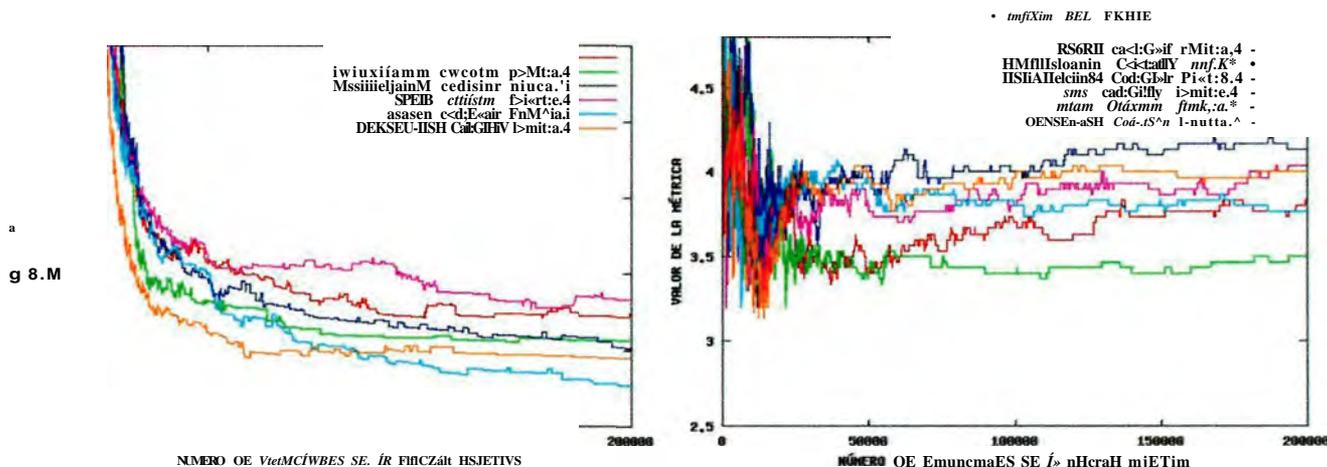


Figura 4.189. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 0.4% y tamaño de población 200.

Caso Y) Diferentes algoritmos: Probabilidad Mutación 0.8%; Estrategia Steady-State (4.190 a 4.192)

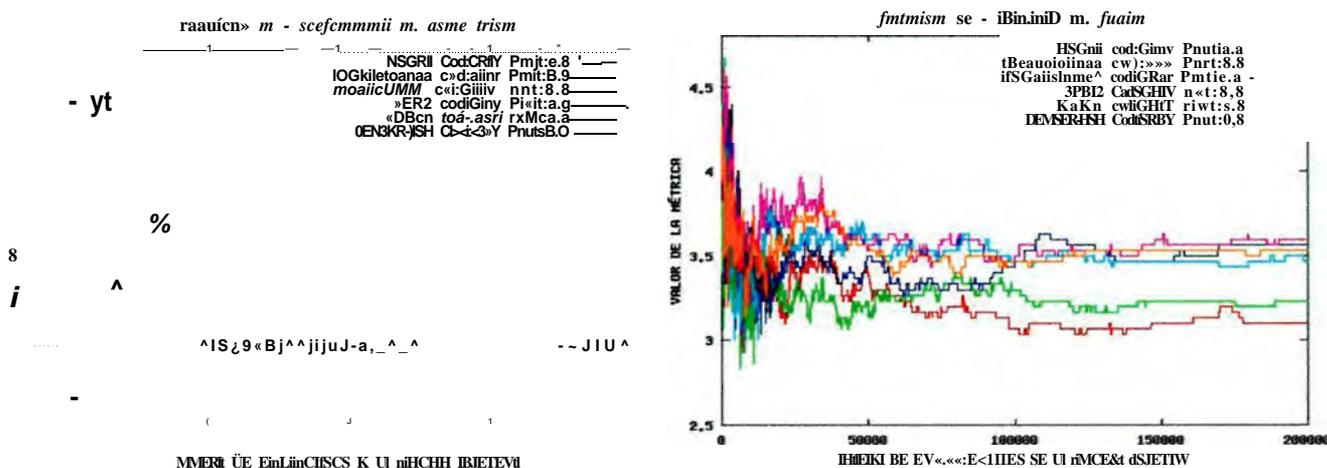


Figura 4.190. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 0.8% y tamaño de población 50.

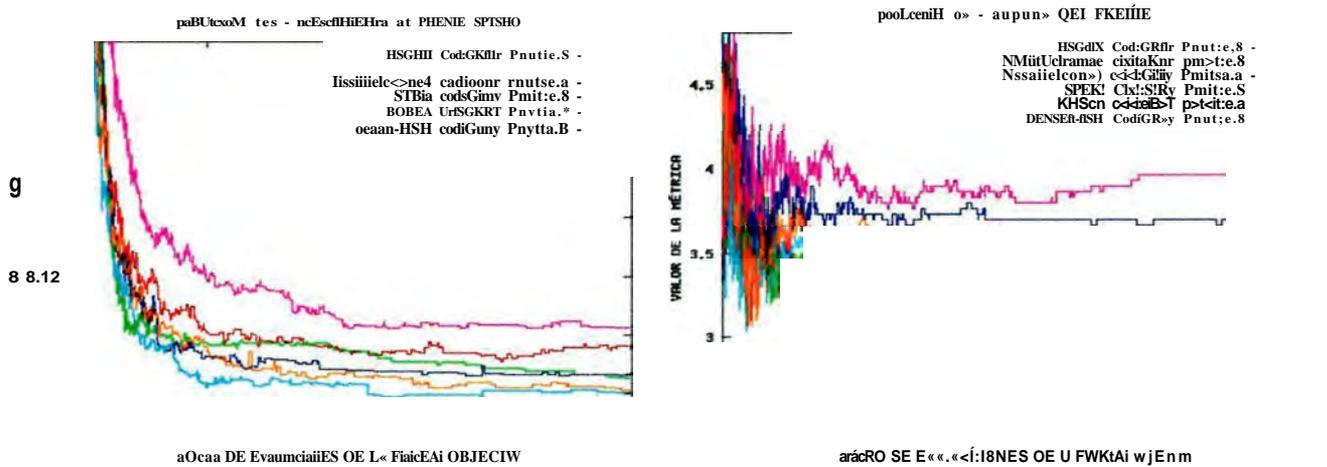


Figura 4.191. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 0.8% y tamaño de población 100.

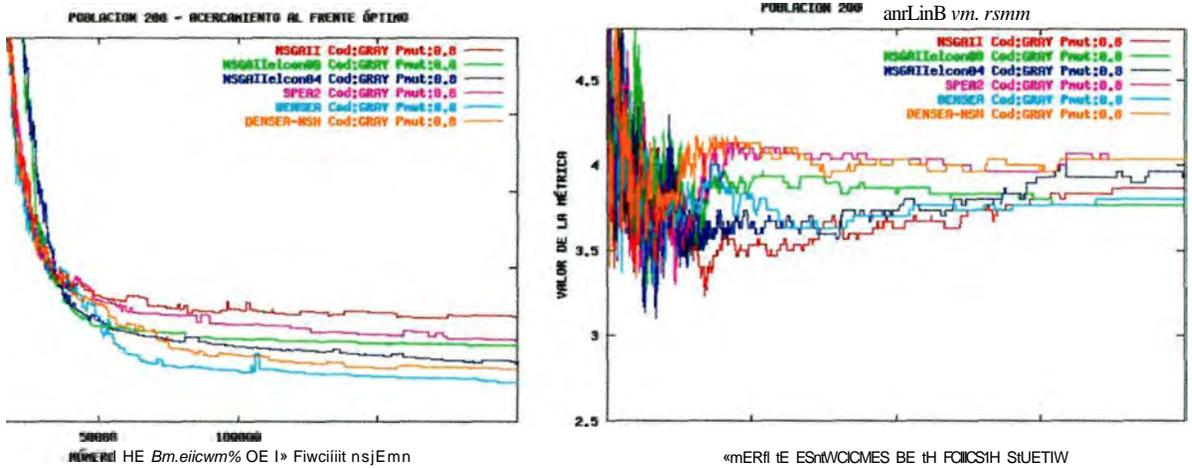


Figura 4.192. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 0.8% y tamaño de población 200.

4.7.3 Análisis de Resultados. La Diversidad Poblacional

En la tabla 4.35, se muestran los resultados acumulados para la métrica acercamiento al frente, considerando cada uno de los casos test (X e Y), los algoritmos generacionales y de estado estacionario (Gen y StSt), además de los resultados totales acumulados (Tot).

Tabla 4.35. Comparativa entre Algoritmos en Casos X e Y. Resultados Globales Acumulados. Acercamiento al frente.

Casos test X+Y	Pu-Gen	N-Gen	Pu-StSt	N-StSt	PuX	NX	PuY	NY	PuTot	Ntot
SPEA2	4.00	1	3.50	2	4.00	2	3.00	1	3.67	3
NSGAII	3.58	12	3.25	4	3.85	7	3.22	9	3.50	16
<u>NSGAIIeLcoiit.rOO</u>	3.40	10	3.57	14	3.82	11	3.23	13	3.50	24
<u>NSGAIIeL.cont.r04</u>	3.00	14	3.06	16	3.00	16	3.07	14	3.03	30
DENSEA	1.77	18	1.55	18	1.72	18	1.60	18	1.66	36
DENSEA-NSH	1.47	17	1.01	18	1.34	18	2.00	17	1.66	35

Cualitativamente, se aprecia que los algoritmos DENSEA y DENSEA-NSH tienen el mejor índice, y ambos con resultados similares acumulados (PuTot=1.66), además de presentar también el mejor índice de frecuencia (sobre un total de 36 posibles casos, sólo una ausencia). Este resultado es también observable para los resultados conjuntos de cada uno de los casos test (X e Y) por separado, así como en los resultados agrupados por tipo de reemplazamiento poblacional (generacional y de estado estacionario). Ello muestra la robustez de los mismos ante distintos tamaños de problema, reflejados en los dos distintos casos test. En un segundo lugar se encuentra el NSGAII con elitismo controlado y $r=0.4$, con un índice de frecuencia ligeramente inferior de 30 y un PuTot de 3.03. El tercer lugar estaría compartido por los algoritmos NSGAII y NSGAII con elitismo controlado con $r=0.0$. Ambos tienen igual índice promedio de ordenación, si bien la frecuencia de aparición del NSGAII estándar es un tercio inferior (16 frente a 24).

De acuerdo con estos resultados, la característica diferenciadora de los dos primeros algoritmos destacados (DENSEA y DENSEA-NSH), es la presencia de ciertos operadores que fomentan el mantenimiento y creación de diversidad en la población, destacando entre ellos la eliminación de

Si observamos los resultados conjuntos de ambos casos test X e Y para la métrica amplitud del frente en la tabla 37, vemos que el algoritmo SPEA2 obtiene el mejor índice de calidad en ambos métodos de reemplazamiento (generacional y estado estacionario), si bien con una robustez media (aparece sólo en la mitad de las ocasiones, $N = 20$). El algoritmo NSGAII con elitismo controlado y $r=0.4$ es el que tiene mayor número de apariciones (35 sobre un total de 36) si bien con peor índice de calidad que los algoritmos DENSEA y DENSEA-NSH, que poseen mejor índice (2.43 y 2.13 frente a 2.80) con un ligero decremento del número de apariciones (28 y 31, respectivamente). Ello unido a su amplia superioridad en acercamiento al frente, les coloca como los algoritmos más competitivos en los problemas estructurales multicriterio ensayados.

Tabla 4.37. Comparativa entre Algoritmos en Casos X e Y. Resultados Globales Acumulados. Amplitud Frente

Caso test X+Y	Pu-Gen	N-Gen	Pu-StSt	N-StSt	PuY	NY
SPEA2	1.80	10	1.30	10	1.50	20
NSGAII	3.40	10	3.80	5	3.53	15
<u>NSGAIIel.cont.r00</u>	3.17	6	2.78	9	2.93	15
<u>NSGAIIel.cont.r04</u>	2.65	17	2.94	18	2.80	35
DENSEA	2.43	14	2.43	14	2.43	28
DENSEA-NSH	2.00	15	2.25	16	2.13	31

Tras haber analizado los algoritmos propuestos se destaca la importancia de garantizar una adecuada diversidad poblacional como constituyente de una adecuada exploración del algoritmo y un desempeño mejor del mismo, aspecto especialmente destacable cuando el espacio de búsqueda tiene componentes discretos.

Este hecho evidenciado en este problema estructural a través del mejor comportamiento de los algoritmos DENSEA y DENSEA-NSH se ha visto muy recientemente también apoyado por resultados en otros problemas discretos, tales como paisajes-MNK, en Aguirre y Tanaka (2005) [4.2], donde se emplea la eliminación de duplicados en el algoritmo NSGAII; y en problemas de optimización combinatoria multiobjetivo (problema de la mochila o 'knapsack'), en Nojima et al. (2005) [4.36], donde se eliminan las soluciones sobrelapadas. En ambos casos se abordan problemas discretos multiobjetivo donde se observan resultados que se benefician de la eliminación de duplicados.

4.8 La MultiObjetivización del Problema de Optimización Estructural de Masa Restringida

Los últimos métodos multiobjetivo evolutivos han evidenciado en múltiples aplicaciones reales resolviendo problemas complejos que son capaces de proporcionar soluciones, sin sobrecoste significativo comparados con optimizadores monocriterio [4.11]. Además, recientemente se ha tratado de aprovechar las nuevas posibilidades de optimización abiertas por los métodos multicriterio evolutivos para mejorar la búsqueda en problemas de optimización monocriterio. Entre otras referencias se destacan las siguientes, que introducen conceptos como 'multiobjetivización' o 'criterios ayudantes':

En Knowles et al. [4.31] se proponen dos vías para disminuir el número de óptimos locales en una búsqueda: 1º) Añadiendo nuevos objetivos al problema que permitan resolverlo como un problema multiobjetivo de tal modo que el mayor número de óptimos locales posible quede comprendido en el frente de Pareto, donde el óptimo global deberá estar incluido; o 2º) Descomponiendo el problema en subproblemas más sencillos, cuyas soluciones serán los objetivos a optimizar en el problema multiobjetivo. De este modo se pretende aumentar el número de caminos hacia el óptimo global no accesibles en una optimización monocriterio. Esta transformación es denominada 'multiobjetivización'. En la multiobjetivización la función escalar a optimizar en un problema monocriterio es sustituida por una función vectorial donde el problema multiobjetivo resultante tiene óptimos de Pareto coincidentes con los óptimos del problema original. Dos ejemplos son resueltos aplicando multiobjetivización: la función H-IFF (hierarchical-if-and-only-if-function) y el problema del viajante (travelling salesman). En la primera función es empleada la primera estrategia de multiobjetivización, mientras que en el segundo problema se sigue la segunda. Los resultados comparativos de distintos algoritmos 'hill-climbing' - sin emplear ningún operador de cruce- muestran mejores resultados mediante la multiobjetivización.

En Abbass y Deb [4.1] se propone la introducción de un nuevo criterio que permita la preservación de la diversidad del problema a resolver. Se proponen los siguientes posibles criterios adicionales: la maximización de la inversa de la función objetivo, maximización / minimización de un valor aleatorio asignado a cada cromosoma en su creación, y la maximización de la edad del cromosoma. Mediante la introducción de una segunda función objetivo se fomenta la conservación de soluciones que de otro modo se perderían durante la evolución del algoritmo. Se realiza un exhaustivo estudio en el caso del criterio edad del cromosoma, donde a cada uno se le asigna un valor atendiendo al orden de creación del mismo. En ausencia de mutación, la aproximación multiobjetivo mantiene mejor la diversidad genética y también logra mejores resultados frente a la monocriterio. También en E. de Jong et al. [4.30] y en S. Bleuer et al. [4.5] se aplica una programación genética multiobjetivo con un criterio añadido como introductor de diversidad en la población para resolver simultáneamente la minimización del tamaño del árbol solución y la resolución del problema de n-paridad.

En M. Jensen [4.28] se destaca cómo a través de la inclusión de nuevos criterios adicionales denominados 'criterios ayudantes' (helper-objectives) y resolviendo el problema como un problema multicriterio se puede contribuir a la disminución de algunas dificultades en optimización monocriterio como son: 1º) evitación de mínimos locales; 2º) mantenimiento de la diversidad a un nivel adecuado; 3º) identificación de buenos 'bloques de construcción' (building blocks). Los criterios ayudantes deben

cumplir el requisito de estar en conflicto con el criterio principal; en caso contrario la optimización simultánea del criterio principal y ayudante equivaldrá a la optimización monocriterio. Aplica este concepto a la optimización de un problema de asignación de tareas (job shop scheduling), donde genera como posible(s) criterio(s) ayudante(s) la minimización del tiempo de tareas individuales. Este criterio es dinámico, variando la tarea ayudante con el transcurso del algoritmo hasta cubrirlas todas. Consigue con un criterio ayudante mejorar la solución obtenida para una batería de 18 casos test distintos ensayados frente a la resolución del problema con algoritmos genéticos monocriterio.

Desde esta perspectiva se plantea a continuación una comparación del problema monocriterio de minimización de masa estructural restringida frente a ese mismo problema con el criterio ayudante del número de distintos tipos de perfiles.

Supongamos que nuestro objetivo es el diseño óptimo estructural con mínima masa restringida, en ese caso, ¿obtendremos alguna ventaja si resolvemos el problema mediante un método evolutivo multicriterio utilizando como criterio adicional el número de distintos tipos de perfiles?

Este criterio adicional cumple el requisito de estar en conflicto con el criterio principal, además de ser un criterio que origina un fi-ente de soluciones no dominadas que contiene la solución óptima del problema monocriterio buscado. El hecho de que durante la evolución del algoritmo multicriterio se mantengan soluciones adicionales proporcionará una diversidad poblacional diferente a la originada en la optimización monocriterio. Se analizará a continuación qué efectos origina sobre la optimización.

Se compara un algoritmo genético generacional elitista (en la población se conservan las dos mejores soluciones, elitismo de los dos mejores individuos) con los algoritmos DENSE A y DENSEA-NSH. En el caso del algoritmo genético monocriterio, se ha empleado tanto cruce de un punto (algoritmo clásico) como cruce uniforme, así como código Gray. Se muestran resultados para tres tamaños poblacionales: 50, 100 y 200 individuos y para cuatro probabilidades de mutación: 0.8%, 1.5%, 3% y 6% en el caso test X y 0.4%, 0.8%, 1.5% y 3% en el caso test Y.

Caso Test X

Se muestra en la tabla 4.38 para 20.000 evaluaciones la media y varianza sobre 30 ejecuciones independientes de los resultados finales obtenidos. En todos los algoritmos se obtuvo como mejor solución la óptima de mínima masa de 3324.32 kg. Se destaca en negrita y cursiva el mejor promedio para cada supuesto (desglosados según tamaño poblacional y probabilidad de mutación).

Tabla 4.38. Resultados promedios para 20.000 evaluaciones sobre 30 ejecuciones independientes en Caso Test X.

<i>Población</i>	<i>Generacional-1 pío</i>		<i>Generacional- Uniforme</i>		<i>DENSEA</i>		<i>DENSEA-NSH</i>	
	Media	Varianza	Media	Varianza	Media	Varianza	Media	Varianza
<i>50</i>								
Pmut 0.8%	3354.572	1871.463	3346.686	387.305	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
Pmut 1.5%	3348.763	482.695	3338.705	298.355	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
Pmut 3.0%	3330.926	149.614	3330.074	91.239	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
Pmut 6.0%	3328.844	30.700	3328.090	28.426	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
<i>Población</i>	Media	Varianza	Media	Varianza	Media	Varianza	Media	Varianza
<i>100</i>								
Pmut 0.8%	3346.564	463.680	3335.869	221.293	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
Pmut 1.5%	3340.416	333.517	3333.565	137.628	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
Pmut 3.0%	3329.975	31.979	3329.221	31.411	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
Pmut 6.0%	3327.336	25.015	3327.713	26.862	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
<i>Población</i>	Media	Varianza	Media	Varianza	Media	Varianza	Media	Varianza
<i>200</i>								
Pmut 0.8%	3339.139	244.344	3329.221	31.411	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
Pmut 1.5%	3328.844	30.700	3327.336	25.015	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
Pmut 3.0%	3328.090	28.426	3328.467	29.705	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>
Pmut 6.0%	3324.697	4.122	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>	<i>3324.320</i>	<i>0.000</i>

Se muestra en las figuras (4.199 a 4.201) donde se puede observar el valor promedio de la masa restringida a lo largo de toda la evolución del algoritmo, agrupadas por tipo de algoritmo y tamaño poblacional. En cada una de las figuras, se observa en la parte superior de la misma los algoritmos monocriterio, en su parte izquierda con cruce de un punto y en su parte derecha con cruce uniforme. En la parte inferior de las mismas se observan los algoritmos DENSEA (parte izquierda) y DENSEA-NSH (parte derecha).

Caso X) Comparativa de Algoritmo mono-objetivo frente a multi-objetivo:

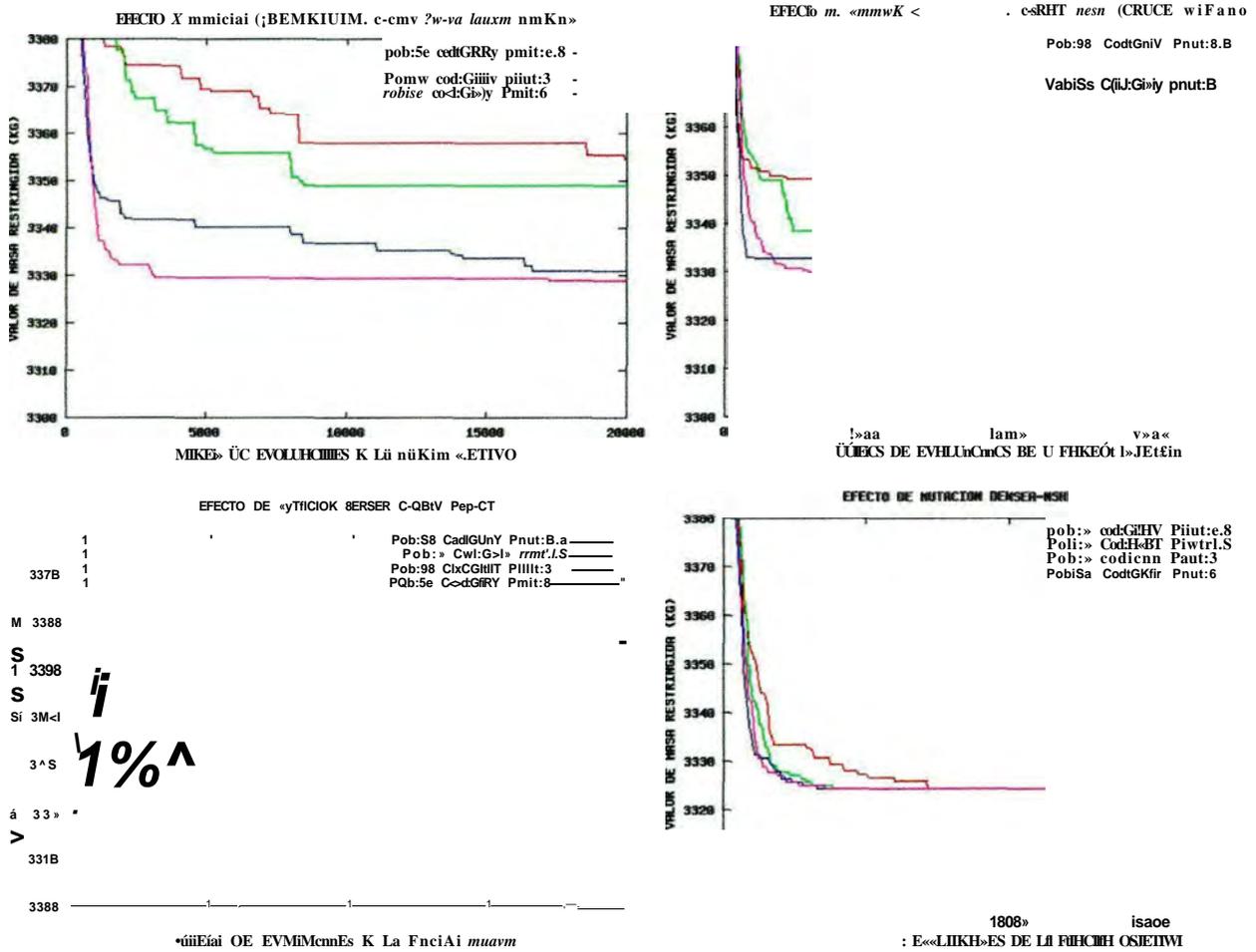


Figura 4.199. Convergencia Masa Restringida en Caso X. Comparativa algoritmo genético monocriterio generacional con cruce de un punto frente a cruce uniforme y algoritmos multicriterio DENSEA y DENSEA-NSH. Tamaño población 50

Caso X) Comparativa de Algoritmo mono-objetivo frente a multi-objetivo:

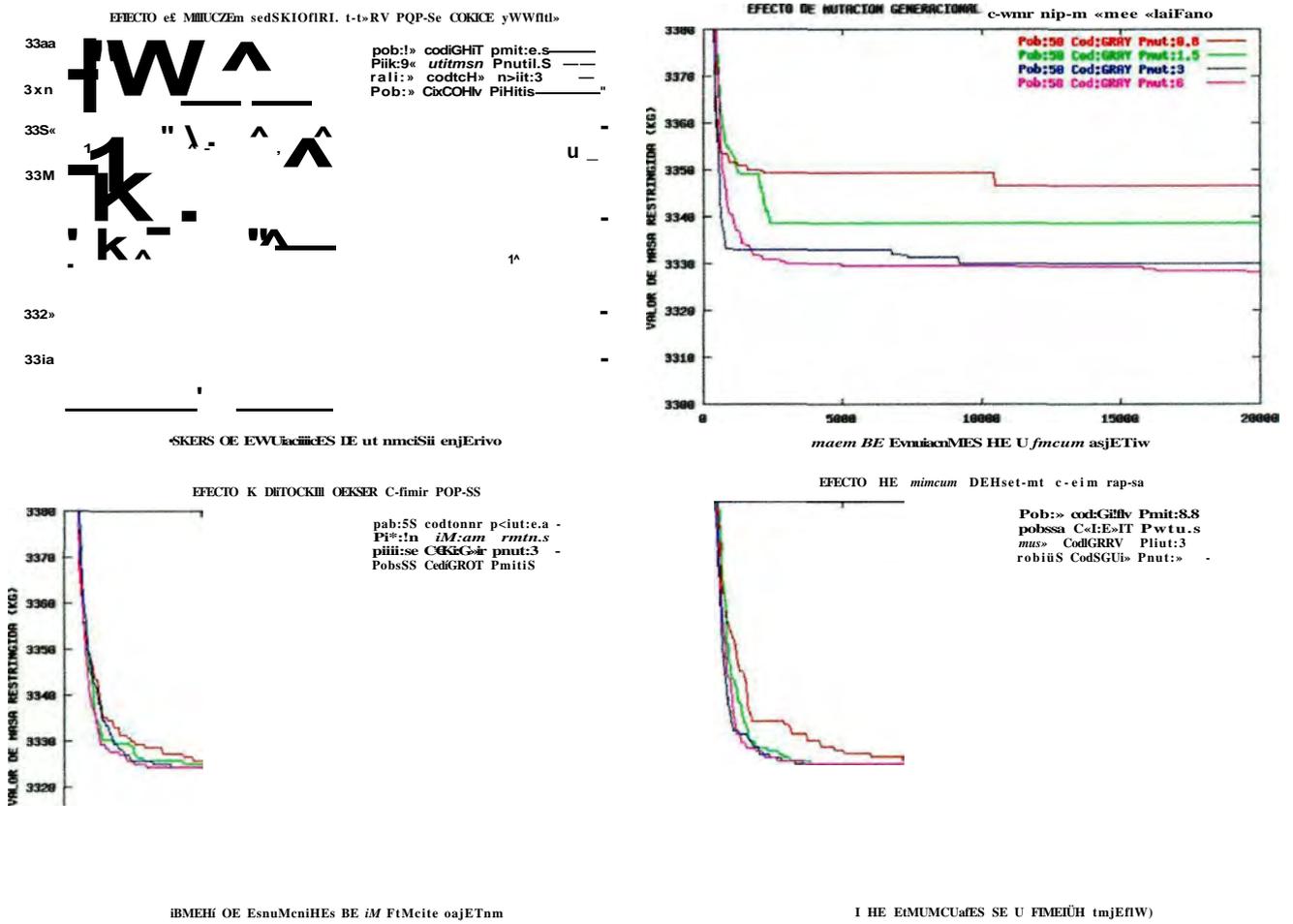


Figura 4.199. Convergencia Masa Restringida en Caso X. Comparativa algoritmo genético monocriterio generacional con cruce de un punto frente a cruce uniforme y algoritmos multicriterio DENSEA y DENSEA-NSH. Tamaño población 50

Capítulo 4. Contribuciones

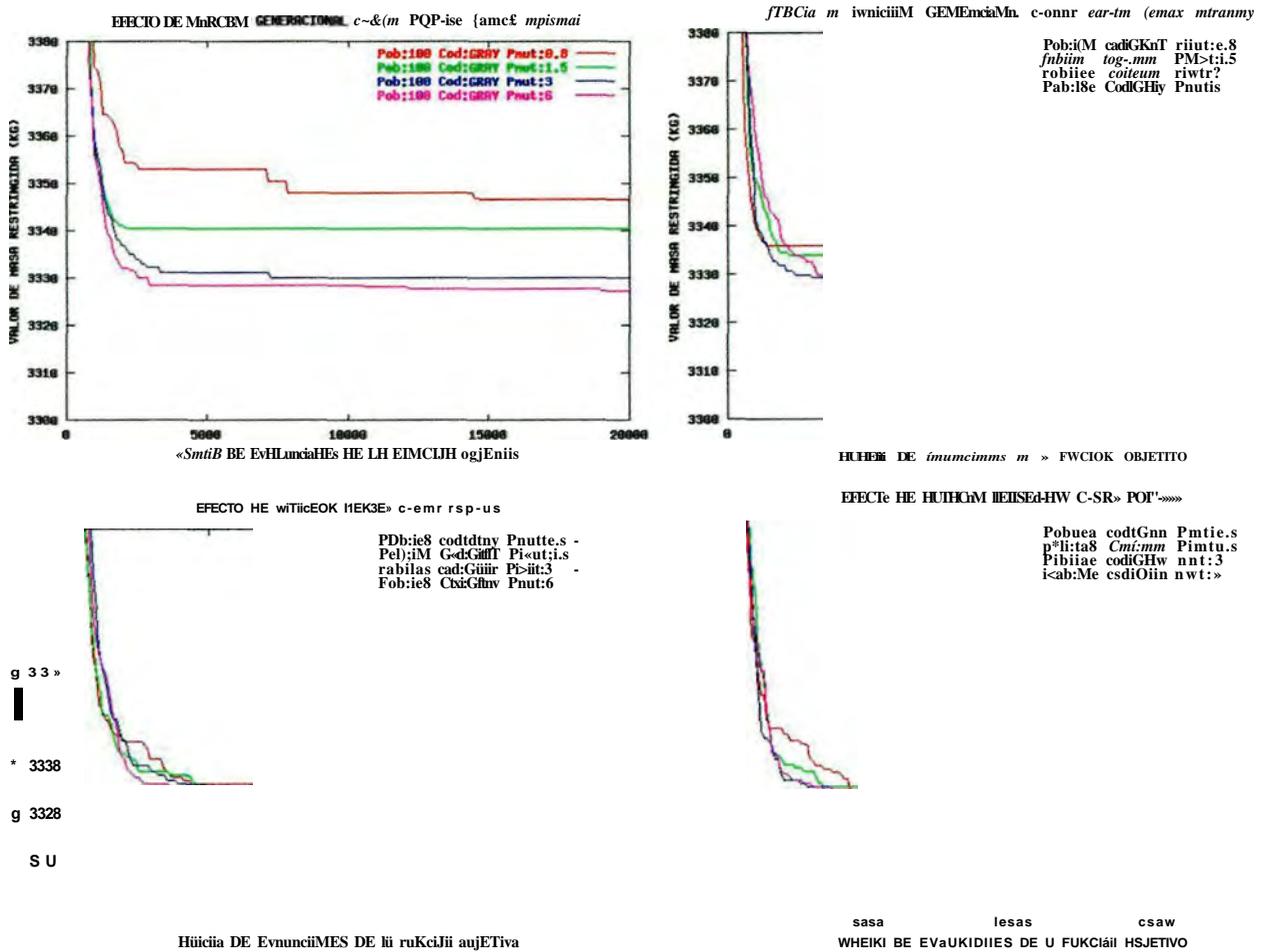


Figura 4.200. Convergencia Masa Restringida en Caso X. Comparativa algoritmo genético monocriterio generacional con cruce de un pinto frente a cruce uniforme y algoritmos multicriterio DENSEA y DENSEA-NSH. Tamaño población 100.

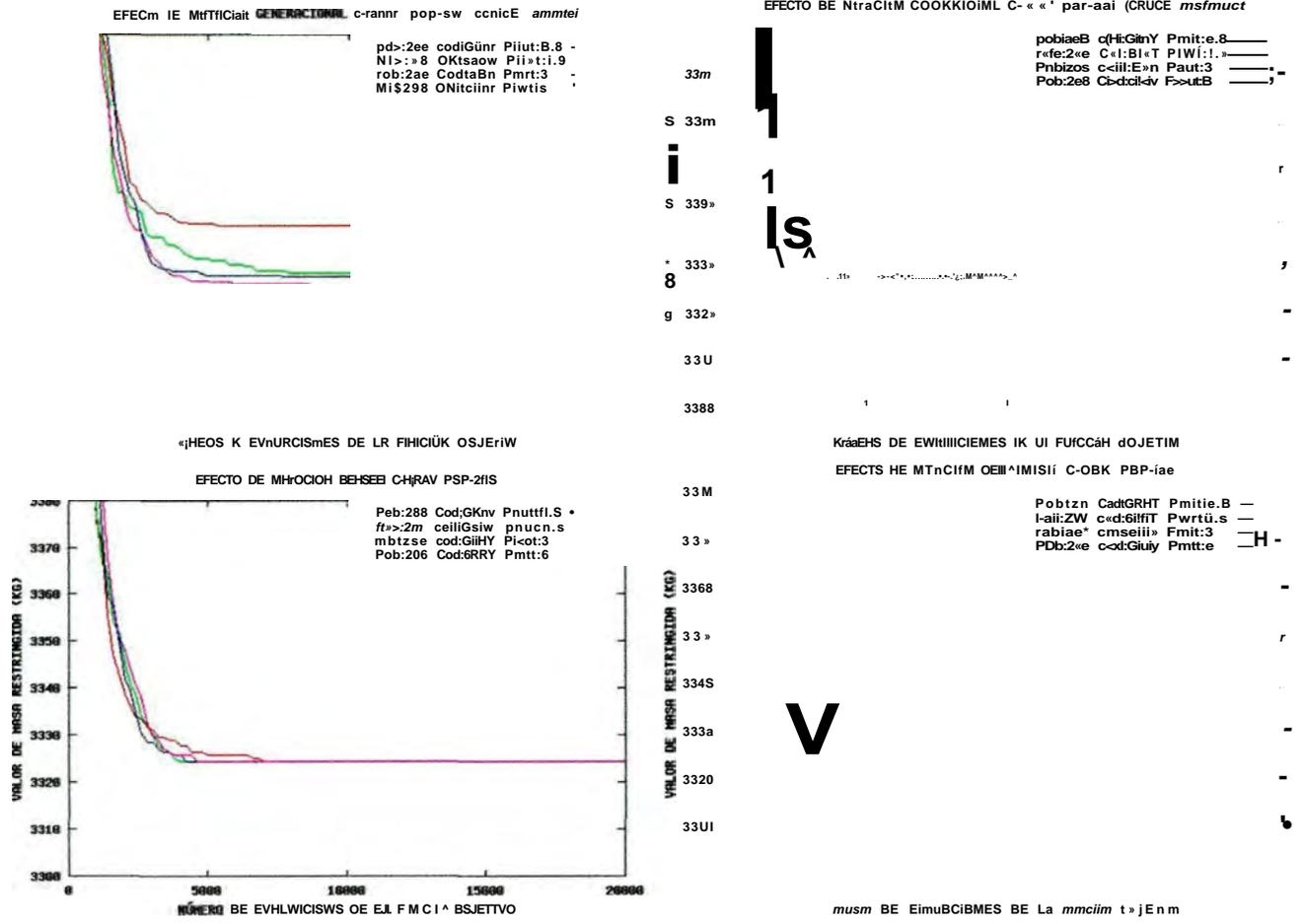


Figura 4.201. Convergencia Masa Restringida en Caso X. Comparativa algoritmo genético monocriterio generacional con cruce de un punto frente a cruce uniforme y algoritmos multicriterio DENSEA y DENSEA-NSH. Tamaño población 200.

Se muestran a continuación las figuras (4.202 a 4.204) donde se puede observar el valor promedio de la masa restringida a lo largo de toda la evolución del algoritmo, agrupadas por tipo de algoritmo y tamaño poblacional. En cada una de las figuras, se observa en la parte superior de la misma los algoritmos monocriterio, en su parte izquierda con cruce de im punto y en su parte derecha con cruce uniforme. En la parte inferior de las mismas se observan los algoritmos DENSEA (parte izquierda) y DENSEA-NSH (parte derecha).

D. CY)Comparativa de Mono-objetivo fi-ente a Multi-objetivo:

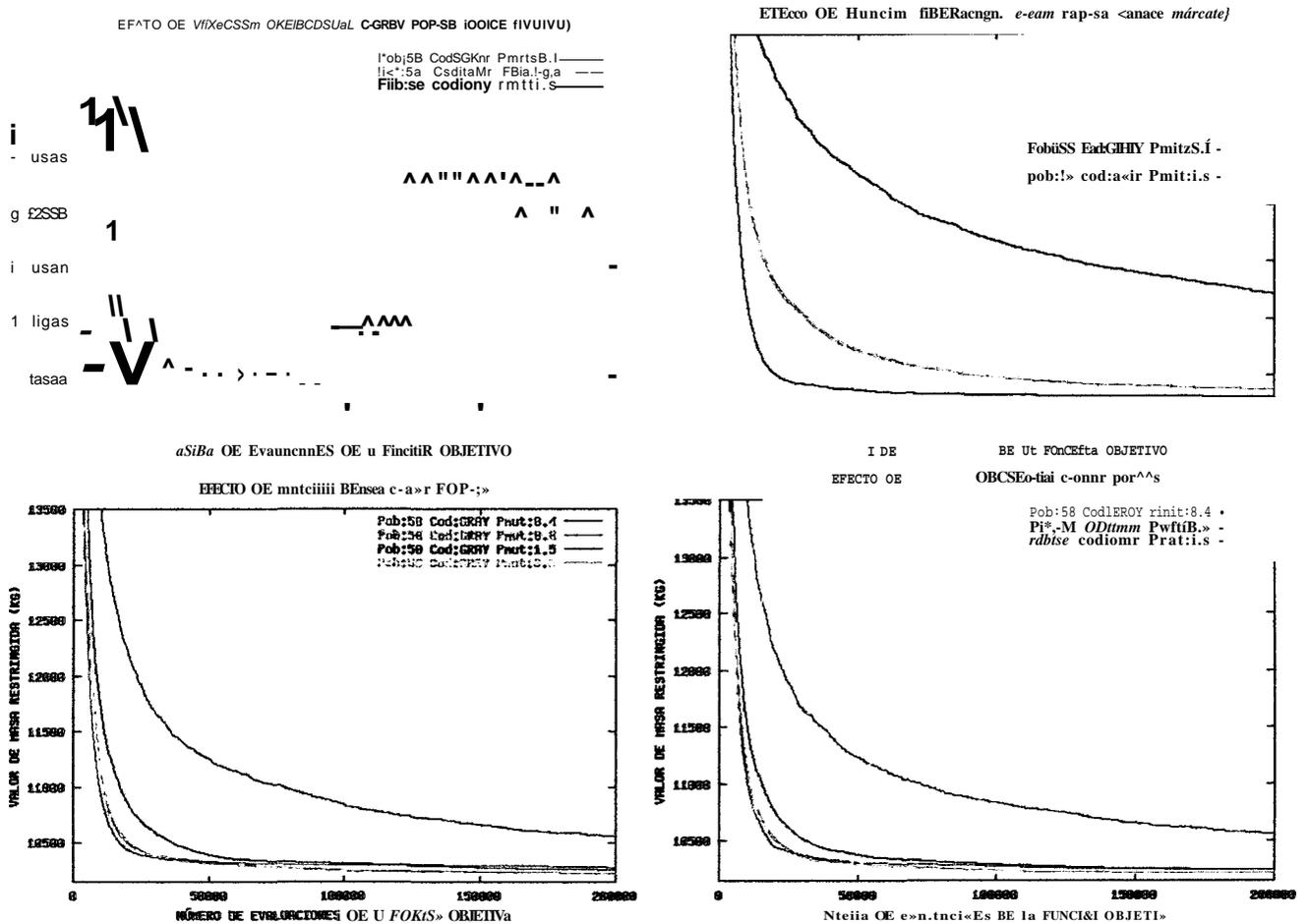


Figura 4.202. Convergencia Masa Restringida en Caso Y. Comparativa algoritmo genético monocriterio generacional con cruce de im punto frente a cruce uniforme y algoritmos multicriterio DENSEA y DENSEA-NSH. Tamaño población 50.

Se muestran a continuación las figuras (4.202 a 4.204) donde se puede observar el valor promedio de la masa restringida a lo largo de toda la evolución del algoritmo, agrupadas por tipo de algoritmo y tamaño poblacional. En cada una de las figuras, se observa en la parte superior de la misma los algoritmos monocriterio, en su parte izquierda con cruce de un punto y en su parte derecha con cruce uniforme. En la parte inferior de las mismas se observan los algoritmos DENSE A (parte izquierda) y DENSEA-NSH (parte derecha).

D. CY)Comparativa de Mono-objetivo frente a Multi-objetivo:

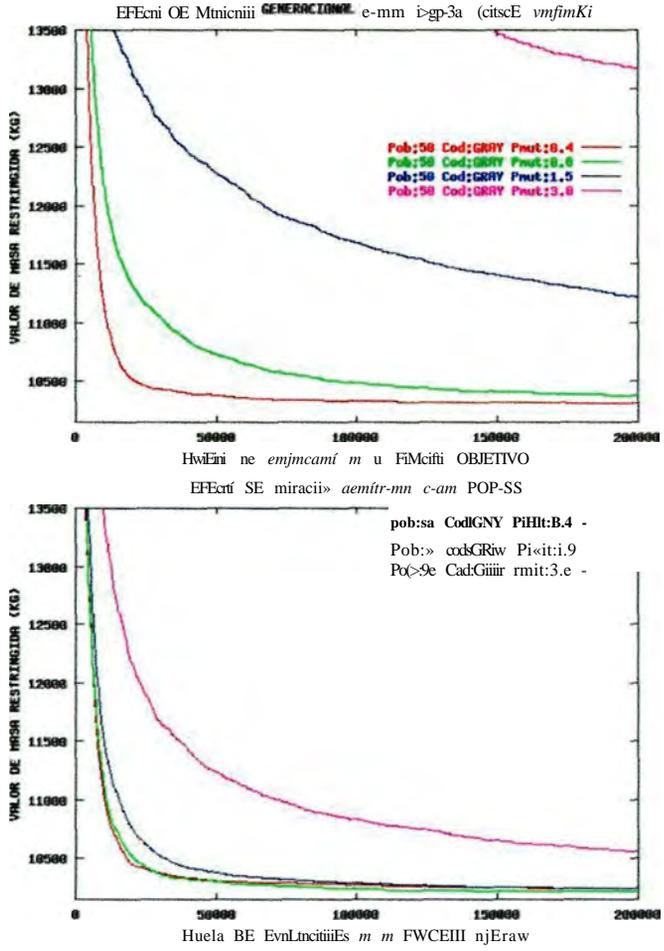
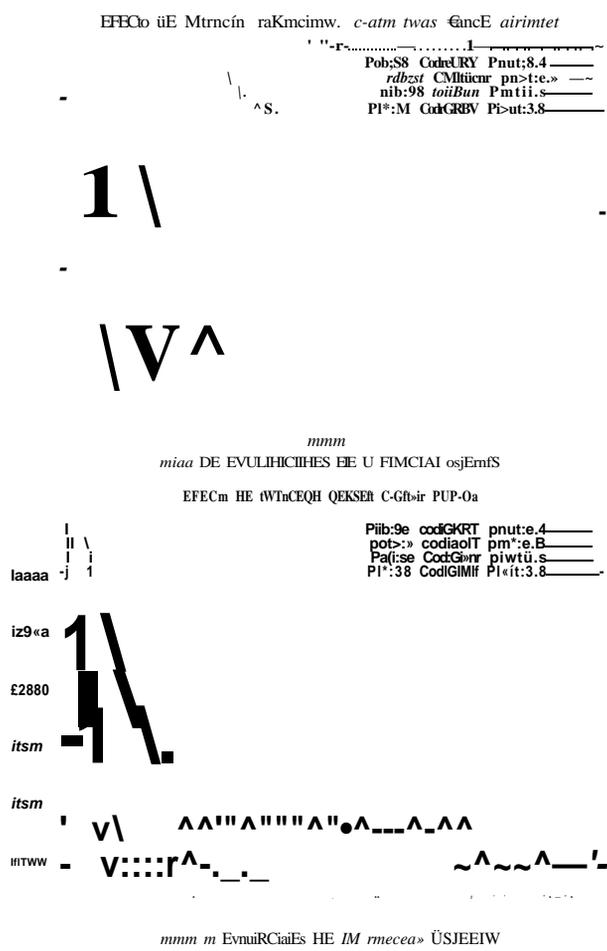


Figura 4.202. Convergencia Masa Restringida en Caso Y. Comparativa algoritmo genético monocriterio generacional con cruce de un punto frente a cruce uniforme y algoritmos multicriterio DENSEA y DENSEA-NSH. Tamaño población 50.

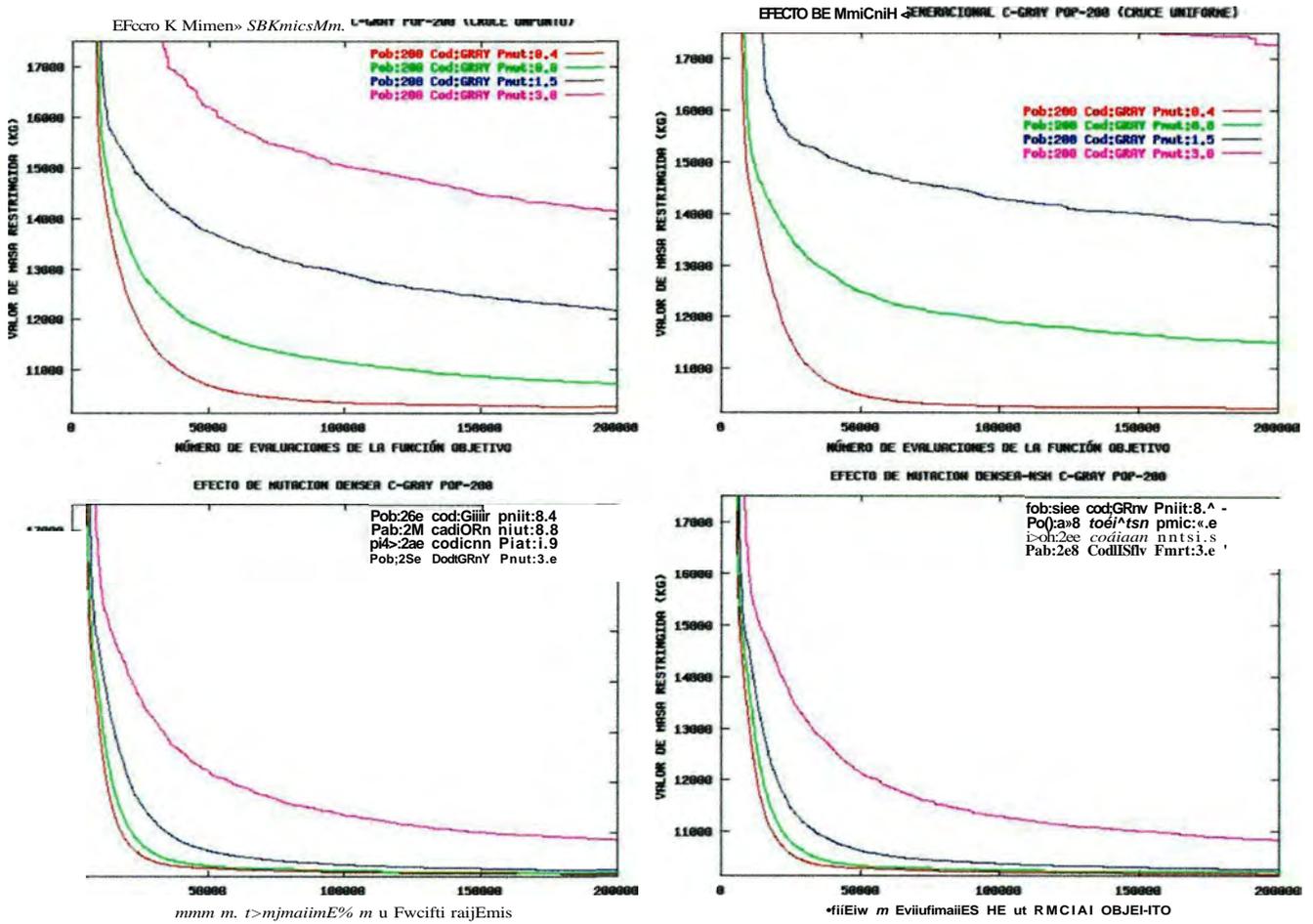


Figura 4.204. Convergencia Masa Restringida en Caso Y. Comparativa algoritmo genético monocriterio generacional con cruce de un punto frente a cruce xmiforme y algoritmos multicriterio DENSEA y DENSEA-NSH. Tamaño población 200.

4.8.2 Análisis de Resultados

Observando los resultados finales de la convergencia en las tablas de los casos test X e Y (4.31 a 4.33), podemos concluir que en ambos casos, tanto los valores de los promedios como los valores de la mejor solución obtenida, superan claramente a los conseguidos por el algoritmo monocriterio. Así, en el caso test X, se observa cómo ante todos los distintos tamaños poblacionales y todas las probabilidades de mutación ensayadas y para cada una de las treinta ejecuciones independientes, los algoritmos DENSEA y DENSEA-NSH son capaces de localizar la solución óptima en un número de evaluaciones mucho menor al ensayado (20.000). En el caso test Y, las diferencias tanto en promedio como en mejores soluciones obtenidas son también notables frente al caso monocriterio. En las figuras (4.199 a 4.201 para el caso X, 4.202 a 4.204 para el caso Y) se puede constatar cómo no solamente son mejores los valores finales, sino que esa ventaja se mantiene a lo largo de toda la convergencia. Se destaca que si bien las gráficas se han realizado en función del número de evaluaciones fimeionales, si se efectúa la comparación en función del tiempo de cálculo, los algoritmos multicriterio y el monocriterio de estado estacionario presentan tiempos análogos, puesto que la sobrecarga de cálculo de la selección multicriterio se ve compensada con el mayor número de generaciones que es preciso evaluar en una

estrategia de estado estacionario donde cada generación se evalúan únicamente dos individuos. Así, una evaluación del algoritmo monocriterio de estado estacionario para 200 individuos y 200.000 evaluaciones estructurales requiere en un ordenador Xeon biprocesador a 1.7 Ghz de 156 segundos, mientras que los algoritmos DENSEA y DENSEA-NSH también están en tomo a esos 150 segundos (ver tabla 4.43 en la sección 4.9 con tiempos detallados de estos y otros algoritmos multicriterio evolutivos). Por tanto las conclusiones obtenidas para el número de evaluaciones funcionales son extrapolables en términos del tiempo de ejecución igualmente.

Se destaca asimismo otra ventaja de los algoritmos DENSEA y DENSEA-NSH, que es su mayor robustez ante los cambios en la probabilidad de mutación frente al algoritmo monocriterio. Se puede apreciar cómo el incremento de la probabilidad de mutación afecta en mucho menor grado en los algoritmos DENSEA y DENSEA-NSH, donde la variabilidad de los resultados es menos acusada. Las figuras 4.199 a 4.204 están agrupadas por tamaños poblacionales y sus leyendas verticales están escaladas equitativamente para favorecer la comparación visual clara. Si comparamos cada una de las figuras, se observa que la variación de las probabilidades de mutación óptimas según se alejan de los valores óptimos, origina peores resultados en la optimización monocriterio, mientras que su influencia es mucho menor en la optimización multicriterio, siendo el algoritmo en este último caso mucho más robusto. Este efecto se aprecia para los dos casos test considerados, X e Y.

Podemos por tanto concluir tras analizar los resultados obtenidos que la introducción del número de distintos tipos de perfiles como 'criterio ayudante' en el problema de la minimización restringida de la masa en estructuras de barras mediante los algoritmos DENSEA y DENSEA-NSH, permite obtener soluciones más precisas que la optimización monocriterio pura. También es un método más robusto ante el parámetro de probabilidad de mutación en la optimización evolutiva.

4.9 *Análisis del Tratamiento Poblacional: Generacional frente a Estado-Estacionario en optimización multiobjetivo estructural*

Un algoritmo genético de estado estacionario genera pocos individuos (usualmente uno o dos) en cada generación, siendo insertados en la población progenitora sustituyendo al mismo número de individuos en ella. Normalmente esta sustitución se realiza en lugar de individuos con función objetivo de mala calidad, de tal manera que el elitismo está implícito en este tipo de estrategia reproductiva.

Los algoritmos genéticos de estado estacionario (steady-state) fueron originados simultánea e independientemente en 1988-89 por Whitley [4.46] [4.47] y Syswerda [4.42]. En este último [4.42], se establece una comparativa de tres tipos de cruce: un punto, dos puntos y uniforme, introduciendo marginalmente a los algoritmos de estado estacionario, aunque presentándolos con ciertas ventajas frente a los generacionales. Se generan dos individuos filiales por generación. Whitley introduce el algoritmo genético denominado GENITOR, que combina dos diferencias fundamentales con los algoritmos genéticos clásicos: la selección se realiza en función de la ordenación de los individuos según su función objetivo y no en función directamente de la misma, y se introduce el concepto de reproducción individual sustituyendo a la generacional. En cada generación un único individuo filial es generado e introducido en la población (el término estado estacionario de Syswerda sería el que tomaría popularidad aunque el concepto introducido es el mismo, en este caso un individuo, en el anterior, dos). En una publicación posterior de Syswerda [4.41] se comparan ambos sistemas reproductivos: el generacional y el de estado estacionario, que resultan ser equivalentes cuando la sustitución del nuevo individuo de la población es aleatoria en el estado estacionario, pero muy ventajosa para el estado estacionario cuando la sustitución se hace eliminando el/los individuo/s con peor función objetivo (la presión de selección aumenta). Se destaca también que el estado estacionario tiene implícito elitismo, que ayuda a preservar la información genética, y que permite el aprovechamiento de la información aportada por los nuevos individuos en el mismo momento de ser generados sin necesidad de esperar a que sea evaluada toda una nueva generación para la sustitución. En problemas dinámicos, donde el valor de la función objetivo es variable para *un* mismo individuo con el trascurso del tiempo, este tipo de algoritmo resulta ser de menor robustez que el algoritmo genético generacional, característica no perteneciente al problema estructural que nos concierne.

En variedad de problemas ha sido declarada la ventaja de los algoritmos genéticos de estado estacionario frente a los generacionales [4.42]. En el caso del problema estructural monocriterio de minimización restringida de la masa habiendo sido comparadas tres estrategias de algoritmos genéticos: generacional, CHC [4.14][4.15] y steady-state, resultó como más ventajosa la estrategia de estado estacionario, al proporcionar soluciones de más calidad en menor número de evaluaciones [4.24] [4.25]. Por ello ha sido implementada una versión de estado estacionario para cada uno de los algoritmos evolutivos multiobjetivo considerados, cuya comparación con las versiones generacionales se mostrará en la presente sección.

En optimización evolutiva multicriterio no es habitual el uso de la estrategia de estado estacionario. Uno de los primeros ejemplos de aplicación de la misma, lo podemos encontrar en Meneghetti et al. [4.34], donde se resuelve el diseño óptimo multicriterio de procesos de fundición, considerando los criterios de dureza Brinell (maximizar), masa (minimizar) y porosidad (minimizar). Igualmente otro algoritmo multicriterio evolutivo con estrategia de estado estacionario se puede encontrar en Weicker

et al. [4.44], donde se presenta el algoritmo stEAPT (steady-state evolutionary algorithm with Pareto tournaments). Se presenta como ventaja del reemplazamiento de estado estacionario que los nuevos individuos son introducidos inmediatamente y favorecen una más rápida velocidad de evolución. Se resuelve el diseño óptimo de la localización de estaciones base de transmisión con asignación de frecuencia, considerando la minimización del coste y la minimización de la interferencia. Se comparan los resultados con los algoritmos SPEA2 y NSGAI, obteniéndose ligeramente mejores resultados con estos dos últimos algoritmos. También el algoritmo 8-MOEA, propuesto en [2.20] es un algoritmo de estado estacionario que busca un desempeño veloz sin pérdida de eficacia. Se compara frente al NSGAI y un NSGAI con el operador de clustering del SPEA, en diversos casos test estándar multiobjetivo, con resultados satisfactorios, si bien la influencia de su tratamiento poblacional sobre el resultado conjunto no es señalada.

Mediante la estrategia de estado estacionario, la eliminación de duplicados puede realizarse de forma estricta a un coste bajo. En el caso de los algoritmos DENSEA y DENSEA-NSH, las versiones de estado estacionario tienen implementada la eliminación estricta de duplicados, entendidos éstos en el sentido genotípico: aunque dos soluciones coincidan en el espacio funcional (mismos valores de la función objetivo), sólo se consideran duplicadas si sus genes son idénticos (es decir, si los tipos de perfiles en cada una de las barras son los mismos). Igualmente se hace notar que siendo las comparaciones en términos de número de evaluaciones de la función objetivo, en el caso de estado estacionario, el número de generaciones será mayor al computarse un menor número de evaluaciones por generación (dos frente al tamaño de la población: 50, 100 ó 200), con lo cual se produce un sobre coste debido a la selección multicriterio por cada generación. Este sobre coste será más significativo cuanto menor sea el tiempo de cálculo necesario de cada evaluación de la función objetivo y por tanto, cuanto menor sea su proporción frente al tiempo de cálculo de la selección multicriterio de soluciones no dominadas. El incremento de coste será por tanto mayor proporcionalmente en el caso de estructuras de pequeño tamaño que frente a estructuras de tamaño mayor.

En el caso test X, las tablas 4.41 (I y II) resumen los resultados comparativos entre las estrategias generacional y estado estacionario en términos de número de veces que se alcanza el frente óptimo (N), número de evaluaciones promedio de esos frentes (Me), y desviación típica (Dt). Se señala en negrita el valor de N mejor para cada uno de los casos considerados. Para los mejores valores de mutación (3% y 6%) en este caso X, en las poblaciones de 50 y 100 individuos se obtienen mejores resultados con la estrategia de estado estacionario, mientras que en la población de 200 individuos es ligeramente mejor la estrategia generacional, si bien las diferencias entre ellas son pequeñas.

Capítulo 4. Contribuciones

Tabla 4.41.(1). Comparativa Generacional / Estado Estacionario en Caso X.

Caso test X			famPob 50			TamPob 100			TamPob 200			Mf	Total	
Probab.	Mutación (%)		N	Me	Dt	N	Me	Dt	N	Me	Dt	N	Me	Dt
JPEA2	0.8%	Gen	18	3888	4889	18	4066	4545	24	2158	608	60	3249	3347
		stst	13	8687	6105	13	4525	5371	19	3140	4360	45	5142	5278
	1.5%	Gen	17	5544	5969	23	3752	4709	24	2425	813	64	3730	3830
		Stst	15	4657	4474	17	4737	5478	21	3963	5297	53	4407	5083
	3.0%	Gen	23	5513	5981	25	4508	5469	27	2533	975	75	4105	4141
		Stst	26	5202	5981	26	4180	5100	27	3731	4188	79	4362	5089
6.0%	Gen	25	2966	2202	28	2157	1603	29	3317	1597	82	2813	1800	
	stst	28	4561	4805	26	4192	4703	30	2298	1733	84	3638	3747	
ÍSGAII	0.8%	Gen	15	4736	4811	18	3867	4409	23	2869	2250	56	3689	3823
		Stst	5	6652	7174	17	4222	4891	24	3401	4095	46	4057	5386
	1.5%	Gen	21	4526	5593	26	3542	4296	23	2400	907	70	3461	3598
		Stst	21	8231	6809	21	5908	5879	22	2887	3531	64	5631	5406
	3.0%	Gen	27	3459	4192	25	3512	3935	26	2453	872	78	3140	2999
		Stst	25	3775	3328	26	2777	4707	25	3661	3847	76	3396	3960
6.0%	Gen	29	2965	2937	30	2380	1657	30	3186	2376	89	2842	2323	
	Stst	30	2538	3147	29	3319	3573	28	2924	2620	87	2922	3113	
ÍSGAIIelco.00	0.8%	Gen	18	5341	5775	17	3770	5578	23	4460	4749	58	4531	5367
		Stst	17	4465	5707	14	5385	6772	21	4365	4552	52	4672	5677
	1.5%	Gen	19	4150	5767	24	3037	3155	26	3830	3044	69	3642	3988
		Stst	21	5720	5563	24	4786	5494	28	3571	3661	73	4588	4906
	3.0%	Gen	26	6401	5323	26	3461	3924	29	3206	1000	81	4313	3415
		Stst	27	3788	3980	28	3736	4631	30	3017	2407	85	3498	3672
6.0%	Gen	27	2264	2065	25	2464	933	30	3626	960	82	2823	1319	
	Stst	30	3262	3038	30	2839	2581	29	4025	3446	89	3368	3021	
^SGAIIelco.04	0.8%	Gen	29	5213	4186	26	5519	4773	28	4900	3950	83	5203	4303
		Stst	27	7322	5875	30	4195	4299	24	3747	2639	81	5104	4271
	1.5%	Gen	30	2725	2305	30	4623	5239	30	4366	3437	90	3904	3660
		Stst	30	4545	2995	30	4139	3560	29	3877	3656	89	4190	3403
	3.0%	Gen	29	2767	2679	30	4130	4726	30	3940	2442	89	3621	3282
		stst	30	3080	2914	30	3768	4008	29	3101	1960	89	3318	2960
6.0%	Gen	30	1861	813	29	4031	3082	30	3646	1532	89	3169	1809	
	stst	30	2386	2305	30	2811	2023	30	3129	1325	90	2775	1884	
)ENSEA	0.8%	Gen	30	2021	1721	30	2756	1844	30	3106	1453	90	2627	1672
		Stst	30	2262	2342	30	2467	1888	30	2353	1252	90	2360	1827
	1.5%	Gen	30	1628	1363	30	2306	1626	30	2906	843	90	2280	1277
		Stst	30	1622	1330	30	2000	1289	30	2383	929	90	2001	1182
	3.0%	Gen	30	1533	998	30	2416	1297	30	2786	824	90	2245	1039
		Stst	30	1569	900	30	2096	1061	30	2502	1298	90	2055	1086
6.0%	Gen	30	1325	620	30	2130	898	30	3360	930	90	2271	816	
	Stst	30	1673	1048	30	2010	865	30	2666	1327	90	2116	1080	

Capítulo 4. Contribuciones

Tabla 4.41 (II). Comparativa Generacional / Estado Estacionario en Caso X.

Caso test X		50			100			200			Total				
Probab. Mutación (%)		N	Me	Dt	N	Me	Dt	N	Me	Dt	N	Me	Dt		
) ENSEA-NSH	0.8%	Gen	30	2470	1986	30	2293	1173	30	2560	895	90	2441	1351	
		stst	30	3144	2648	30	1665	867	30	2853	1704	90	2554	1739	
	1.5%	Gen	30	1576	900	30	1993		30	3200	1138	90	2256	100 ^	
		Stst	30	1841	1737	30	1709	931	30	2732	1404	90	2094	135 ^	
	3.0%	Gen	30	1225	711	30	1720	620	30	2626	631	90	1857	654 j	
		Stst	30	1497	1119	30	1807	1098	30	2691	1144	90	1998	112 fl	
	6.0%	Gen	30	1383	671	30	1840	649	30	2986	977	90	2069	765	
		Stst	30	1301	687	30	1856	667	30	2729	992	90	1962	782	
	total	0.8%	Gen	140	3736	3185	139	3610	2994	158	3338	1986	437	3552	2775
			stst	122	4770	3698	134	3401	3012	148	3236	2649	404	3754	3195
		1.5%	Gen	147	3034	2894	163 ^	3183	2965	163	3233	1580	473	3154	2537
			Stst	147	4102	3195	152	3650	3184	160	3203	2752	459	3639	3064
3.0%		Gen	165	3330	2964	166	3243	3002	172	2940	994	503	3168	2346	
		Stst	168	3073	2786	170	3033	3169	171	3090	2184	509	3065	2725	
6.0%		Gen	171	2095	1411	172	2496	1323	179	3353	1243	522	2658	1341	
		stst	178	2598	2307	175	2804	2199	177	2956	1670	530	2785	2080	

Se muestran en las figuras 4.205 a 4.228 los resultados comparativos a lo largo de toda la evolución de las métricas, tanto de aproximación al frente como de amplitud del mismo. Posteriormente se exponen las figuras donde se comparan los frentes no dominados finales 4.229 a 4.240 (en ambos casos ver el *Anexo*). Estas últimas figuras tienen el formato de candelabro ('candlestick'), donde se expresan los extremos máximos y mínimos del número de evaluaciones computado para la obtención del frente de Pareto, como los puntos extremos de las líneas exteriores a cada columna. Cada una está centrada en la media de la distribución y su altura indica el valor de la desviación típica. El número contenido en cada columna es indicativo del número total de veces sobre 30 que se ha alcanzado el frente de Pareto, compuesto por tres soluciones en este caso test X. Cuanto más abajo esté la columna, indicará menor número medio de evaluaciones necesarias para la obtención del frente óptimo, y cuanto más achatada, menores oscilaciones tendrá la muestra.

A la vista de los resultados, no es posible establecer que alguna de las dos estrategias de reemplazamiento sea mejor que la otra, al menos en este caso test X.

Caso Y) Comparativa Generacional / SteadyState:

Métricas de acercamiento al frente, comparando 3 tamaños poblacs. de cada algoritmo Prob 0.4%:

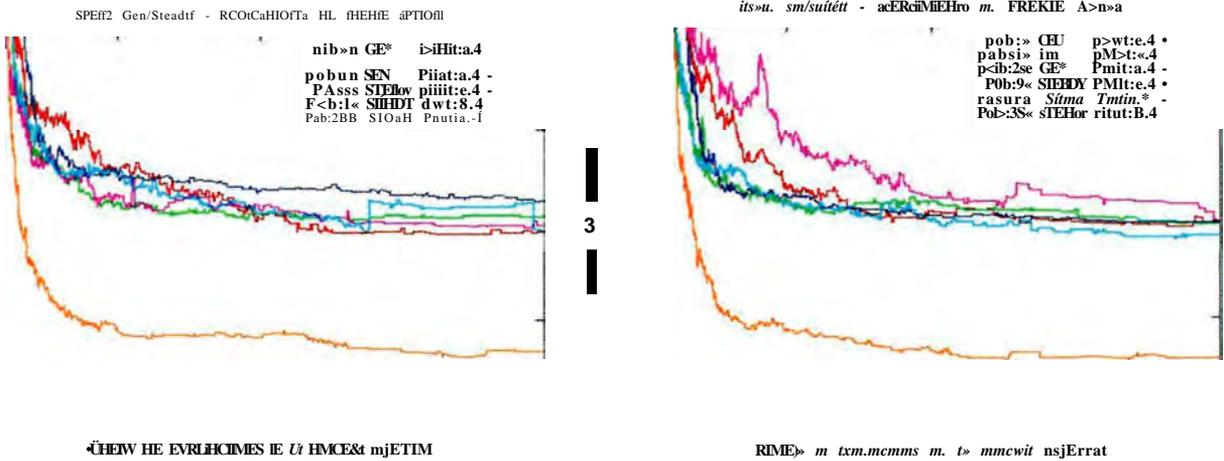


Figura 4.241. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en CasoY. Probabilidad de mutación 0.4%. Algoritmos SPEA2 y NSGAI.

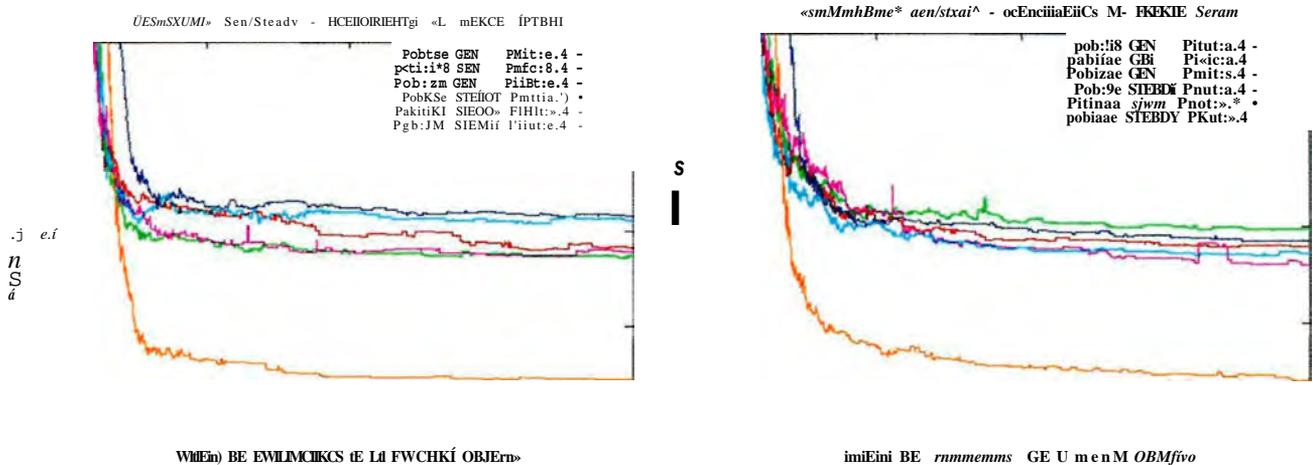


Figura 4.242. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en CasoY. Probabilidad de mutación 0.4%. Algoritmos NSGAI con elitismo controlado con r=0.0 y r=0.4.

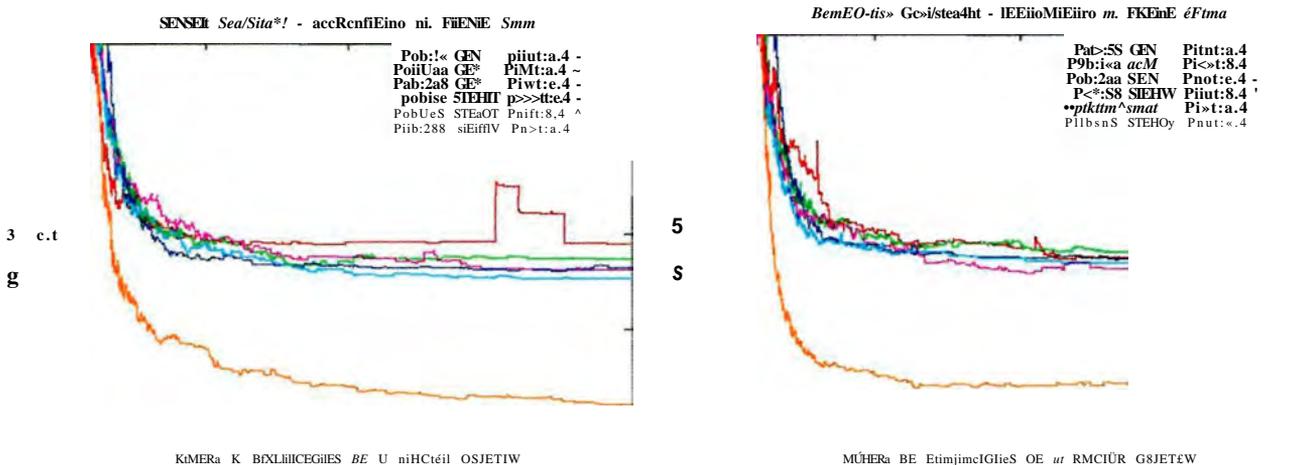


Figura 4.243. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en CasoY. Probabilidad de mutación 0.4%. Algoritmos DENSEA y DENSEA-NSH.

Métricas de acercamiento al frente, comparando 3 tamaños poblacs. de cada algoritmo Prob 0.8%:

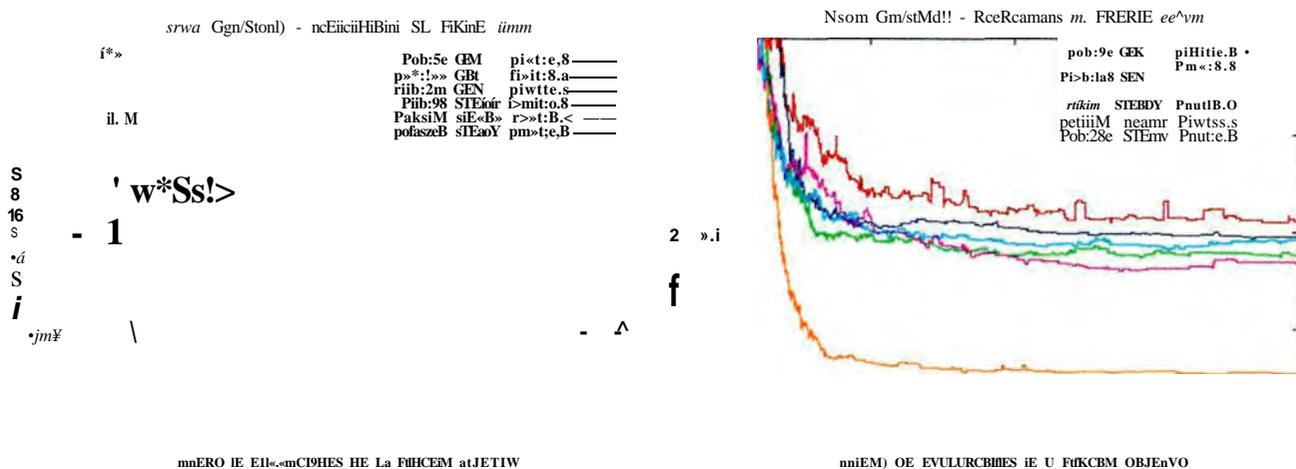


Figura 4.244. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 0.8%. Algoritmos SPEA2 y NSGAII.

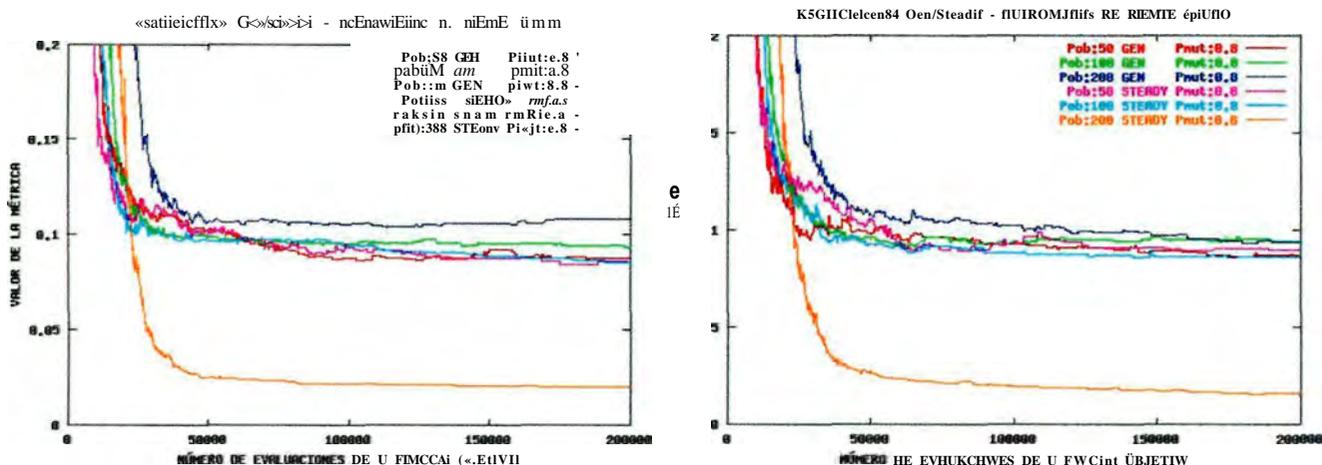


Figura 4.245. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 0.8%. Algoritmos NSGAII con elitismo controlado con $r=0.0$ y $r=0.4$.

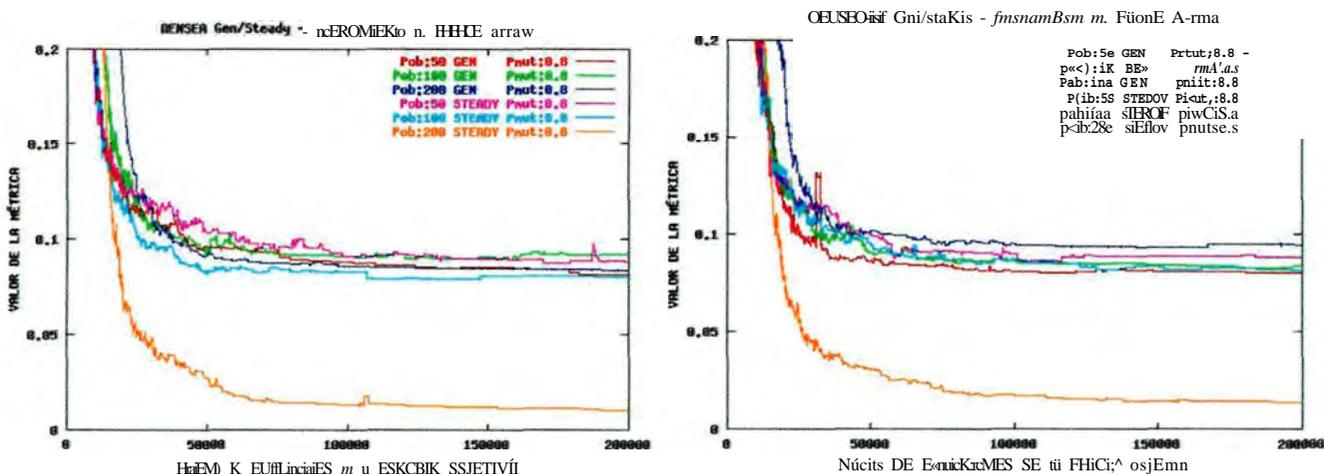


Figura 4.246. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 0.8%. Algoritmos DENSEA y DENSEA-NSH.

Métricas de acercamiento al frente, comparando 3 tamaños poblacs. de cada algoritmo Prob 1.5%:

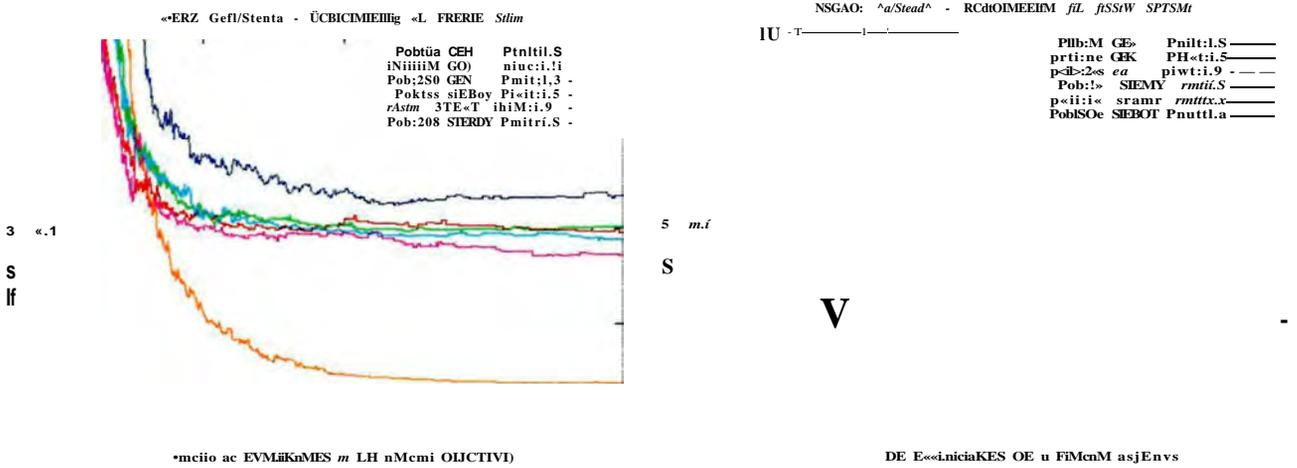


Figura 4.247. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 1.5%. Algoritmos SPEA2 y NSGAI.

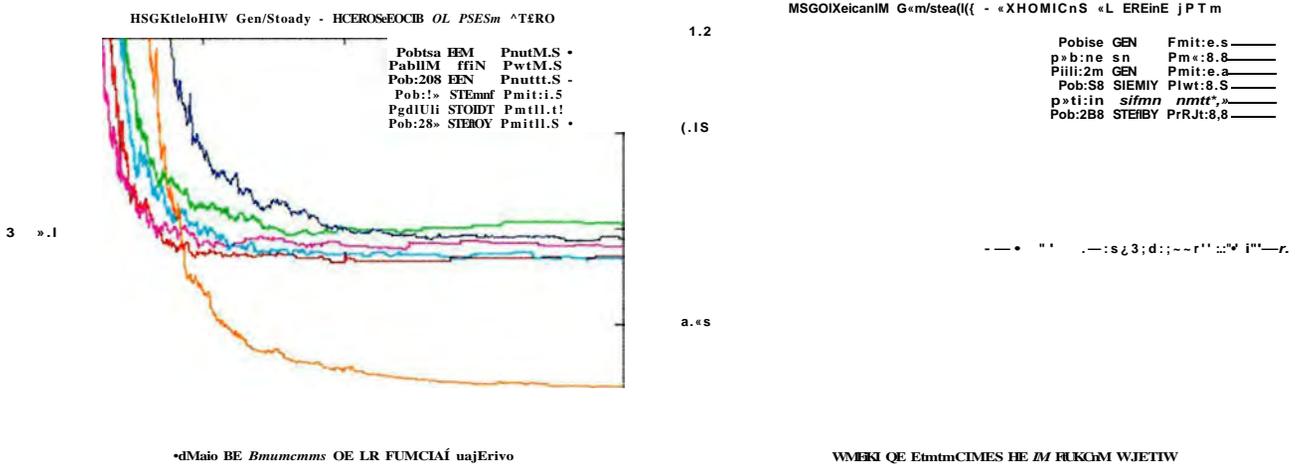


Figura 4.248. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 1.5%. Algoritmos NSGAI con elitismo controlado con $r=0.0$ y $r=0.4$.

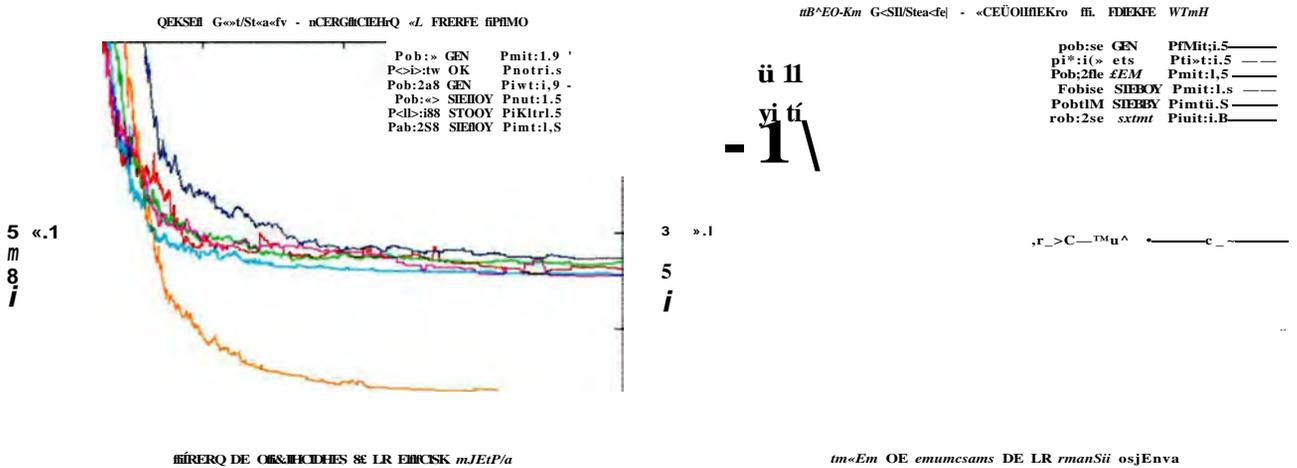


Figura 4.249. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 1.5%. Algoritmos DENSEA y DENSEA-NSH.

Métricas de amplitud del frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 0.4%:

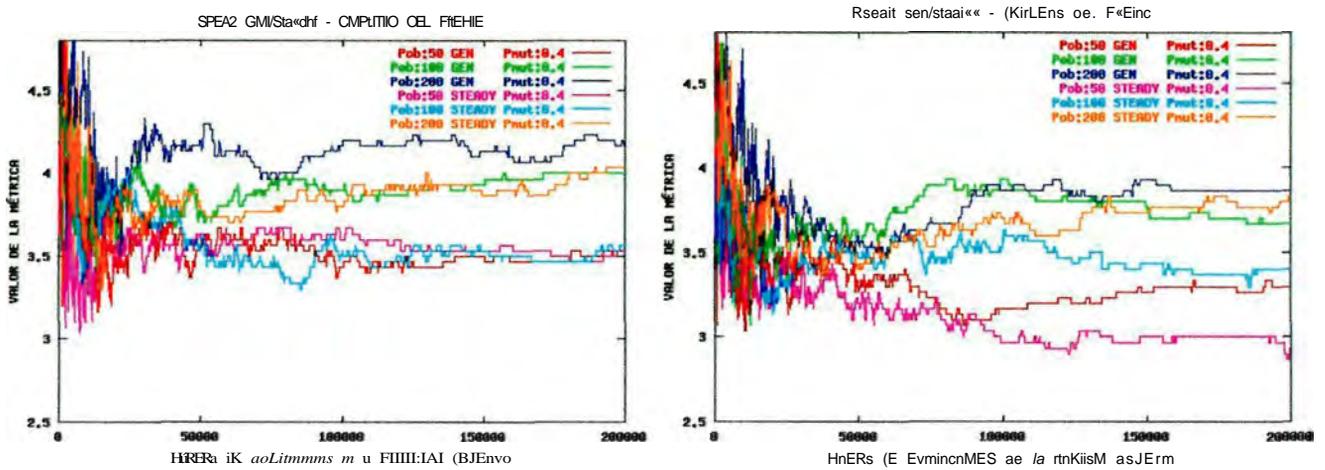


Figura 4.253. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 0.4%. Algoritmos SPEA2 y NSGAI1.

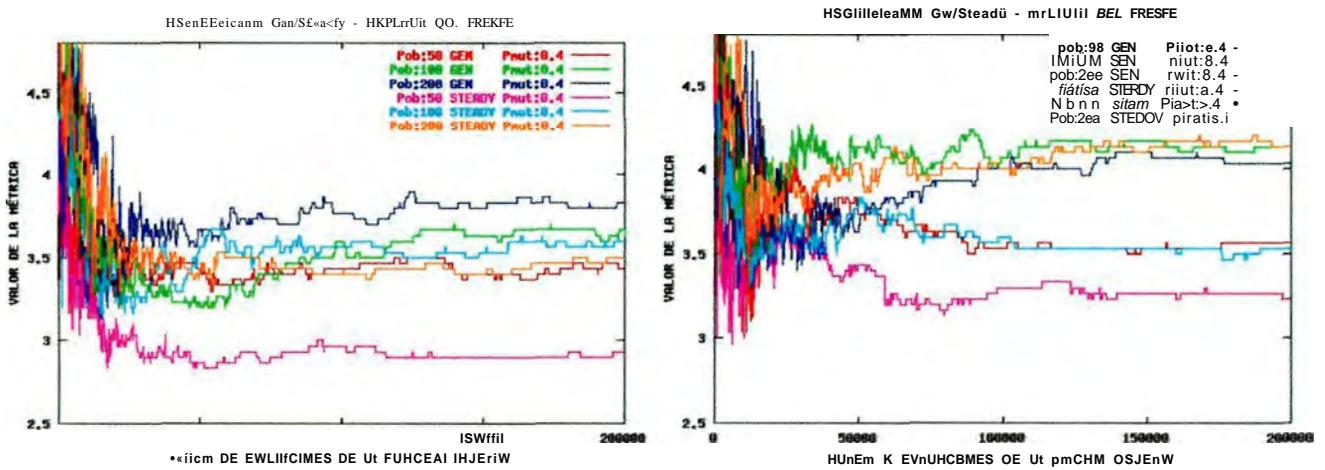


Figura 4.254. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 0.4%. Algoritmos NSGAI1 con elitismo controlado $r=0.0$ y $r=0.4$.

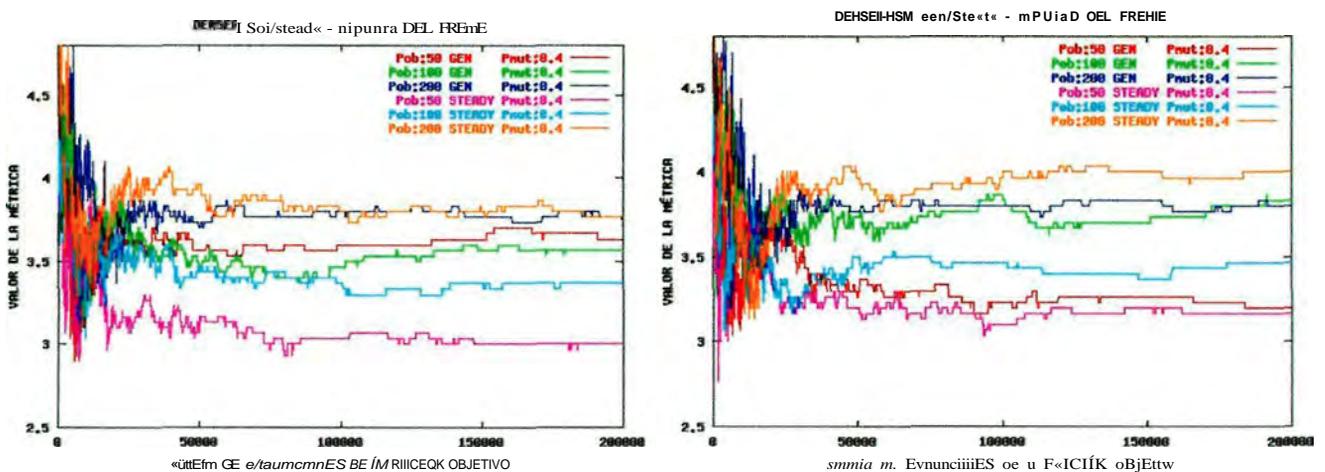


Figura 4.255. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 0.4%. Algoritmos DENSEA y DENSEA-NSH.

Capítulo 4. Contribuciones

Métricas de amplitud del frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 0.8%:

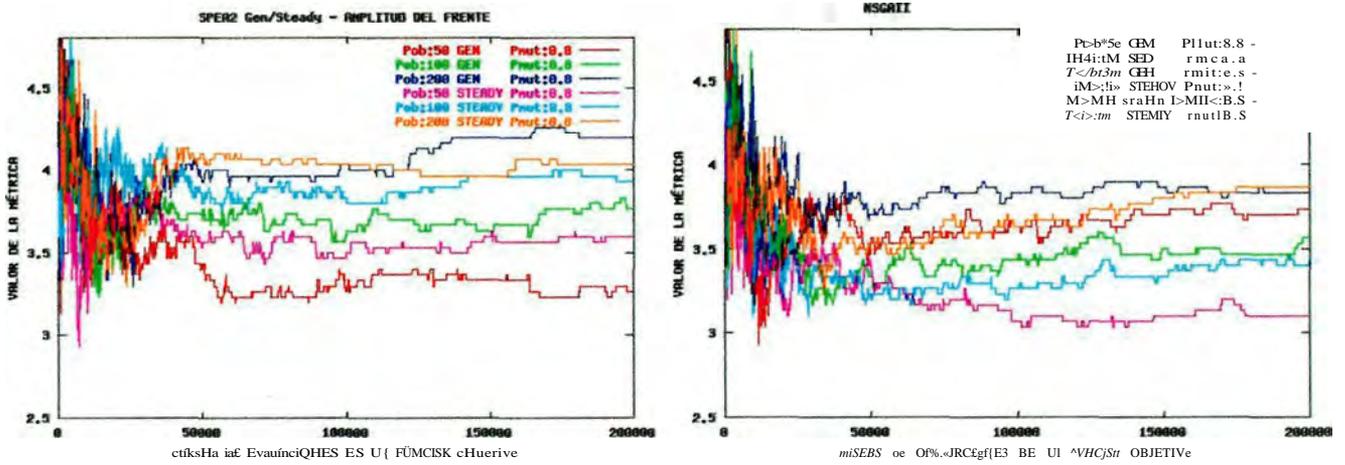


Figura 4.256. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 0.8%. Algoritmos SPEA2 y NSGAI.

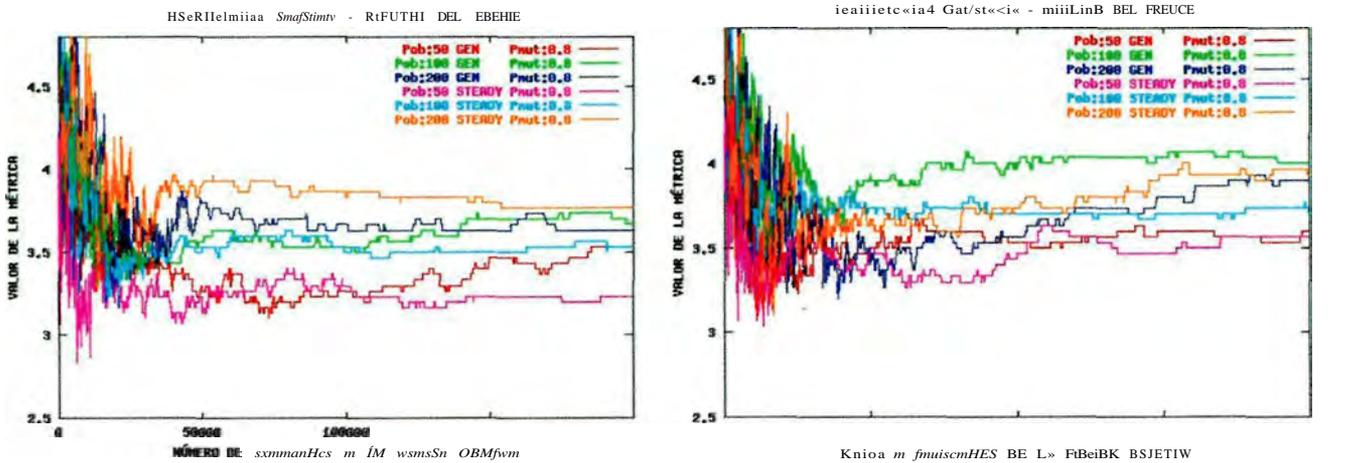


Figura 4.257. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 0.8%. Algoritmos NSGAI con elitismo controlado $r=0.0$ y $r=0.4$.

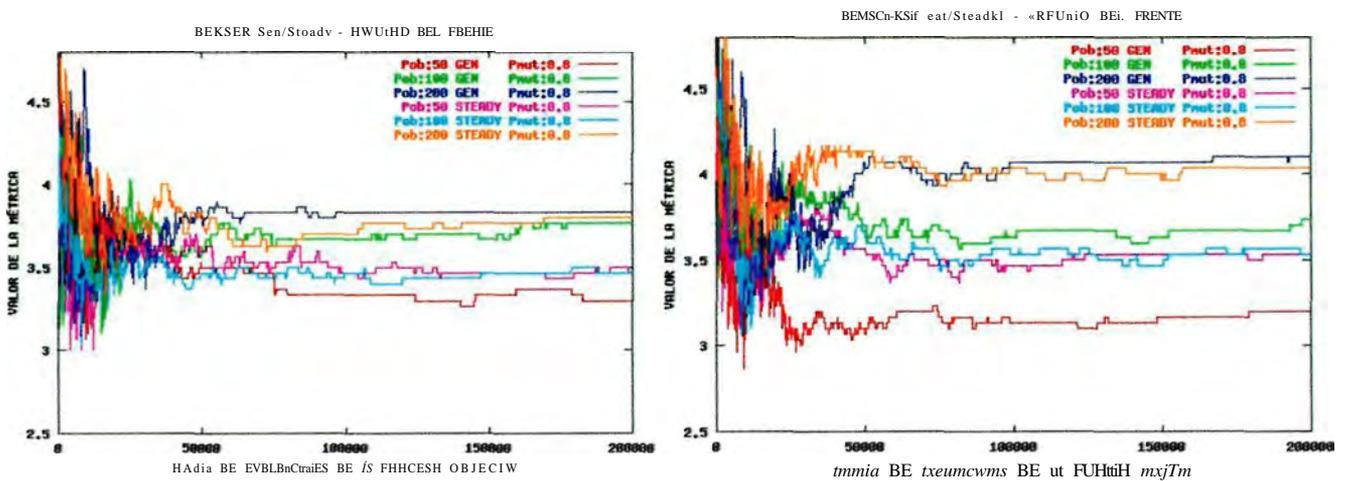


Figura 4.258. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 0.8%. Algoritmos DENSEA y DENSEA-NSH.

Métricas de amplitud del frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 1.5%:

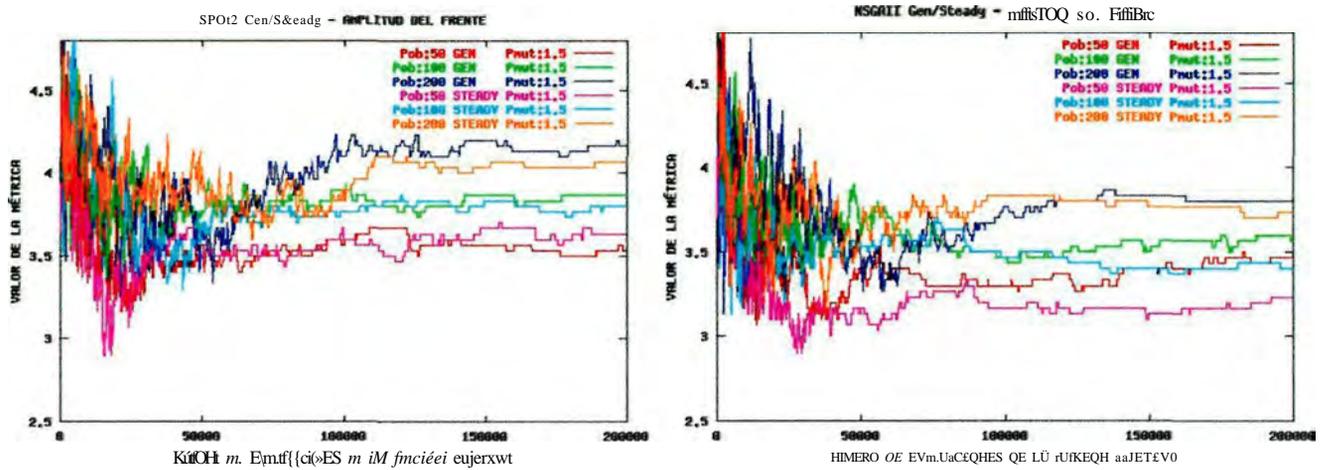


Figura 4.259. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 1.5%. Algoritmos SPEA2 y NSGAII.

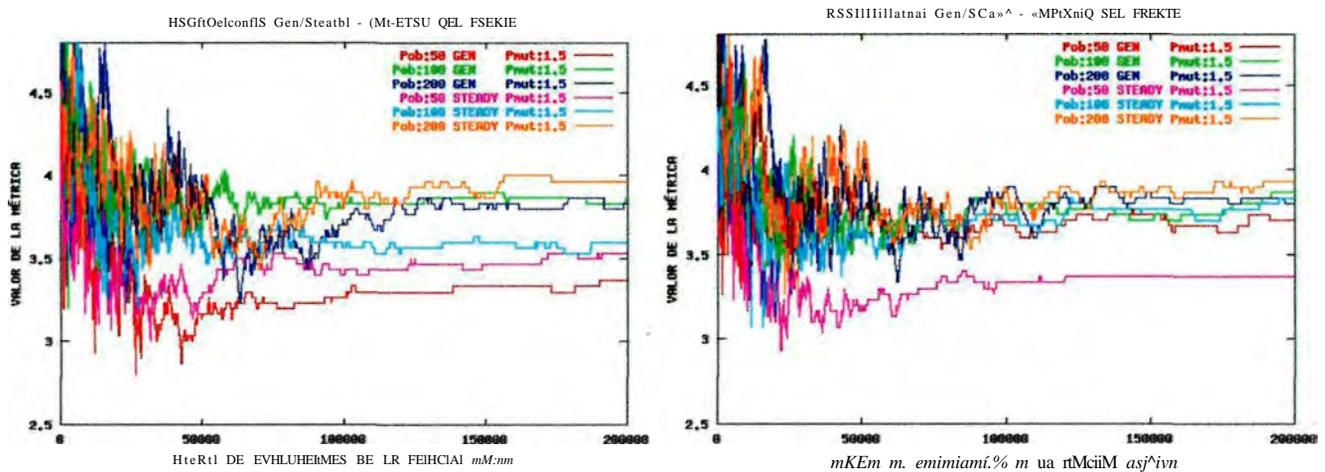


Figura 4.260. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 1.5%. Algoritmos NSGAII con elitismo controlado $r=0.0$ y $r=0.4$.

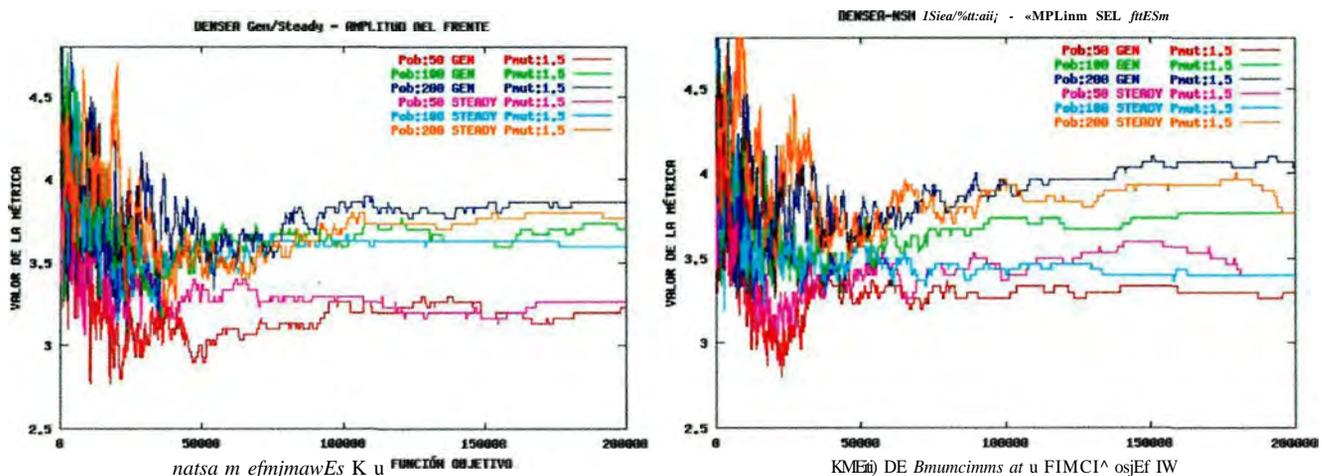


Figura 4.261. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 1.5%. Algoritmos DENSEA y DENSEA-NSH.

En el caso test Y, se puede apreciar a partir de la observación de las figuras 4.241 a 4.264, cómo la estrategia de estado estacionario en el caso de tamaño de población de 200 individuos es claramente superior en todos los algoritmos ensayados y para todas las probabilidades de mutación ensayadas, en términos de la métrica acercamiento al frente óptimo. (De estas figuras se han incluido las correspondientes a la probabilidad del 3% en el *Anexo*). Es claro que con un tamaño poblacional de 200 individuos, la estrategia de estado estacionario proporciona un fi-ente más cercano al óptimo durante toda la evolución del algoritmo a partir de la evaluación funcional 25000 como se observa en las figuras 2.241 a 2.252.

Tabla 4.42. Comparativa de estrategia de reemplazamiento de métrica Amplitud de Frente en Caso Y para población de 200 individuos.

Algoritmo	Probabilidad de Mutación	Mejor Estrategia Reemplazamiento para métrica Amplitud de Frente
SPEA2	0.4%	Generacional
	0.8%	Generacional
	1.5%	Generacional
	3.0%	Estado Estacionario
NSGA-II	0.4%	Generacional
	0.8%	
	1.5%	Generacional
	3.0%	Estado Estacionario
NSGA-II	0.4%	Generacional
Elitismo controlado r=0.0	0.8%	Estado Estacionario
	1.5%	Estado Estacionario
	3.0%	
NSGA-II elitismo controlado r=0.4	0.4%	Estado Estacionario
	0.8%	Estado Estacionario
	1.5%	Estado Estacionario
	3.0%	
DENSEA	0.4%	
	0.8%	Generacional
	1.5%	Generacional
	3.0%	
DENSEA-NSH	0.4%	Estado Estacionario
	0.8%	Generacional
	1.5%	Generacional
	3.0%	Estado Estacionario

Si se considera la métrica amplitud del fi-ente para el caso concreto de 200 individuos citado con anterioridad, se comparan las estrategias de estado estacionario y generacional en la tabla 4.42. A

partir de ella podemos contabilizar 10 casos de superioridad de la estrategia generacional, 9 casos de superioridad de la estrategia de estado estacionario y 5 casos no concluyentes. Si los discriminamos por algoritmo empleado, entonces obtenemos superioridad de la estrategia generacional en el SPEA2, NSGAI y DENSEA, superioridad de la estrategia de estado estacionario en el NSGAI con elitismo controlado y $r=0.0$, así como en el NSGAI con elitismo controlado y $r=0.4$; además en el caso del algoritmo DENSEA-NSH, hay igualdad entre estrategias de reemplazamiento. Por consiguiente, en principio no se puede apreciar ventaja o desventaja alguna en términos de la métrica amplitud del frente entre las estrategias de reemplazamiento generacional y de estado estacionario para la población de 200 individuos, donde la amplia superioridad en la métrica de acercamiento al frente es notoria en términos del número de evaluaciones de la función objetivo.

En términos del equilibrio exploración-explotación, desde la perspectiva del balance diversidad poblacional frente a presión de selección tal y como es señalada en [4.46], parece crucial que un aumento del tamaño de la población hasta 200 individuos proporciona una diversidad poblacional inicial suficientemente alta para que la presión de selección mayor de la estrategia de estado estacionario permita un acercamiento al frente de soluciones no dominadas más rápido (mayor pendiente) en promedio y obteniendo soluciones más cercanas al frente no dominado óptimo.

Este tamaño de población se adapta a otros resultados preliminares en optimización de estructuras de nudos rígidos con algoritmos genéticos monoobjetivo por Chen y Rajan [4.8], donde entre otras conclusiones se propone como tamaño poblacional apropiado aquel cercano al tamaño del cromosoma. En este caso test Y, donde el tamaño del cromosoma es de 220 bits, el tamaño poblacional de 200 individuos es de su orden de magnitud, presentando una ventaja significativa la estrategia de estado estacionario mostrada en esta sección.

Comparando los algoritmos entre si para la estrategia de 200 individuos y reemplazamiento de estado estacionario (ver figuras 4.189 ó 4.192), son los algoritmos DENSEA y DENSEA-NSH los mejores. No obstante, la elección correcta del tipo de reemplazamiento y un tamaño poblacional adecuado, son de mayor influencia incluso que la probabilidad de mutación (dentro de los márgenes 0.4 a 1.5%) y que el tipo de algoritmo, puesto que cualquier algoritmo de entre SPEA2, NSGAI y los NSGAI con elitismo controlado empleando la estrategia de estado estacionario y población de 200 individuos es superior en términos de la métrica acercamiento al frente no dominado óptimo frente a los algoritmos DENSEA y DENSEA-NSH con otra estrategia, cuando la comparación se realiza en términos del número de evaluaciones funcionales y para este caso test Y.

Si en lugar del número de evaluaciones funcionales, se consideran los tiempos de ejecución como criterio comparativo, entonces se debe tener en cuenta el sobrecoste introducido por el aumento del número de generaciones (y el consiguiente aumento del número de veces que debemos ordenar por el criterio de no-dominancia y de diversidad del frente). Para el caso test Y, se muestra en la tabla 4.43 el tiempo en segundos de una ejecución en un ordenador Xeon biprocesador 1.7 Ghz de cada uno de los algoritmos, en sus versiones generacional y de estado estacionario (cada una de ellas supone la evaluación de 200.000 estructuras distintas).

Tabla 4.43. Tiempo en segundos de una ejecución del caso Y con 200.000 evaluaciones estructurales para un tamaño de población de 200 individuos.

Algoritmo	Generacional	Estado Estacionario
SPEA2	167	601
NSGA-II	156	230
NSGA-II elitcont r=0.0	152	238
NSGA-II elitcont r=0.4	152	246
DENSEA	159	1375
DENSEA-NSH	152	220

También se destaca que en caso de paralelización es más aprovechable la estrategia generacional, donde en cada generación se puede distribuir un mayor número de soluciones (tantas como el tamaño de la población), frente a la estrategia de estado estacionario, donde en cada generación sólo se generan 1 ó 2. Por tanto, la ventaja práctica de la estrategia de estado estacionario, considerando el factor de menor tiempo de cálculo a igualdad de calidad de la solución, en la resolución del problema multicriterio sería cuando no se dispusiera de la posibilidad de paralelización, y además la función objetivo tratada fuera suficientemente costosa como para compensar el sobrecoste del mayor número de generaciones y con ellas, el mayor número de evaluaciones de la ordenación de no-dominancia y de diversidad del frente.

En optimización multiobjetivo la estrategia de estado estacionario no es claramente ventajosa como ocurría en optimización monoobjetivo [4.24][4.25]. Es en el caso test Y, y para la población de 200 individuos, cuando se aprecia un mejor comportamiento de la métrica acercamiento al frente, si las comparaciones son en términos del número de evaluaciones de la función objetivo. En optimización multicriterio se deben tener en cuenta varios factores: aumenta el coste de cada generación al considerarse la selección por nodominancia además de los operadores de diversidad del frente (distancia crowding, operador de truncamiento, etc.). Ello implica un aumento del tiempo de cálculo necesario por generación. Además en multiobjetivo existe un menor hueco generacional ('generation gap') desde el punto de vista de de Jong y Sarma [4.29], puesto que el elitismo inherente en él es de múltiples individuos (tantos como miembros haya en la frontera óptima de individuos no-dominados) y por tanto su reemplazamiento generacional está a medio camino entre el de estado estacionario puro y el generacional en optimización monocriterio (la diferencia es menor frente a la optimización monocriterio, donde el elitismo en el caso del reemplazamiento generacional era de dos individuos).

No obstante, si consideramos la mejora en calidad de la solución obtenida (sobre todo en términos de mejor acercamiento al fi-ente óptimo), el incremento de tiempo de computación debido a la estrategia de estado estacionario puede ser asumible para un cálculo convencional (incremento desde 152 a 220 segundos en el caso del algoritmo DENSEA-NSH), siendo entonces la estrategia recomendada.

4.10 Comparación de los Algoritmos DENSEA frente a DENSEA-NSH en optimización multiobjetivo estructural

4.10.1 Caso Test X

En todos los casos correspondientes a este caso test X, para las treinta ejecuciones independientes ensayadas, se alcanza el frente óptimo de Pareto. Los valores de los promedios y desviaciones típicas del número de evaluaciones se muestran en la tabla 4.44, donde en la columna de la izquierda se expresan los promedios y en la columna de la derecha las desviaciones típicas de cada uno de los algoritmos. Se destaca en negrita el valor mejor entre los dos algoritmos comparados en cada caso (DENSEA o DENSEA-NSH).

Tabla 4.44. Promedio de la métrica acercamiento al frente óptimo en 30 ejecuciones independientes del caso test X para DENSEA y DENSEA-NSH (columna izquierda: Media y columna derecha: *Desviación Típica*).

Tam. Pob.	50				100				200			
	Med.	<i>D.Ti</i>	Med.	<i>D.Ti</i>	Med.	<i>D.Ti</i>	Med.	<i>D.Ti</i>	Med.	<i>D.Ti</i>	Med.	<i>D.Ti</i>
Pmut. 0.8%	DENSEA		DENSEANSH		DEN SEA		DENSEANSH		DEN SEA		DENSEANSH	
Generac.	2021	1721	2470	1986	2756	1844	2293	1173	3106	1453	2560	895
Esta. Estac.	2262	2342	3144	2648	2467	1888	1665	867	2353	1252	2853	1704
Pmut. 1.5%	DENSEA		DENSEANSH		DEN SEA		DENSEANSH		DEN SEA		DENSEANSH	
Generac.	1628	1363	1576	900	2306	1626	1993	985	2906	843	3200	1138
Esta. Estac.	1622	1330	1841	1737	2000	1289	1709	931	2383	929	2732	1404
Pmut. 3.0%	DENSEA		DENSEANSH		DEN SEA		DENSEANSH		DEN SEA		DENSEANSH	
Generac.	1533	998	1225	711	2416	1297	1720	620	2786	824	2626	631
Esta. Estac.	1569	900	1497	1119	2096	1061	1807	1098	2502	1298	2691	1144
Pmut. 6.0%	DENSEA		DENSEANSH		DEN SEA		DENSEANSH		DEN SEA		DENSEANSH	
Generac.	1325	620	1383	671	2130	898	1840	649	3360	930	2986	977
Esta. Estac.	1673	1048	1301	687	2010	865	1856	667	2666	1327	2729	992

Contabilizando a partir de la tabla 4.44 el número de veces que es mejor un algoritmo que otro, se obtienen las tablas 4.45 y 4.46, agrupados los resultados por probabilidades de mutación, y por tamaños poblacionales, además del resultado total. La tabla 4.45 expone los resultados de los promedios, y la tabla 4.46 los resultados de las desviaciones típicas.

De los resultados de promedio (tabla 4.45) se puede concluir para el caso test X, que en las probabilidades de mutación de 0.8% y 1.5%, existe igualdad en los resultados, mientras que las probabilidades altas favorecen al DENSEA-NSH ligeramente. Analizando los resultados agrupados por tamaño poblacional, se observa que en la población de 50 individuos existe igualdad entre ambos algoritmos, si bien en el tamaño de 100 individuos la victoria corresponde claramente al DENSEA-NSH y en la población de 200 individuos existe una ligera ventaja del DENSEA. Los resultados en conjunto, resultan igualados para la estrategia de estado estacionario, y ligeramente favorables al DENSEA-NSH en la estrategia generacional.

Las conclusiones son muy similares si estudiamos las desviaciones típicas en la tabla 4.46. En este caso test X, en las probabilidades de mutación bajas, de 0.8% y 1.5%) hay igualdad de resultados, sin embargo las probabilidades de mutación mayores son ligeramente ventajosas para el DENSEA-NSH.

Capítulo 4. Contribuciones

Si analizamos los resultados considerando el tamaño de la población, observamos una ligera ventaja del DENSEA-NSH con población de 200, gran dominio en la población de 100 y ligera desventaja en la población de 50. Los resultados en conjunto, resultan igualados para la estrategia de estado estacionario, y ligeramente favorables al DENSEA-NSH en la estrategia generacional.

Los mejores resultados de entre todas las estrategias ensayadas se obtienen para una población de 50 individuos y una probabilidad de mutación del 6%. En ese caso es vencedor el algoritmo DENSEA en la estrategia generacional y el DENSEA-NSH en la estrategia de estado estacionario; las diferencias entre ellos no obstante, son pequeñas.

Tabla 4.45. Número de casos favorables entre DENSEA y DENSEA-NSH para la media en el caso test X. Desglosado por probabilidad de mutación y por tamaño poblacional.

Probabilidad Mutación	0.8%		1.5%		3%		6%	
	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generacional	1	2	1	2	0	3	1	2
Estado Estacionario	2	1	2	1	1	2	1	2
Total	3	3	3	3	1	5	2	4
Tamaño Población	50		100		200		Total	
	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generacional	2	2	0	4	1	3	3	9
Estado Estacionario	2	2	0	4	4	0	6	6
Total	4	4	0	8	5	3	9	15

Tabla 4.46. Número de casos favorables entre DENSEA y DENSEA-NSH para la desviación típica en el caso test X. Desglosado por probabilidad de mutación y por tamaño poblacional.

Probabilidad Mutación	0.8%		1.5%		3%		6%	
	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generacional	1	2	1	2	0	3	1	2
Estado Estacionario	2	1	2	1	2	1	0	3
Total	3	3	3	3	2	4	1	5
Tamaño Población	50		100		200		Total	
	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generacional	2	2	0	4	1	3	3	9
Estado Estacionario	3	1	1	3	2	2	6	6
Total	5	3	1	7	3	5	9	15

Se muestran en las figuras 4.265 a 4.267 las comparativas entre ambos algoritmos desglosadas por tamaño poblacional y tipo de reemplazamiento generacional para el caso X en función del número de evaluaciones de la función objetivo necesarias para alcanzar el frente de Pareto óptimo. Las figuras 4.268 a 4.279 que muestran la evolución de la convergencia de los distintos casos se incluyen en el Anexo.

Caso X) DENSEA vs DENSEA-NSH:

Número de evaluaciones en alcanzar frente óptimo (medias y desviaciones típicas):

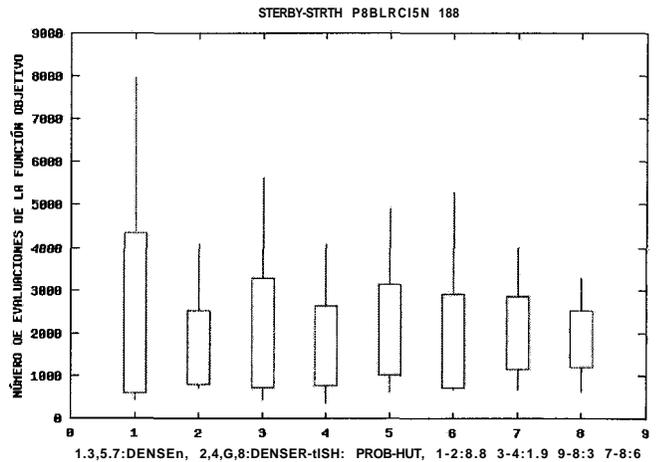
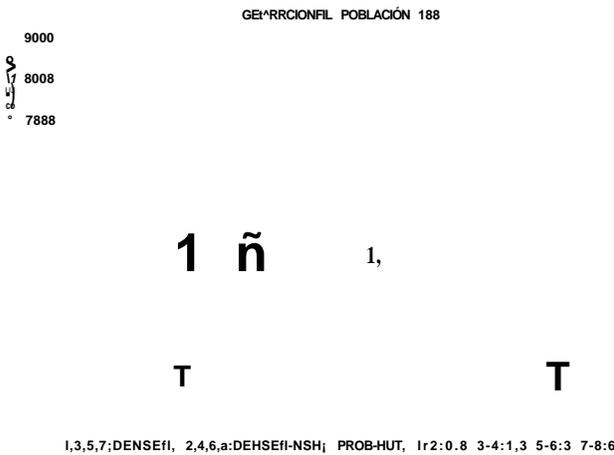
GENERACIONES POBLACION 50

STEDV-STRTE POBLACION 50

1,3,5,7:DENSEn, 2,4,6,8:DEHSEf1-NSH; PROB-HUT, 1-2:0.8 3-4:1.5 5-6:3 7-8:6

1,3,5,7:DEHSEf1, 2,4,6,8:DENSER-NSH; PROB-HUT, 1-2:8.8 3-4:1.5 5-6:3 7-8:6

Figura 4.265. Comparativa de algoritmos DENSEA / DENSEA-NSH en Caso X. Gráficas sobre Frente de Pareto. Tamaño población 50.

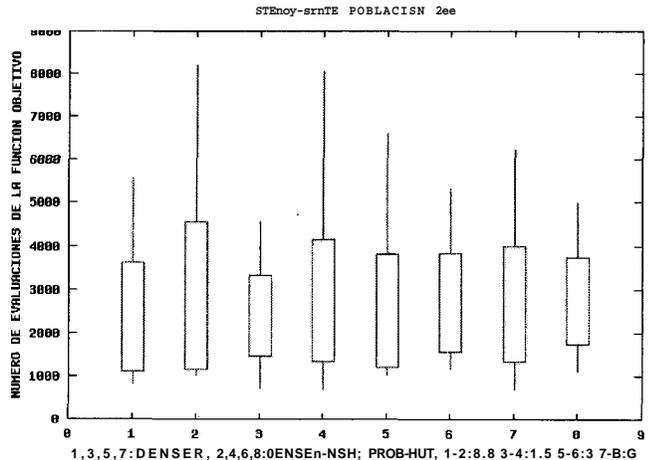


1,3,5,7:DENSEf1, 2,4,6,8:DEHSEf1-NSH; PROB-HUT, 1-2:0.8 3-4:1,3 5-6:3 7-8:6

1,3,5,7:DENSEn, 2,4,6,8:DENSER-NSH; PROB-HUT, 1-2:8.8 3-4:1.9 9-8:3 7-8:6

Figura 4.266. Comparativa de algoritmos DENSEA / DENSEA-NSH en Caso X. Gráficas sobre Frente de Pareto. Tamaño población 100.

GENERACIONES POBLACION 288



1,3,5,7:DENSEn, 2,4,6,8:DEHSEf1-NSH; PROB-HUT, 1-2:8.8 3-4:1.5 5-6:3 7-8:1

1,3,5,7:DENSER, 2,4,6,8:DEHSEf1-NSH; PROB-HUT, 1-2:8.8 3-4:1.5 5-6:3 7-8:6

Figura 4.267. Comparativa de algoritmos DENSEA / DENSEA-NSH en Caso X. Gráficas sobre Frente de Pareto. Tamaño población 200.

4.10.2 Caso Test Y

En las figuras 4.280 a 4.291, de las que se muestran las correspondientes a las probabilidades de mutación de 0.4% a 0.8% (figuras 4.280 a 4.285, el resto se encuentra en el *Anexo*) podemos observar un comportamiento similar de los algoritmos DENSEA y DENSEA-NSH a lo largo de toda la evolución en el caso test Y. Se distinguen dos tendencias diferenciadas, la de los algoritmos generacionales y la de los algoritmos de estado estacionario.

Caso Y) DENSEA vs DENSEA-NSH: Pmutación del 0.4% y 3 tamaños de población:

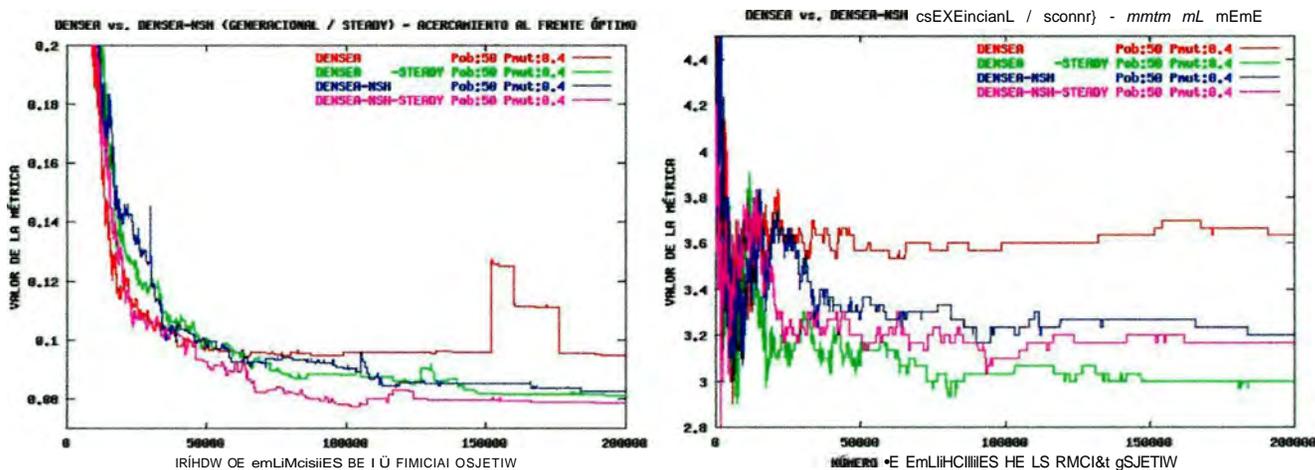


Figura 4.280. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 0.4%. Tamaño población 50.

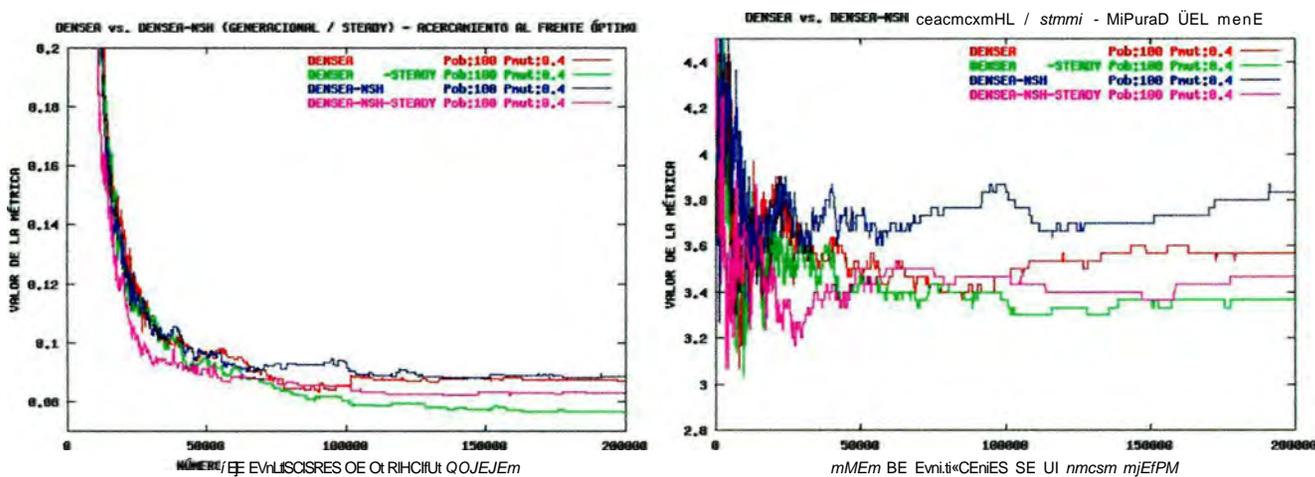


Figura 4.281. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 0.4%. Tamaño población 100.

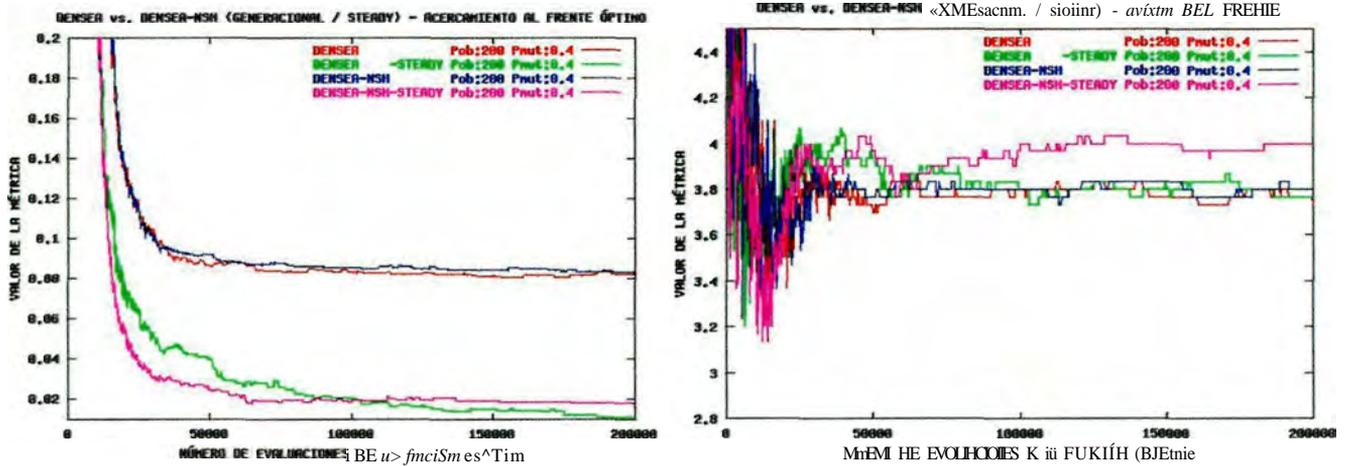


Figura 4.282. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 0.4%. Tamaño población 200.

Caso Y) DENSEA vs DENSEA-NSH: Pmutación del 0.8% y 3 tamaños de población:

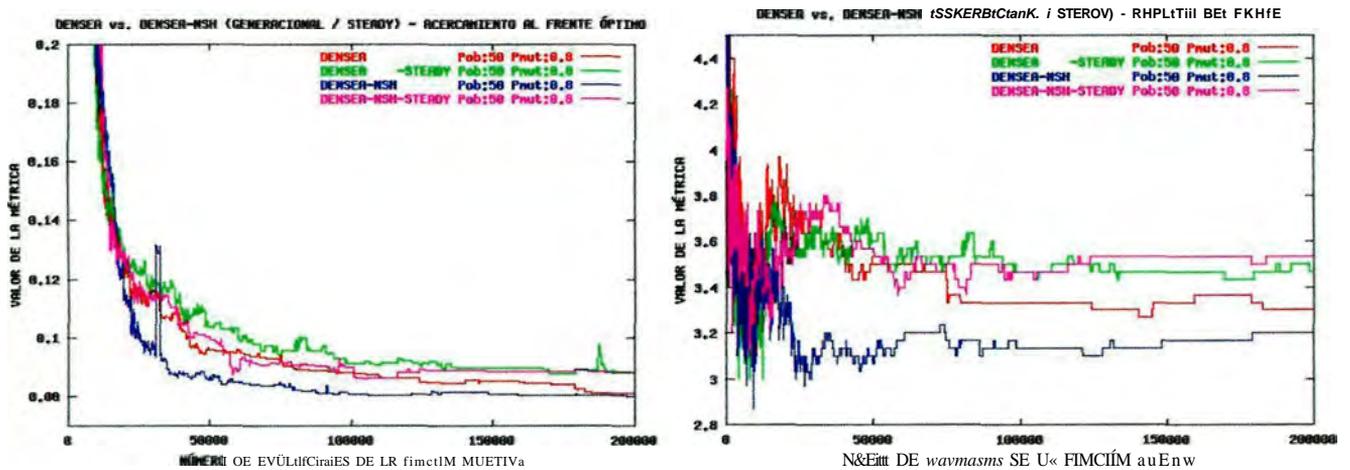


Figura 4.283. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 0.8%. Tamaño población 50.

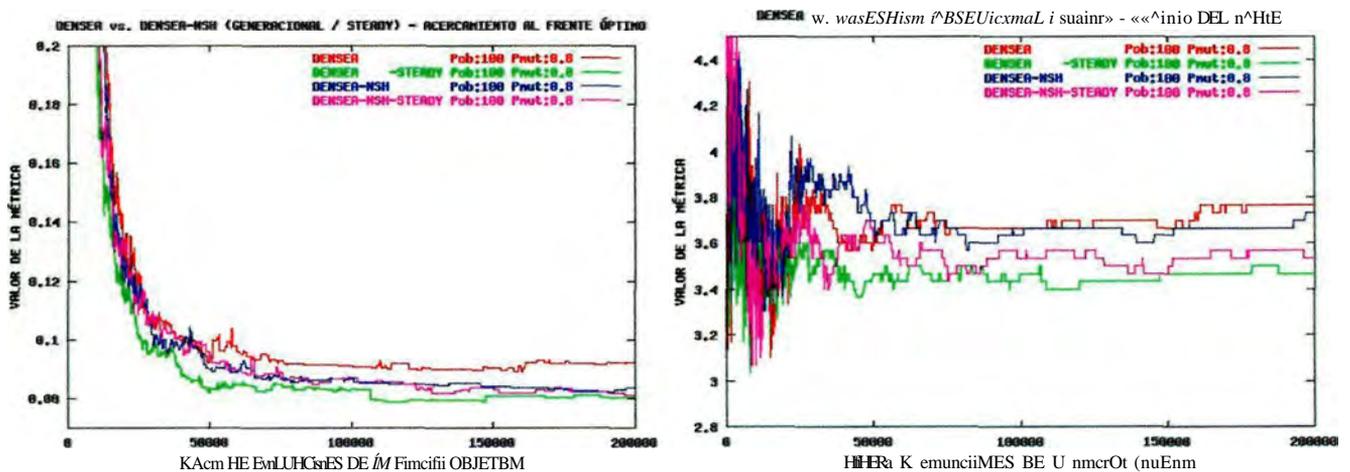


Figura 4.284. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 0.8%. Tamaño población 100.

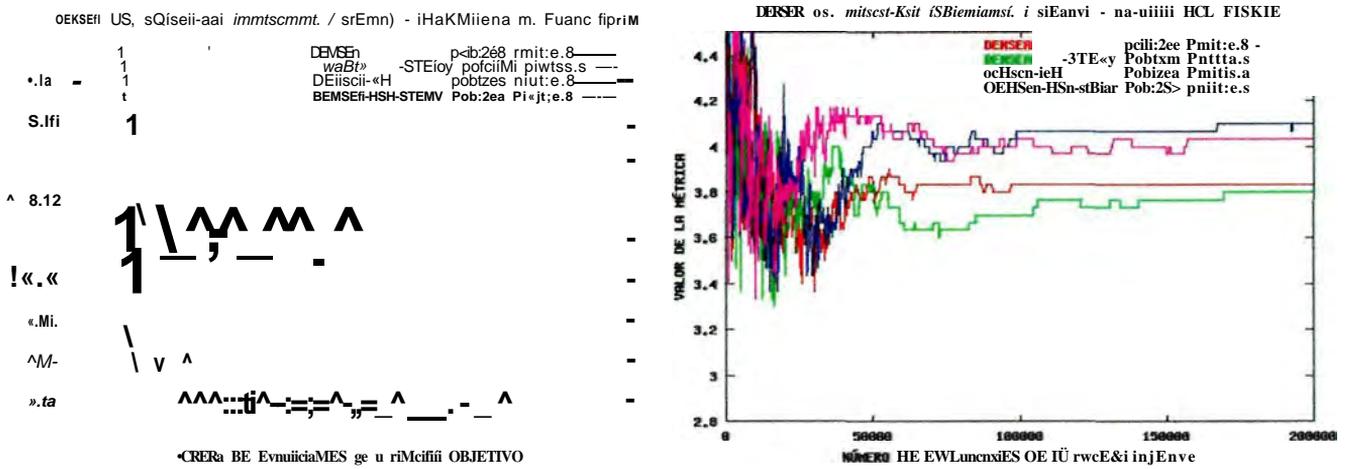


Figura 4.285. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 0.8%. Tamaño población 200.

En las tablas 4.47 a 4.52 se efectúa un análisis sobre los valores finales de las métricas indicadoras de la aproximación al fi-ente óptimo y de la amplitud del frente, en el punto final de convergencia del algoritmo (evaluación funcional 200.000), tanto en promedio como en varianza. Las tablas 4.47 y 4.48 muestran el promedio y varianza respectivamente, de la métrica acercamiento al fi-ente óptimo, mientras que las tablas 4.49 y 4.50 muestran el promedio y varianza respectivamente, de la métrica amplitud del fi-ente. En la tabla 4.51 se comparan los resultados de las cuatro tablas anteriores, resumiéndose en la tabla 4.52.

Tabla 4.47. Promedio de la métrica acercamiento al frente óptimo en 30 ejecuciones independientes del caso test Y para DENSEA y DENSEA-NSH tras 200.000 evaluaciones.

Tam. Pob.		50		100		200	
Pmut. 0.4%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH	
	Generac.	0.09460	0.08249	0.08716	0.08830	0.08318	0.08320
	Esta. Estac.	0.08116	0.07860	0.07645	0.08292	0.01080	0.01782
Pmut. 0.8%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH	
	Generac.	0.08100	0.08014	0.09217	0.08369	0.08375	0.09457
	Esta. Estac.	0.08805	0.08819	0.08029	0.08109	0.01015	0.01363
Pmut. 1.5%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH	
	Generac.	0.08099	0.09907	0.08480	0.08553	0.08693	0.09475
	Esta. Estac.	0.07773	0.08724	0.07865	0.07839	0.01525	0.01694
Pmut. 3.0%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH	
	Generac.	0.10375	0.10545	0.12002	0.12175	0.13094	0.12958
	Esta. Estac.	0.08941	0.09104	0.10850	0.10579	0.07023	0.07476

Capítulo 4. Contribuciones

Tabla 4.48. Varianza de la métrica acercamiento al frente óptimo en 30 ejecuciones independientes del caso test Y para DENSEA y DENSEA-NSH tras 200.000 evaluaciones.

Tam. Pob.	50		100		200	
Pmut. 0.4%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	0.00152	0.00068	0.00075	0.00071	0.00038	0.00037
Esta. Estac.	0.00053	0.00059	0.00037	0.00032	0.00005	0.00055
Pmut. 0.8%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	0.00057	0.00025	0.00186	0.00063	0.00012	0.00037
Esta. Estac.	0.00107	0.00202	0.00066	0.00048	0.00006	0.00005
Pmut. 1.5%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	0.00036	0.01003	0.00047	0.00030	0.00012	0.00020
Esta. Estac.	0.00029	0.00069	0.00021	0.00025	0.00004	0.00009
Pmut. 3.0%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	0.00069	0.00056	0.00044	0.00103	0.00044	0.00050
Esta. Estac.	0.00068	0.00037	0.00028	0.00047	0.00045	0.00098

Tabla 4.49. Promedio de la métrica amplitud del frente en 30 ejecuciones independientes del caso test Y para DENSEA y DENSEA-NSH tras 200.000 evaluaciones.

Tam. Pob.	50		100		200	
Pmut. 0.4%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	3.63	3.20	3.57	3.83	3.73	3.80
Esta. Estac.	3.00	3.17	3.37	3.47	3.77	4.00
Pmut. 0.8%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	3.30	3.20	3.77	3.73	3.83	4.10
Esta. Estac.	3.47	3.53	3.47	3.53	3.80	4.03
Pmut. 1.5%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	3.23	3.30	3.70	3.77	3.87	4.03
Esta. Estac.	3.27	3.40	3.60	3.40	3.77	3.77
Pmut. 3.0%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	3.27	3.10	3.67	3.67	3.80	3.80
Esta. Estac.	3.20	3.17	3.63	3.53	4.10	4.17

Tabla 4.50. Varianza de la métrica amplitud del frente en 30 ejecuciones independientes del caso test Y para DENSEA y DENSEA-NSH tras 200.000 evaluaciones.

Tam. Pob.	50		100		200	
Pmut. 0.4%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	0.765	0.560	0.845	0.739	0.396	0.293
Esta. Estac.	0.867	0.739	0.766	0.582	0.179	1.000
Pmut. 0.8%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	0.610	0.560	0.912	0.462	0.139	0.157
Esta. Estac.	1.116	0.582	0.516	0.516	0.627	0.432
Pmut. 1.5%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	0.512	0.477	0.410	0.246	0.116	0.099
Esta. Estac.	0.396	0.640	0.240	0.373	0.246	0.246
Pmut. 3.0%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	0.729	0.757	0.956	0.622	1.160	1.027
Esta. Estac.	0.760	0.672	0.766	0.582	0.890	0.739

Tabla 4.51. DENSEA frente a DENSEA-NSH en caso Y. Mejor algoritmo considerando: promedio (Me) y varianza (De) de métricas acercamiento al frente óptimo {Ac} y amplitud del frente (Am), sobre 30 ejecuciones de cada caso.

Tam. Pob.	50		100		200	
Pmut. 0.4%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	AmMe	AcMe, AcDe, AmDe	AcMe	AcDe, AmMe, AmDe	AcMe	AcDe, AmMe, AmDe
Esta. Estac.	AcDe	AcMe, AmMe, AmDe	AcMe	AcDe, AmMe, AmDe	AcMe, AcDe, AmDe	AmMe
Pmut. 0.8%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	AmMe	AcMe, AcDe, AmDe	AmMe	AcMe, AcDe, AmDe	AcMe, AcDe, AmDe	AmMe
Esta. Estac.	AcMe, AcDe	AmMe, AmDe	AcMe	AcDe, AmMe	AcMe	AcDe, AmMe, AmDe
Pmut. 1.5%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	AcMe, AcDe	AmMe, AmDe	AcMe	AcDe, AmMe, AmDe	AcMe, AcDe	AmMe, AmDe
Esta. Estac.	AcMe, AcDe, AmDe	AmMe	AcDe, AmMe, AmDe	AcMe	AcMe, AcDe	
Pmut. 3.0%	DENSEA	DENSEANSH	DENSEA	DENSEANSH	DENSEA	DENSEANSH
Generac.	AcMe, AmMe, AmDe	AcDe	AcMe, AcDe	AmDe	AcDe	AcMe, AmDe
Esta. Estac.	AcMe, AmMe	AcDe, AmDe	AcDe, AmMe	AcMe, AmDe	AcMe, AcDe	AmMe, AmDe

Tabla 4.52. Contabilización de mejores valores sobre tabla anterior.

	DENSEA	DENSEANSH
Promedio Acercamiento (<i>AcMé</i>)	17	7
Varianza Acercamiento { <i>AcDé</i> }	13	11
Promedio Amplitud (AmMe)	7	14
Varianza Amplitud (AmDe)	5	17
<i>Total Promedios</i>	24	21
<i>Total Varianzas</i>	18	28
<i>Total Global</i>	42	49

Para este caso test Y, a partir de la tabla 4.52 se concluye que el algoritmo DENSEA es ligeramente superior en términos de la métrica de acercamiento, tanto en promedio de la métrica (valor del número de mejores casos es de 17 frente a 7) como en varianza (valor del número de mejores casos es de 13 frente a 11). En cambio, el algoritmo DENSEA-NSH es ligeramente superior en términos de la métrica de amplitud de frente, tanto en promedio (valor del número de mejores casos es de 14 frente a 7) como en varianza (valor del número de mejores casos es de 17 frente a 5). El operador distribuidor de la diversidad a lo largo del frente de soluciones no-dominadas implementado en el DENSEA-NSH, basado en la distancia del espacio ñinccional sobre la segunda ñincción objetivo únicamente (número de distintos tipos de perfiles), parece a la vista de los resultados analizados de este caso test Y, favorecer una mejora en la amplitud de los frentes obtenidos. No obstante, ello se consigue a costa de perjudicar ligeramente el acercamiento al mejor frente.

Si nos fijamos en la mejor combinación de los casos ensayados para este caso test Y -que resulta ser la de tamaño de población de 200 individuos y estrategia de reemplazamiento de estado estacionario con las probabilidades de mutación entre 0.4% y 1.5%- se puede observar la misma conclusión que la ofrecida por el conjunto de casos en el párrafo anterior: el algoritmo DENSEA presenta un mejor valor de la métrica de aproximación al frente en promedio y varianza, si bien el algoritmo DENSEA-NSH

muestra una amplitud del frente ligeramente mayor en promedio (si bien las diferencias son pequeñas en valor).

Los resultados comparativos entre los algoritmos DENSEA y DENSEA-NSH en cualquier caso no permiten establecer diferencias notables, siendo su comportamiento cualitativamente de similar magnitud y mejor que el resto de algoritmos comparados en las secciones anteriores para ambos casos test ensayados, X e Y. Se destaca que el algoritmo DENSEA-NSH es totalmente competitivo frente al DENSEA, habiéndose implementado una independencia de parámetros en el operador distribuidor de la diversidad, habiéndose además simplificado su tiempo de computación sin perjuicio del rendimiento, y en ocasiones habiendo mostrado mejora del mismo, tanto en acercamiento al frente óptimo como en amplitud.

4.11 El Operador Renacimiento AutoAdaptativo en optimización estructural. Análisis frente al Renacimiento

El equilibrio exploración explotación es la tarea principal a conseguir en la resolución de problemas de búsqueda. Una explotación excesiva de la información de alta calidad obtenida implica una peor exploración del espacio de búsqueda, siendo la población finita. Análogamente, una exploración excesiva puede provocar una pérdida de eficiencia en el proceso, aproximándolo a una búsqueda aleatoria. El comportamiento de evolución de un algoritmo evolutivo puede ser analizado desde el punto de vista de este equilibrio, tal como señala Holland [4.27].

Un modo de abordar con el equilibrio exploración explotación en algoritmos evolutivos es por medio de dos factores principales que controlan la evolución [4.46]: *presión de selección* y *diversidad de la población*. Ambos están inversamente relacionados: una presión de selección alta implica una rápida pérdida de diversidad poblacional, a causa de un excesivo foco de la búsqueda evolutiva en los mejores miembros de la población; por el contrario, el mantenimiento de la diversidad poblacional puede neutralizar los efectos de una presión de selección excesiva. La mayoría de los parámetros que son usados para ajustar las estrategias de una búsqueda evolutiva son realmente términos indirectos de ajuste de la presión de selección y la diversidad poblacional.

Un operador que introduce diversidad en la población es la reinicialización. Este operador ha sido sugerido en Goldberg [4.18] y consiste en la creación de una nueva población inicial tras el estancamiento del algoritmo genético, en la cual el mejor individuo de la población anterior es insertado. De ese modo, ese individuo es trasmisor de la información genética obtenida con la ejecución inicial y simultáneamente los nuevos individuos generados aleatoriamente contribuyen a la diversidad poblacional, permitiendo de ese modo una continuación de la evolución. Este operador de reinicialización ha sido usado en muchos algoritmos evolutivos, como el microalgoritmo genético [4.32] o el algoritmo CHC [4.14][4.15].

Otro factor que influye sobre la diversidad poblacional es el tamaño de la población. El tamaño de la población juega un importante papel en la diversidad poblacional, y por tanto, en las tareas exploratorias del algoritmo: mayores tamaños poblacionales son asociados con menor velocidad de convergencia del algoritmo, pero también menor convergencia prematura de la población. Menores tamaños poblacionales pueden conducir a convergencia prematura, y su carencia de diversidad poblacional tener que ser corregida con otros operadores que la incrementan, como mayores probabilidades de mutación o el operador de reinicialización (micro-algoritmo genético). Un estudio acerca de la influencia del tamaño de la población y su interacción con el operador renacimiento aplicado a optimización discreta estructural será mostrado con posterioridad en esta sección.

Una aplicación para optimizar la estimación de parámetros ópticos en cristales líquidos emplea la reducción sucesiva de los intervalos de búsqueda de las variables fenotípicas como operador de mejora por Mikulin [4.35].

El código Delta [4.48], es el precursor más similar del operador renacimiento: cuando la convergencia es alcanzada, el algoritmo conserva el mejor valor de entre aquellos obtenidos en ese punto y realiza una reinicialización. La nueva población cambia su codificación y las variables fenotípicas se transforman en valores incrementales con respecto al mejor individuo, reduciendo su tamaño de bits (y consiguientemente su longitud de cromosoma).

El operador renacimiento consiste en una reinicialización y una reducción del intervalo de las variables fenotípicas del cromosoma (lo que implica una reducción del espacio de búsqueda) y de la longitud del cromosoma. Fue introducido y empleado por Miguel Galante [4.16] en optimización de barras de nudos articulados [4.17]. Cuando la población converge, es aplicado y extermina todos los miembros de la población excepto el mejor individuo. Crea una nueva población y recodifica el intervalo de las variables, que es reducido y centrado en las variables del mejor individuo. Un operador similar aplicado a la optimización de barras de nudos articulados denominado 'Método de las dos Fases' es usado en Rajeev Krishnamoorthy [4.38], con cromosomas de longitud variable.

Los Algoritmos Evolutivos son herramientas de optimización global gracias a trabajar con una población de soluciones en lugar de con un único individuo, pero a causa de su estocacidad, la localización del óptimo global no está garantizada. Por tanto, diferentes ejecuciones encaminadas a resolver un problema concreto pueden conducir a distintas soluciones óptimas finales (variación de los resultados cuando varias ejecuciones son lanzadas). Una alternativa para reducir esta variación es realizar una búsqueda más allá, tras el estancamiento del algoritmo genético. Hay autores que realizan una búsqueda local, considerando como punto de partida el mejor valor encontrado por el algoritmo evolutivo (algoritmos meméticos). Sin embargo, el operador renacimiento sustituye la búsqueda local por una búsqueda evolutiva, manteniendo sus características globales y concentrada en un espacio de búsqueda menor del dominio de las variables y centrado en los valores de las variables del mejor individuo.

El operador renacimiento autoadaptativo está basado en el operador renacimiento introducido por Galante en [4.16] y aplicado a optimización de estructuras de barras articuladas. Consiste en la aplicación del renacimiento consecutivamente en la misma ejecución, tantas veces como pueda mejorar la convergencia. Se aplica tras la saturación de la población repetidamente, tantas veces como sea necesario, para evitar el estancamiento en un mínimo local y para mejorar los resultados finales en términos de velocidad y precisión. El criterio automático para activar este operador propuesto es el estancamiento de la población (parámetro RP que considera el número de evaluaciones funcionales sin mejora en el mejor individuo en términos del valor de la función objetivo). Un valor apropiado de este parámetro es esencial para obtener el mejor comportamiento de convergencia. Consideraciones acerca de cómo seleccionar este parámetro son discutidas a continuación en esta sección. Un valor de compromiso obtenido para los casos test tratados es la elección de 5000 evaluaciones. Después de que esta condición de saturación sea satisfecha, el renacimiento es aplicado. Él inicia una nueva población y una nueva búsqueda, después de la cual el renacimiento es nuevamente aplicado de forma repetida, y así sucesivamente. Después de que tras su aplicación, las soluciones obtenidas no mejoran, el criterio de parada es alcanzado. Resultados comparativos mostrando las ventajas del operador de renacimiento autoadaptativo frente al operador renacimiento son presentadas a continuación, aplicadas a la optimización discreta de estructuras de nudos rígidos.

El algoritmo evolutivo considerado en los siguientes resultados experimentales, es el que mejor resultados proporciona de entre varias estrategias de algoritmos evolutivos monoobjetivo (generacional, estado-estacionario, CHC) en los dos casos test ensayados, ver [4.24][4.25]. El tamaño de la población considerado es de 50 individuos y con una probabilidad de mutación del 3%. Es un algoritmo genético de estado-estacionario sin duplicados en la población, y donde la sustitución de dos individuos en la

población es en lugar de los peores de la generación anterior. Se emplea cruce uniforme, probabilidad de cruce 1.0 y selección estocástica universal (SUS).

4.11.1 Influencia del tamaño de la población y la reducción espacial

Inicialmente, se muestra en las figuras (4.292 a 4.295) una comparativa de un tamaño de población medio de 50 individuos con otro pequeño de 8 individuos. Se presentan algunos resultados variando el parámetro de saturación RP que define cuándo se aplica el renacimiento autoadaptativo. Los valores de RP =3000, 6000, 9000 y 12000 son mostrados, definiendo el número de evaluaciones de la función en que no ocurre mejora del mejor individuo de la población. Las figuras reflejan el promedio de 30 ejecuciones independientes y un modelo incluyendo pandeo y peso propio es considerado.

Se aprecia que en cada caso el tamaño poblacional pequeño permite una convergencia inicial más rápida, pero unos peores resultados finales, conduciendo a una cantidad incrementada de valores subóptimos (su línea de convergencia es siempre superior respecto del algoritmo de población media). Esto puede ser explicado porque la calidad de la mejor inicial solución previa a la aplicación del primer renacimiento depende mucho del tamaño de la población, influencia que se incrementa con grandes espacios de búsqueda, que requieren alta diversidad poblacional para evitar la convergencia prematura (el caso test tiene un espacio de búsqueda de 2^{20}).

Con posterioridad, se muestran algunas figuras (4.296 a 4.298) comparando el algoritmo con renacimiento autoadaptativo y el algoritmo con únicamente reinicialización en una población con 8 individuos (renacimiento autoadaptativo frente a microalgoritmo genético). Ambos métodos emplean la reinicialización como un modo de incrementar la diversidad poblacional tras la saturación de la población, pero el microalgoritmo genético no emplea reducción del espacio, como hace el renacimiento autoadaptativo. El efecto beneficioso de la reducción de espacio introducido por el renacimiento se muestra claramente en las figuras, tanto en términos de velocidad de convergencia como en términos de calidad de los resultados finales (e independientemente del valor del parámetro RP).

Capítulo 4. Contribuciones

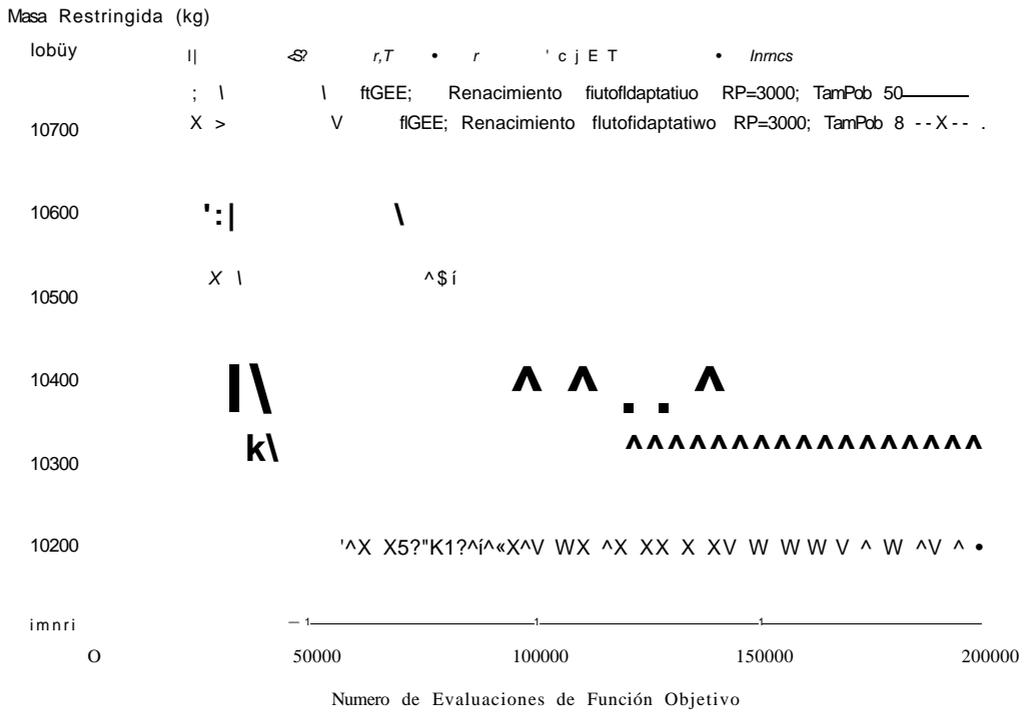


Figura 4.292. Influencia de Tamaño Poblacional. Promedio de 30 ejecuciones independientes. RP=3000.

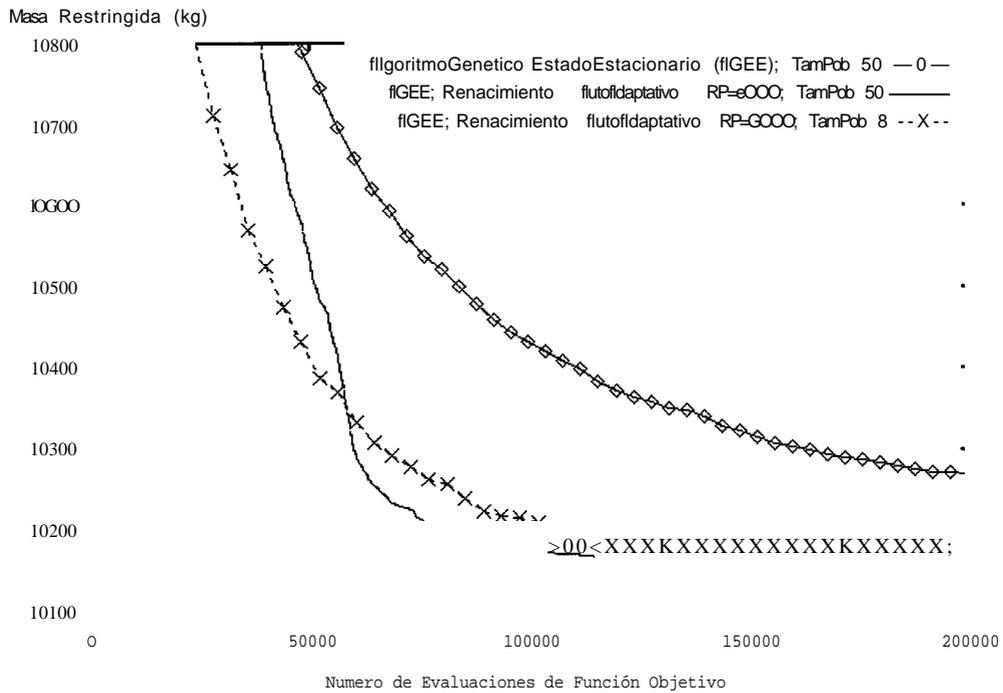


Figura 4.293. Influencia de Tamaño Poblacional. Promedio de 30 ejecuciones independientes. RP=6000.

Capítulo 4. Contribuciones

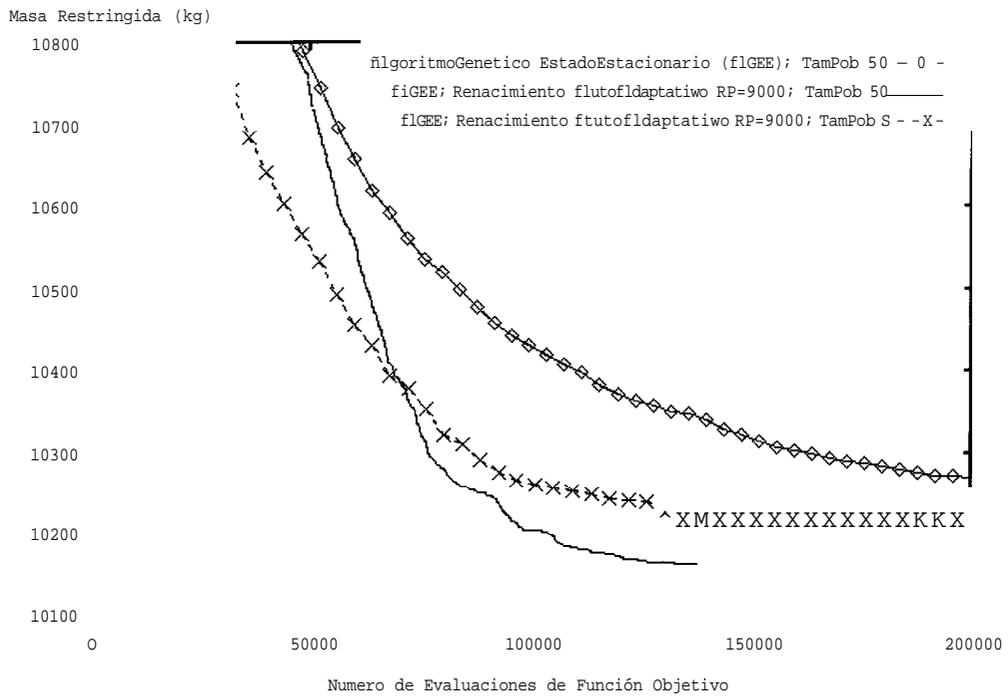


Figura 4.294. Influencia de Tamaño Poblacional. Promedio de 30 ejecuciones independientes. RP=9000.

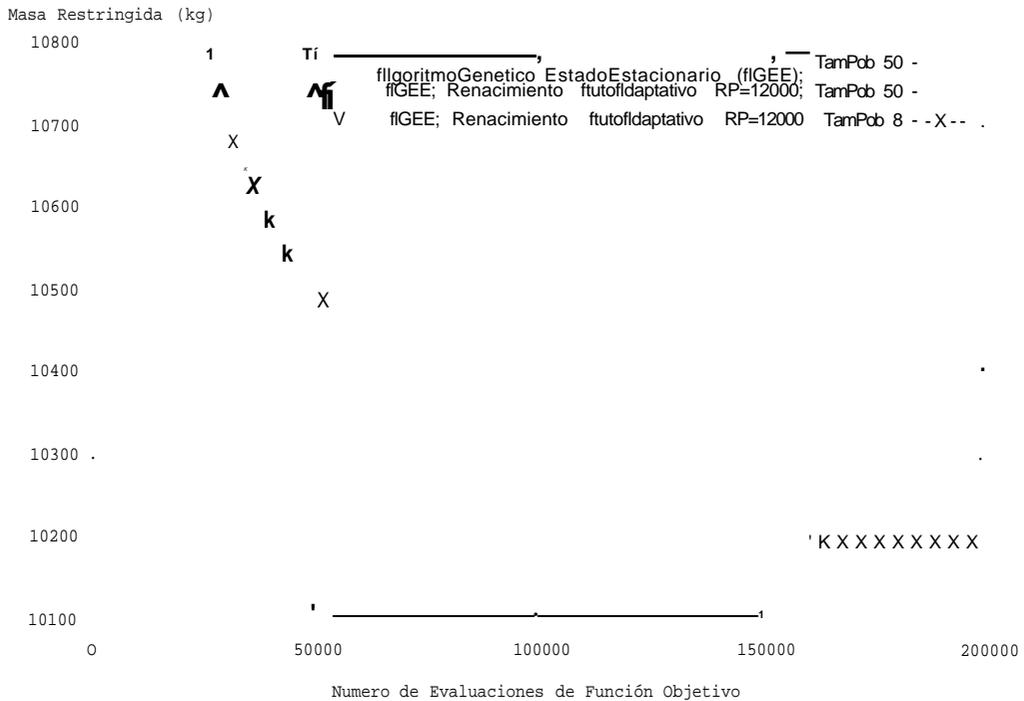


Figura 4.295. Influencia de Tamaño Poblacional. Promedio de 30 ejecuciones independientes. RP=12000.

Capítulo 4. Contribuciones

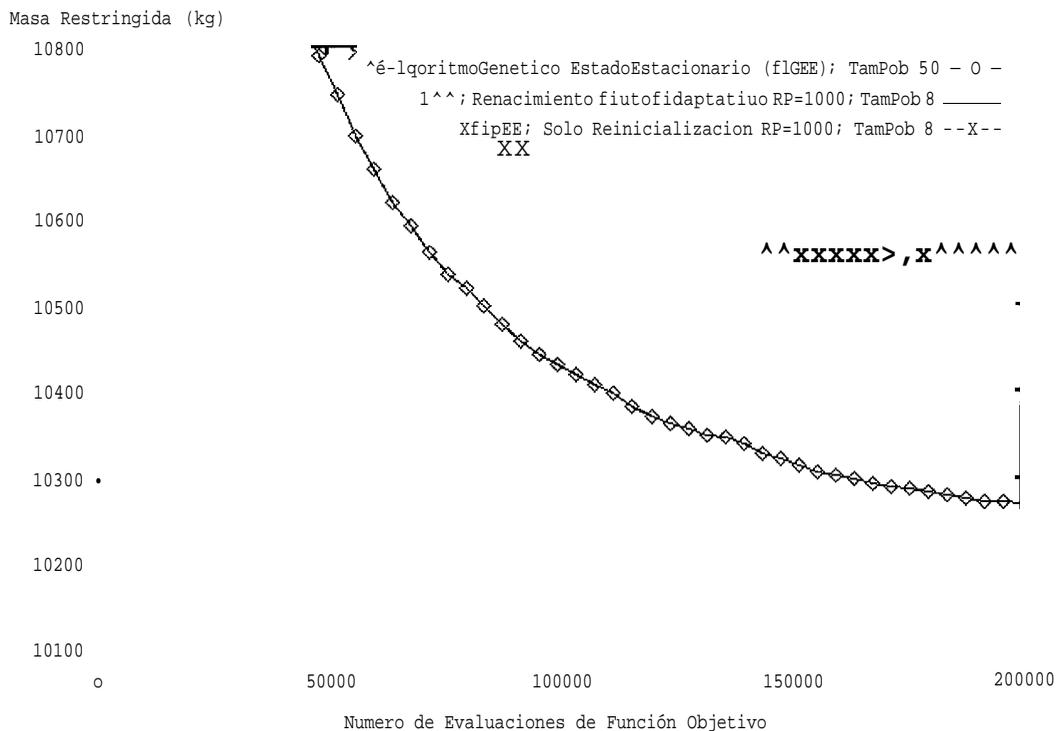


Figura 4.296. Influencia de la Reducción Espacial. Promedio de 30 ejecuciones independientes. RP=1000

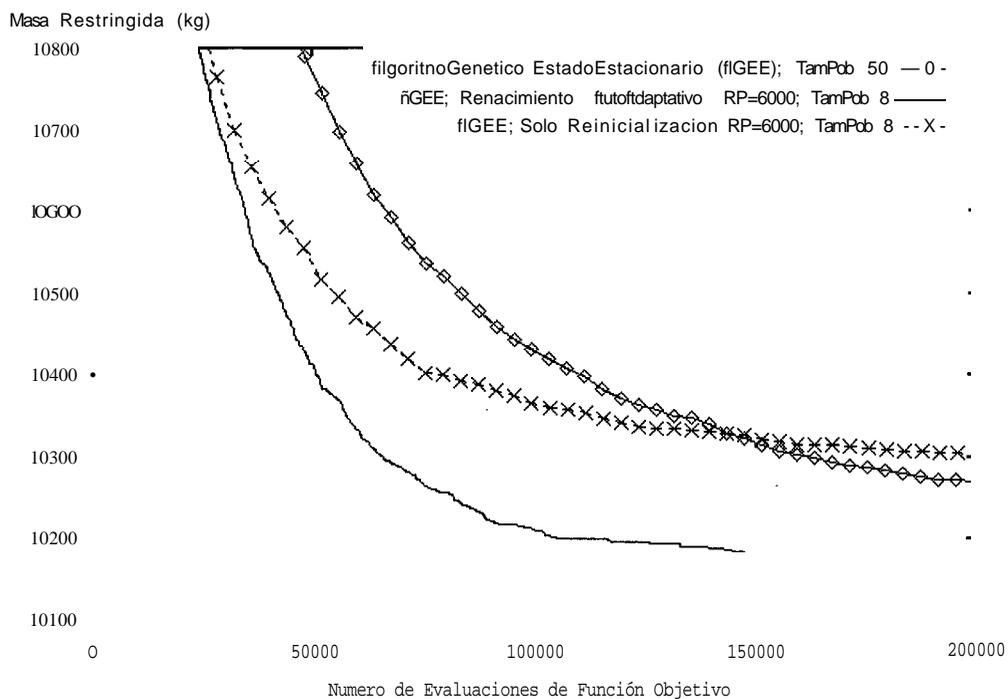


Figura 4.297. Influencia de la Reducción Espacial. Promedio de 30 ejecuciones independientes. RP=6000

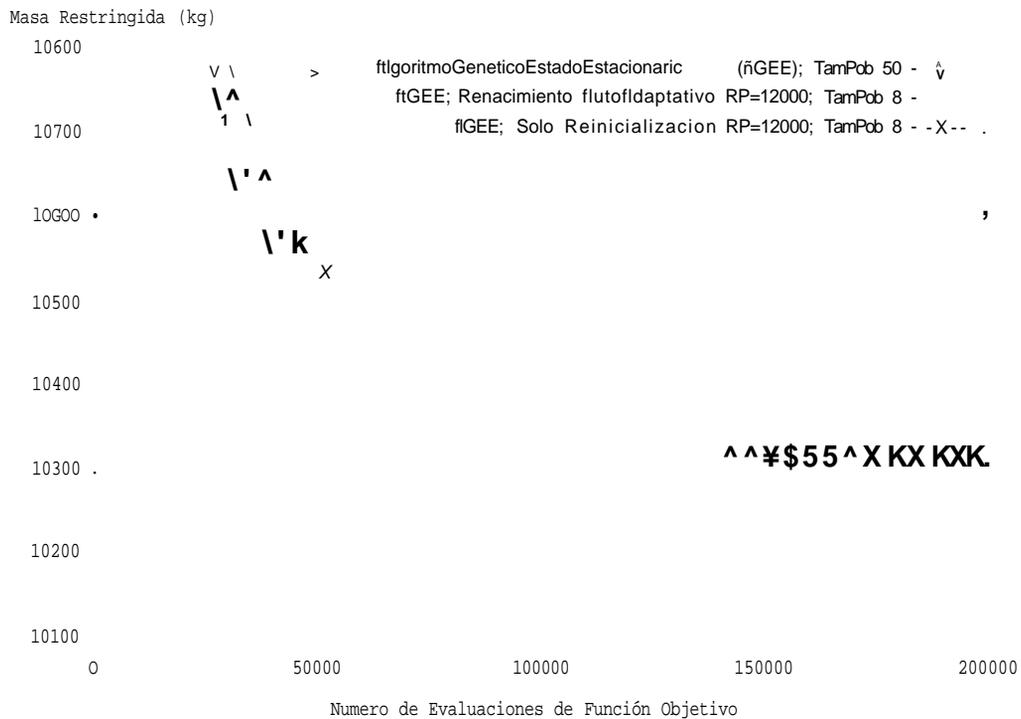


Figura 4.298. Influencia de la Reducción Espacial. Promedio de 30 ejecuciones independientes. RP=12000

4.11.2 Influencia del parámetro de saturación de la población (RP)

El momento de aplicar el renacimiento debe ser elegido de tal modo que permita al algoritmo completar su convergencia. Si es aplicado demasiado pronto, podría conducir a un renacimiento prematuro que tomara un subóptimo como punto central de la reducción del espacio de búsqueda y condujera a una convergencia prematura con soluciones subóptimas. Si es aplicado tarde, estaríamos alargando la convergencia innecesariamente.

Las figuras 4.299 y 4.300 reflejan los valores promedios (figura 4.299) y varianzas (figura 4.300) de 30 ejecuciones independientes del modelo real de optimización discreta del caso test Y. Se han seleccionado cuatro valores del parámetro de saturación poblacional RP: 3000, 5000, 9000 y 12000, definiendo el número de evaluaciones consecutivas de la función objetivo que no suponen cambio en el mejor individuo de la población. Se aprecia cómo el valor de 3000 conduce a mayor convergencia prematura y mayor varianza que los otros, así como el valor de 12000 proporcional la más lenta velocidad de convergencia.

Una selección adecuada del parámetro de saturación es un valor que proporcione una convergencia rápida sin pérdida de precisión final en la solución.

El valor reportado de 5000 tiene una velocidad de convergencia más rápida que 9000 y 12000 sin pérdida significativa de precisión, tal como se observa en la figura 4.299 (sin embargo, su varianza es ligeramente peor que la de estos dos valores).

Para seleccionar un valor apropiado del parámetro de saturación, la longitud de cromosoma es el factor principal a considerar, porque el tamaño del espacio de búsqueda (en este caso 220 bits) principalmente determina la velocidad de convergencia. Longitudes de convergencia menores permiten menores valores de RP.

Capítulo 4. Contribuciones

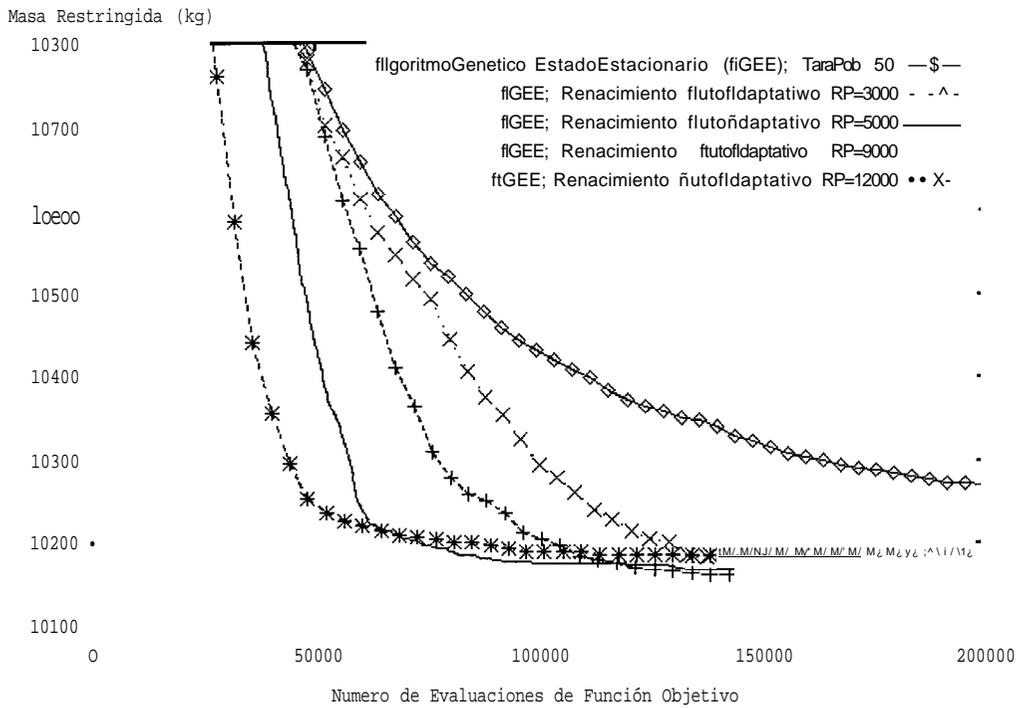


Figura 4.299. Influencia del parámetro RP. Promedio de 30 ejecuciones independientes.

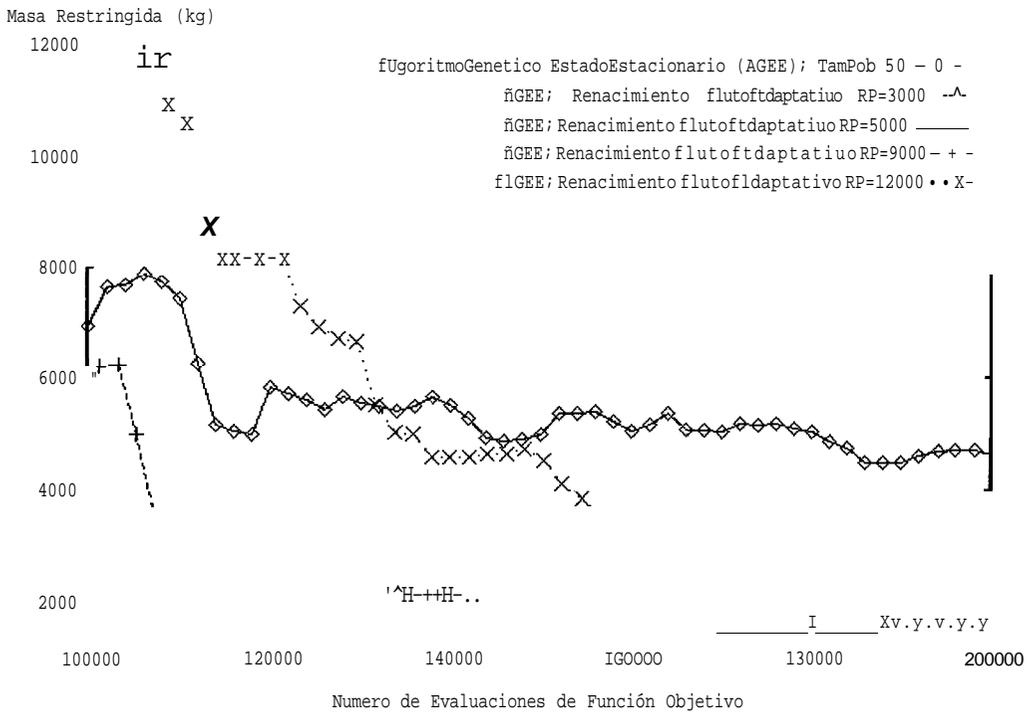


Figura 4.300. Influencia del parámetro RP. Varianza de 30 ejecuciones independientes.

Dos casos test estructurales adicionales son analizados aquí. El segundo tiene 45 barras y 30 nudos, y se encuentra representado en la figura 4.301, considerando que su dimensionamiento, cargas y características son las mismas que el caso test Y representado en la figura 3.4 (incluyendo la carga uniforme en las vigas). El mejor valor encontrado en 30 ejecuciones independientes es de 8290.6 kg

con una restricción de 5.9 kg. Los resultados promedios de media y varianza se representan en las figuras 4.302 y 4.303.

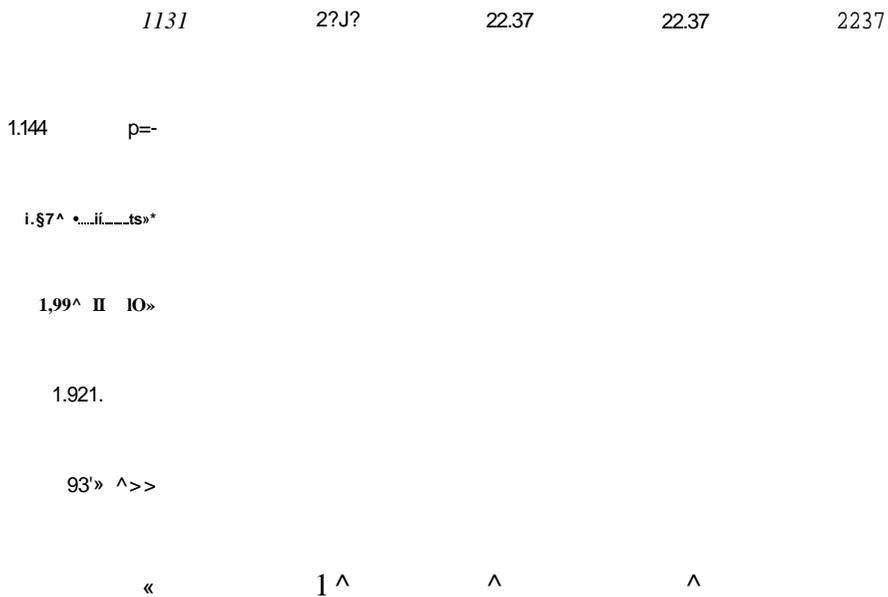


Figura 4.301. Nuevo Caso Test Y2.

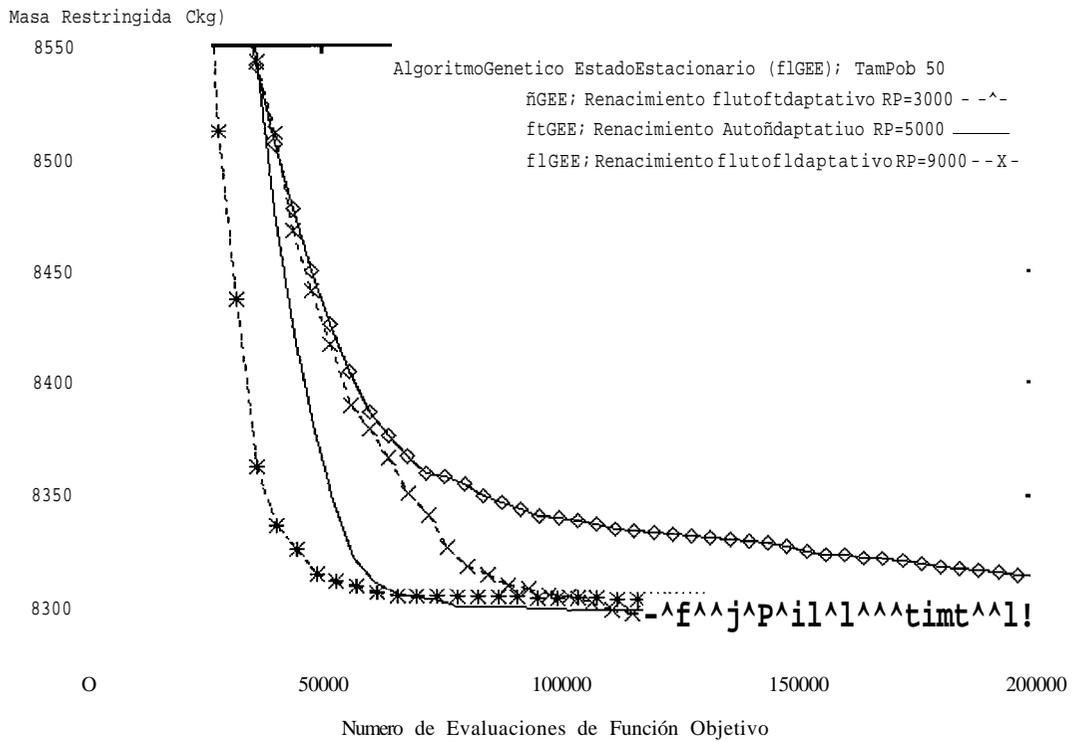


Figura 4.302. Influencia del parámetro RP. Promedio de 30 ejecuciones independientes. Caso Test Y2.

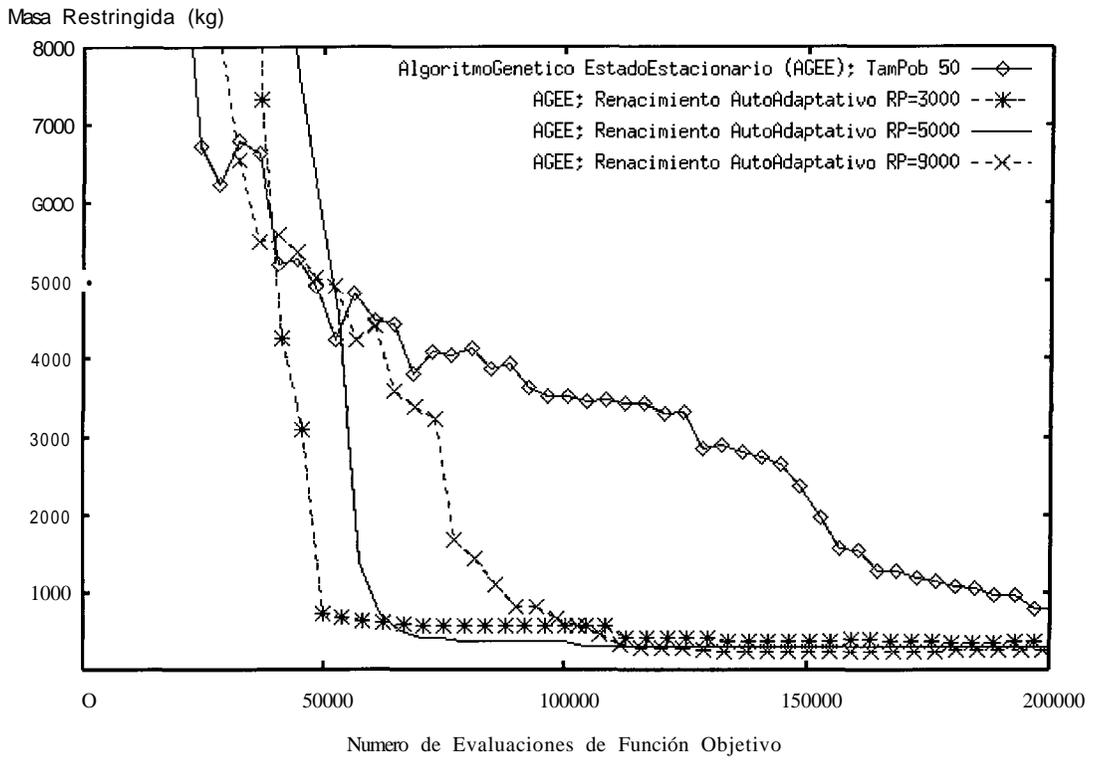


Figura 4.303. Influencia del parámetro RP. Varianza de 30 ejecuciones independientes. Caso Test Y2.

El segundo nuevo caso test analizado tiene 35 barras y 24 nudos, y está representado en la figura 4.304, considerando que dimensiones, cargas y características son las mismas que el caso test Y. El mejor valor encontrado en 30 ejecuciones independientes es de 6405.9 kg con una restricción de 9.7 kg. Los resultados promedios de media y varianza están representados en las figuras 4.305 y 4.306.

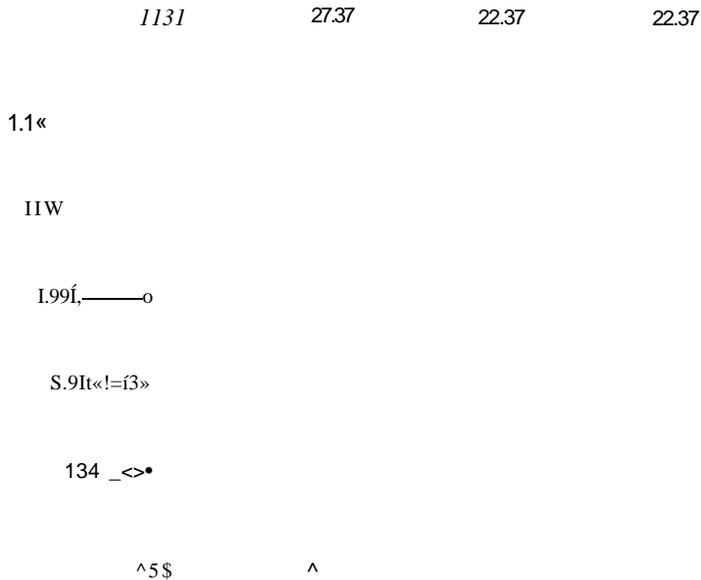


Figura 4.304. Nuevo Caso Test Y3.

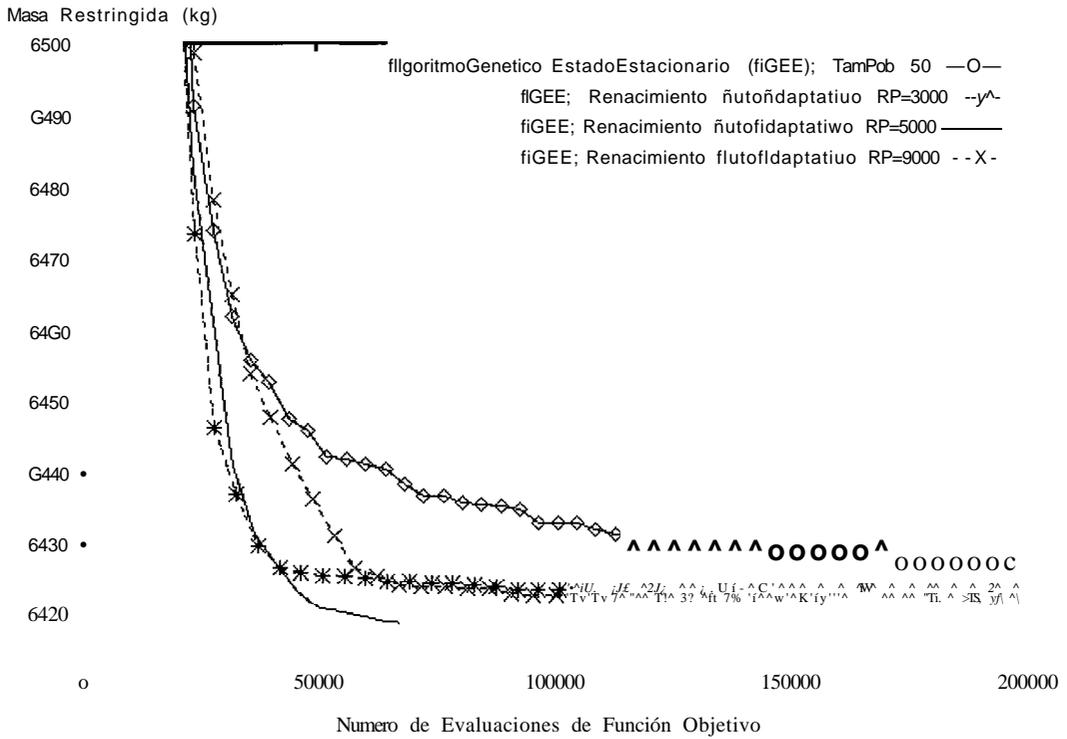


Figura 4.305. Influencia del parámetro RP. Promedio de 30 ejecuciones independientes. Caso Test Y3.

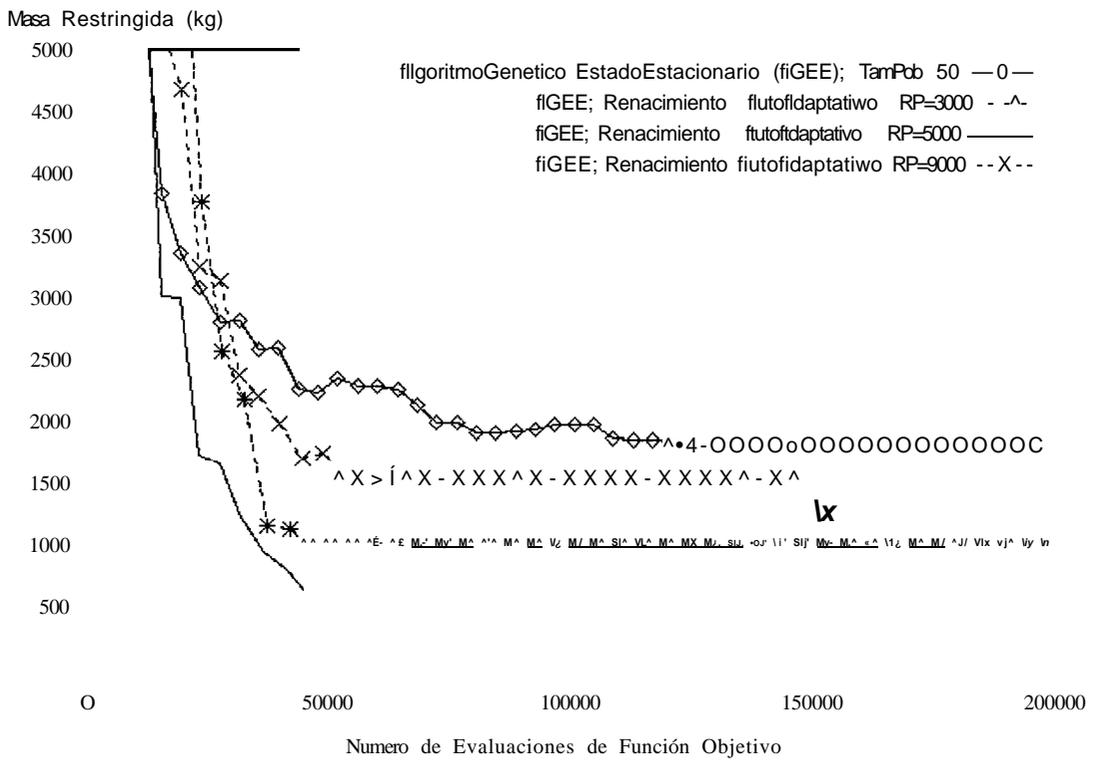


Figura 4.306. Influencia del parámetro RP. Varianza de 30 ejecuciones independientes. Caso Test Y3.

Ambos casos test incluidos aquí refuerzan el hecho de que el renacimiento autoadaptativo puede obtener no sólo más rápidamente soluciones optimizadas de menor masa (en términos de valores

promedio), sino también soluciones más robustas (en términos de menor varianza en las figuras) como puede verse en las figuras 4.302, 4.303 (correspondientes al primer nuevo caso test), y 4.305, 4.306 (correspondientes al segundo nuevo caso test).

4.11.3 Comparando Renacimiento frente a Renacimiento AutoAdaptativo

Comparado frente al renacimiento, un operador de renacimiento autoadaptativo, que aplica el renacimiento consecutivamente, mejora el comportamiento de convergencia del algoritmo, como puede observarse en las figuras 4.307 a 4.312. Éstas reflejan el promedio sobre 30 ejecuciones independientes del modelo real del caso test Y. Valores diferentes del parámetro de saturación RP son consideradas (define el criterio de saturación para la aplicación del renacimiento).

Las figuras 4.308 a 4.312 muestran cómo el renacimiento autoadaptativo provoca una convergencia más rápida y precisa que la aplicación del renacimiento y que el algoritmo estándar.

El procedimiento de la reducción de espacio es aplicado tras la saturación del algoritmo. El operador renacimiento es diseñado como una herramienta que permite una búsqueda posterior tras la saturación del algoritmo, precisamente, para evitar la convergencia prematura y la pérdida del óptimo global. Aplica un nuevo proceso de búsqueda tras la saturación del algoritmo evolutivo, manteniendo el mejor individuo encontrado hasta el momento y el resto de la población es reinicializado (incrementando la diversidad en la población: énfasis en la exploración), pero reduciendo el espacio de búsqueda alrededor de las variables del mejor individuo (énfasis en la explotación). Es posible, no obstante, tras su aplicación y tras alcanzar la saturación de la población, que el óptimo global aún no haya sido encontrado. El renacimiento autoadaptativo implementa aplicaciones sucesivas del renacimiento para reducir este efecto.

Por tanto, el caer en una región de mínimo local es consecuencia de una mala convergencia del algoritmo evolutivo. Puede ser provocada por una aplicación prematura del renacimiento si el parámetro RP es demasiado corto y no permite una convergencia completa. El operador renacimiento autoadaptativo, mediante sucesivas aplicaciones del renacimiento contribuye a reducir este posible efecto (como puede verse por ejemplo en las figuras 4.307 y 4.308) donde el renacimiento conduce a valores subóptimos (sin embargo una elección adecuada del parámetro RP palia este efecto perjudicial).

Capítulo 4. Contribuciones

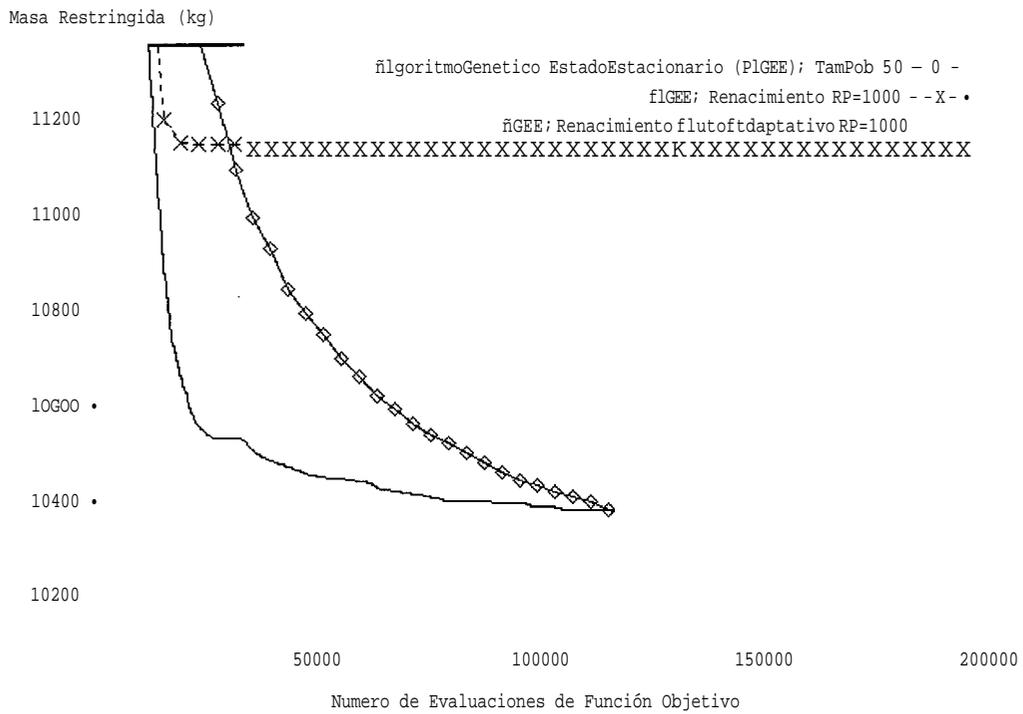


Figura 4.307. Renacimiento frente a Renacimiento AutoAdaptativo. Promedio de 30 ejecuciones independientes.
RP=1000

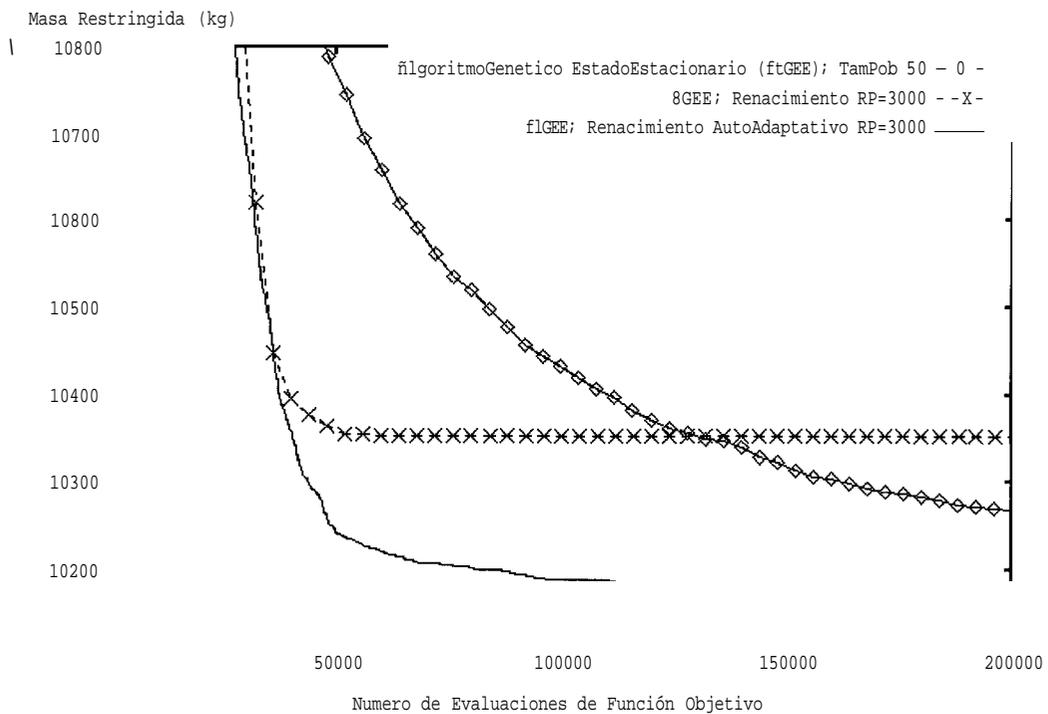


Figura 4.308. Renacimiento frente a Renacimiento AutoAdaptativo. Promedio de 30 ejecuciones independientes.
RP=3000

Capítulo 4. Contribuciones

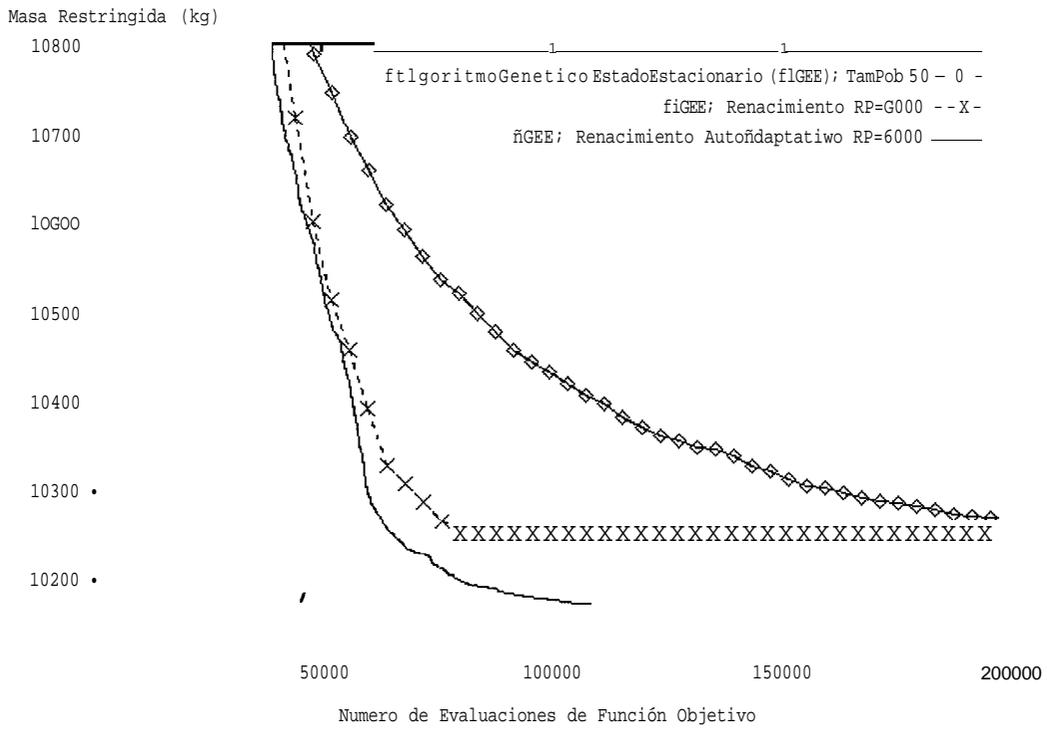


Figura 4.309. Renacimiento frente a Renacimiento AutoAdaptativo. Promedio de 30 ejecuciones independientes. RP=6000

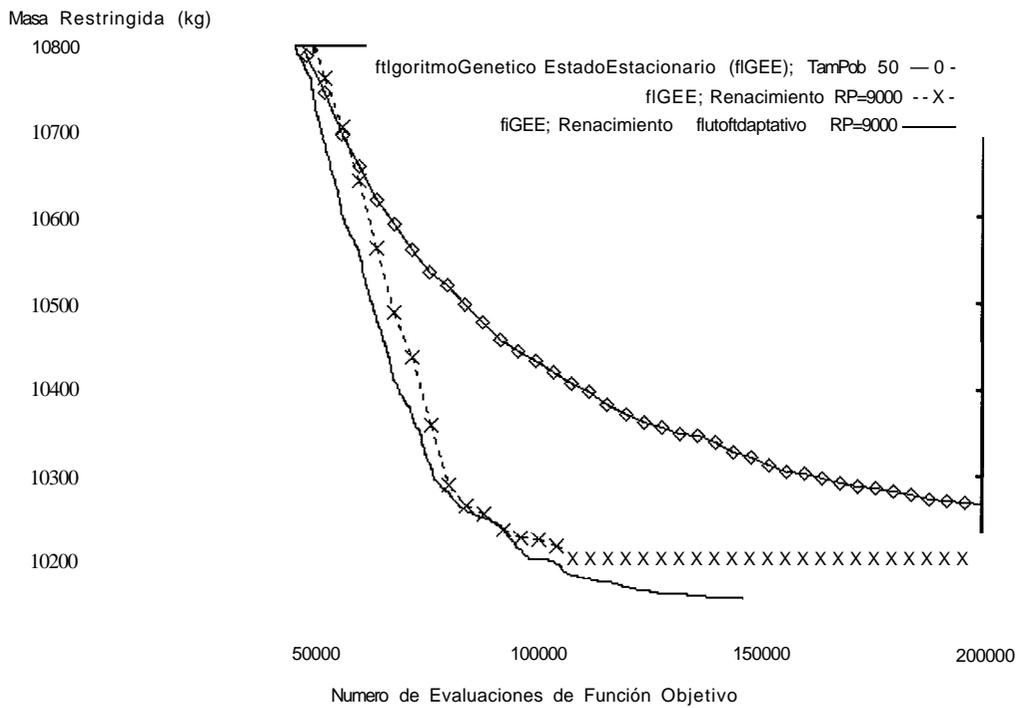


Figura 4.310. Renacimiento frente a Renacimiento AutoAdaptativo. Promedio de 30 ejecuciones independientes. RP=9000

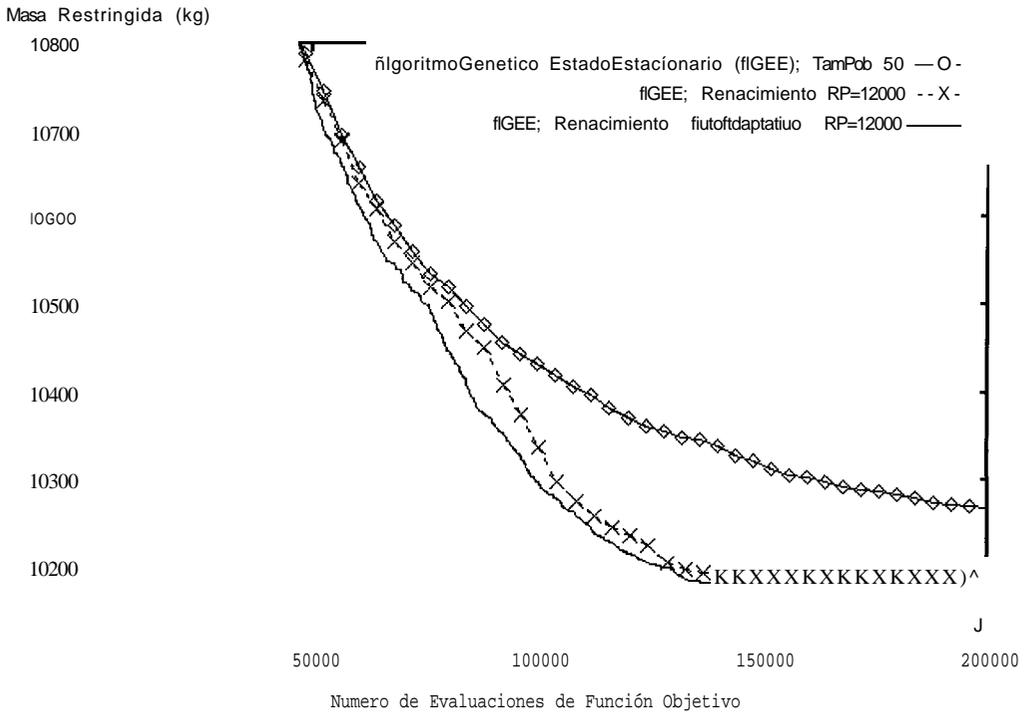


Figura 4.311. Renacimiento frente a Renacimiento AutoAdaptativo. Promedio de 30 ejecuciones independientes. RP=12000

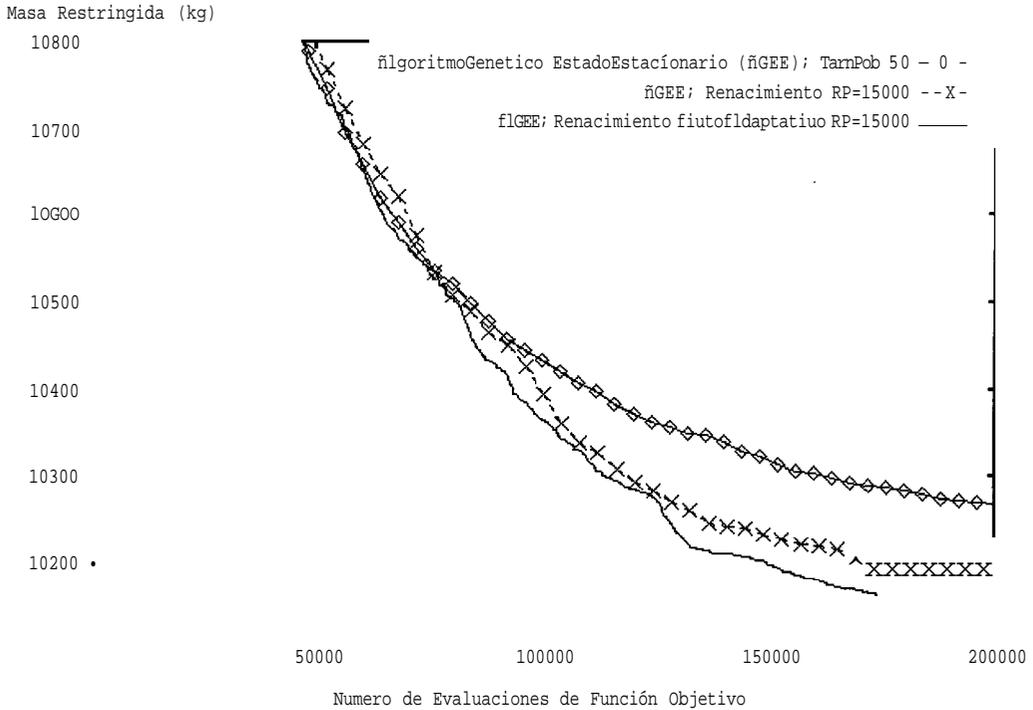


Figura 4.312. Renacimiento frente a Renacimiento AutoAdaptativo. Promedio de 30 ejecuciones independientes. RP=15000

Mediante el renacimiento autoadaptativo se pueden alcanzar soluciones más precisas y más rápidamente para la resolución de la optimización del problema de masa restringida.

4.11.4 Introducción de soluciones de alta calidad monocriterio en la población inicial multicriterio.

La introducción de soluciones de alta calidad en la población inicial, es un método que en optimización evolutiva monocriterio ha sido frecuentemente utilizado. Por ejemplo, la reinicialización de la población, como operador en sí mismo, o formando parte de un tipo de algoritmo evolutivo, como puede ser el caso del micro-algoritmo genético, se basa en ese principio. Se conserva el mejor individuo obtenido hasta el momento, y se crea una nueva población de forma aleatoria donde se incluye esa mejor solución. También el propio operador renacimiento o el renacimiento autoadaptativo descrito con anterioridad en esta sección, se valen de este mismo concepto. Un operador equivalente en optimización evolutiva multicriterio conduciría al mantenimiento de todo el frente de soluciones no-dominadas (las mejores soluciones encontradas en términos del criterio de Pareto) en una nueva población con individuos aleatorios en número restante para completar la población (cuando ello fuera posible, puesto que en muchos casos sólo el frente de soluciones no-dominadas alcanzaría por sí mismo a completar el tamaño poblacional totalmente).

A continuación se muestra un procedimiento inspirado en esa reinicialización, que consigue mejorar el rendimiento de la optimización multicriterio. Se propone la inserción en la población inicial de una solución de alta calidad en términos de uno de los objetivos a optimizar, y a partir de ella, iniciar la optimización multicriterio [4.3][4.19][4.20]. Cuanto mejor sea la solución insertada, mayor ventaja adoptará el proceso multicriterio: así pues, este proceso se beneficiará de soluciones de alta calidad obtenidas, por ejemplo a partir del renacimiento autoadaptativo introducido en este capítulo. Este operador ha sido mostrado en las anteriores secciones como ventajoso en su aplicación en el algoritmo e incluso ventajoso respecto al renacimiento clásico.

Anexo. Otras Tablas y Figuras

Se han incluido en este anexo aquellas tablas y figuras que no se consideran imprescindibles a la hora de realizar la exposición de la tesis, y que por su extensión para más claridad del desarrollo se han agrupado aquí.

La numeración de las mismas es concordante con el resto del trabajo, e irán numeradas de acuerdo con su ordenación natural en el mismo.

Tabla 4.5. Resultados Comparativos Código Gray / Binario en el NSGA con reemplazamiento generacional

Tamaño Población	Código	Probabilidad Mutación	Nº alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	6	6250.0	2586.5
50	Binary	1.5%	22	7309.1	4903.6
50	Binary	3%	27	8113.0	5547.2
50	Binary	6%	8	8275.0	5481.8
50	Gray	0.8%	13	9800.0	4002.6
50	Gray	1.5%	23	7821.7	4320.8
50	Gray	3%	30	6166.7	4166.7
50	Gray	6%	3	4750.0	4213.3
100	Binary	0.8%	12	6825.0	2998.4
100	Binary	1.5%	18	8000.0	5598.3
100	Binary	3%	25	6812.0	3973.4
100	Binary	6%	9	8600.0	4976.8
100	Gray	0.8%	9	10266.7	4566.3
100	Gray	1.5%	24	7170.8	3761.4
100	Gray	3%	30	7040.0	3932.6
100	Gray	6%	5	12900.0	6085.1
200	Binary	0.8%	14	10685.7	4111.1
200	Binary	1.5%	12	7450.0	3033.3
200	Binary	3%	25	8520.0	3621.3
200	Binary	6%	12	9900.0	4584.4
200	Gray	0.8%	7	9314.3	4984.4
200	Gray	1.5%	18	11244.4	5124.9
200	Gray	3%	29	9531.0	4897.4
200	Gray	6%	12	12133.3	3186.8

Capítulo 4. Contribuciones

Tabla 4.6. Resultados Comparativos Código Gray / Binario en el NSGAI con reemplazamiento generacional

Tamaño Población	Código	Probabilidad Mutación	N° alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	22	2263.6	3130.8
50	Binary	1.5%	19	1197.4	875.8
50	Binary	3%	22	1284.1	1043.2
50	Binary	6%	24	2981.2	4222.7
50	Gray	0.8%	15	4736.7	4811.6
50	Gray	1.5%	21	4526.2	5593.7
50	Gray	3%	27	3459.3	4192.5
50	Gray	6%	29	2965.5	2937.6
100	Binary	0.8%	21	1795.2	757.5
100	Binary	1.5%	24	1420.8	969.5
100	Binary	3%	24	2033.3	2688.7
100	Binary	6%	26	1615.4	636.5
100	Gray	0.8%	18	3866.7	4409.5
100	Gray	1.5%	26	3542.3	4296.0
100	Gray	3%	25	3512.0	3935.1
100	Gray	6%	30	2380.0	1657.8
200	Binary	0.8%	29	2048.3	539.9
200	Binary	1.5%	28	2321.4	1285.5
200	Binary	3%	27	2266.7	1003.7
200	Binary	6%	28	3485.7	3233.9
200	Gray	0.8%	23	2869.6	2250.1
200	Gray	1.5%	23	2400.0	907.9
200	Gray	3%	26	2453.8	872.8
200	Gray	6%	30	3186.7	2376.5

Tabla 4.7. Resultados Comparativos Código Gray / Binario en el NSGAI con Elitismo Controlado y $r=0.0$ con reemplazamiento generacional

Tamaño Población	Código	Probabilidad Mutación	N° alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	18	2647.2	4092.4
50	Binary	1.5%	20	1605.0	1733.7
50	Binary	3%	24	2810.4	4137.6
50	Binary	6%	28	2692.9	3659.6
50	Gray	0.8%	18	5341.7	5775.8
50	Gray	1.5%	19	4150.0	5767.0
50	Gray	3%	26	6401.9	5323.6
50	Gray	6%	27	2264.8	2065.8
100	Binary	0.8%	24	3266.7	3394.9
100	Binary	1.5%	27	2444.4	2472.3
100	Binary	3%	24	2137.5	1521.7
100	Binary	6%	30	2853.3	2586.1
100	Gray	0.8%	17	3770.6	5578.7
100	Gray	1.5%	24	3037.5	3155.3
100	Gray	3%	26	3461.5	3924.9
100	Gray	6%	25	2464.0	933.3
200	Binary	0.8%	29	4110.3	3707.1
200	Binary	1.5%	29	2820.7	1094.6
200	Binary	3%	29	3386.2	2718.0
200	Binary	6%	29	3213.8	848.4
200	Gray	0.8%	23	4460.9	4749.2
200	Gray	1.5%	26	3830.8	3044.1
200	Gray	3%	29	3206.9	1000.0
200	Gray	6%	30	3626.7	960.2

Capítulo 4. Contribuciones

Tabla 4.8. Resultados Comparativos Código Gray / Binario en el NSGAI con Elitismo Controlado y $r=0.4$ con reemplazamiento generacional

Tamaño Población	Código	Probabilidad Mutación	Nº alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	26	1780.8	2653.7
50	Binary	1.5%	27	3288.9	4640.8
50	Binary	3%	25	1588.0	1452.5
50	Binary	6%	30	2920.0	3279.0
50	Gray	0.8%	29	5213.8	4186.3
50	Gray	1.5%	30	2725.0	2305.2
50	Gray	3%	29	2767.2	2679.2
50	Gray	6%	30	1861.7	813.2
100	Binary	0.8%	28	3278.6	3064.2
100	Binary	1.5%	29	2658.6	2293.4
100	Binary	3%	29	2869.0	2878.2
100	Binary	6%	30	3616.7	4154.1
100	Gray	0.8%	26	5519.2	4773.0
100	Gray	1.5%	30	4623.3	5239.7
100	Gray	3%	30	4130.0	4726.0
100	Gray	6%	29	4031.0	3082.9
200	Binary	0.8%	30	2953.3	1644.1
200	Binary	1.5%	30	4246.7	3932.3
200	Binary	3%	29	3889.6	3046.1
200	Binary	6%	29	3758.6	1809.8
200	Gray	0.8%	28	4900.0	3950.2
200	Gray	1.5%	30	4366.7	3437.1
200	Gray	3%	30	3940.0	2442.5
200	Gray	6%	30	3646.7	1532.5

Tabla 4.9. Resultados Comparativos Código Gray / Binario en el SPEA2 con reemplazamiento generacional

Tamaño Población	Código	Probabilidad Mutación	Nº alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	20	1600.0	1762.0
50	Binary	1.5%	20	967.5	754.0
50	Binary	3%	26	2507.7	3914.2
50	Binary	6%	23	1443.5	1408.3
50	Gray	0.8%	18	3888.9	4889.5
50	Gray	1.5%	17	5544.1	5969.9
50	Gray	3%	23	5513.0	5981.2
50	Gray	6%	25	2966.0	2202.0
100	Binary	0.8%	23	1978.3	1866.2
100	Binary	1.5%	23	2534.8	2426.2
100	Binary	3%	21	1990.5	1364.5
100	Binary	6%	26	2811.5	3196.6
100	Gray	0.8%	18	4066.7	4545.7
100	Gray	1.5%	23	3752.2	4709.2
100	Gray	3%	25	4508.0	5469.1
100	Gray	6%	28	2157.1	1603.7
200	Binary	0.8%	25	2360.0	1134.2
200	Binary	1.5%	29	2282.8	1124.7
200	Binary	3%	28	3464.3	3060.8
200	Binary	6%	29	2675.9	1699.8
200	Gray	0.8%	24	2158.3	608.2
200	Gray	1.5%	24	2425.0	813.0
200	Gray	3%	27	2533.3	975.2
200	Gray	6%	29	3317.2	1597.0

Capítulo 4. Contribuciones

Tabla 4.10. Resultados Comparativos Código Gray / Binario en el DENSEA con reemplazamiento generacional

Tamaño Población	Código	Probabilidad Mutación	N° alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	28	2494.6	3411.2
50	Binary	1.5%	30	1618.3	1713.6
50	Binary	3%	30	2785.0	4579.2
50	Binary	6%	30	1628.3	1580.9
50	Gray	0.8%	30	2021.7	1721.6
50	Gray	1.5%	30	1628.3	1363.2
50	Gray	3%	30	1533.3	998.3
50	Gray	6%	30	1325.0	620.9
100	Binary	0.8%	30	1966.7	1166.9
100	Binary	1.5%	30	1533.3	736.4
100	Binary	3%	30	2153.3	1813.2
100	Binary	6%	30	2130.0	1445.7
100	Gray	0.8%	30	2756.7	1844.9
100	Gray	1.5%	30	2306.7	1626.8
100	Gray	3%	30	2416.7	1297.5
100	Gray	6%	30	2130.0	898.6
200	Binary	0.8%	30	2266.7	662.0
200	Binary	1.5%	30	2346.7	860.9
200	Binary	3%	30	2626.7	877.5
200	Binary	6%	30	2613.3	755.3
200	Gray	0.8%	30	3106.7	1453.9
200	Gray	1.5%	30	2906.7	843.4
200	Gray	3%	30	2786.7	824.5
200	Gray	6%	30	3360.0	930.1

Tabla 4.11. Resultados Comparativos Código Gray / Binario en el DENSEA-NSH con reemplazamiento generacional

Tamaño Población	Código	Probabilidad Mutación	N° alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	29	2474.1	3975.1
50	Binary	1.5%	30	2518.3	3017.8
50	Binary	3%	29	2465.5	3516.9
50	Binary	6%	30	1400.0	863.5
50	Gray	0.8%	30	2470.0	1986.4
50	Gray	1.5%	30	1576.7	900.6
50	Gray	3%	30	1225.0	711.7
50	Gray	6%	30	1383.3	671.9
100	Binary	0.8%	30	1910.0	1103.1
100	Binary	1.5%	30	1513.3	835.4
100	Binary	3%	30	1600.0	735.3
100	Binary	6%	30	1773.3	829.0
100	Gray	0.8%	30	2293.3	1173.6
100	Gray	1.5%	30	1993.3	985.9
100	Gray	3%	30	1720.0	620.4
100	Gray	6%	30	1840.0	649.4
200	Binary	0.8%	30	2160.0	629.1
200	Binary	1.5%	30	2280.0	751.2
200	Binary	3%	30	2353.3	910.2
200	Binary	6%	30	2540.0	746.8
200	Gray	0.8%	30	2560.0	895.0
200	Gray	1.5%	30	3200.0	1138.4
200	Gray	3%	30	2626.7	631.9
200	Gray	6%	30	2986.7	977.0

Capítulo 4. Contribuciones

Tabla 4.12. Resultados Comparativos Código Gray / Binario en el NSGAI con reemplazamiento estado-estacionario

Tamaño Población	Código	Probabilidad Mutación	N° alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	19	1878.3	1787.3
50	Binary	1.5%	16	2088.2	2336.1
50	Binary	3%	23	2286.8	3447.0
50	Binary	6%	23	2380.7	3223.8
50	Gray	0.8%	5	6652.0	7174.8
50	Gray	1.5%	21	8231.0	6809.6
50	Gray	3%	25	3775.2	3328.6
50	Gray	6%	30	2538.0	3147.2
100	Binary	0.8%	20	1508.0	1536.5
100	Binary	1.5%	23	2956.8	4281.5
100	Binary	3%	26	2072.0	2830.2
100	Binary	6%	27	2272.4	3363.8
100	Gray	0.8%	17	4222.0	4891.2
100	Gray	1.5%	21	5907.7	5879.3
100	Gray	3%	26	2777.4	4707.4
100	Gray	6%	29	3319.2	3572.8
200	Binary	0.8%	27	2555.3	2477.3
200	Binary	1.5%	23	2251.6	2997.4
200	Binary	3%	25	2572.4	2898.4
200	Binary	6%	29	3604.1	3926.5
200	Gray	0.8%	24	3401.2	4095.3
200	Gray	1.5%	22	2886.5	3530.6
200	Gray	3%	25	3661.2	3847.1
200	Gray	6%	28	2924.1	2620.3

Tabla 4.13. Resultados Comparativos Código Gray / Binario en el NSGAI con Elitismo Controlado y $r=0.0$ con reemplazamiento estado estacionario

Tamaño Población	Código	Probabilidad Mutación	N° alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	15	1614.7	1645.1
50	Binary	1.5%	18	2769.8	3433.8
50	Binary	3%	24	3148.7	4890.8
50	Binary	6%	25	2276.0	2770.2
50	Gray	0.8%	17	4464.9	5707.4
50	Gray	1.5%	21	5720.6	5563.1
50	Gray	3%	27	3788.3	3979.9
50	Gray	6%	30	3262.0	3038.1
100	Binary	0.8%	23	2739.4	3942.2
100	Binary	1.5%	26	2228.2	2304.7
100	Binary	3%	23	2193.3	2292.3
100	Binary	6%	28	1744.9	873.8
100	Gray	0.8%	14	5384.9	6772.0
100	Gray	1.5%	24	4786.2	5493.8
100	Gray	3%	28	3736.3	4630.7
100	Gray	6%	30	2839.3	2580.9
200	Binary	0.8%	27	2687.9	2885.5
200	Binary	1.5%	25	3187.6	1989.6
200	Binary	3%	28	2562.0	1185.9
200	Binary	6%	28	2692.0	1025.7
200	Gray	0.8%	21	4364.9	4551.7
200	Gray	1.5%	28	3571.3	3661.4
200	Gray	3%	30	3016.7	2406.8
200	1 Gray	6%	29	4025.4	3446.4

Capítulo 4. Contribuciones

Tabla 4.14. Resultados Comparativos Código Gray / Binario en el NSGAI con Elitismo Controlado y $r=0.4$ con reemplazamiento estado-estacionario

Tamaño Población	Código	Probabilidad Mutación	N° alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	21	2813.0	3714.2
50	Binary	1.5%	26	3526.6	4769.2
50	Binary	3%	26	3609.7	5242.7
50	Binary	6%	30	2972.0	3168.7
50	Gray	0.8%	27	7321.6	5874.9
50	Gray	1.5%	30	4544.7	2994.6
50	Gray	3%	30	3080.7	2914.0
50	Gray	6%	30	2386.0	2305.6
100	Binary	0.8%	27	3113.9	4147.4
100	Binary	1.5%	24	3990.3	5755.0
100	Binary	3%	30	3189.3	3756.9
100	Binary	6%	29	2489.6	2850.2
100	Gray	0.8%	30	4195.3	4299.2
100	Gray	1.5%	30	4138.7	3560.2
100	Gray	3%	30	3768.0	4008.2
100	Gray	6%	30	2810.7	2022.7
200	Binary	0.8%	28	2827.7	2431.9
200	Binary	1.5%	28	4029.1	5040.1
200	Binary	3%	29	3128.2	2754.2
200	Binary	6%	30	3391.3	3537.0
200	Gray	0.8%	24	3747.0	2638.6
200	Gray	1.5%	29	3877.2	3655.7
200	Gray	3%	29	3101.3	1960.1
200	Gray	6%	30	3129.3	1324.9

Tabla 4.15. Resultados Comparativos Código Gray / Binario en el SPEA2 con reemplazamiento estado-estacionario

Tamaño Población	Código	Probabilidad Mutación	N° alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	16	1285.7	2374.1
50	Binary	1.5%	18	1468.7	2227.4
50	Binary	3%	19	3106.7	4492.2
50	Binary	6%	21	2605.3	3673.9
50	Gray	0.8%	13	8687.4	6105.7
50	Gray	1.5%	15	4657.3	4474.7
50	Gray	3%	26	5202.8	5981.9
50	Gray	6%	28	4561.3	4805.2
100	Binary	0.8%	19	964.1	352.8
100	Binary	1.5%	20	1026.0	826.9
100	Binary	3%	26	2632.8	4093.6
100	Binary	6%	25	1756.4	1782.7
100	Gray	0.8%	13	4525.1	5371.1
100	Gray	1.5%	17	4737.3	5478.0
100	Gray	3%	26	4180.5	5099.8
100	Gray	6%	26	4192.0	4703.6
200	Binary	0.8%	23	2486.3	2438.2
200	Binary	1.5%	26	2598.9	3410.0
200	Binary	3%	27	2933.1	3182.7
200	Binary	6%	30	3052.7	3420.0
200	Gray	0.8%	19	3139.9	4359.7
200	Gray	1.5%	21	3963.0	5296.9
200	Gray	3%	27	3731.6	4188.0
200	Gray	6%	30	2298.0	1733.3

Capítulo 4. Contribuciones

Tabla 4.16. Resultados Comparativos Código Gray / Binario en el DENSEA con reemplazamiento estado-estacionario

Tamaño Población	Código	Probabilidad Mutación	N° alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	27	2617.2	3366.8
50	Binary	1.5%	29	2263.0	2444.7
50	Binary	3%	30	1929.3	2179.5
50	Binary	6%	30	1214.7	930.1
50	Gray	0.8%	30	2262.0	2342.4
50	Gray	1.5%	30	1622.7	1330.6
50	Gray	3%	30	1568.7	899.9
50	Gray	6%	30	1672.7	1048.4
100	Binary	0.8%	30	1566.0	1034.1
100	Binary	1.5%	30	1965.3	1399.9
100	Binary	3%	30	1520.7	1206.0
100	Binary	6%	30	1625.3	806.7
100	Gray	0.8%	30	2467.3	1888.0
100	Gray	1.5%	30	2000.0	1289.5
100	Gray	3%	30	2096.0	1061.3
100	Gray	6%	30	2010.0	865.4
200	Binary	0.8%	30	1874.7	507.6
200	Binary	1.5%	30	1854.7	609.8
200	Binary	3%	30	2237.3	924.8
200	Binary	6%	30	2602.7	965.1
200	Gray	0.8%	30	2352.7	1251.8
200	Gray	1.5%	30	2382.7	929.0
200	Gray	3%	30	2502.0	1298.1
200	Gray	6%	30	2666.0	1327.4

Tabla 4.17. Resultados Comparativos Código Gray / Binario en el DENSEA-NSH con reemplazamiento estado-estacionario

Tamaño Población	Código	Probabilidad Mutación	N° alcanzado frente óptimo	Generación media en alcanzarlo	Desviación Típica
50	Binary	0.8%	29	4149.2	4667.5
50	Binary	1.5%	30	3338.7	4164.9
50	Binary	3%	30	2054.7	2220.2
50	Binary	6%	30	2040.0	1671.9
50	Gray	0.8%	30	3144.0	2648.4
50	Gray	1.5%	30	1840.7	1736.6
50	Gray	3%	30	1496.7	1119.1
50	Gray	6%	30	1300.7	686.9
100	Binary	0.8%	30	1108.0	486.5
100	Binary	1.5%	30	1461.3	909.8
100	Binary	3%	30	1930.7	1553.2
100	Binary	6%	30	1653.3	946.2
100	Gray	0.8%	30	1665.3	867.5
100	Gray	1.5%	30	1709.3	930.6
100	Gray	3%	30	1807.3	1098.3
100	Gray	6%	30	1856.0	666.6
200	Binary	0.8%	30	1932.7	905.1
200	Binary	1.5%	30	2000.7	741.1
200	Binary	3%	30	2026.0	815.9
200	Binary	6%	30	2502.0	1117.2
200	Gray	0.8%	30	2853.3	1704.0
200	Gray	1.5%	30	2732.0	1404.4
200	Gray	3%	30	2690.7	1143.8
200	Gray	6%	30	2728.7	991.6

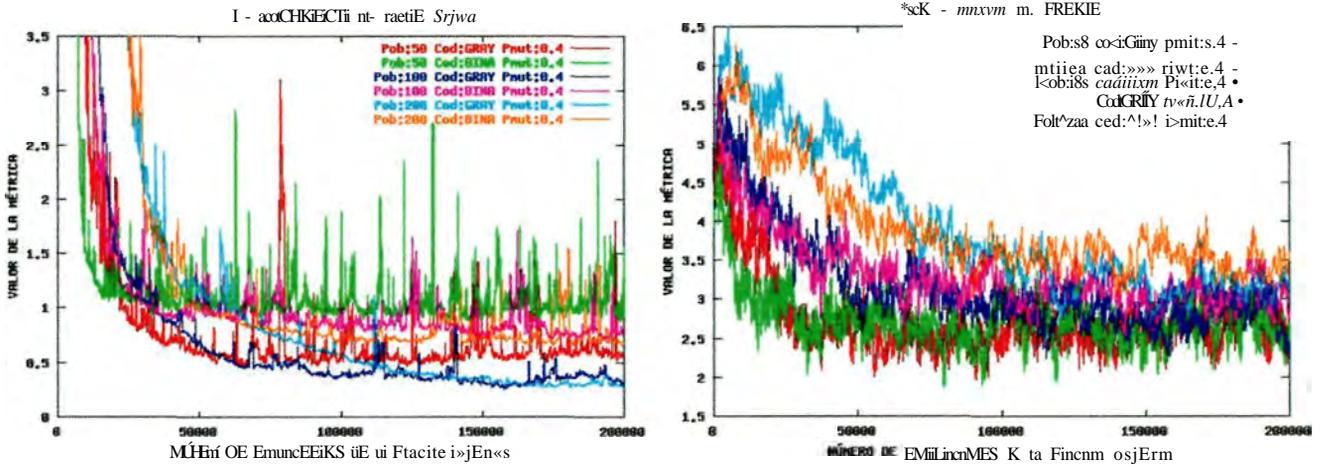


Figura 4.12. Comparativa Código Gray / Binario con NSGA, Pmut 0.4% y reemplazamiento Generacional en Caso Y

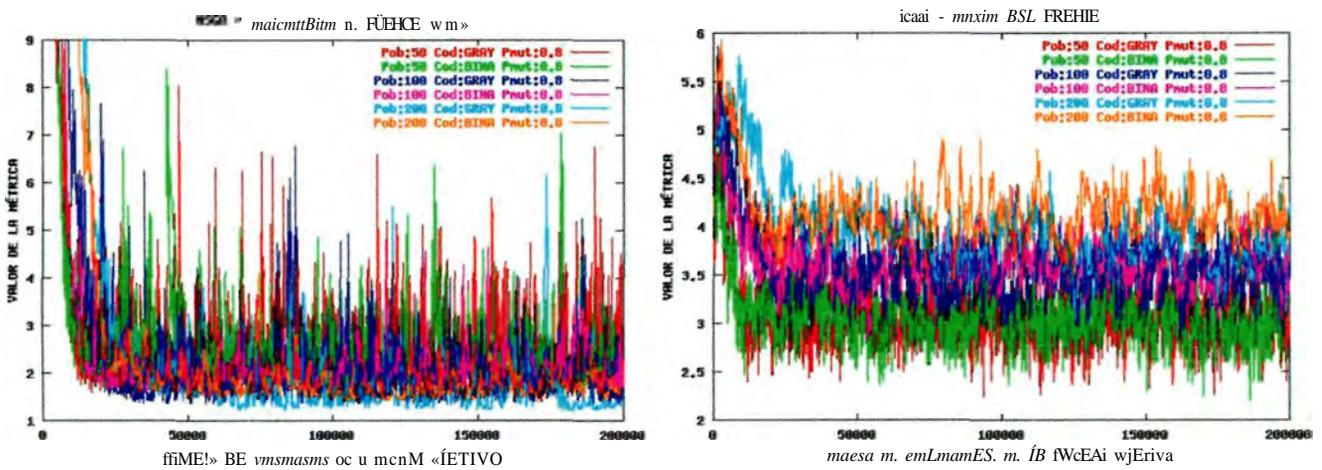


Figura 4.13. Comparativa Código Gray / Binario con NSGA, Pmut 0.8% y reemplazamiento Generacional en Caso Y

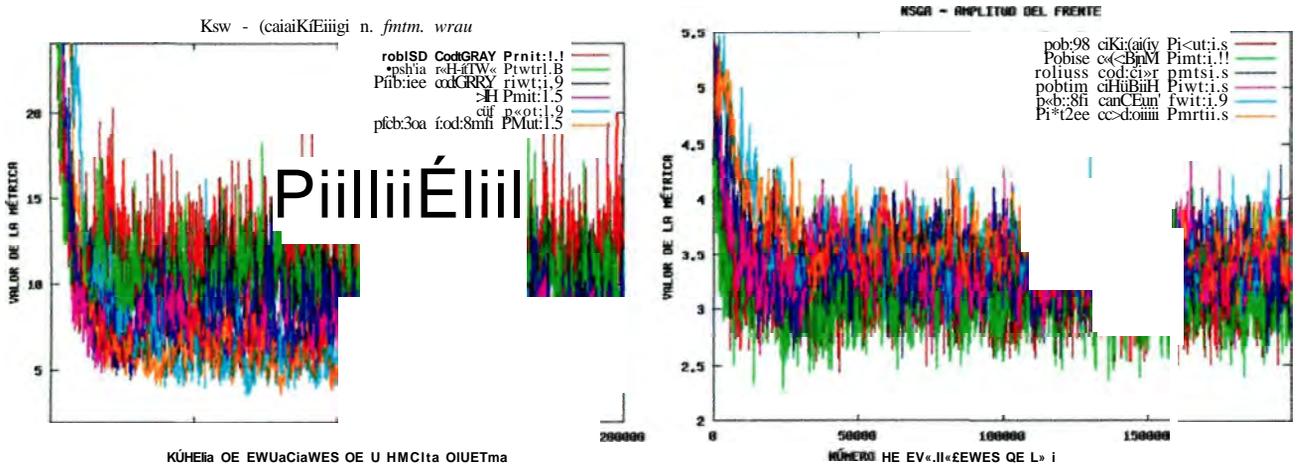


Figura 4.14. Comparativa Código Gray / Binario con NSGA, Pmut 1.5% y reemplazamiento Generacional en Caso Y

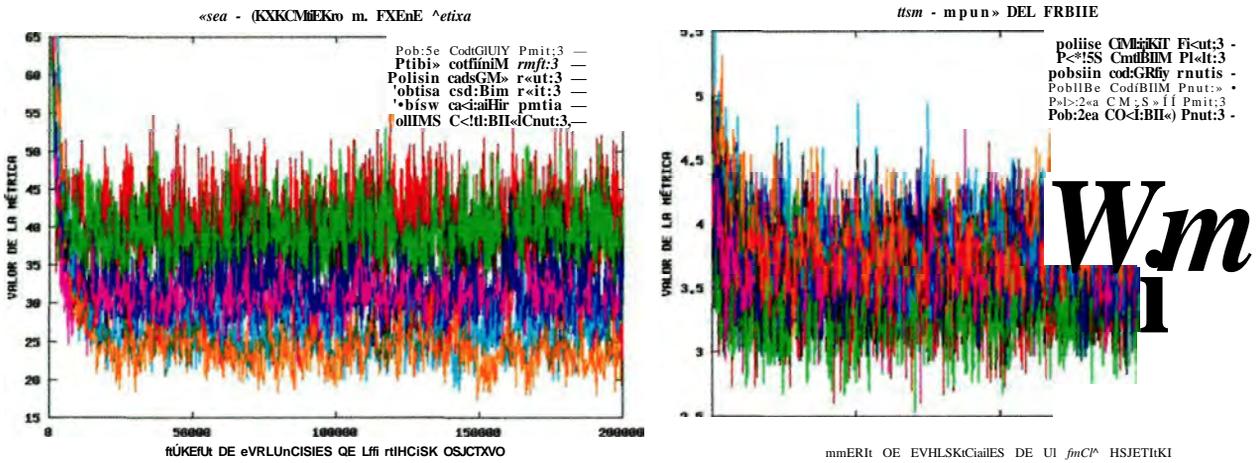


Figura 4.15. Comparativa Código Gray / Binario con NSGA, Pmut 3.0% y reemplazamiento Generacional en Caso Y

Caso Y) BINARIO / GRAY : NSGA-II-ElconOO Mutación 0.4%, 0.8%, 1.5%, 3%

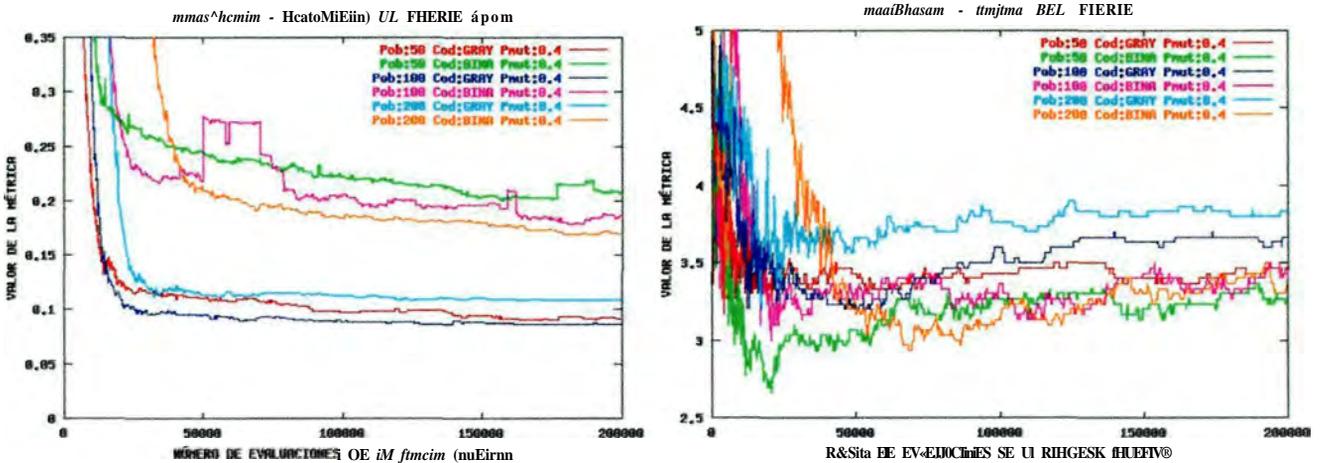


Figura 4.20. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con $r=0.0$, Pmut 0.4% y reemplazamiento Generacional en Caso Y

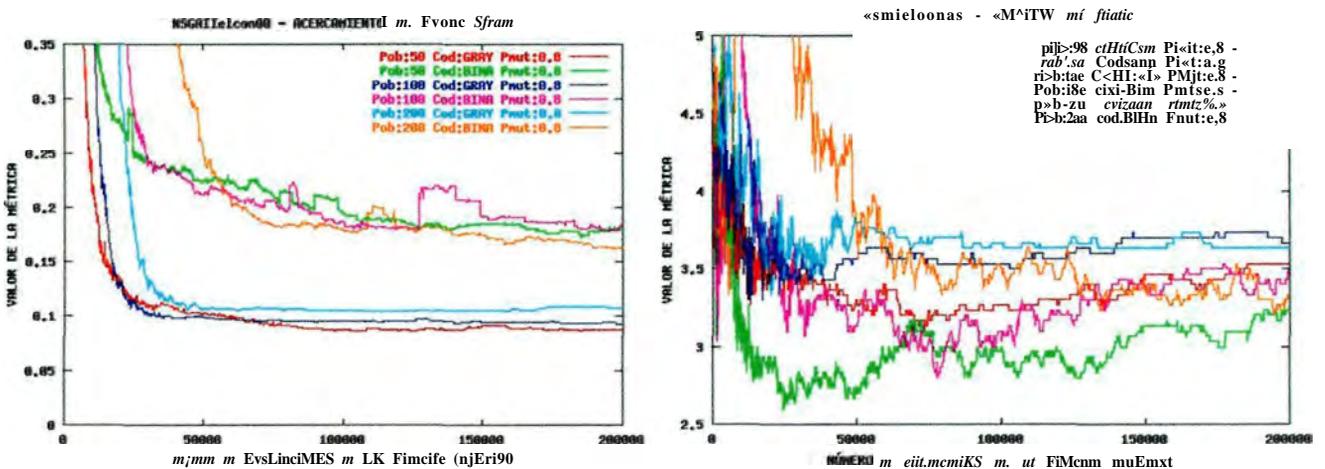


Figura 4.21. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con $r=0.0$, Pmut 0.8% y reemplazamiento Generacional en Caso Y

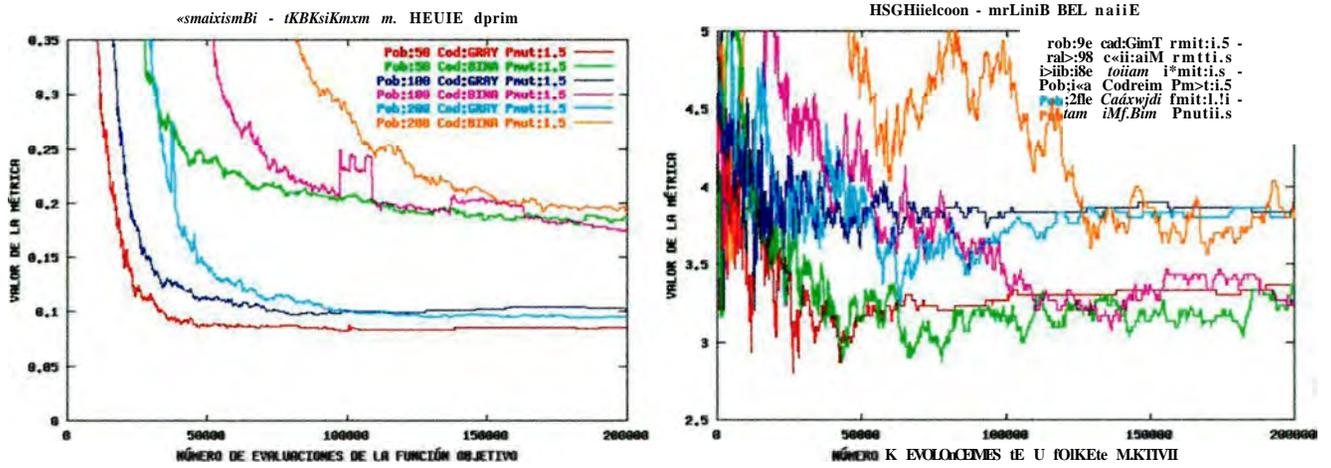


Figura 4.22. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con $r=0.0$, Pmut 1.5% y reemplazamiento Generacional en Caso Y

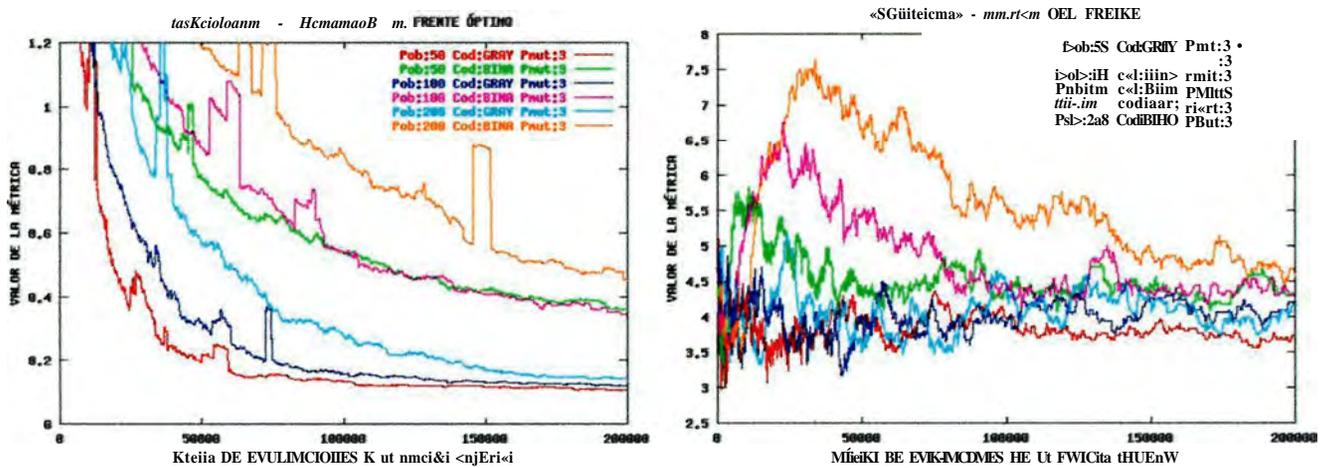


Figura 4.23. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con $r=0.0$, Pmut 3.0% y reemplazamiento Generacional en Caso Y

Caso Y) BINARIO / GRAY : NSGA-II-Elcon04 Mutación 0.4%, 0.8%, 1.5%, 3%

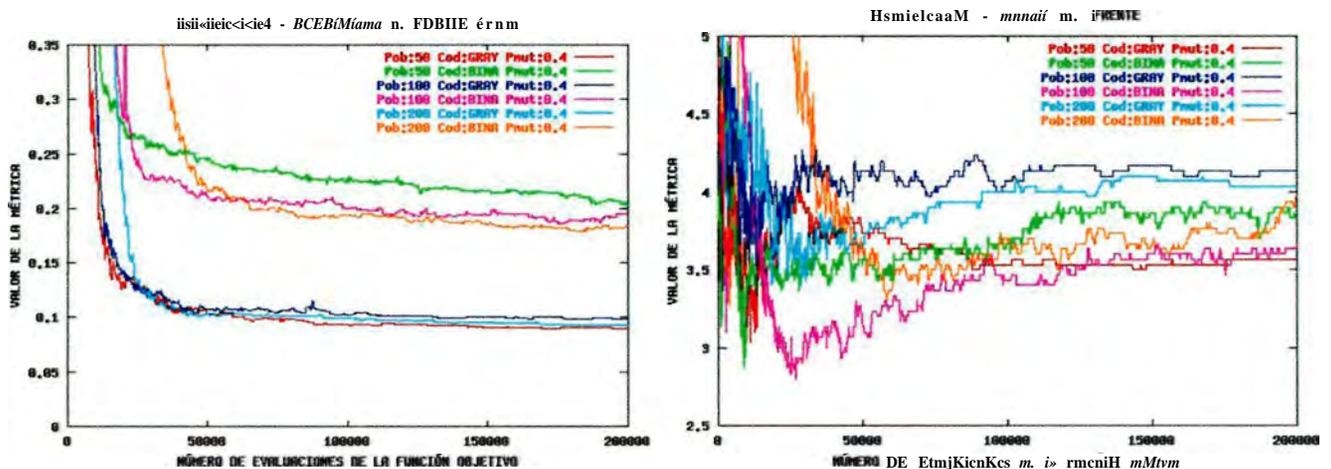


Figura 4.24. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con $r=0.4$, Pmut 0.4% y reemplazamiento Generacional en Caso Y

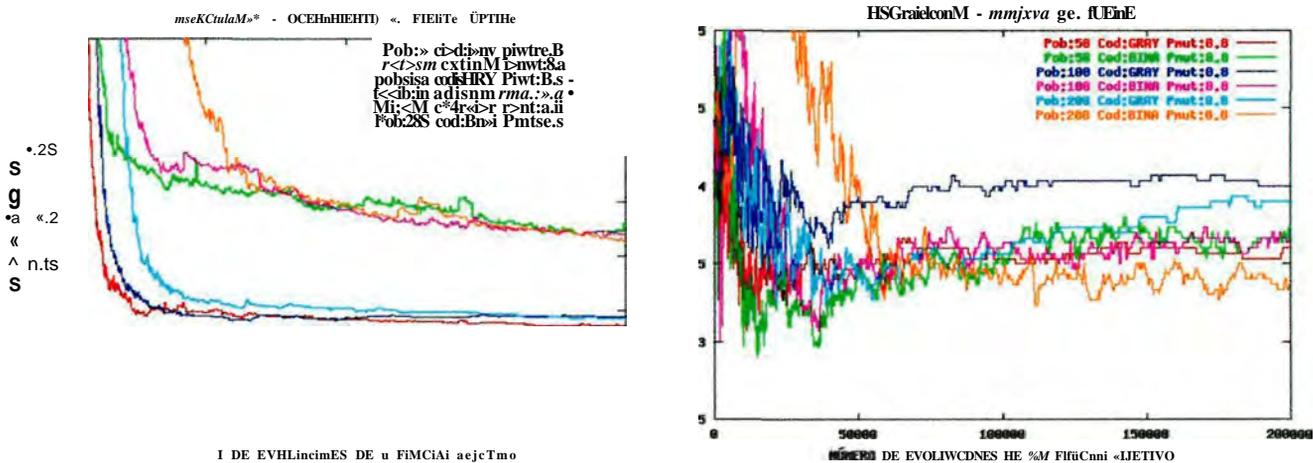


Figura 4.25. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con $r=0.4$, Pmut 0.8% y reemplazamiento Generacional en Caso Y

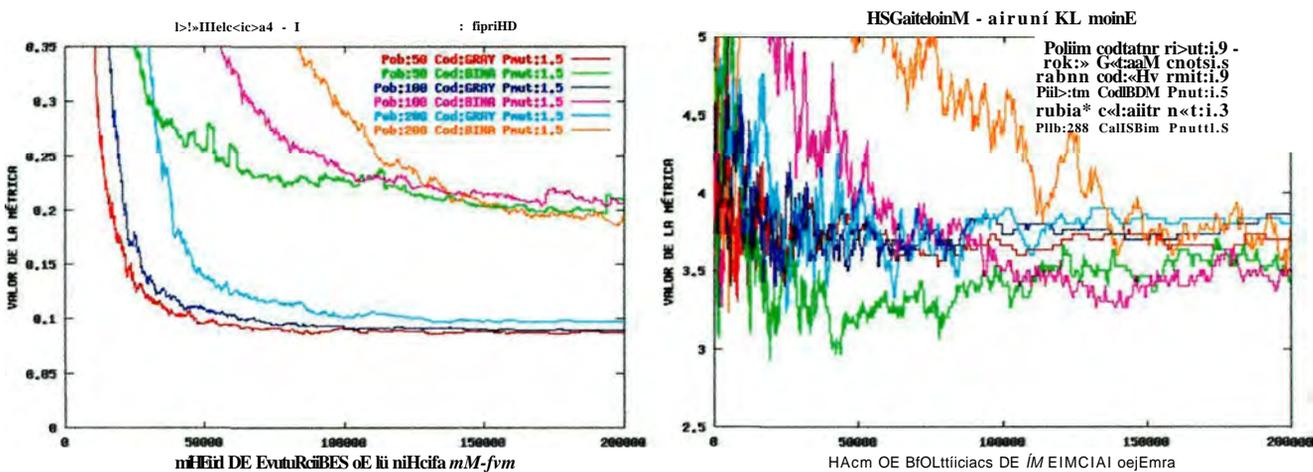


Figura 4.26. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con $r=0.4$, Pmut 1.5% y reemplazamiento Generacional en Caso Y

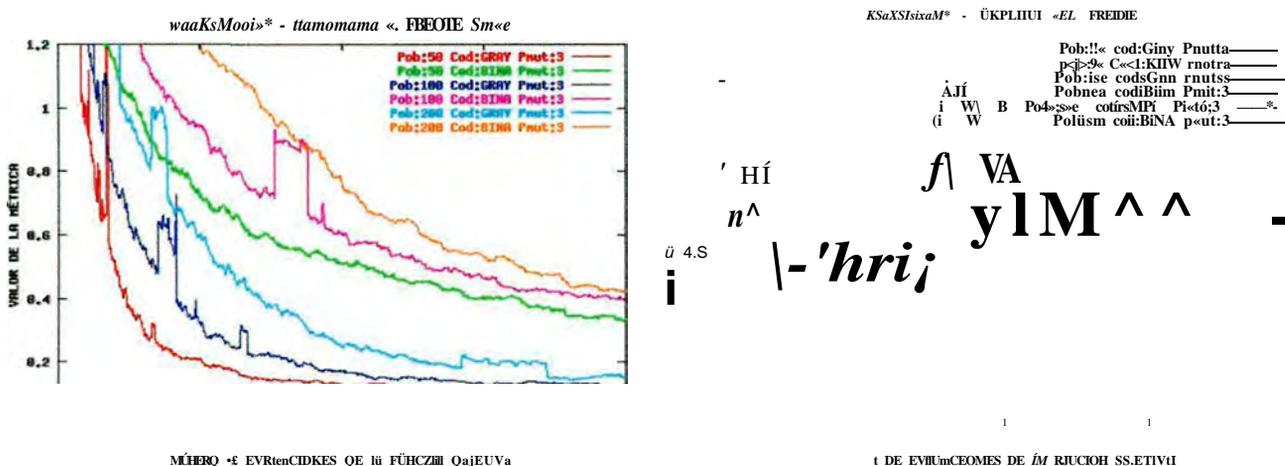


Figura 4.27. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con $r=0.4$, Pmut 3.0% y reemplazamiento Generacional en Caso Y

Caso Y) BINARIO / GRAY : Spea2 Mutación 0.4%, 0.8%, 1.5%, 3%

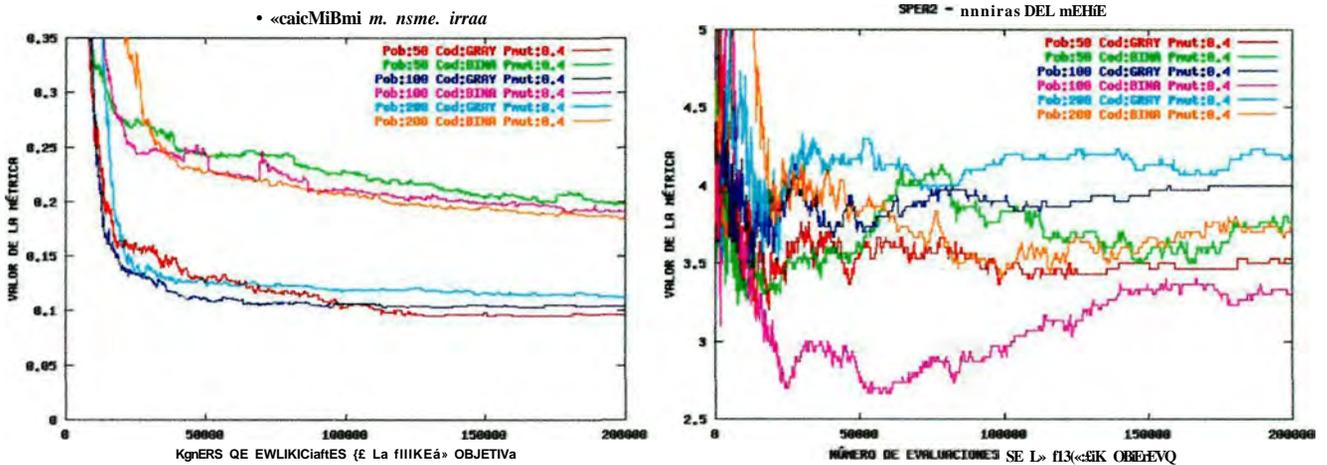


Figura 4.28. Comparativa Código Gray / Binario con SPEA2, Pmut 0.4% y reemplazamiento Generacional en Caso Y

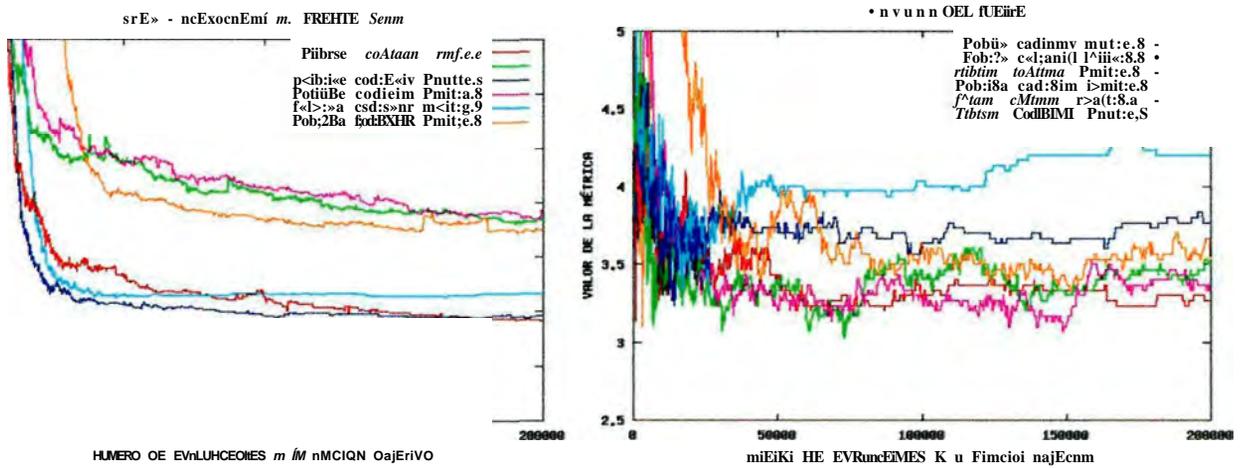


Figura 4.29. Comparativa Código Gray / Binario con SPEA2, Pmut 0.8% y reemplazamiento Generacional en Caso Y

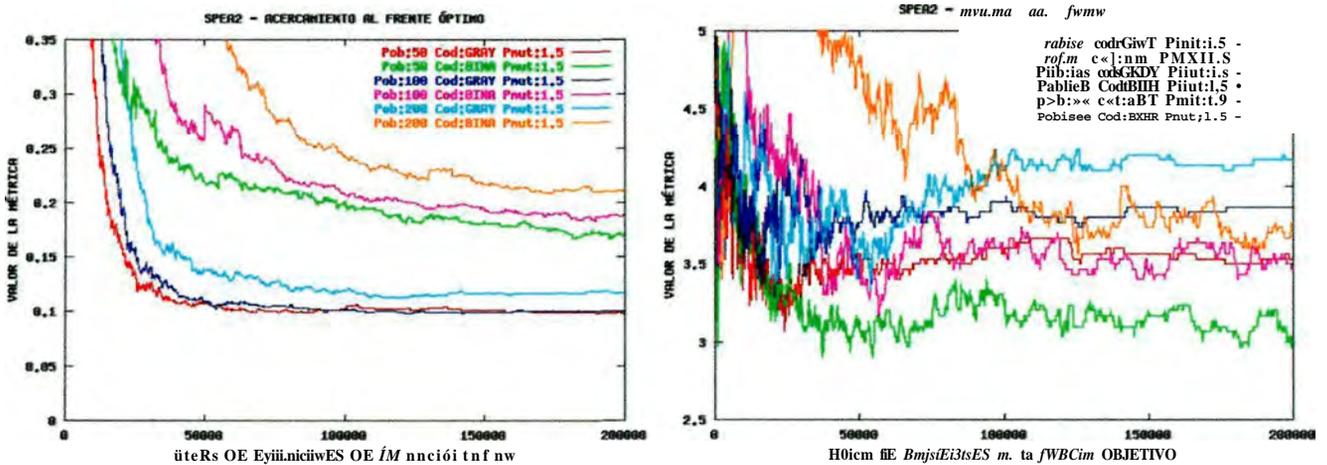


Figura 4.30. Comparativa Código Gray / Binario con SPEA2, Pmut 1.5% y reemplazamiento Generacional en Caso Y

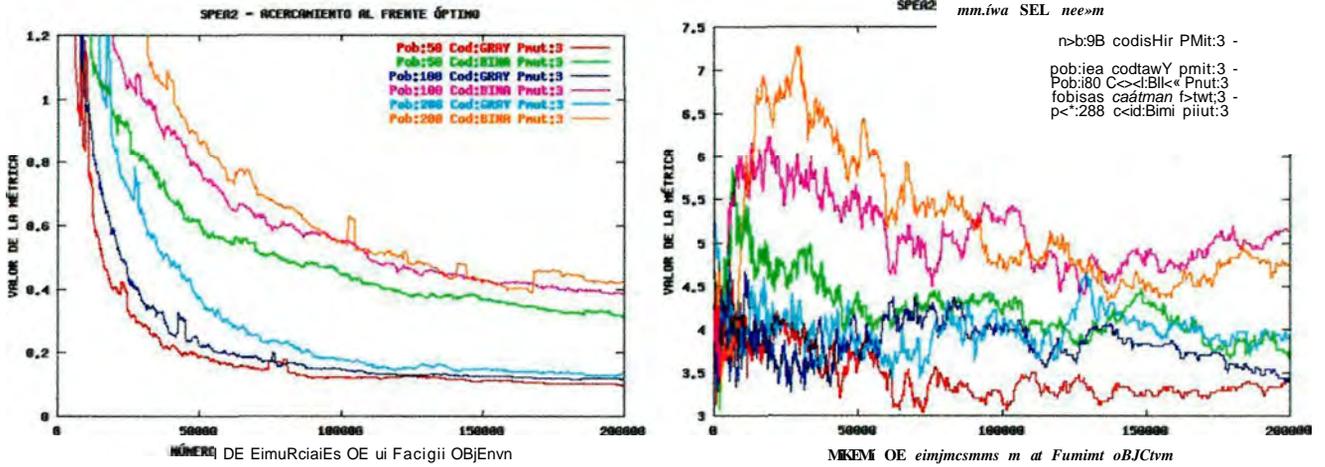


Figura 4.31. Comparativa Código Gray / Binario con SPEA2, Pmut 3.0% y reemplazamiento Generacional en Caso Y

Caso Y) BINARIO / GRAY : DENSEA Mutación 0.4%, 0.8%, 1.5%, 3%

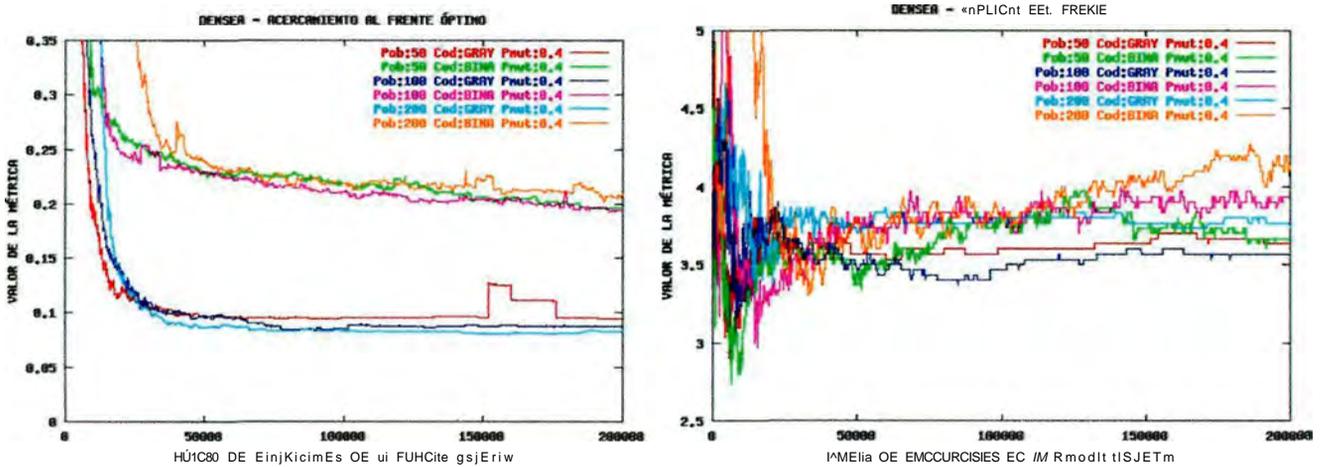


Figura 4.32. Comparativa Código Gray / Binario con DENSEA, Pmut 0.4% y reemplazamiento Generacional en Caso Y

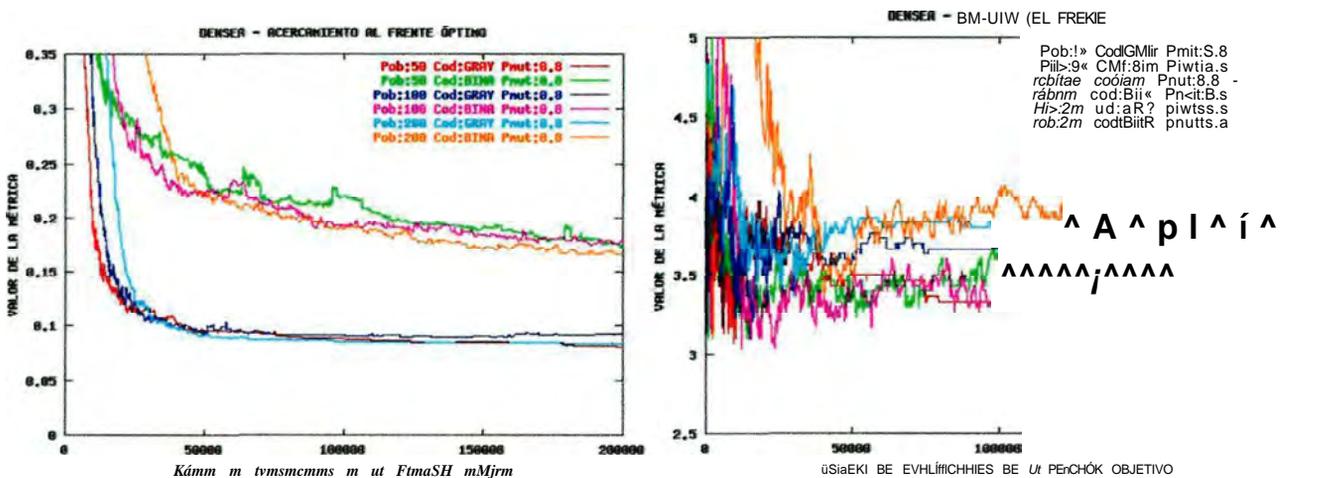


Figura 4.33. Comparativa Código Gray / Binario con DENSEA, Pmut 0.8% y reemplazamiento Generacional en Caso Y

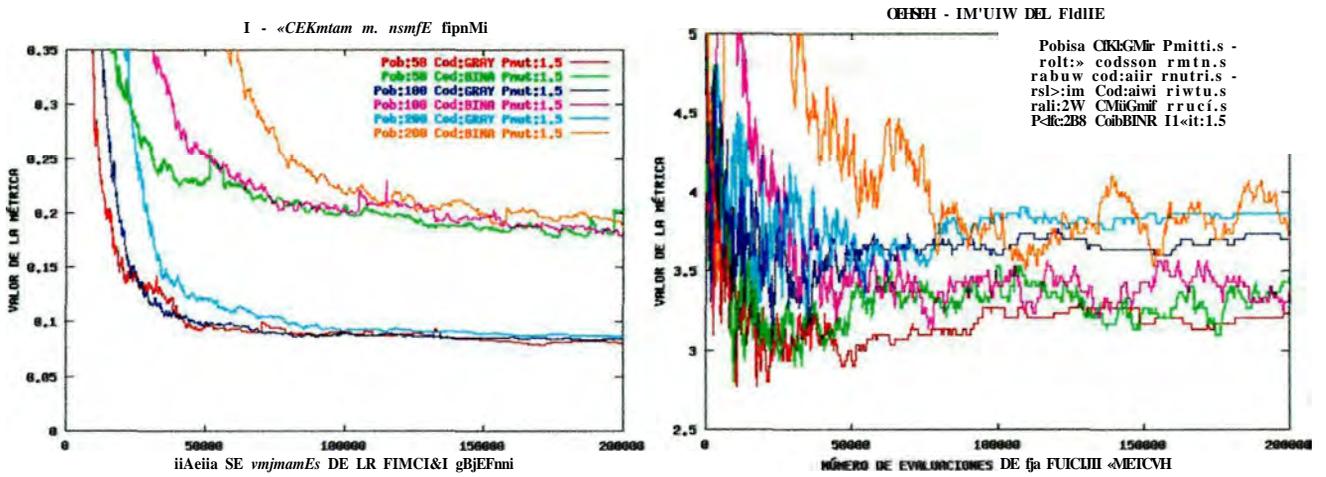


Figura 4.34. Comparativa Código Gray / Binario con DENSEA, Pmut 1.5% y reemplazamiento Generacional en Caso Y

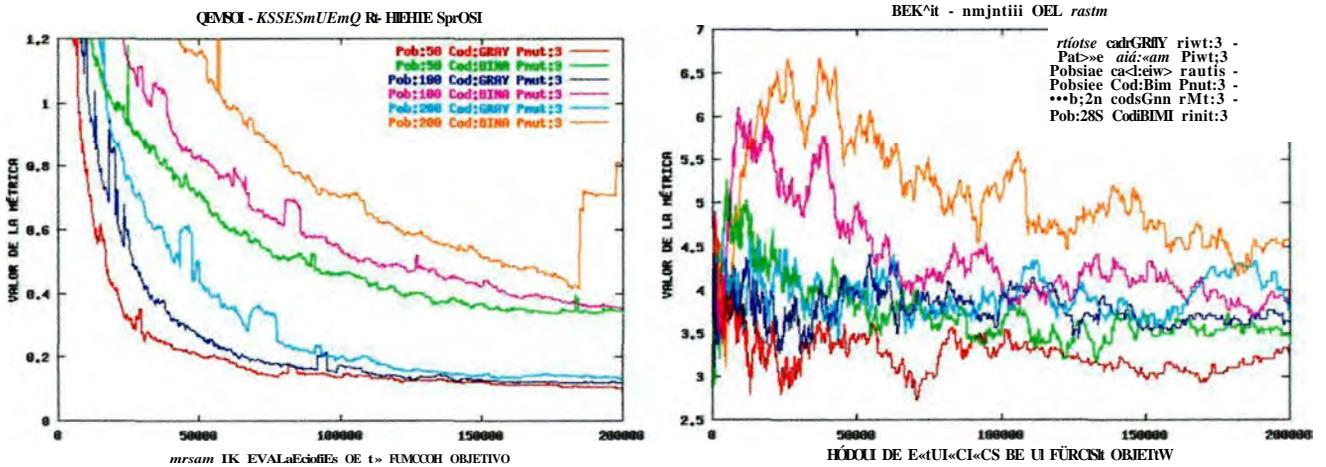


Figura 4.35. Comparativa Código Gray / Binario con DENSEA, Pmut 3.0% y reemplazamiento Generacional en Caso Y

Caso Y)BINARIO/GRAY: NSGA-II-ElconOO Steady-State Mut 0.4%, 0.8%, 1.5%, 3%

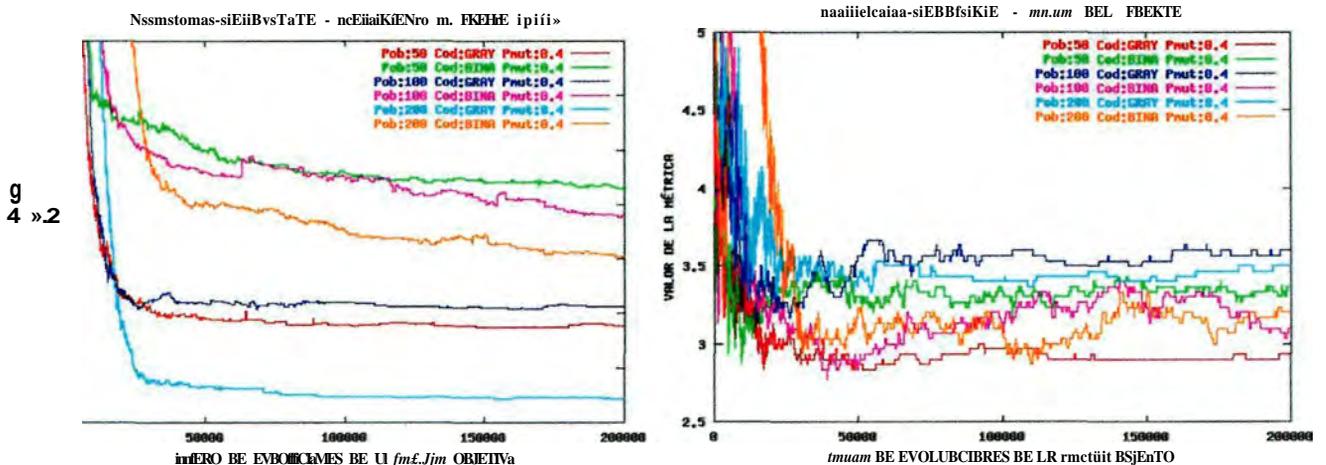
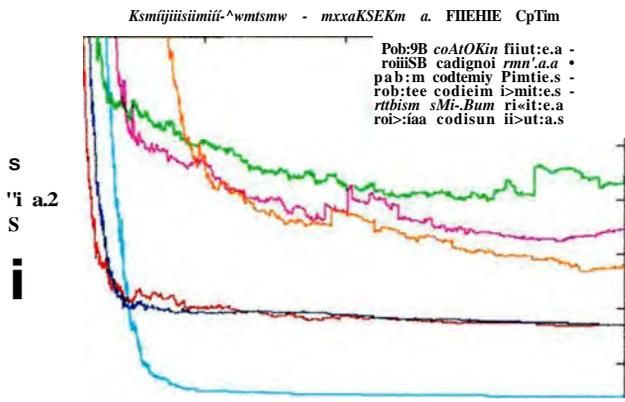
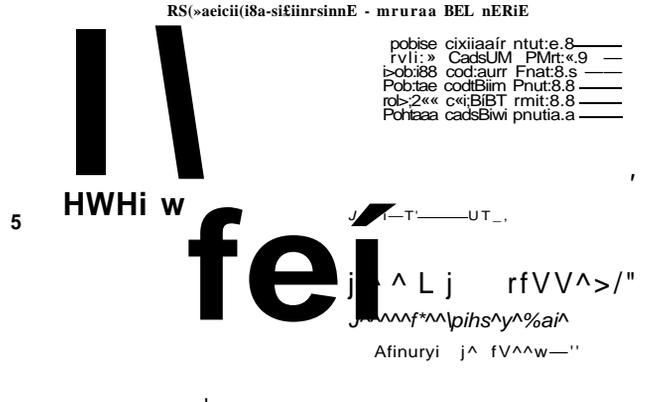


Figura 4.44. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con $r=0.0$, Pmut 0.4% y reemplazamiento EstadoEstacionario en Caso Y



M'iem QE EMLOCZQIES DE UI BMCISC EBIEIVO



NñcHi DE eimmems DE ut niicJH< <>JEn:W

Figura 4.45. Comparativa Código Gray / Binario con NSGAI1 y elitismo controlado con $r=0.0$, Pmut 0.8% y reemplazamiento EstadoEstacionario en Caso Y

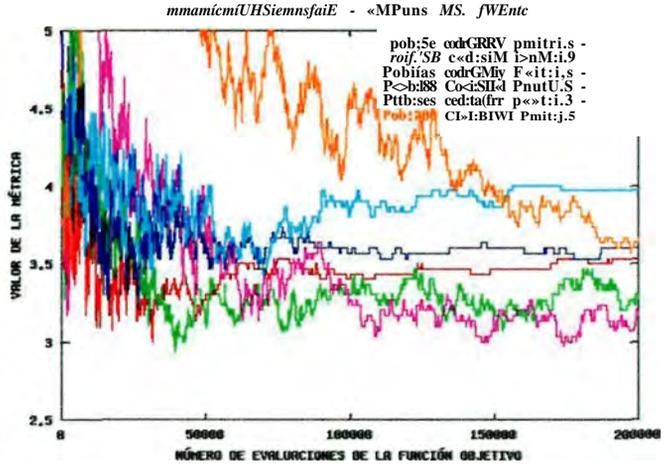
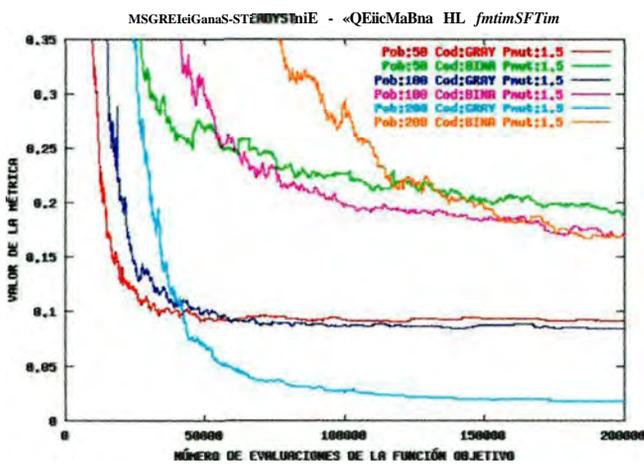
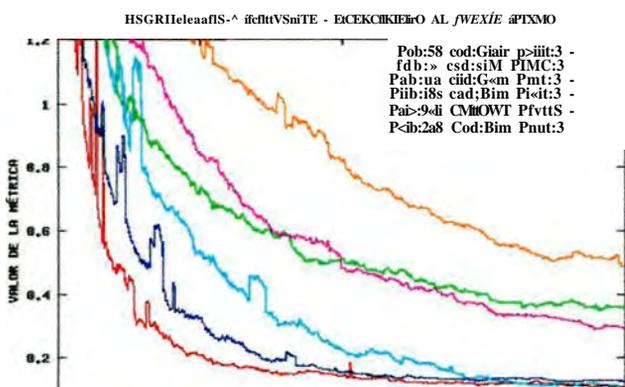
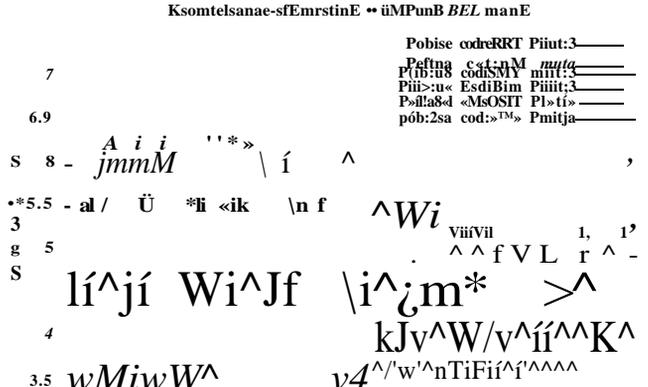


Figura 4.46. Comparativa Código Gray / Binario con NSGAI1 y elitismo controlado con $r=0.0$, Pmut 1.5% y reemplazamiento EstadoEstacionario en Caso Y



KbCRS BE EmUaiCi<ES DE UI FIXEAI IIBJEYIW1



DE EvaLuRcimes BE UI FIMCEON anjEmn

Figura 4.47. Comparativa Código Gray / Binario con NSGAI1 y elitismo controlado con $r=0.0$, Pmut 3.0% y reemplazamiento EstadoEstacionario en Caso Y

Caso Y)BINARIO/GRAY: NSGA-II-Elcon04 Steady-State Mut 0.4%, 0.8%, 1.5%, 3%

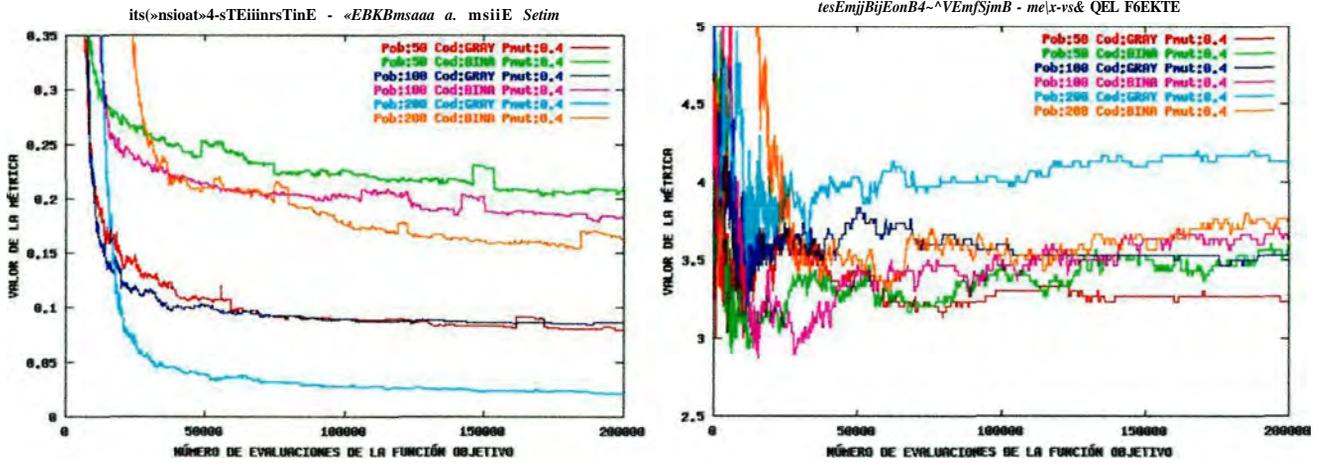


Figura 4.48. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con r=0.4, Pmut 0.4% y reemplazamiento EstadoEstacionario en Caso Y

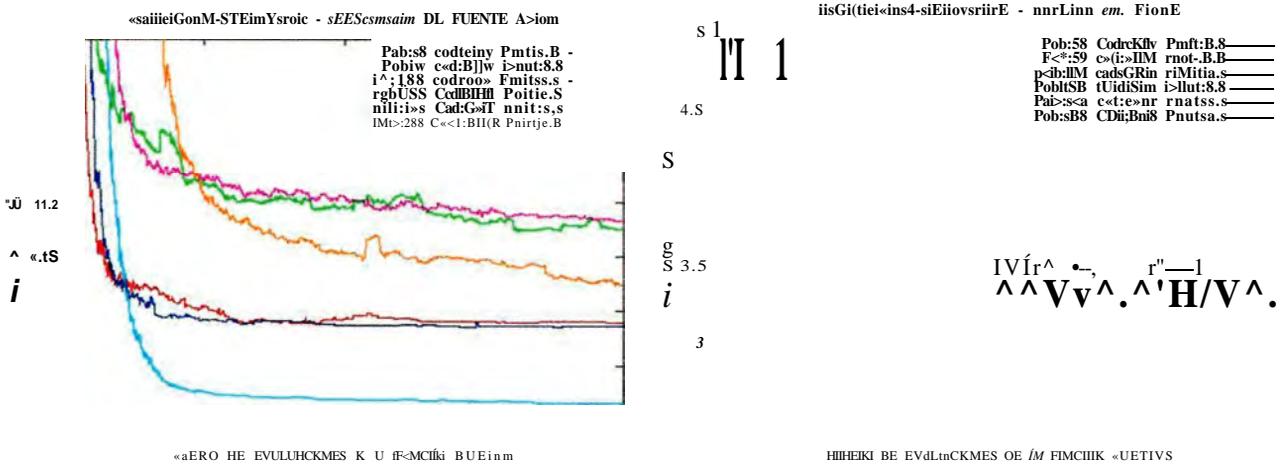


Figura 4.49. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con r=0.4, Pmut 0.8% y reemplazamiento EstadoEstacionario en Caso Y

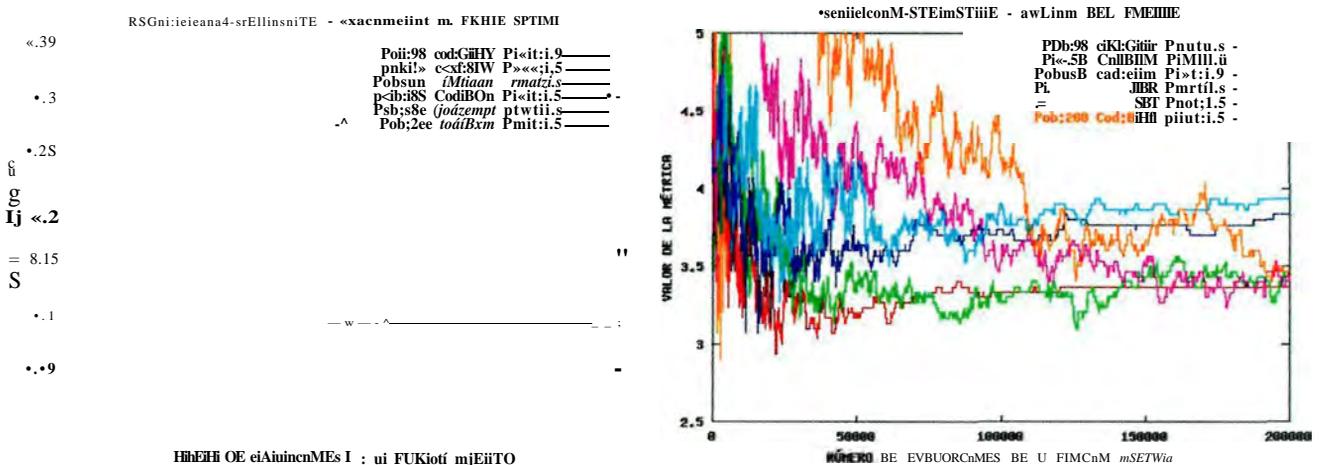


Figura 4.50. Comparativa Código Gray / Binario con NSGAII y elitismo controlado con r=0.4, Pmut 1.5% y reemplazamiento EstadoEstacionario en Caso Y

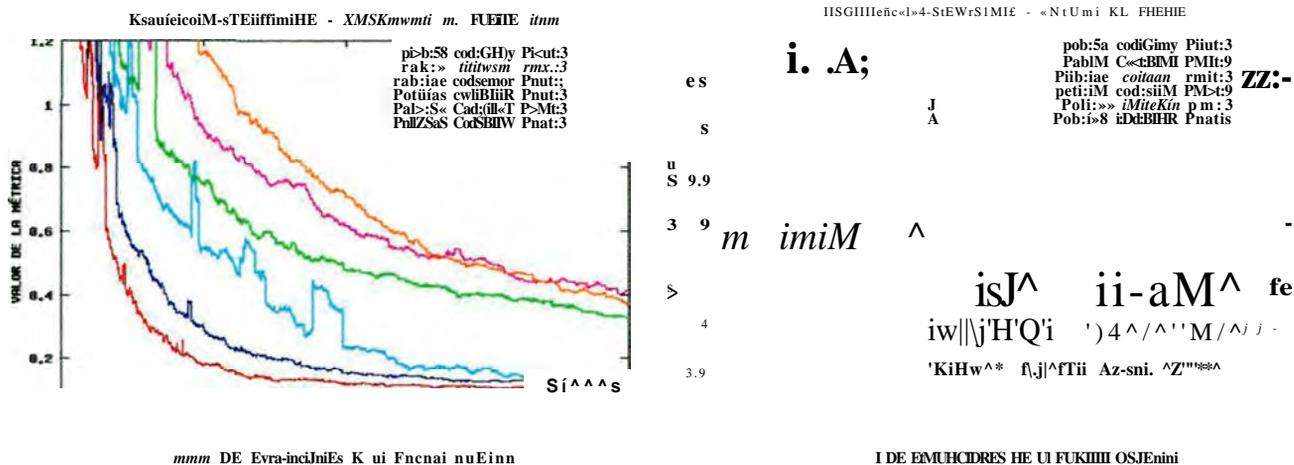


Figura 4.51. Comparativa Código Gray / Binario con NSGAI y elitismo controlado con $r=0.4$, $Pmut$ 3.0% y reemplazamiento EstadoEstacionario en Caso Y

Caso Y) BINARIO / GRAY: Spea2 Steady-State Mut 0.4%, 0.8%, 1.5%, 3%

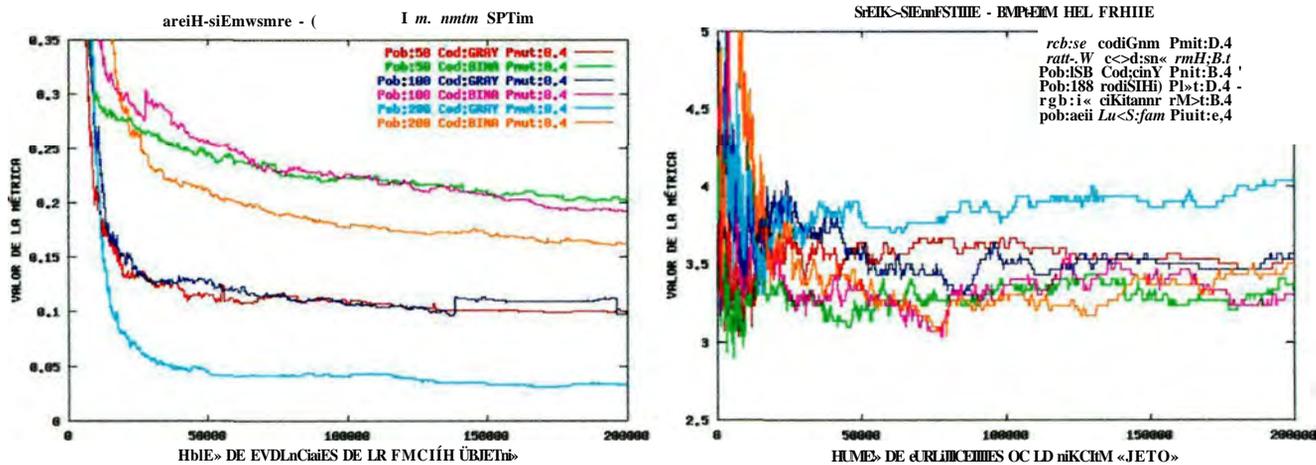


Figura 4.52. Comparativa Código Gray / Binario con SPEA2, $Pmut$ 0.4% y reemplazamiento EstadoEstacionario en Caso Y

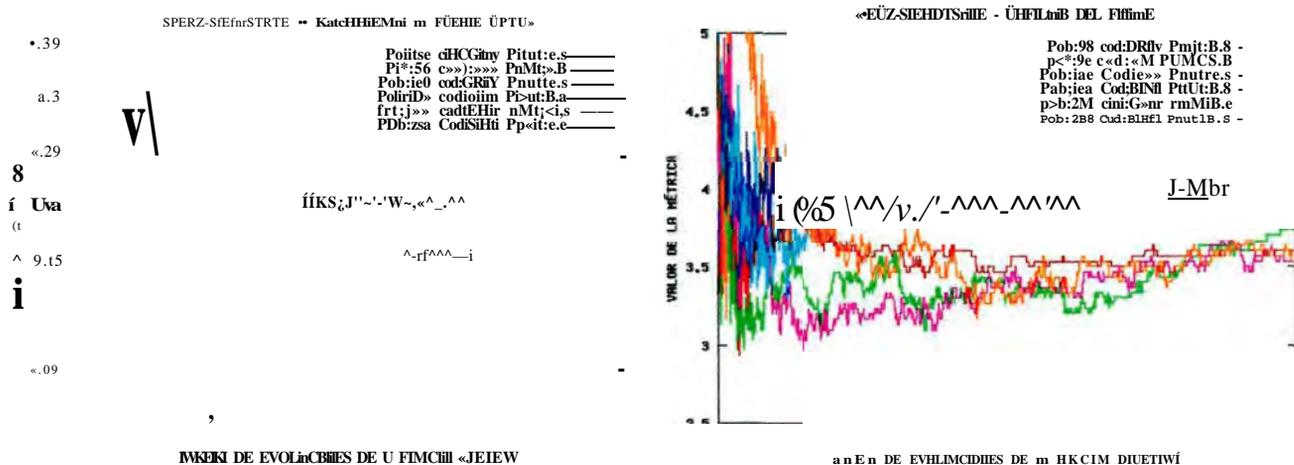


Figura 4.53. Comparativa Código Gray / Binario con SPEA2, $Pmut$ 0.8% y reemplazamiento EstadoEstacionario en Caso Y

Capítulo 4. Contribuciones

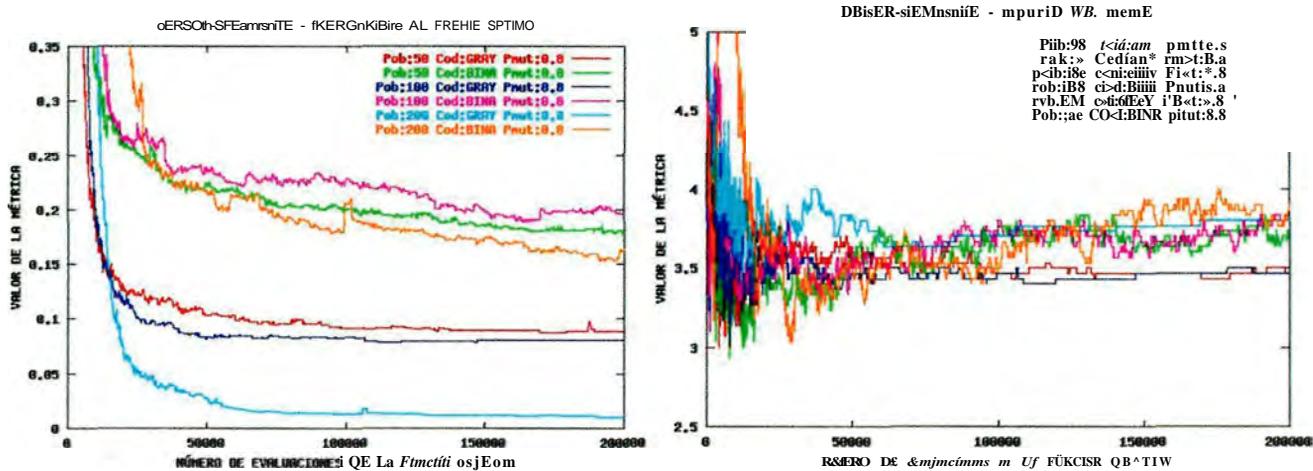


Figura 4.57. Comparativa Código Gray / Binario con DENSEA, Pmut 0.8% y reemplazamiento EstadoEstacionario en Caso Y

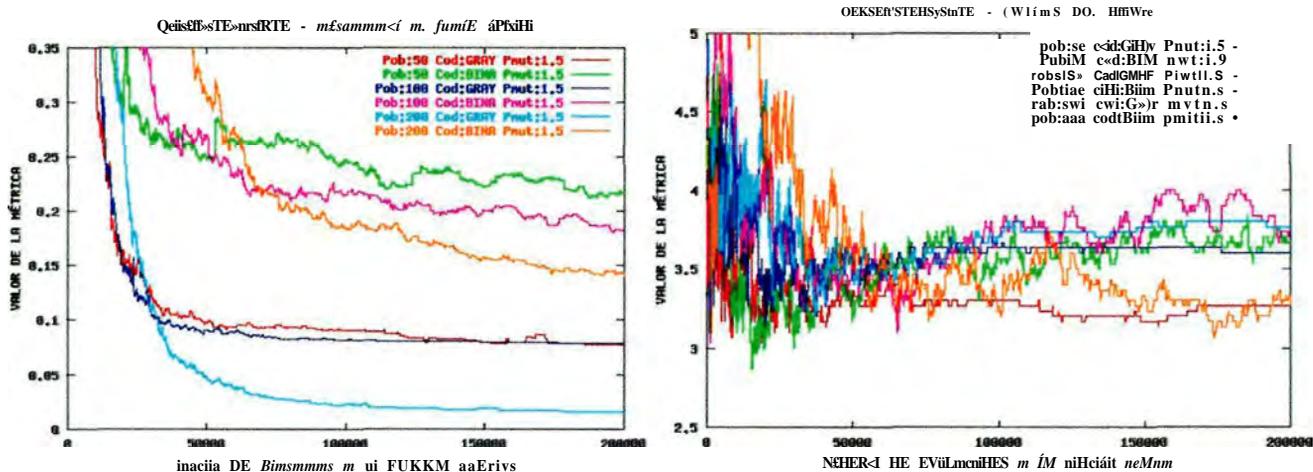


Figura 4.58. Comparativa Código Gray / Binario con DENSEA, Pmut 1.5% y reemplazamiento EstadoEstacionario en Caso Y

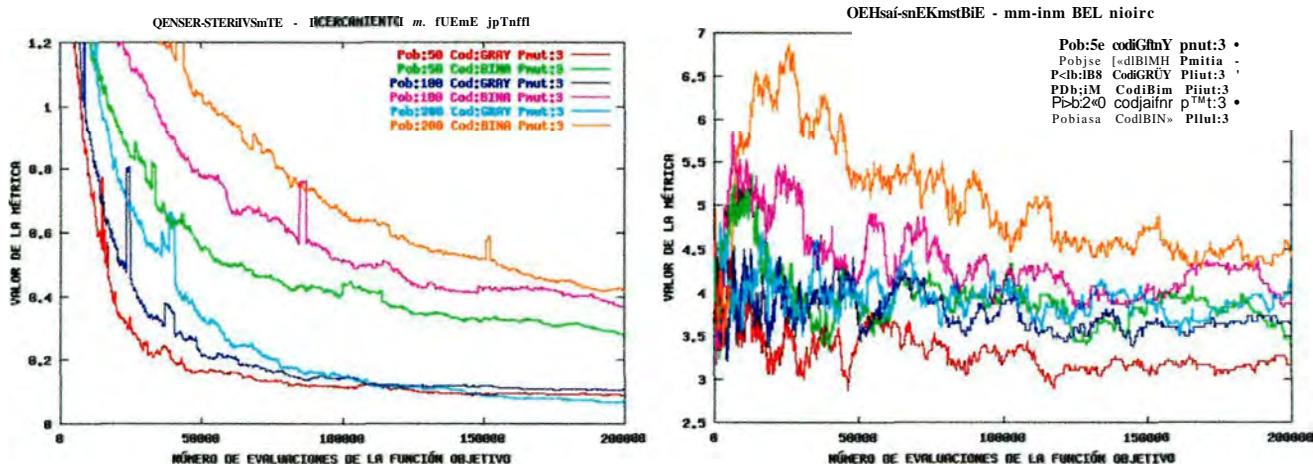


Figura 4.59. Comparativa Código Gray / Binario con DENSEA, Pmut 3.0% y reemplazamiento EstadoEstacionario en Caso Y

Se muestra el algoritmo Spea2 (generacional), con 3 tamaños de población en las figuras 4.77 a 4.79.

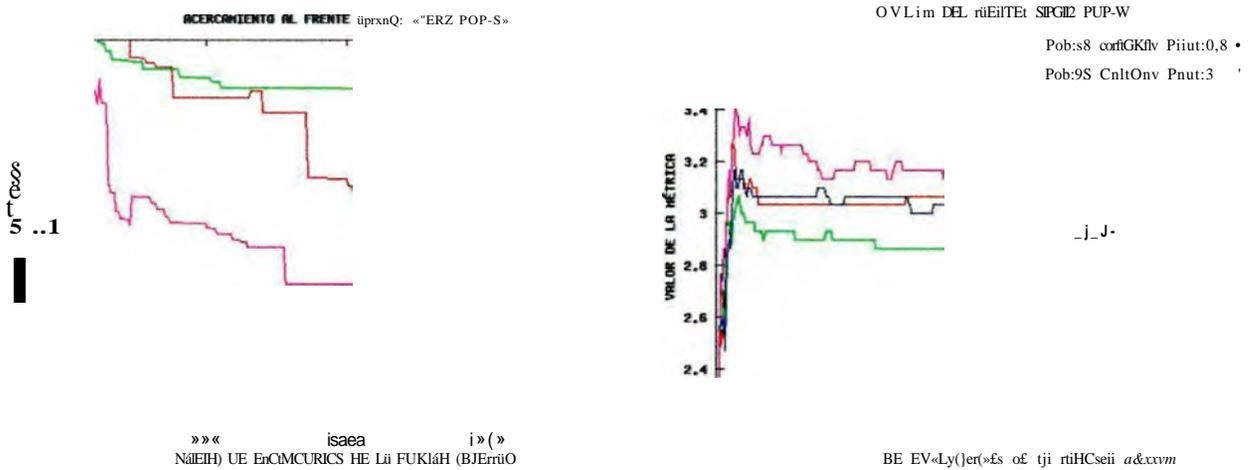


Figura 4.77. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo SPEA2 con tamaño de población 50.

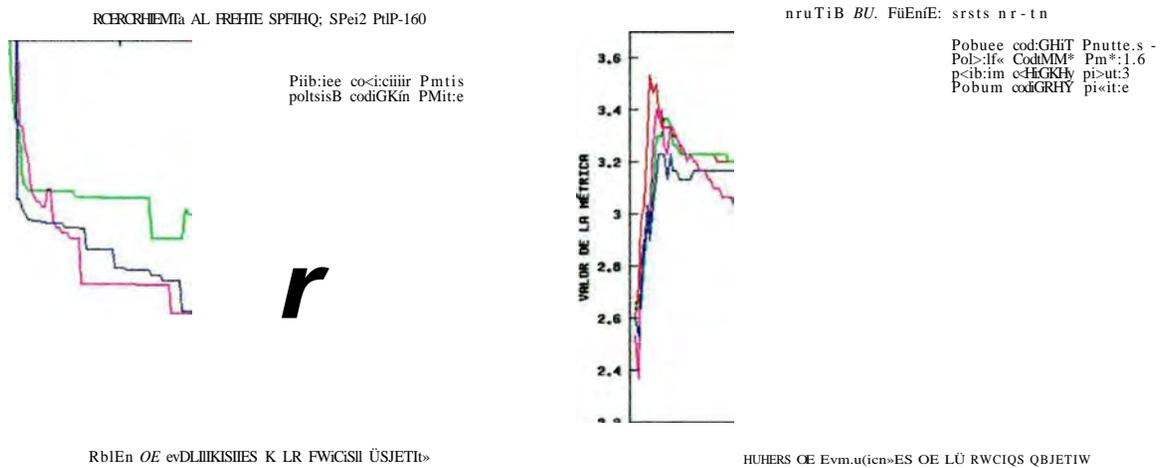


Figura 4.78. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo SPEA2 con tamaño de población 100.

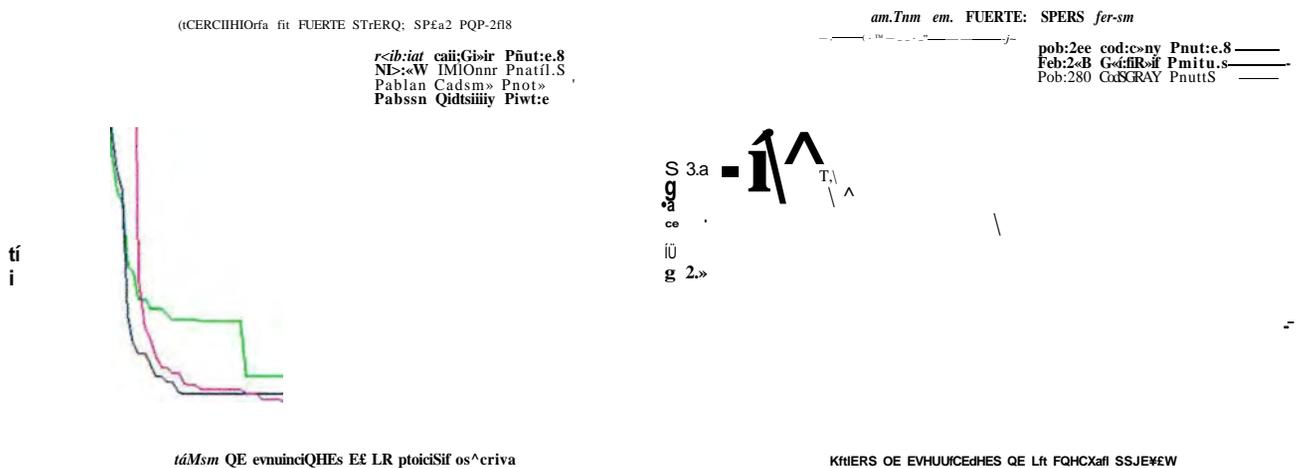


Figura 4.79. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo SPEA2 con tamaño de población 200.

Capítulo 4. Contribuciones

Se muestra el algoritmo Nsgall (generacional), con 3 tamaños de población en las figuras 4.80 a 4.82.

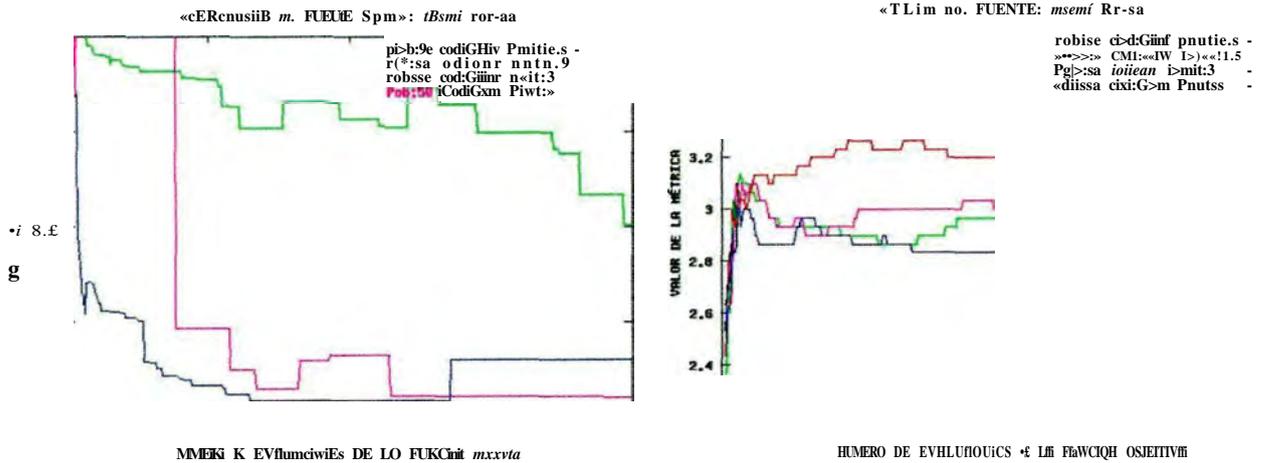


Figura 4.80. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo NSGAII con tamaño de población 50.

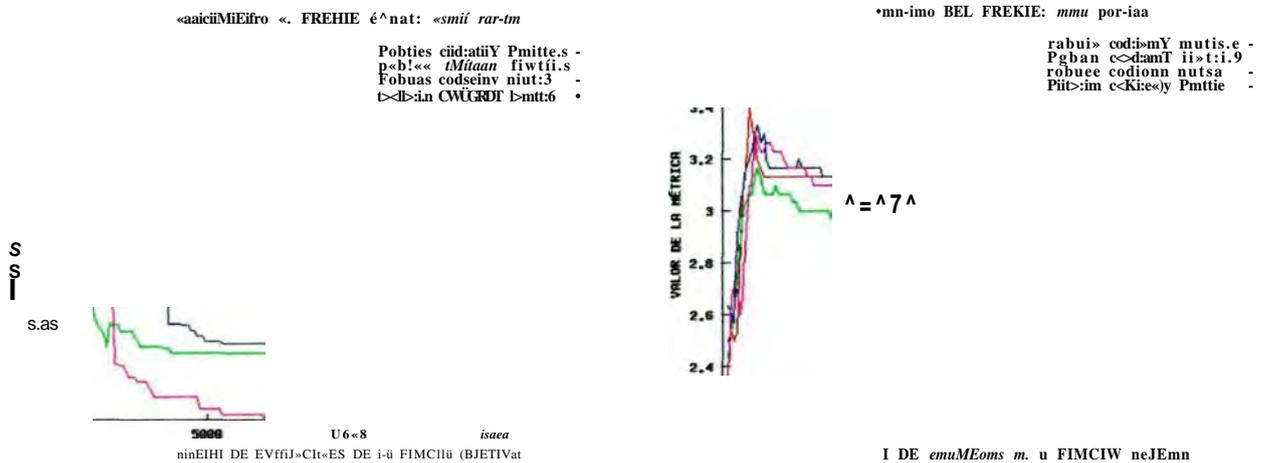


Figura 4.81. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo NSGAII con tamaño de población 100.

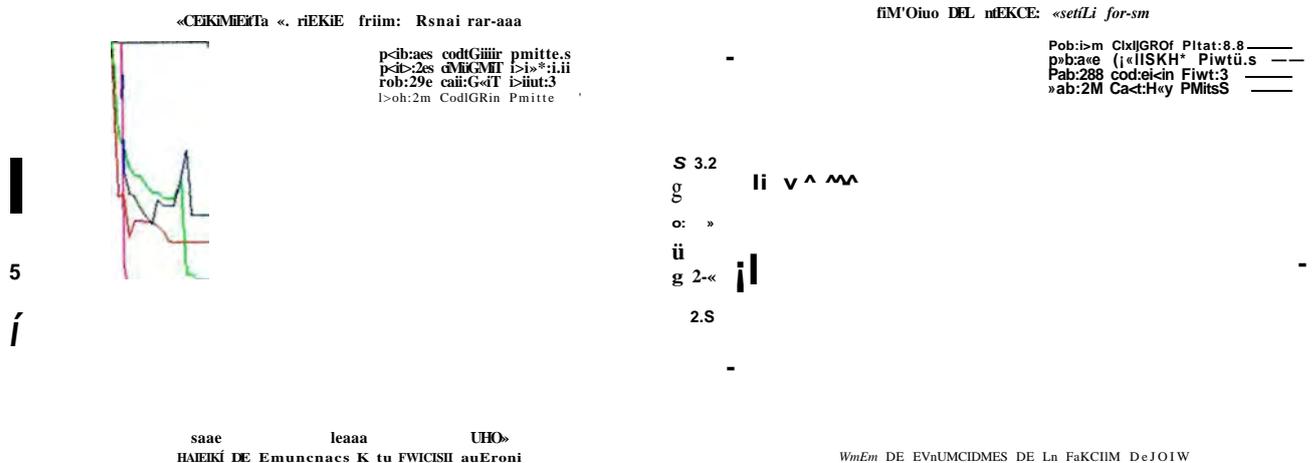


Figura 4.82. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo NSGAII con tamaño de población 200.

Capítulo 4. Contribuciones

Se muestra el algoritmo NsgaiI-ElconOO (generacional), con 3 tamaños de población en las figuras 4.83 a 4.85.

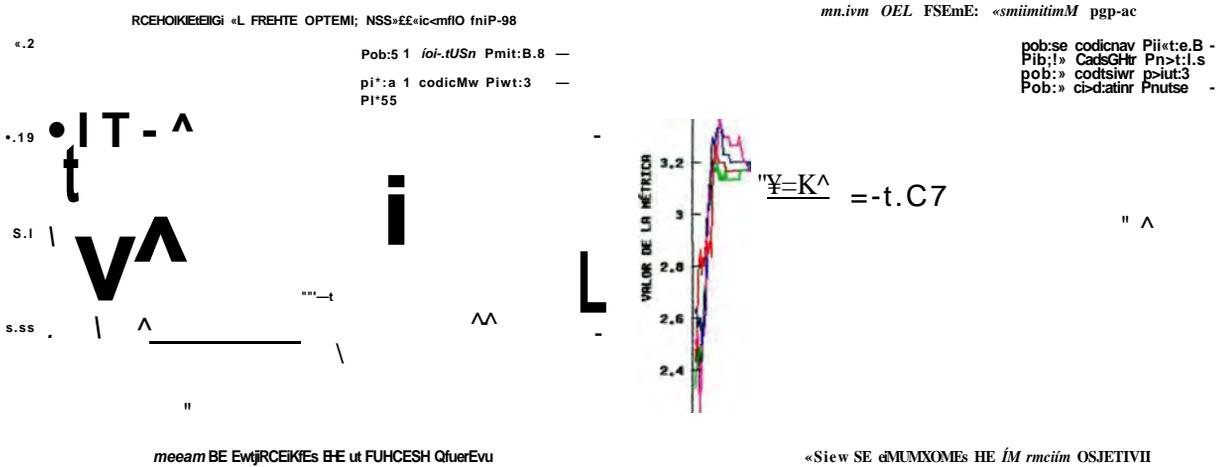


Figura 4.83. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.0$ con tamaño de población 50.

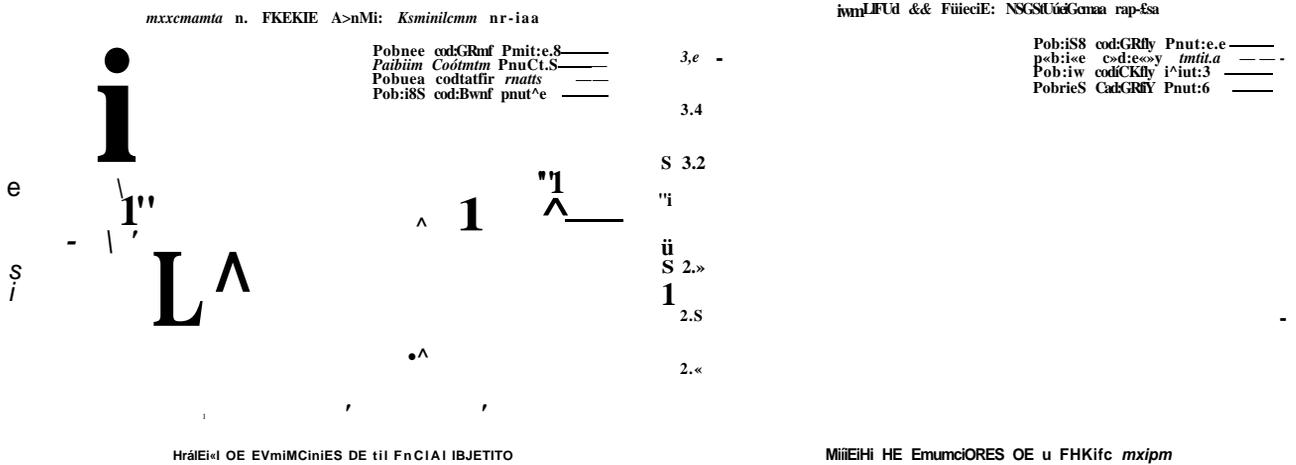


Figura 4.84. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.0$ con tamaño de población 100.

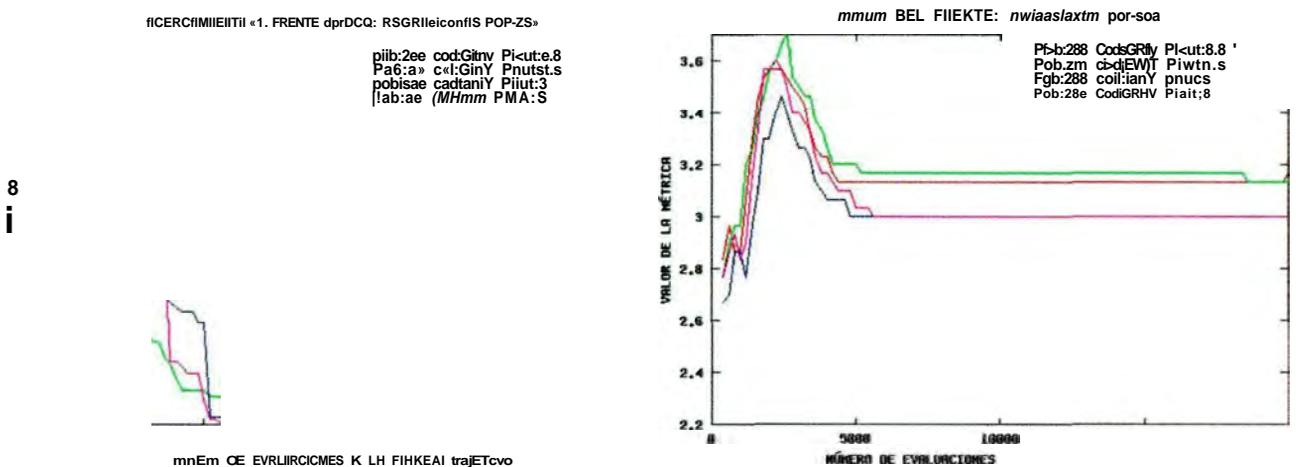


Figura 4.85. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.0$ con tamaño de población 200.

Se muestra el algoritmo NsgaiI-Elcon04 (generacional), con 3 tamaños de población en las figuras 4.86 a 4.88

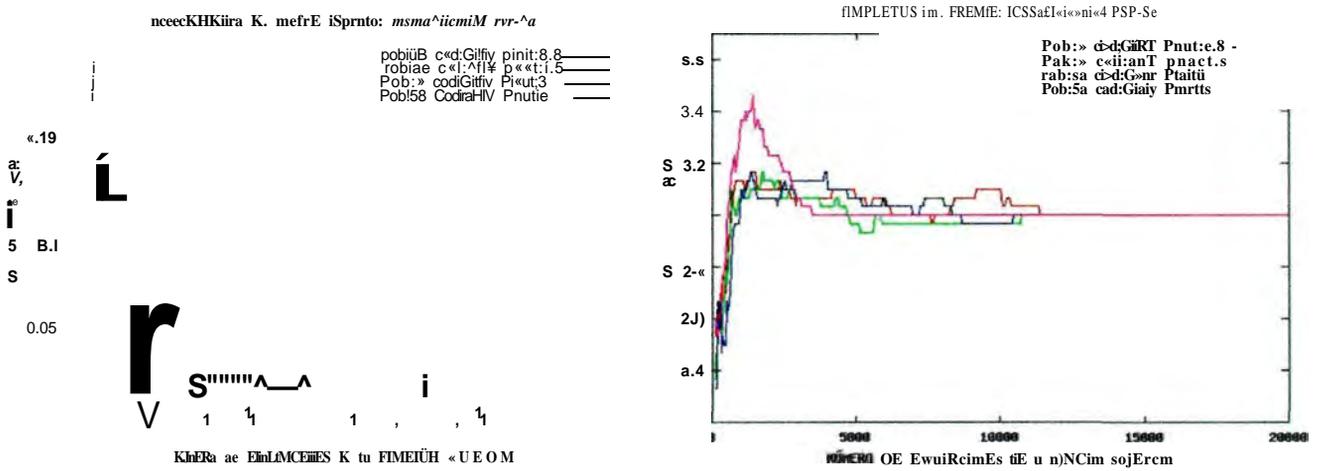


Figura 4.86. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 50.

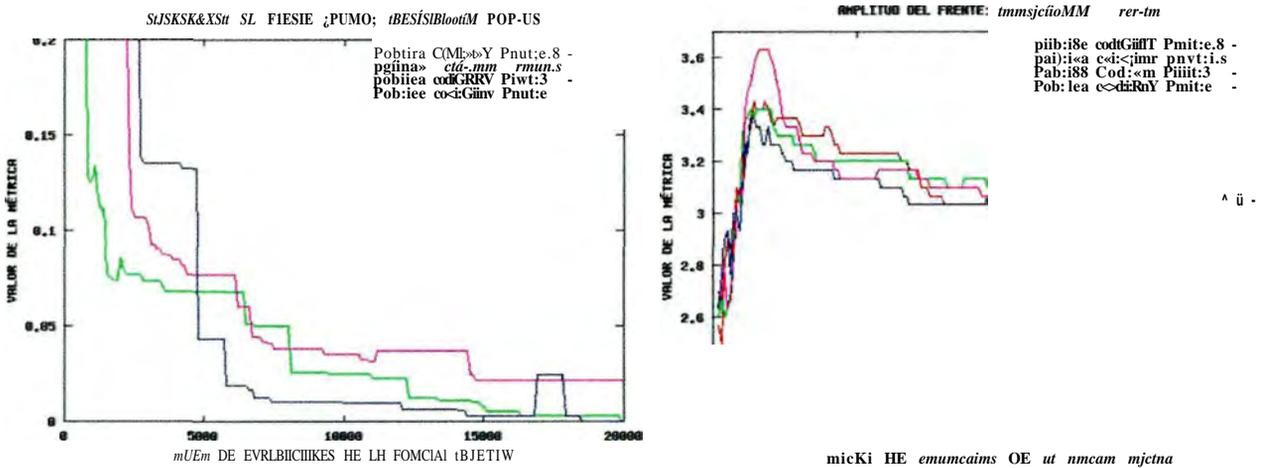


Figura 4.87. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 100.

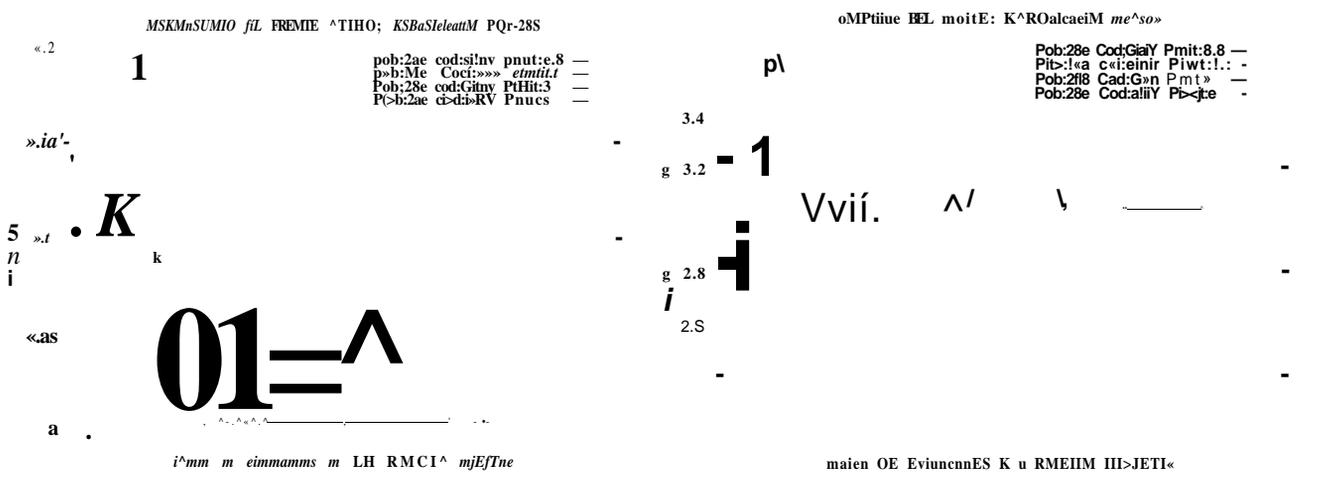


Figura 4.88. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 200.

Se muestra el algoritmo DENSEA (generacional), con 3 tamaños de población en las figuras 4.89 a 4.91.

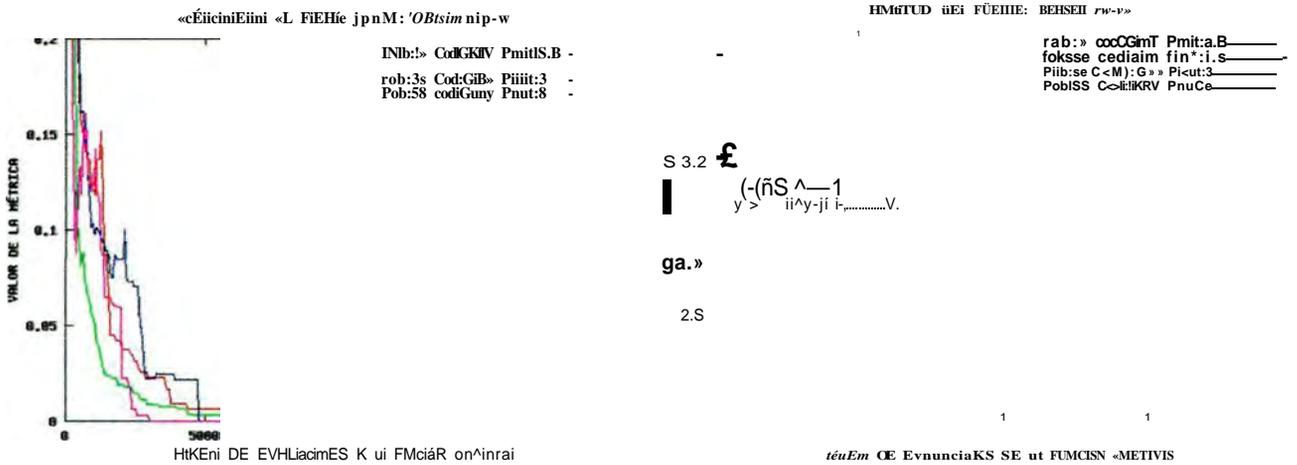


Figura 4.89. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo DENSEA con tamaño de población 50.

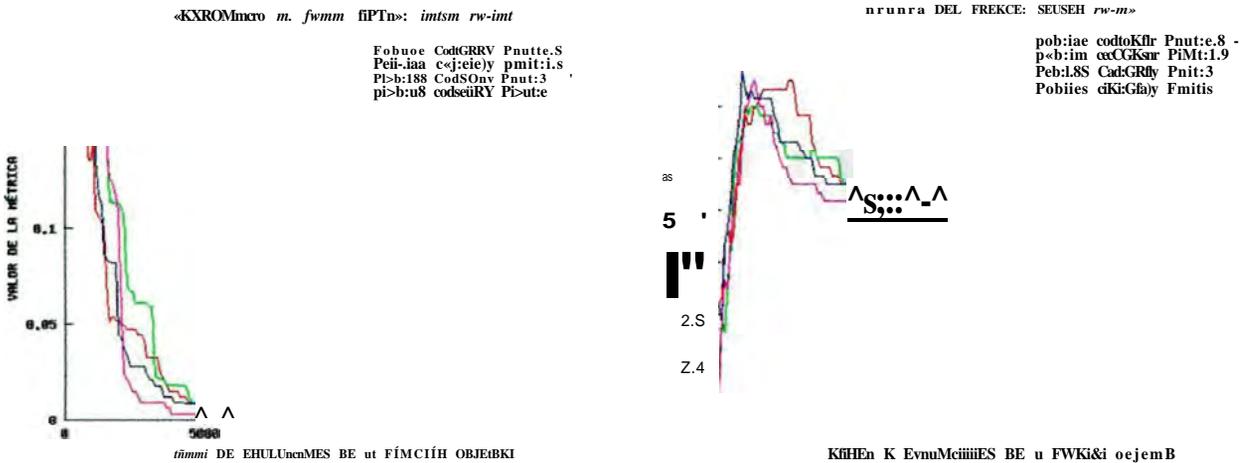


Figura 4.90. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo DENSEA con tamaño de población 100.

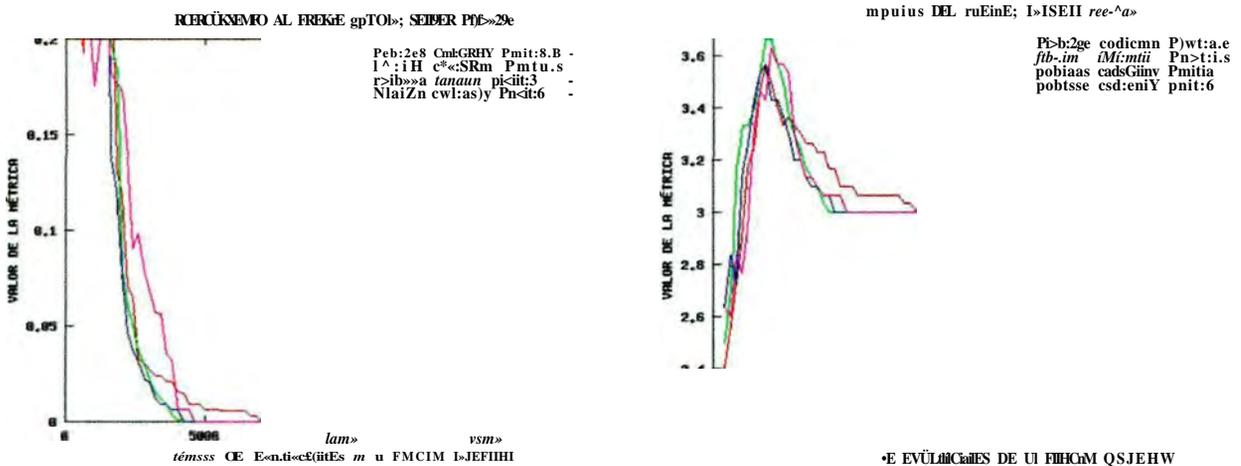


Figura 4.91. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Generacional. Algoritmo DENSEA con tamaño de población 200.

Se muestra el algoritmo Spea2 (steady-state), con 3 tamaños de población en las figuras 4.95 a 4.97.

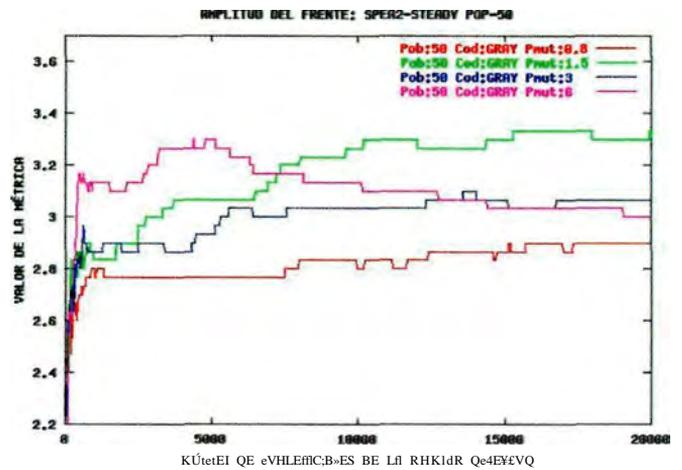
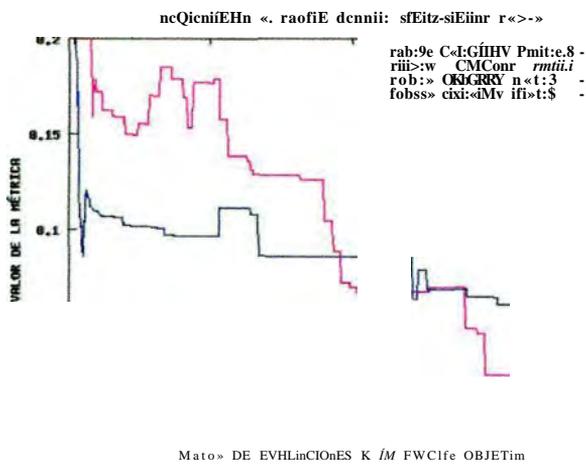


Figura 4.95. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo SPEA2 con tamaño de población 50.

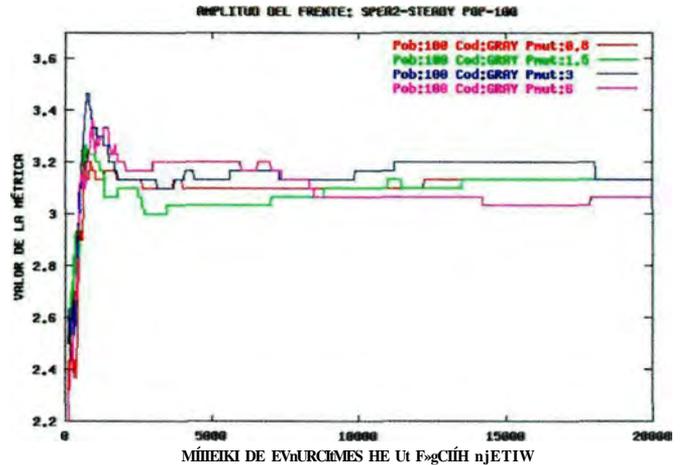
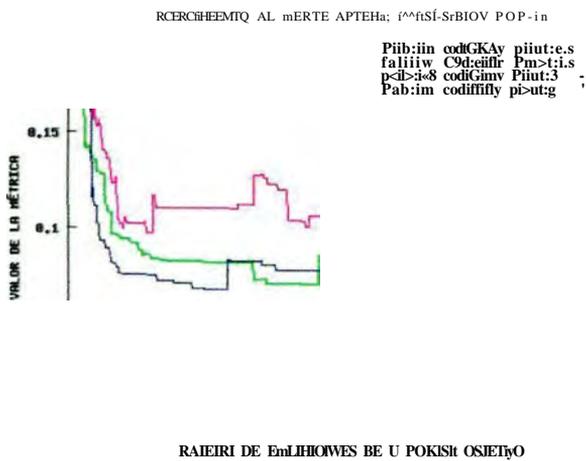


Figura 4.96. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo SPEA2 con tamaño de población 100.

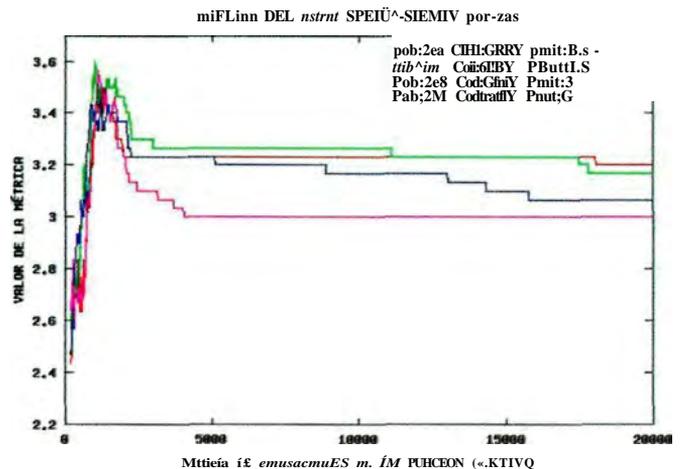
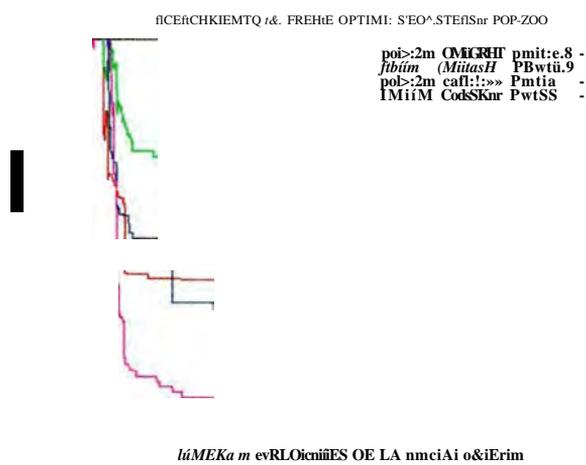


Figura 4.97. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo SPEA2 con tamaño de población 200.

Se muestra el algoritmo Nsgall (steady-state), con 3 tamaños de población en las figuras 4.98 a 4.100.

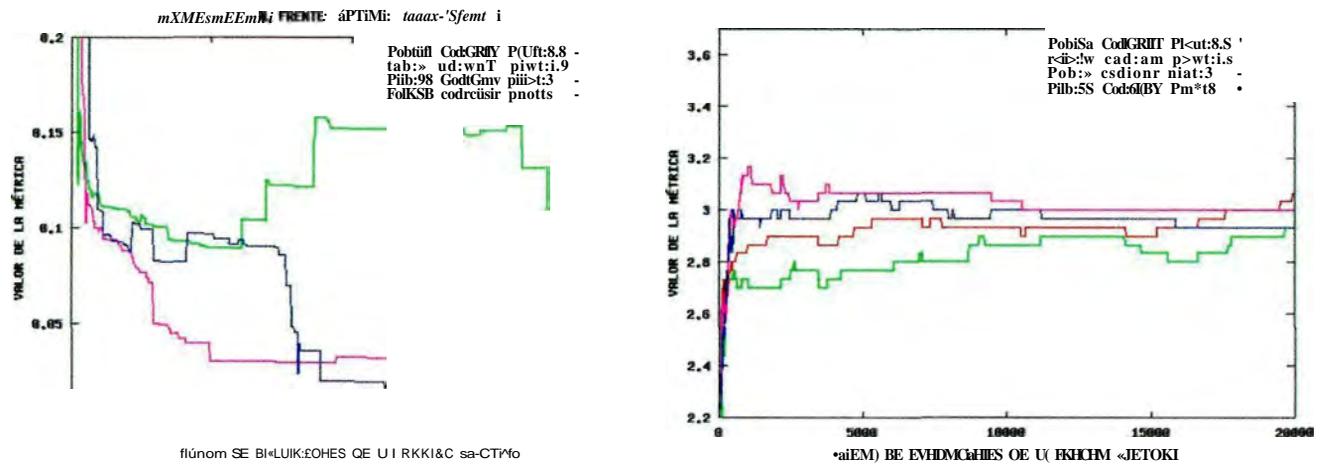


Figura 4.98. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con tamaño de población 50.

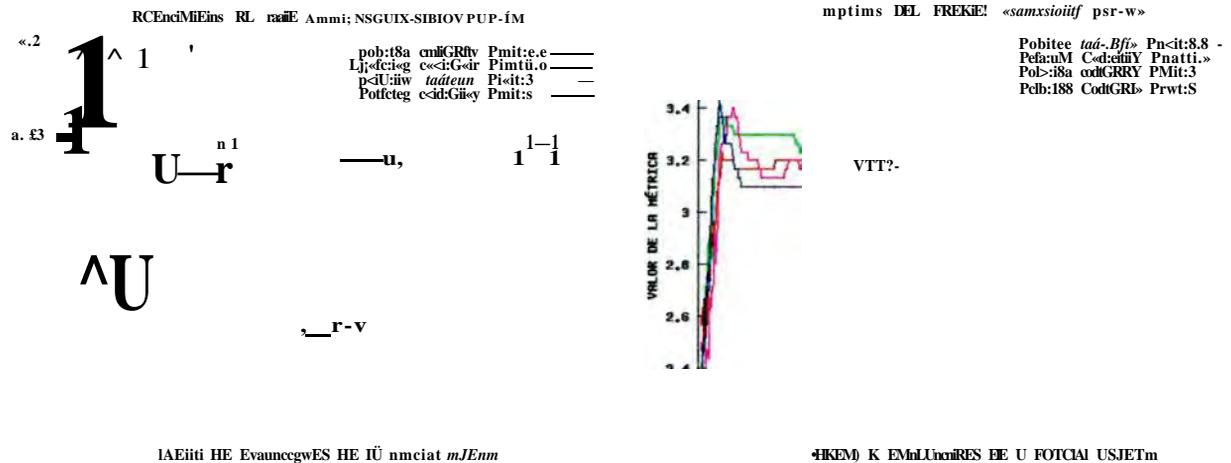


Figura 4.99. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con tamaño de población 100.

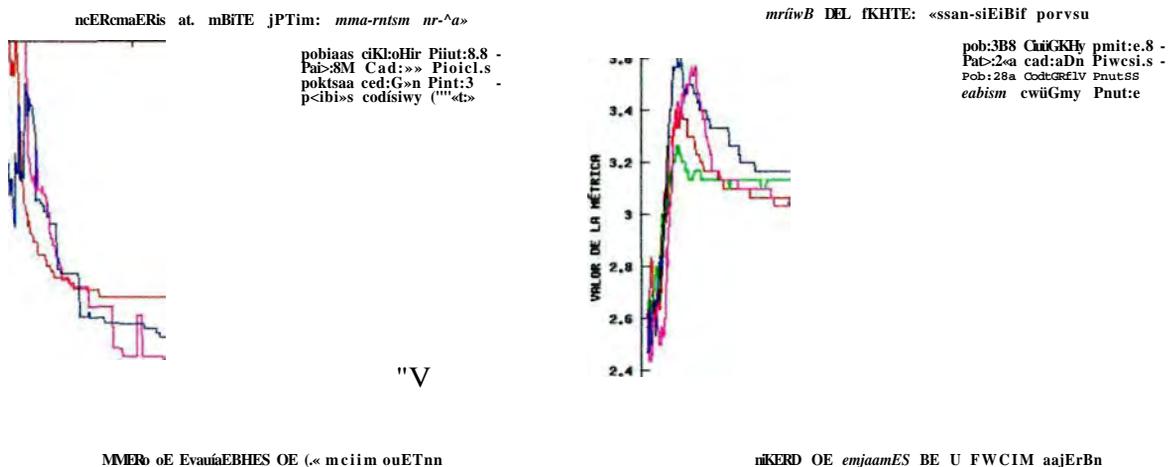


Figura 4.100. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con tamaño de población 200.

Se muestra el algoritmo NsgaII-ElconOO (steady-state), con 3 tamaños de población en las figuras 4.101 a 4.103.

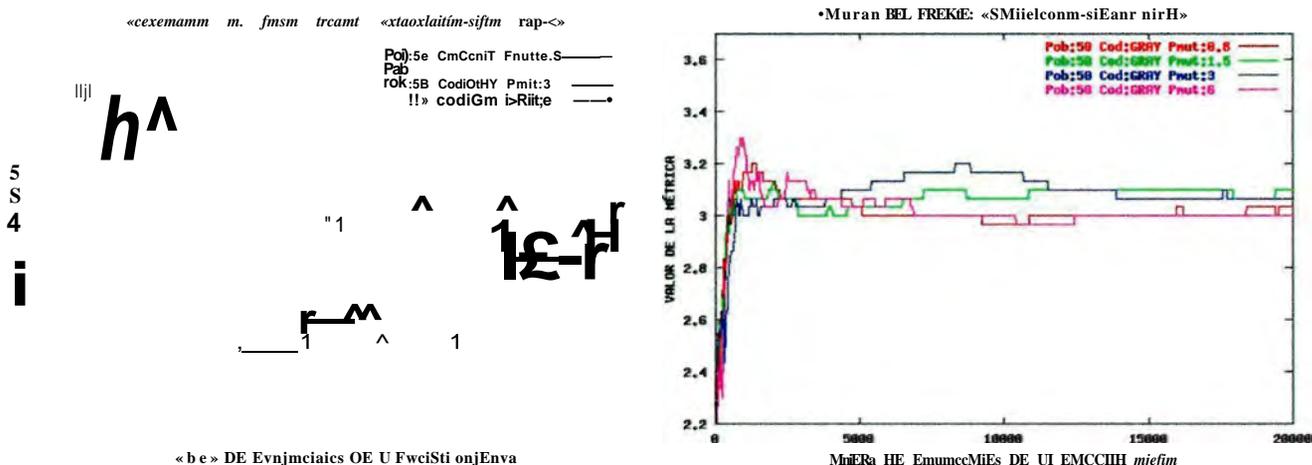


Figura 4.101. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Estado Estacionario. Algoritmo NSGAII con elitismo controlado y $r=0.0$ con tamaño de población 50.

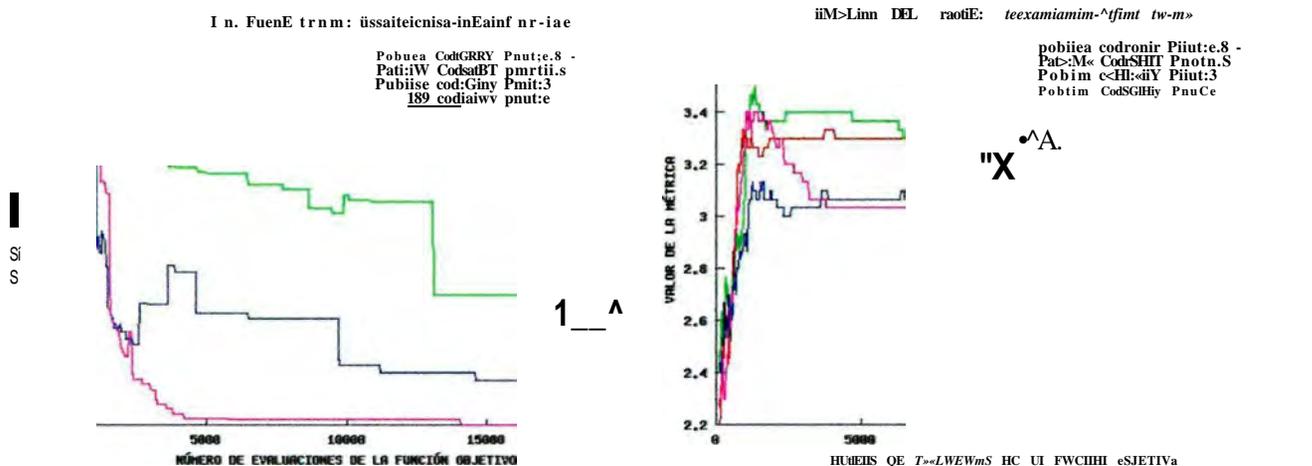


Figura 4.102. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Estado Estacionario. Algoritmo NSGAII con elitismo controlado y $r=0.0$ con tamaño de población 100.

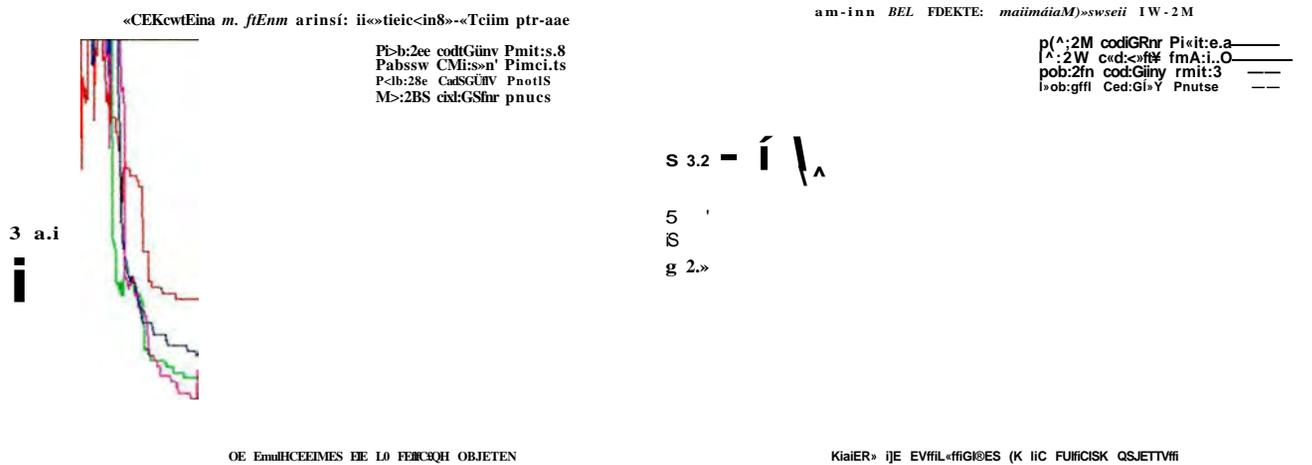


Figura 4.103. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento Estado Estacionario. Algoritmo NSGAII con elitismo controlado y $r=0.0$ con tamaño de población 200.

Capítulo 4. Contribuciones

Se muestra el algoritmo NsgaII-Elcon04 (steady-state), con 3 tamaños de población en las figuras 4.104 a 4.106.

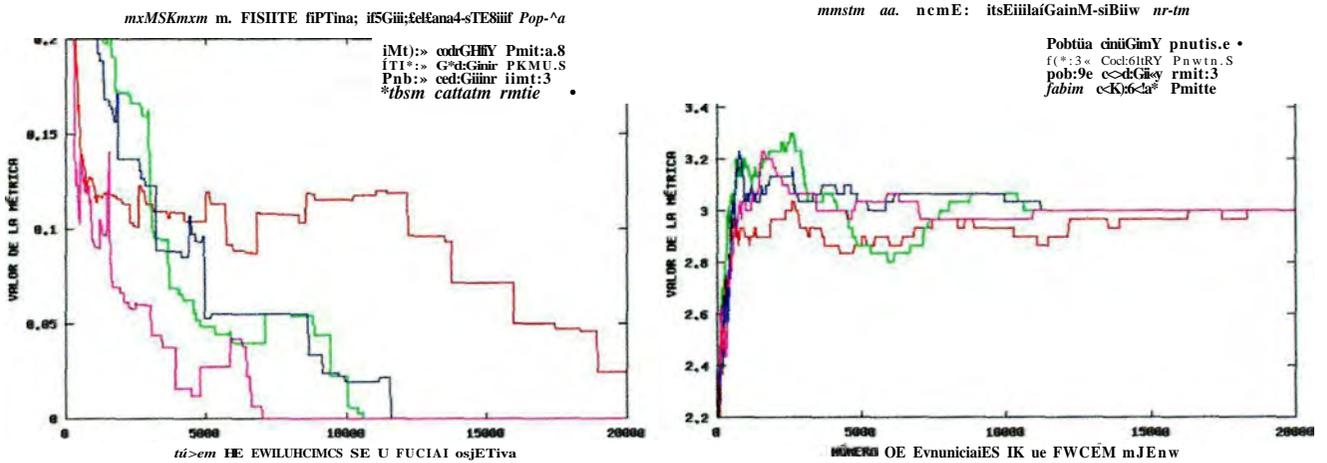


Figura 4.104. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 50.

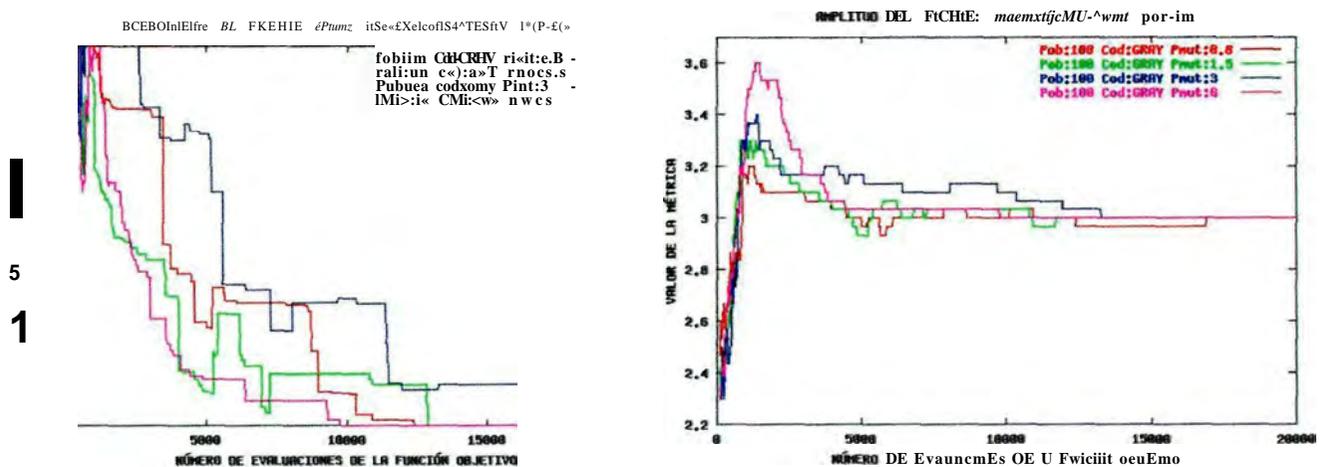


Figura 4.105. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 100.

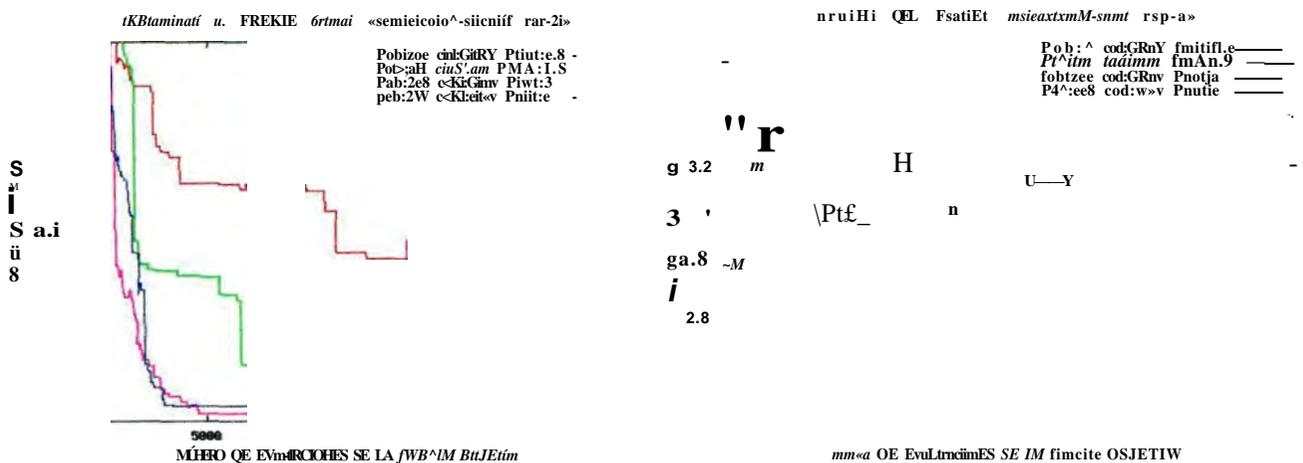


Figura 4.106. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 200.

Capítulo 4. Contribuciones

Se muestra el algoritmo DENSE A (steady-state), con 3 tamaños de población en las figuras 4.107 a 4.109.

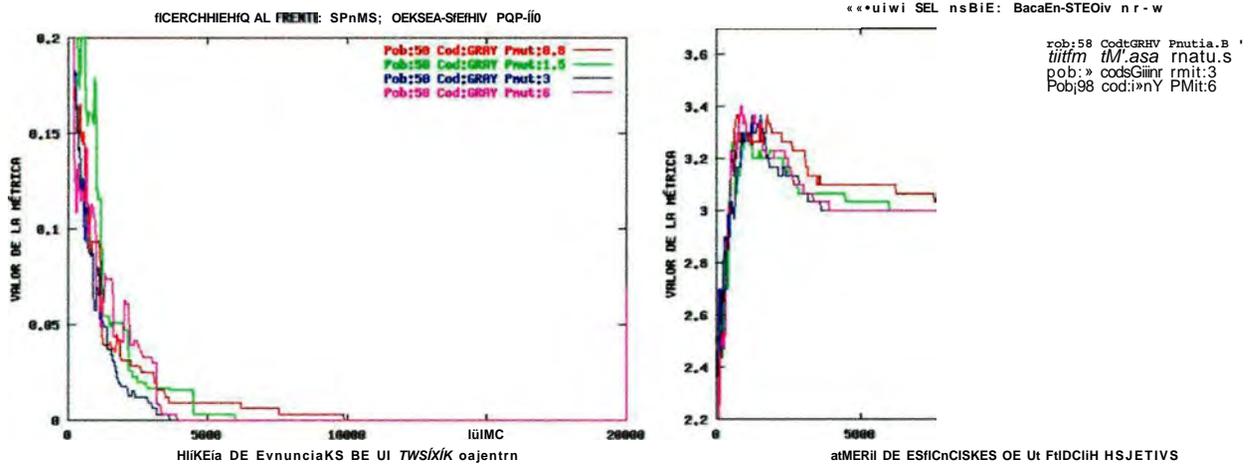


Figura 4.107. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA con tamaño de población 50.

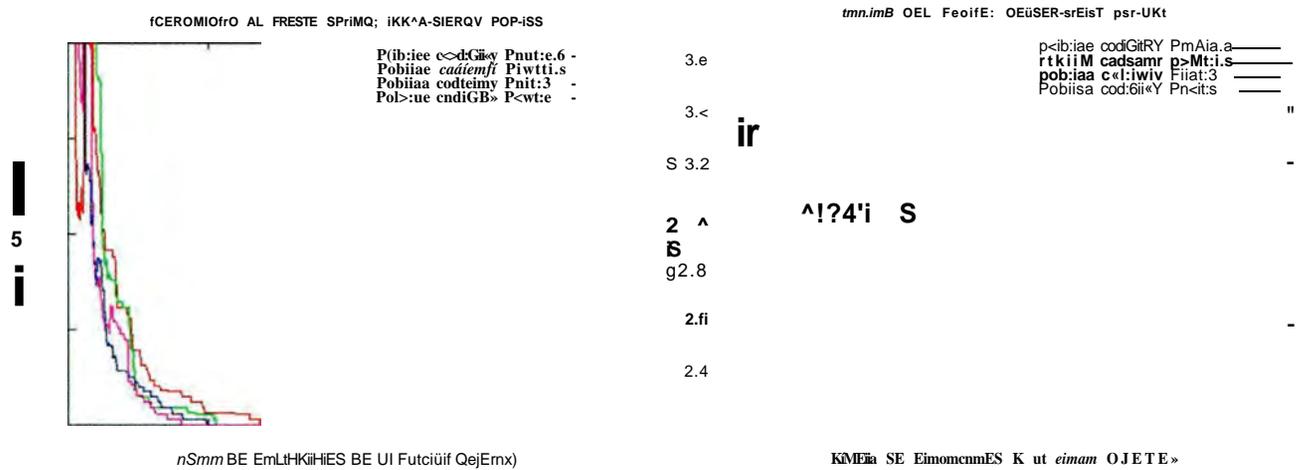


Figura 4.108. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA con tamaño de población 100.

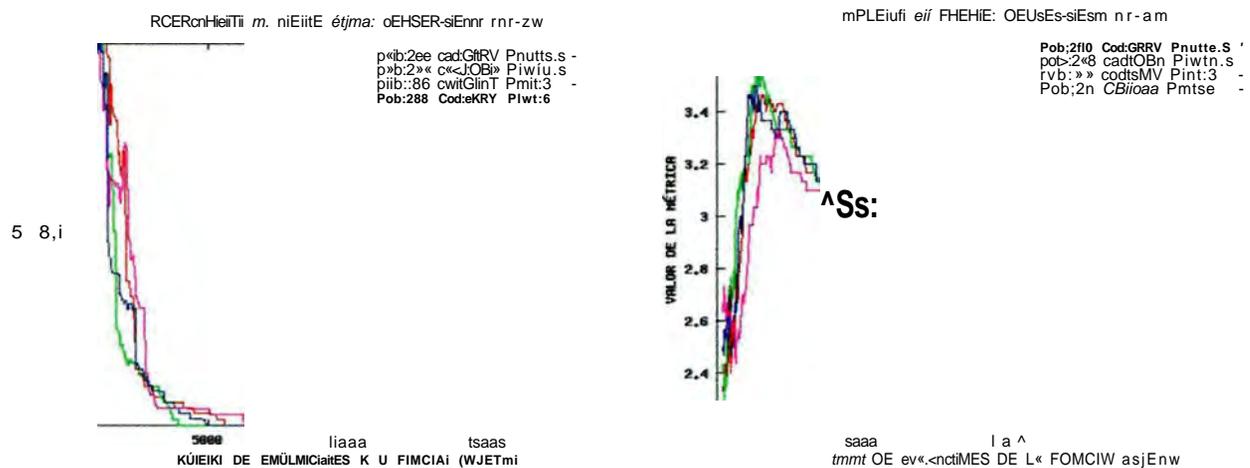


Figura 4.109. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA con tamaño de población 200.

Se muestra el algoritmo DENSEA-NSH (steady-state), con 3 tamaños de población en las figuras 4.110a4.112.

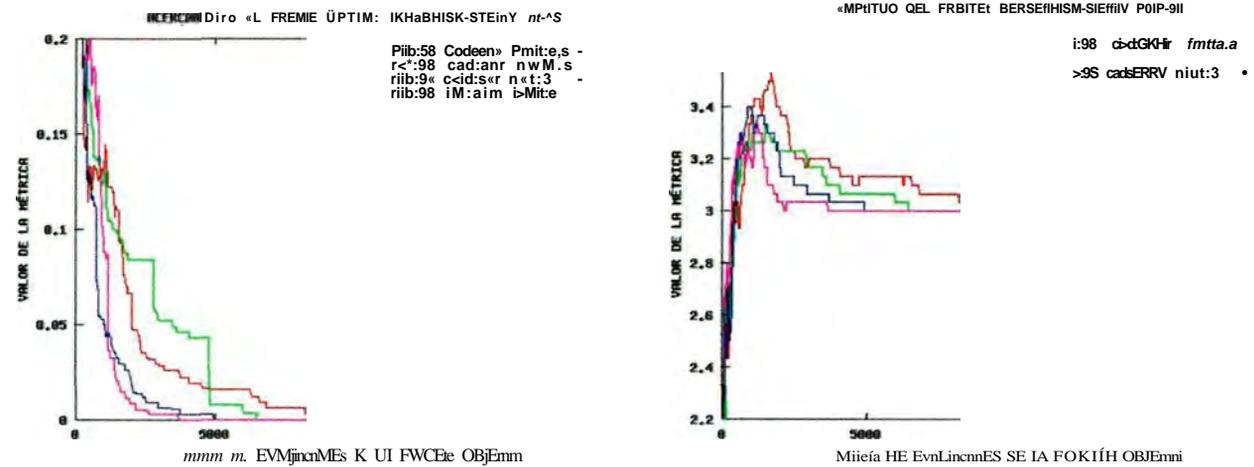


Figura 4.110. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA-NSH con tamaño de población 50.

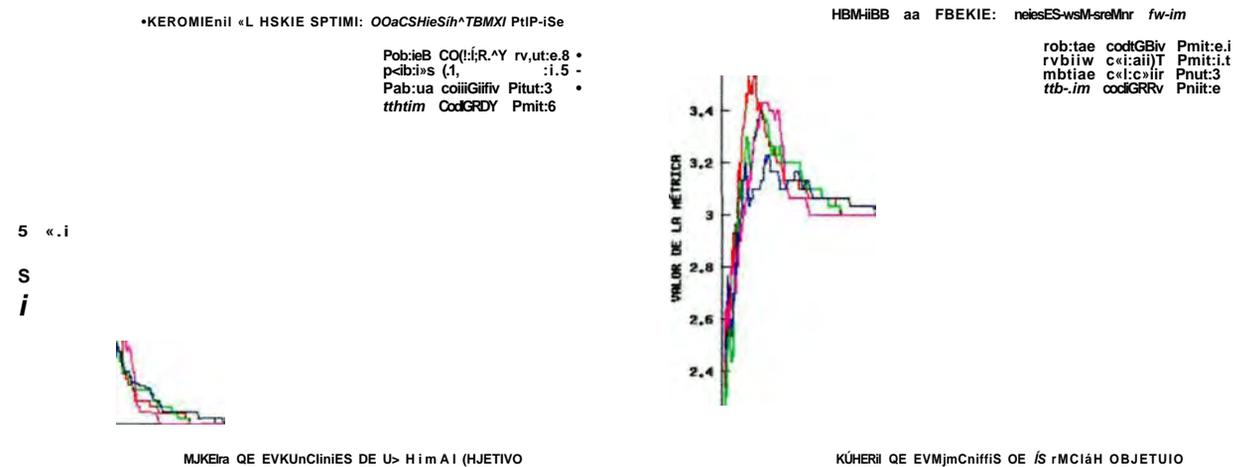


Figura 4.111. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA-NSH con tamaño de población 100.

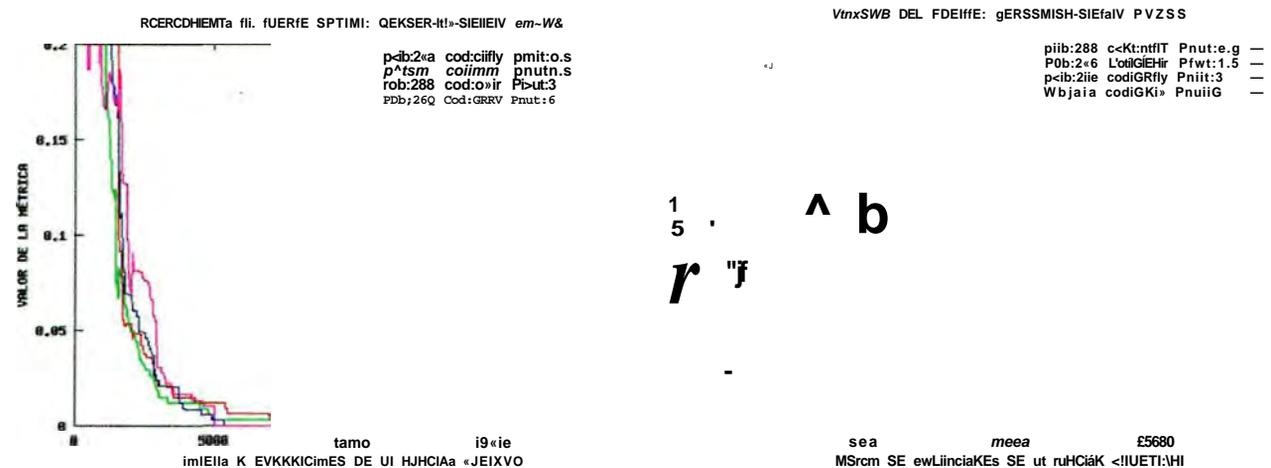


Figura 4.112. Métricas en comparativa de probabilidades de mutación en Caso X. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA-NSH con tamaño de población 200.

Capítulo 4. Contribuciones

Se muestra en el algoritmo Nsgall (generacional) con 3 tamaños de población en las figuras 4.116 a 4.118.

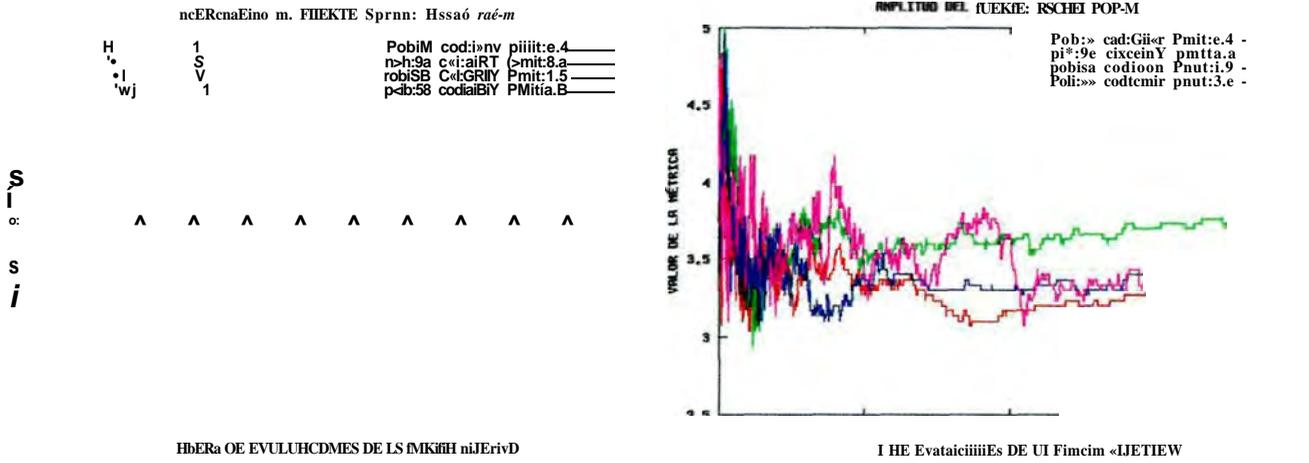


Figura 4.116. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo NSGAI con tamaño de población 50.

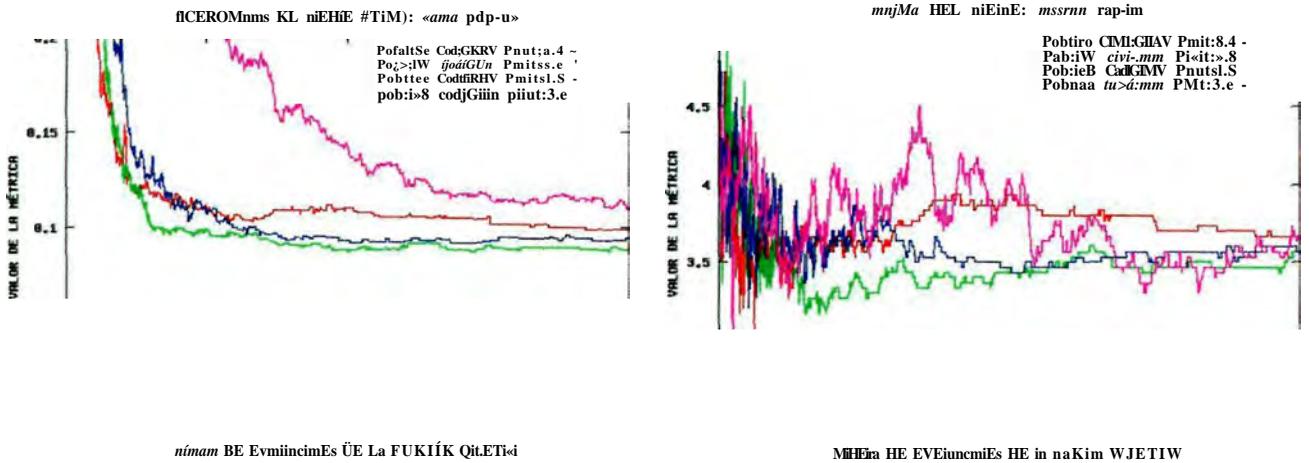


Figura 4.117. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo NSGAI con tamaño de población 100.

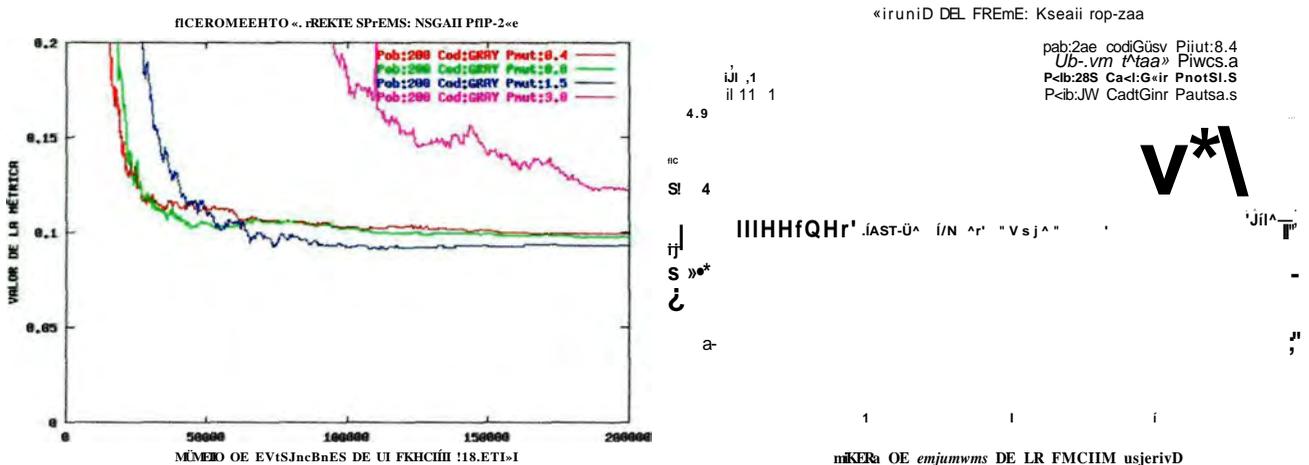


Figura 4.118. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo NSGAI con tamaño de población 200.

Se muestra en el algoritmo NsgaiI-ElconOO (generacional), con 3 tamaños de población en las figuras 4.119a4.121.

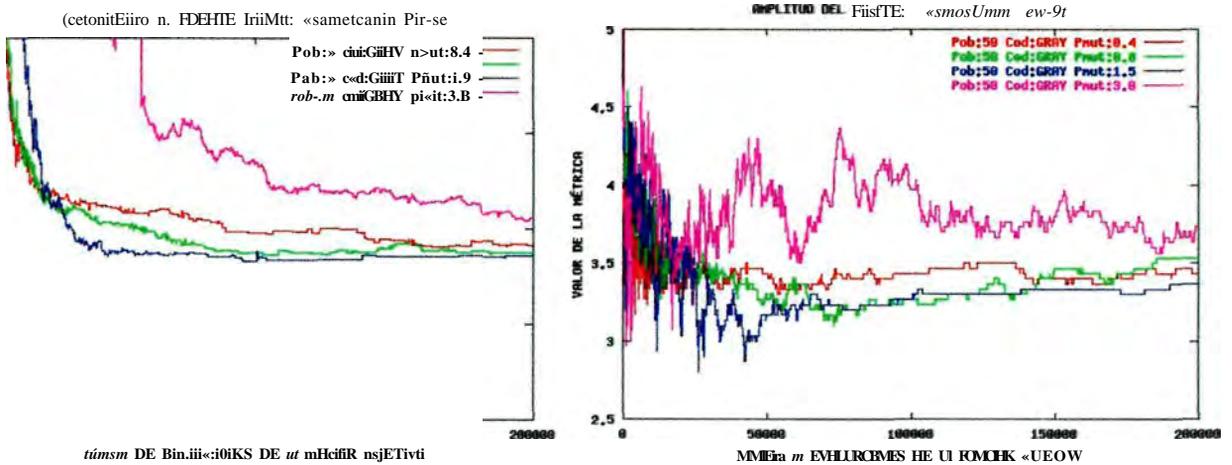


Figura 4.119. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.0$ con tamaño de población 50.

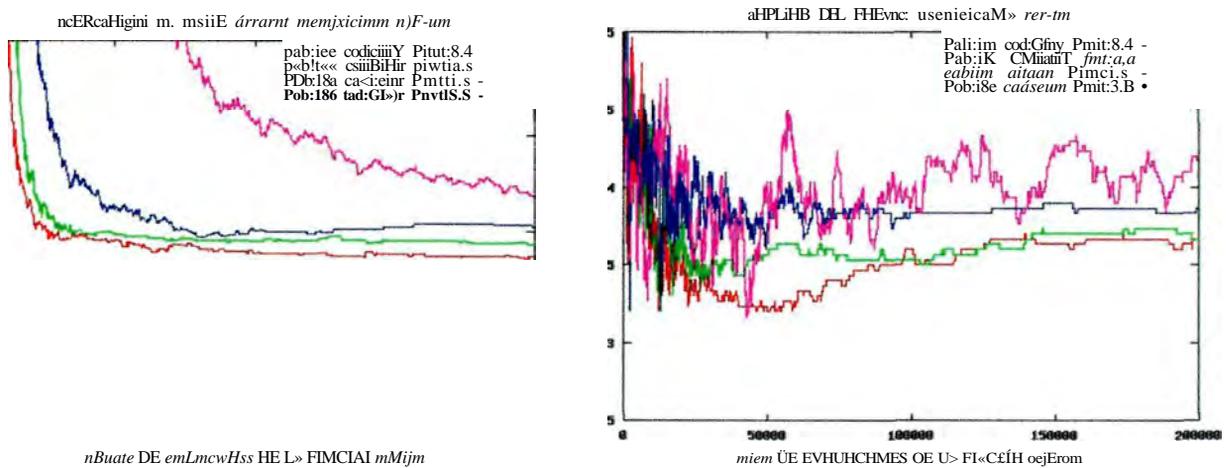


Figura 4.120. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.0$ con tamaño de población 100.

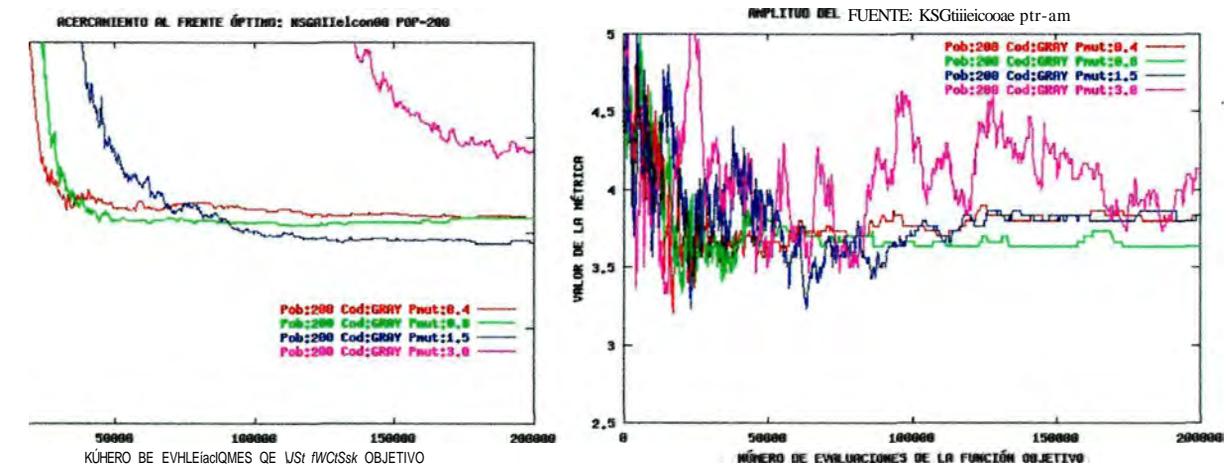


Figura 4.121. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.0$ con tamaño de población 200.

Se muestra en el algoritmo NsgaII-Elcon04 (generacional), con 3 tamaños de población en las figuras 4.122 a 4.124.

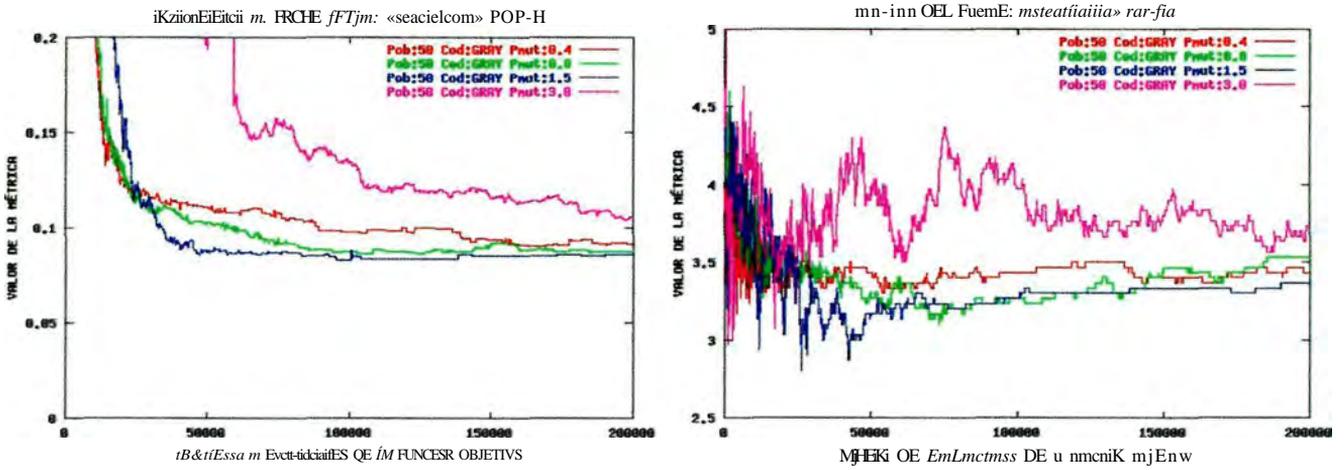


Figura 4.122. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 50.

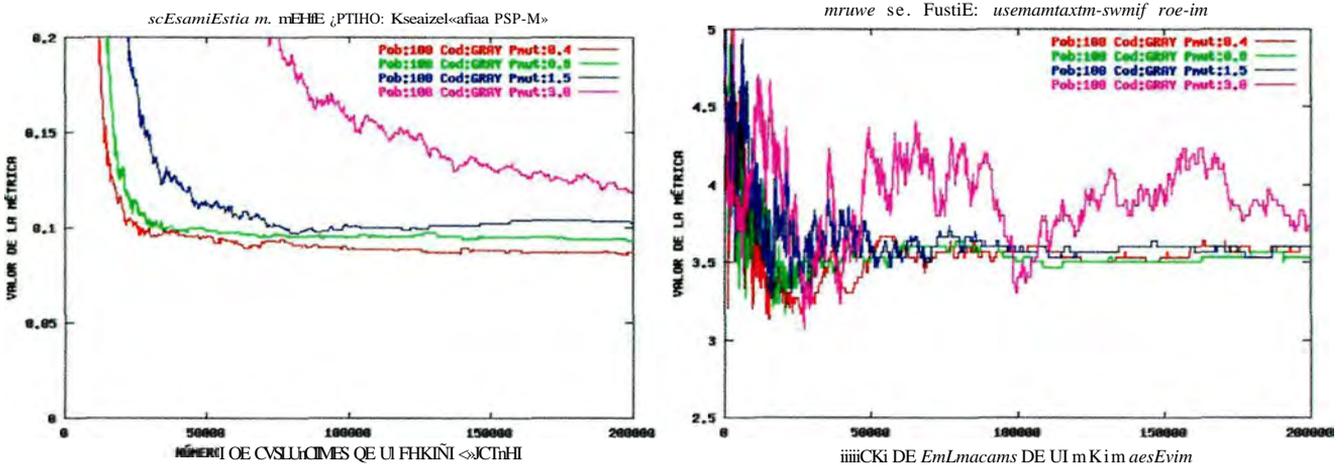


Figura 4.123. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 100.

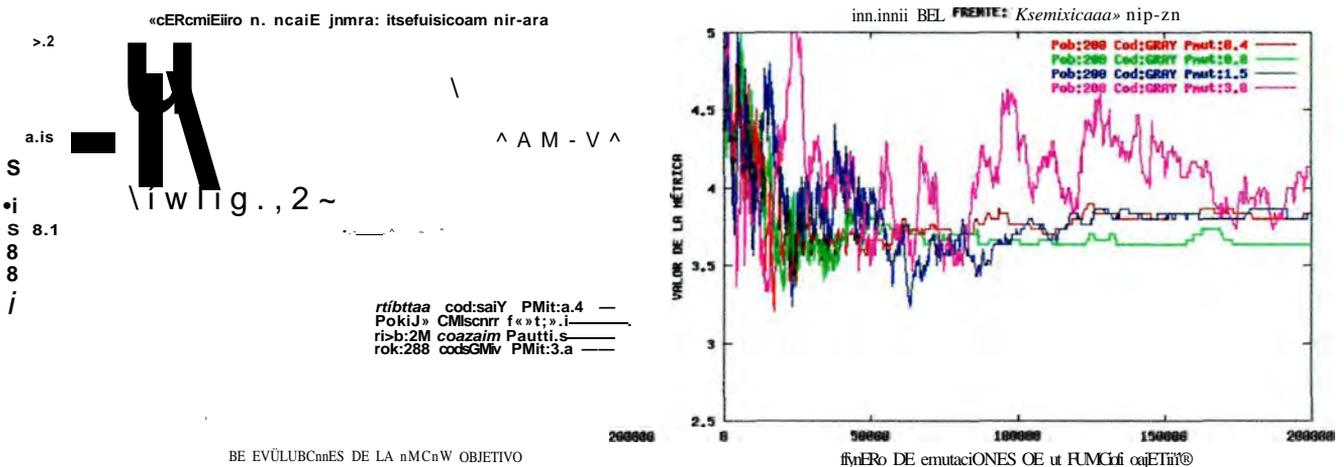


Figura 4.124. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 200.

Capítulo 4. Contribuciones

Se muestra en el algoritmo DENSEA (generacional), con 3 tamaños de población en las figuras 4.125 a 4.127.

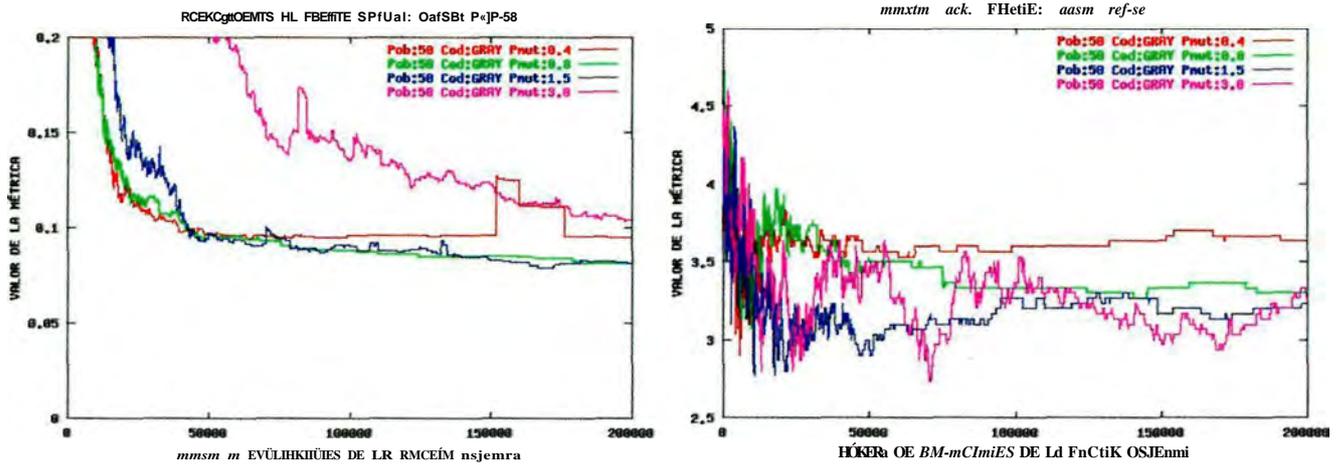


Figura 4.125. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo DENSEA con tamaño de población 50.

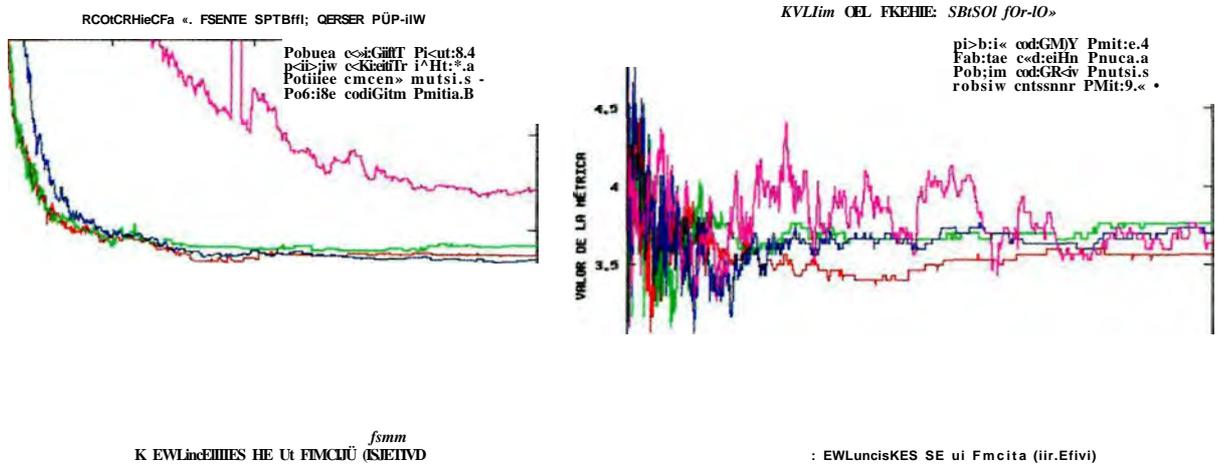


Figura 4.126. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo DENSEA con tamaño de población 100.

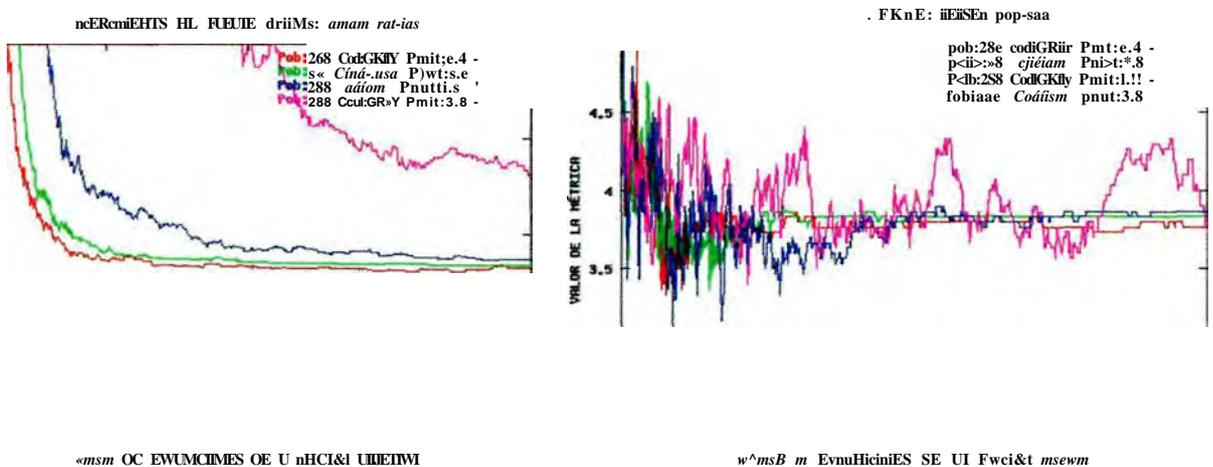


Figura 4.127. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento Generacional. Algoritmo DENSEA con tamaño de población 200.

Capítulo 4. Contribuciones

Se muestra en el algoritmo NSGAI (estado estacionario), con 3 tamaños de población en las figuras 4.134 a 4.136.

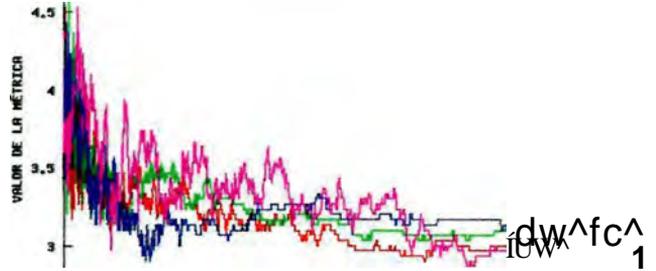
•KXHOIttBini «L FREKTE ÍPTEMI: KSaH-SIEmW Itlp-Sa

mpLiniB BEL FBEKTE; moKa-mEim rsr-vs

Pob:100 Cod:GRRY Pmut:0.4
 rob:» G«:aDnr i>wt:a,«
 rob:» cad:GHr pnuit:1.5

Fobisa cod:ciiiv r<:ut:B.4 -
 tsA'.S» Cod:OKIY fiiut:«.S
 rob:s8 cod:ei«v niut:1.5 -
 >ab:38 cad:cúin f>wt:3.a

- \hat{t} VW-,



KMEem BE CM(KIWI«JMS HE UI FI«CBM WSIETCm

: EVW«MCIBIES HE LK FaHcM OBJETVS

Figura 4.134. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con tamaño de población 50.

RcaiciMEKro «L FKOE éeioit mma-swmm nip-iss

AMPLITUD DEL fwrni: Rsam-STEnir rar-vm

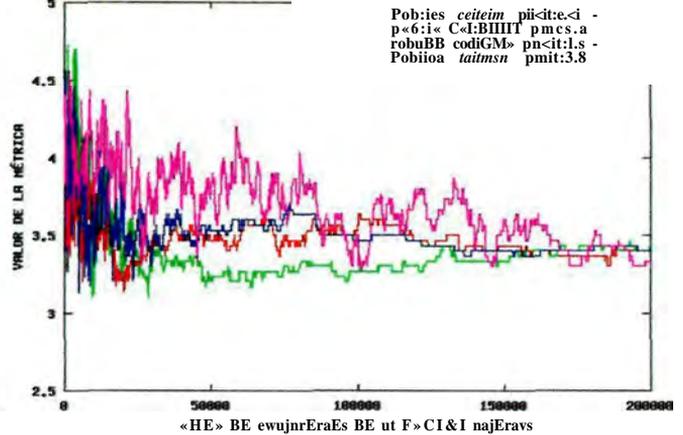
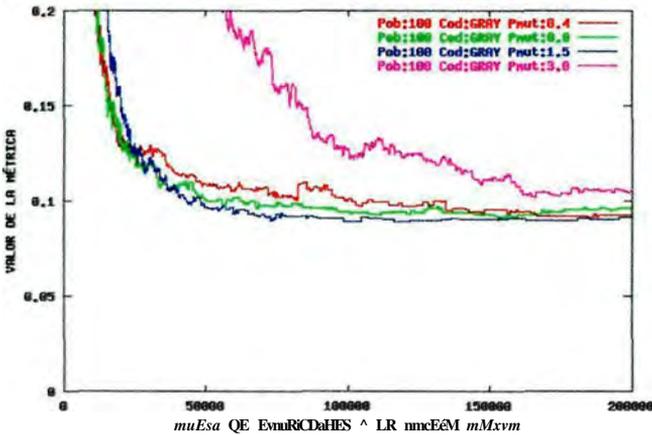


Figura 4.135. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con tamaño de población 100.

RCERCMEEni «. FUEKTE áprnu: Rsam-siEn» rat-^aa

mpLin» BEL FREmE: «saai-^wm» eae-iaa

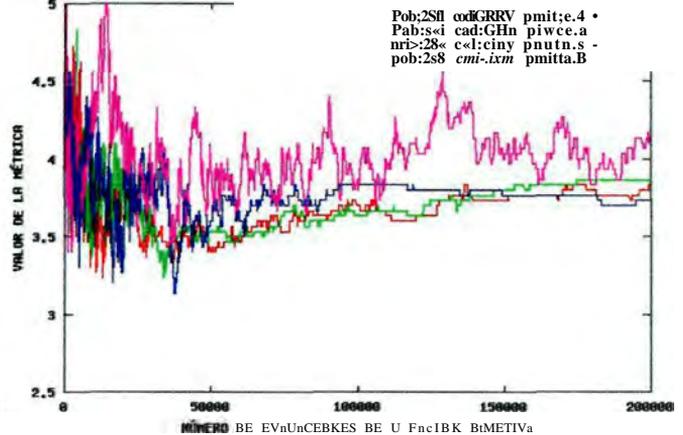
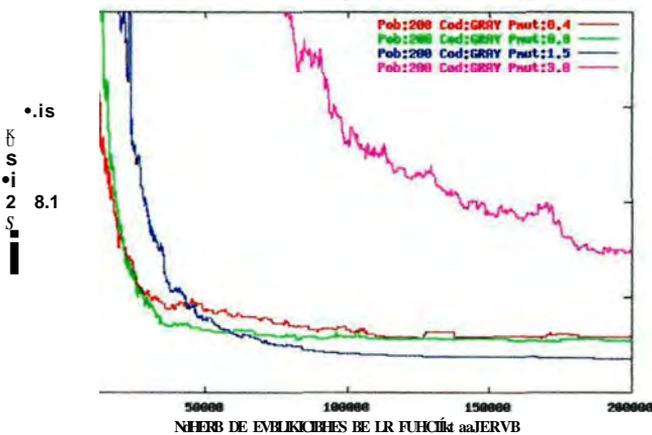


Figura 4.136. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con tamaño de población 200.

Capítulo 4. Contribuciones

Se muestra en el algoritmo NsgaII-ElconOO (steady-state), con 3 tamaños de población en las figuras 4.137 a 4.139.

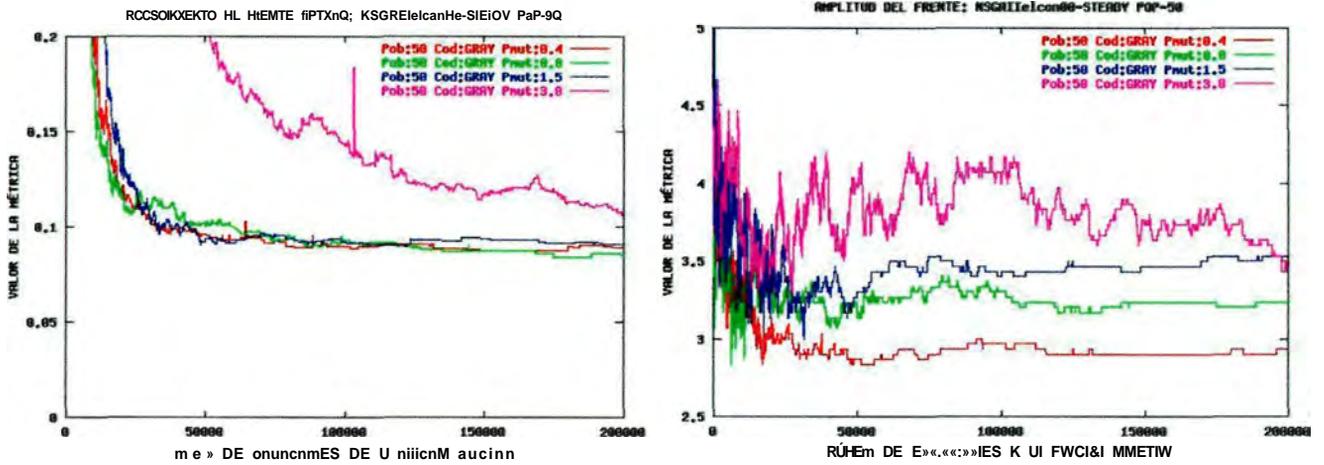


Figura 4.137. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con elitismo controlado y $r=0.0$ con tamaño de población 50.

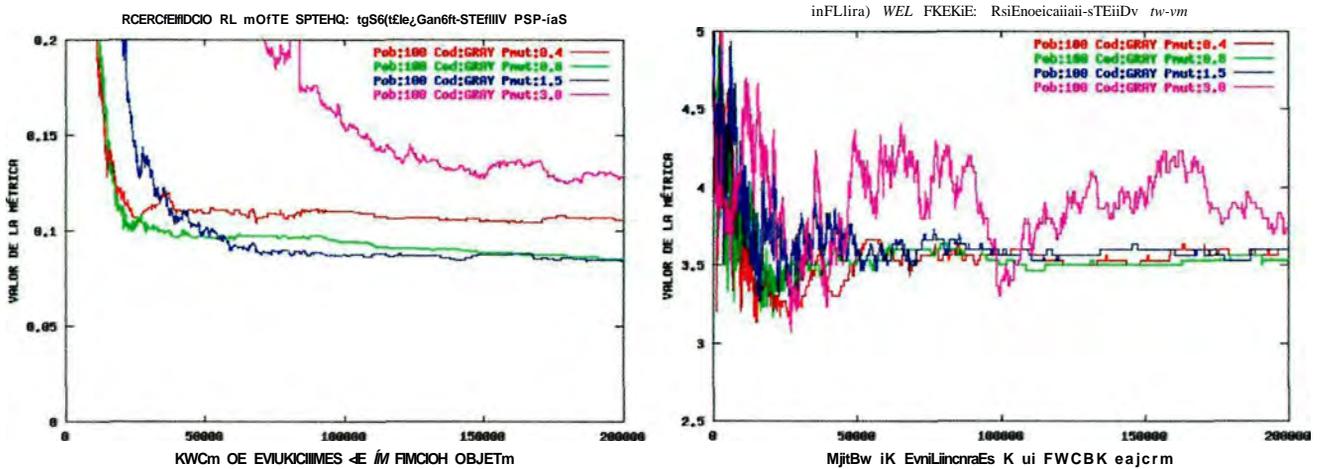


Figura 4.138. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con elitismo controlado y $r=0.0$ con tamaño de población 100.

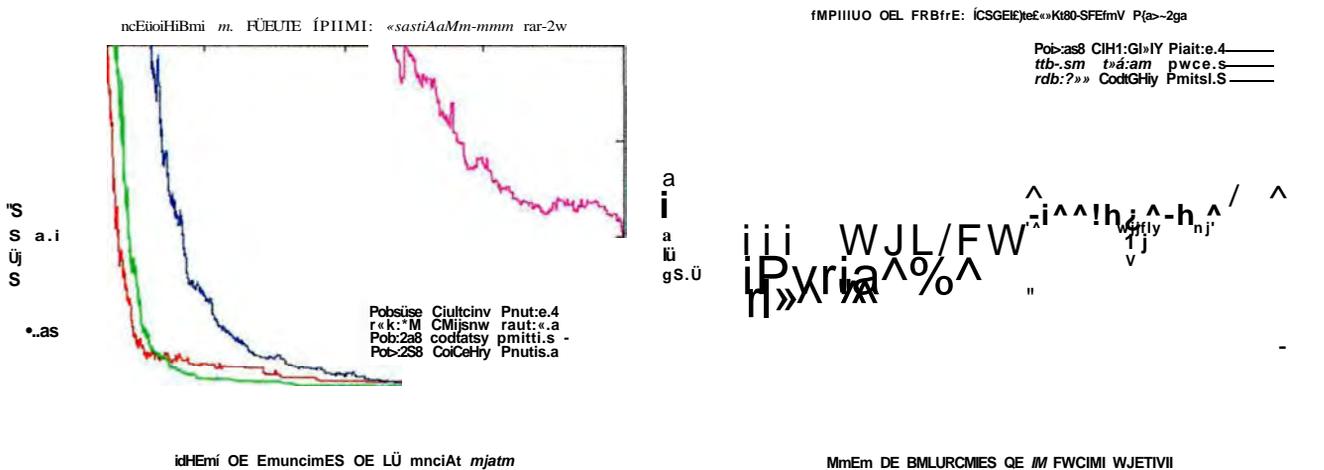


Figura 4.139. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con elitismo controlado y $r=0.0$ con tamaño de población 200.

Se muestra en el algoritmo NsgaII-Elcon04 (steady-state), con 3 tamaños de población en las figuras 4.140 a 4.142.

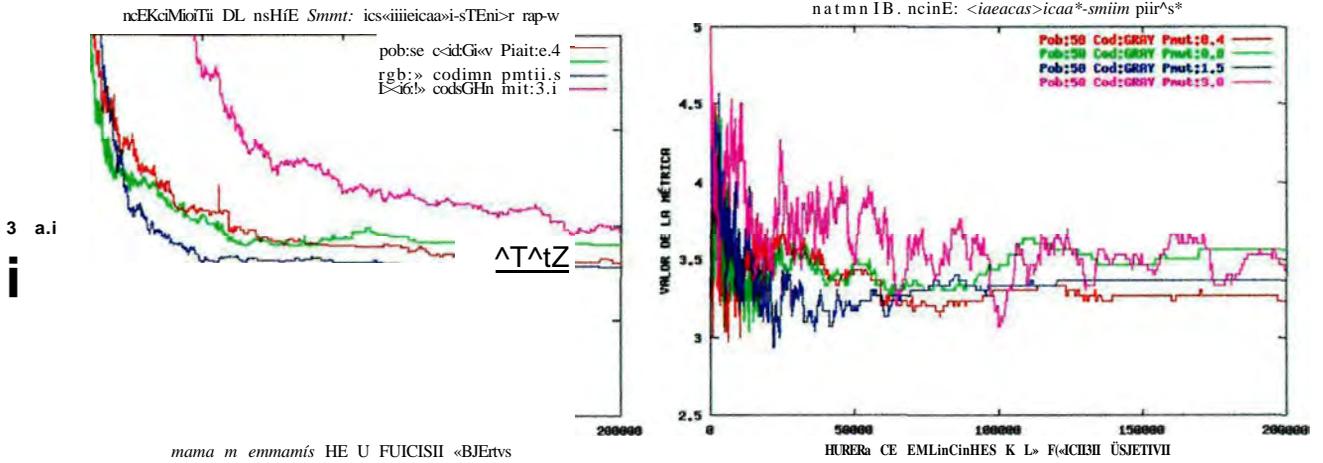


Figura 4.140. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 50.

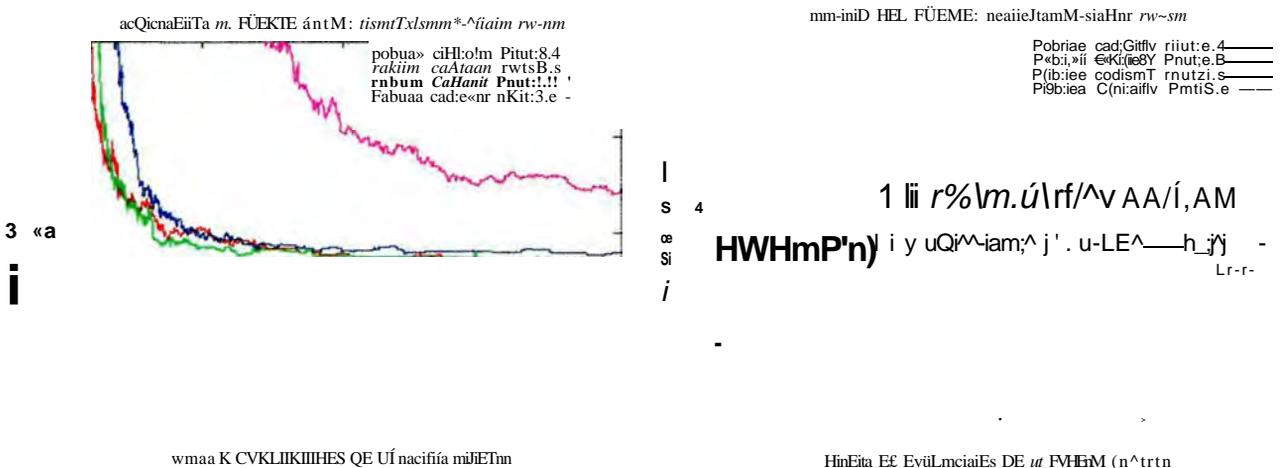


Figura 4.141. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 100.

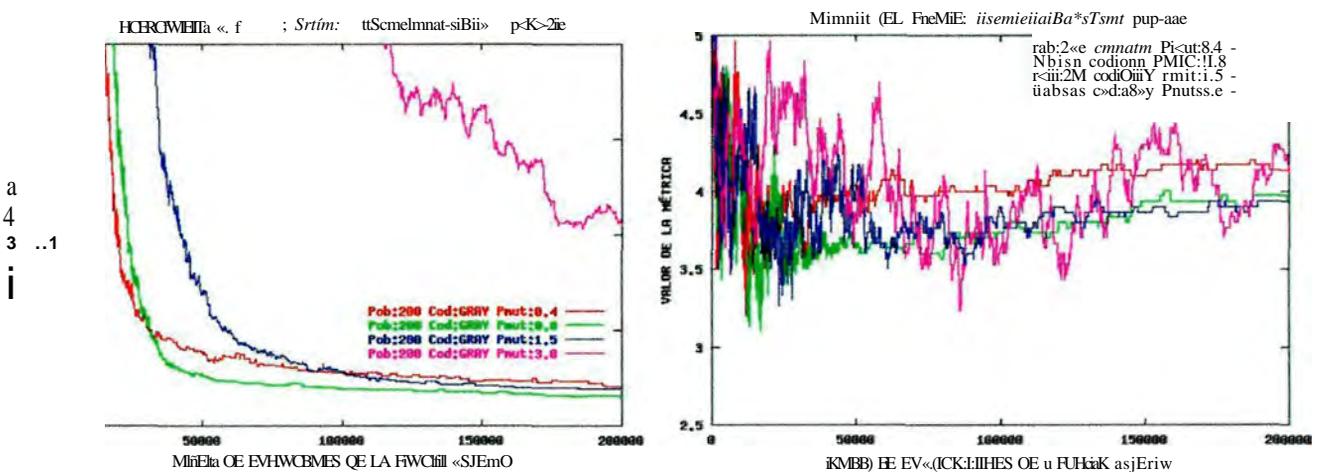


Figura 4.142. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo NSGAI con elitismo controlado y $r=0.4$ con tamaño de población 200.

Se muestra en el algoritmo DENSEA (steady-state), con 3 tamaños de población en las figuras 4.143 a 4.145.

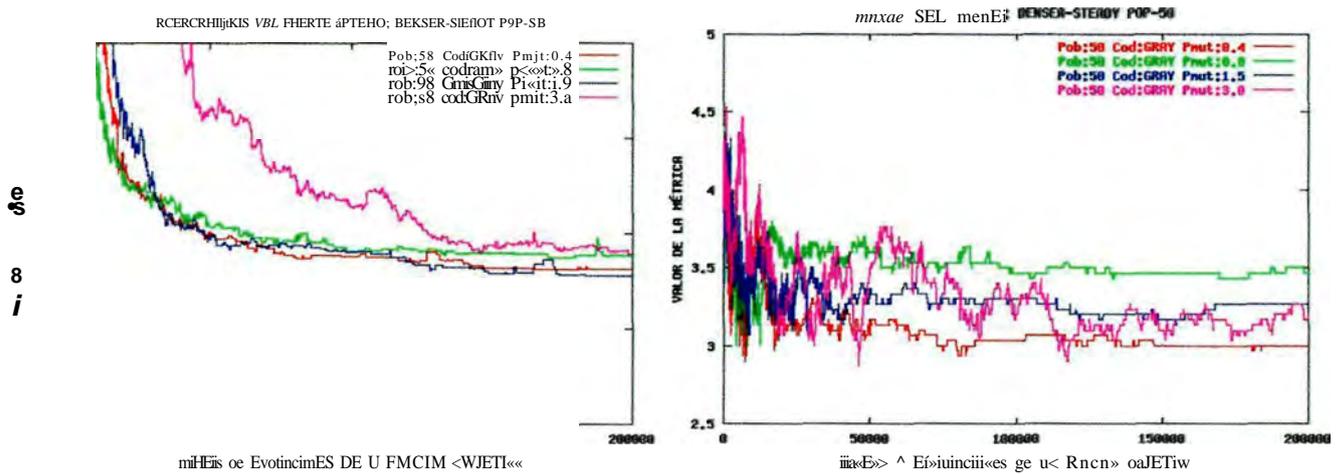


Figura 4.143. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA con tamaño de población 50.

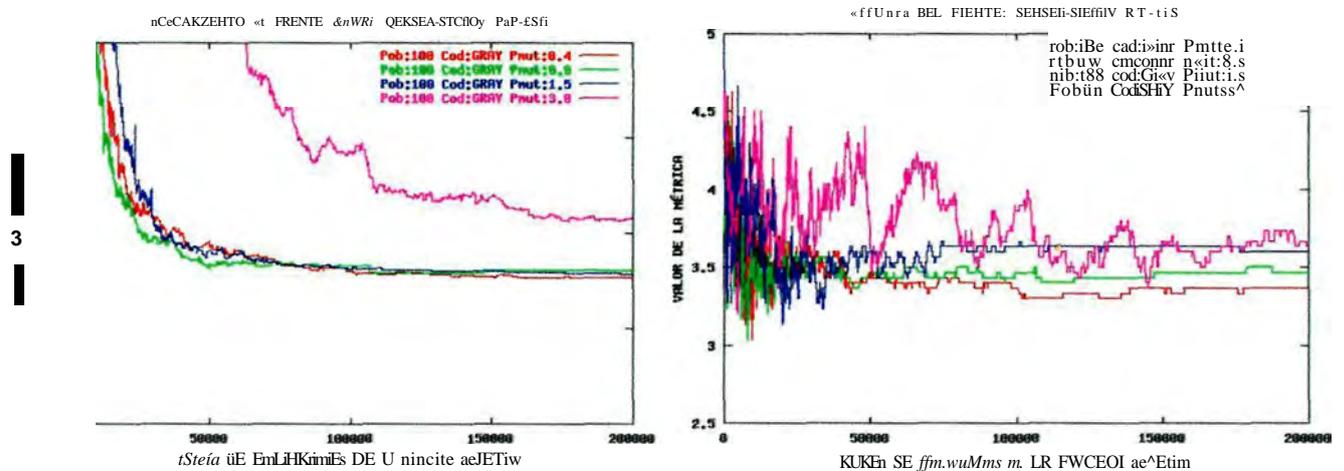


Figura 4.144. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA con tamaño de población 100.

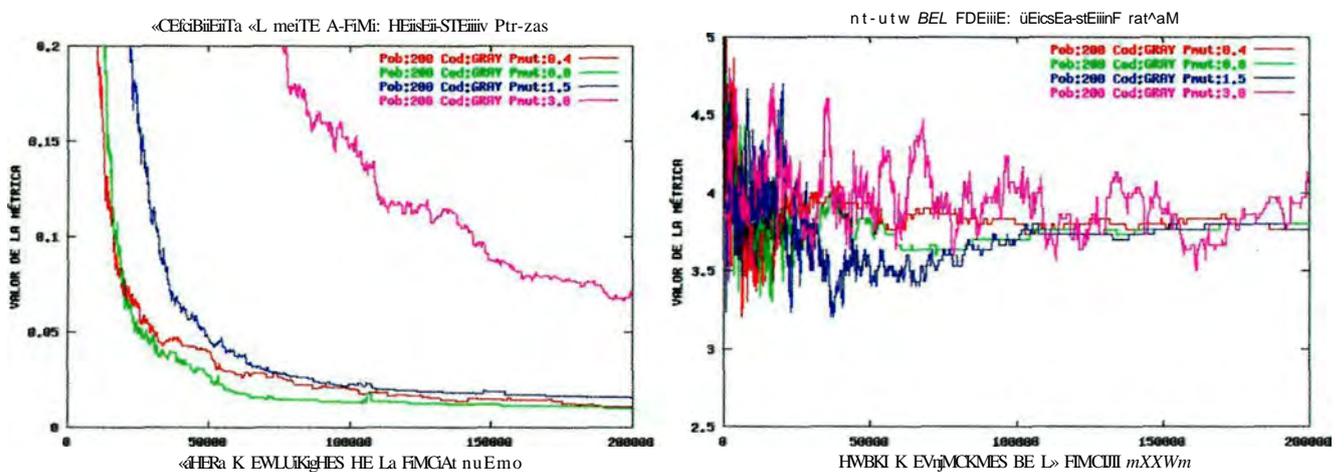


Figura 4.145. Métricas en comparativa de probabilidades de mutación en Caso Y. Reemplazamiento EstadoEstacionario. Algoritmo DENSEA con tamaño de población 200.

Caso X) Diferentes algoritmos: Probabilidad Mutación 0.8%; Estrategia Generacional (4.151 a 4.153)

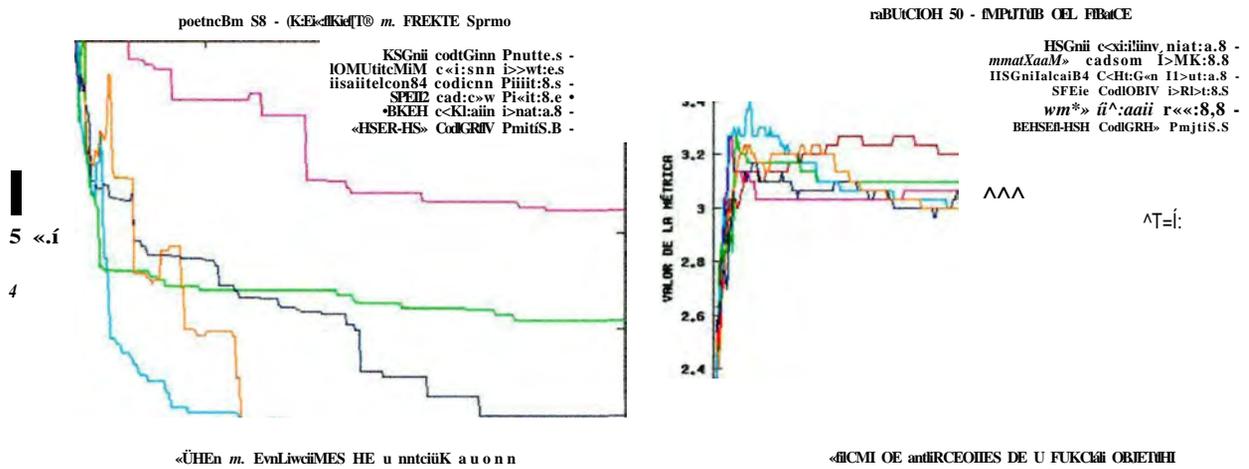


Figura 4.151. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 0.8% y tamaño de población 50.

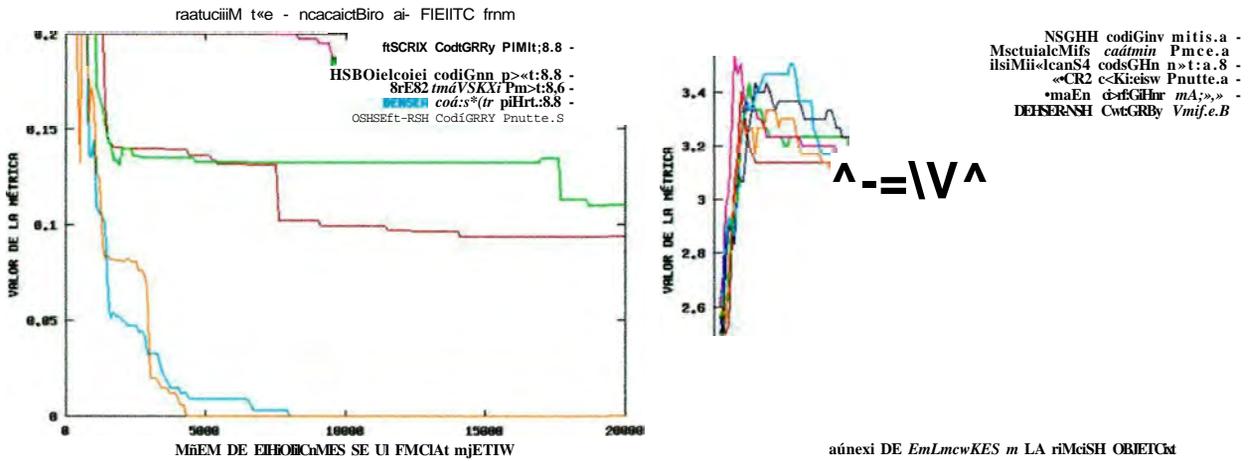


Figura 4.152. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 0.8% y tamaño de población 100.

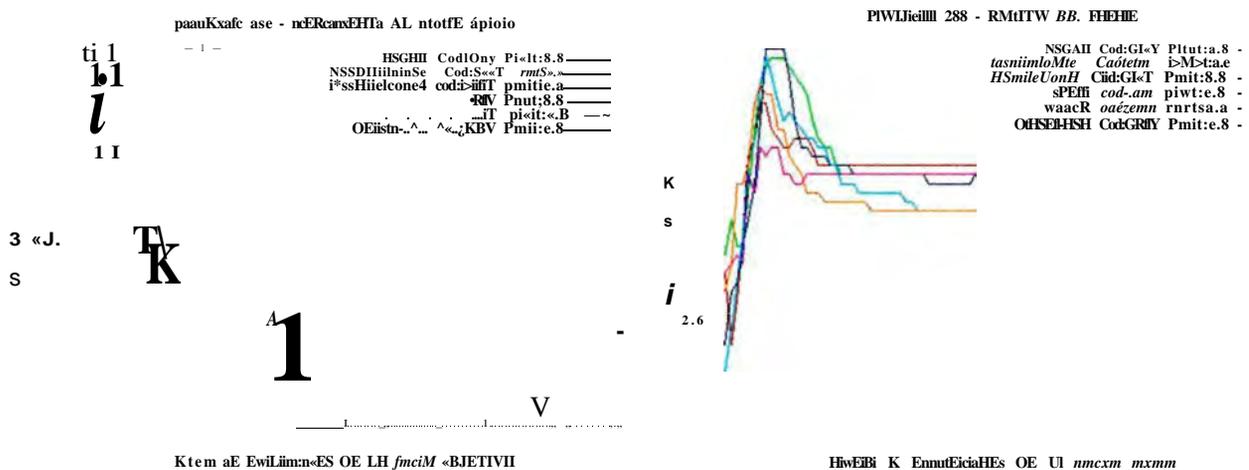


Figura 4.153. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 0.8% y tamaño de población 200.

Caso X) Diferentes algoritmos: Probabilidad Mutación 1.5%; Estrategia Generacional (4.154 a 4.156)

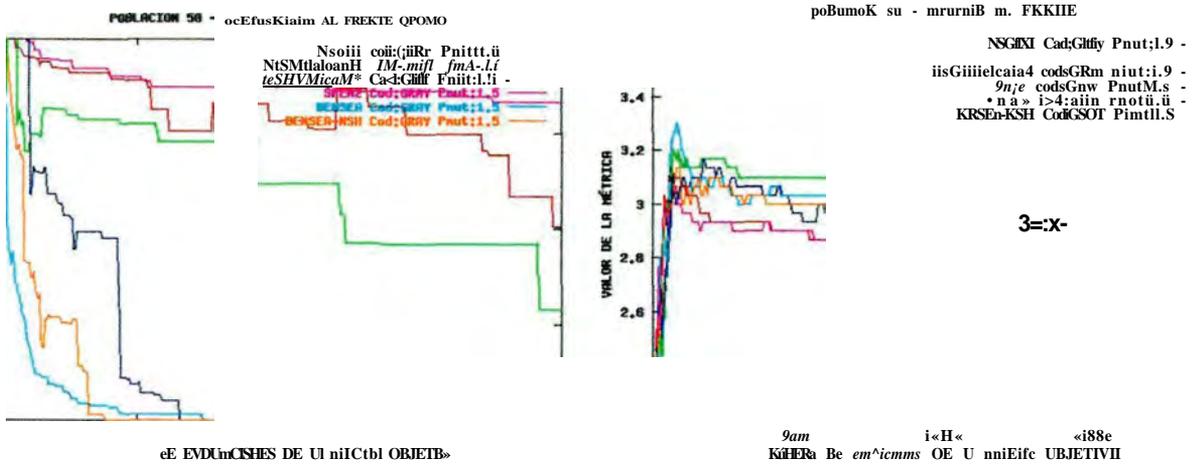


Figura 4.154. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 1.5% y tamaño de población 50.

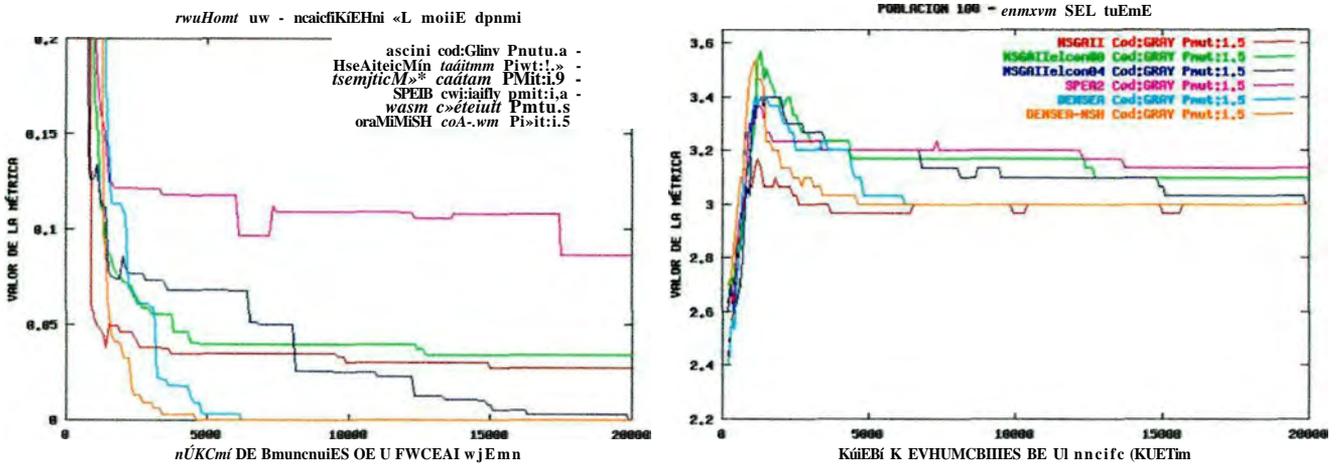


Figura 4.155. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 1.5% y tamaño de población 100.

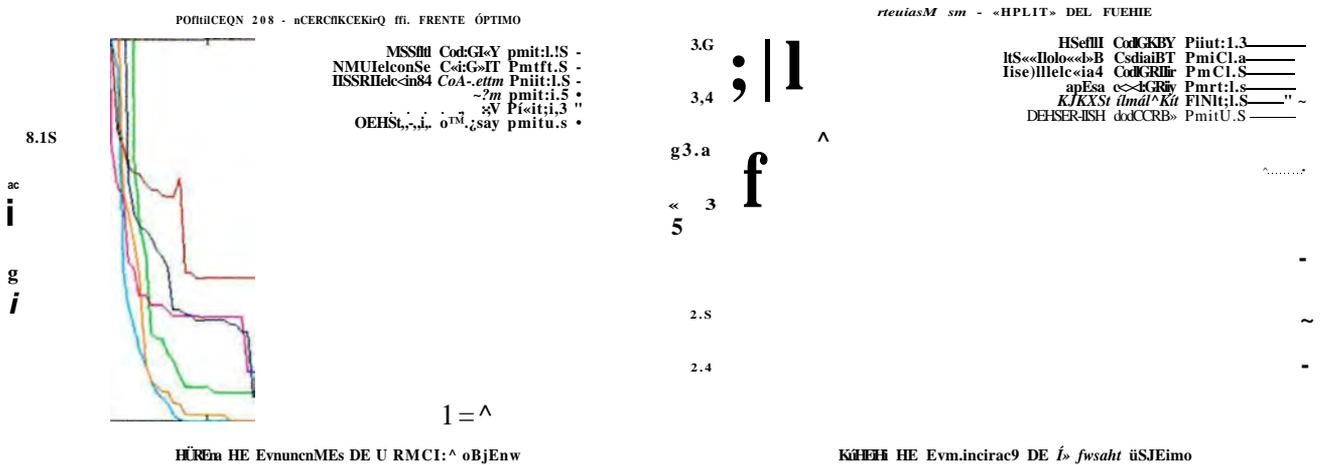


Figura 4.156. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 1.5% y tamaño de población 200.

Caso X) Diferentes algoritmos: Probabilidad Mutación 3%; Estrategia Generacional (4.157 a 4.159)

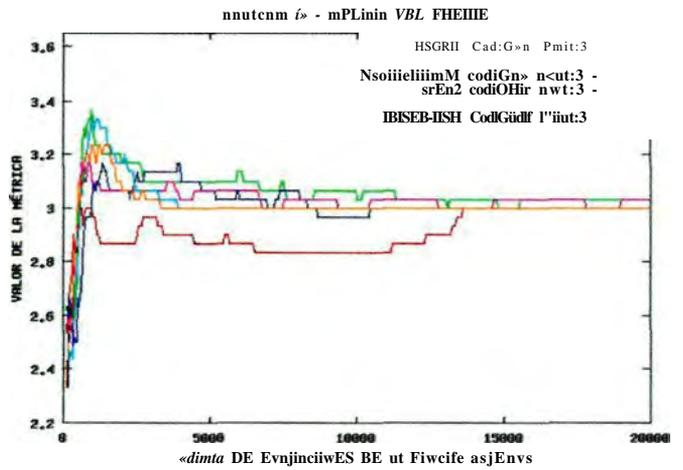
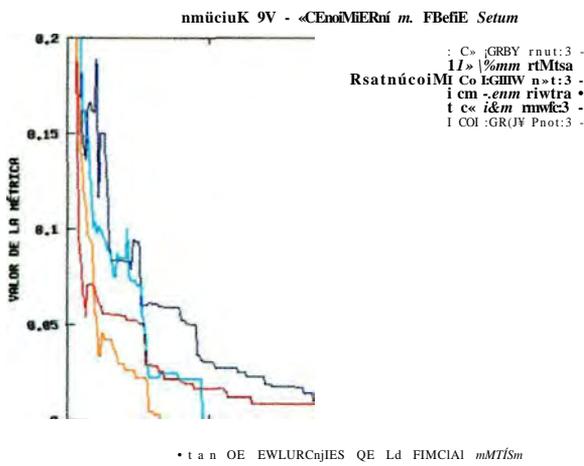


Figura 4.157. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 3.0% y tamaño de población 50.

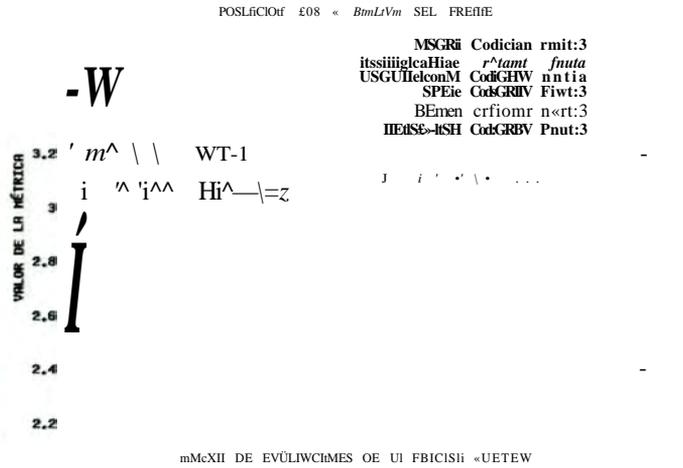
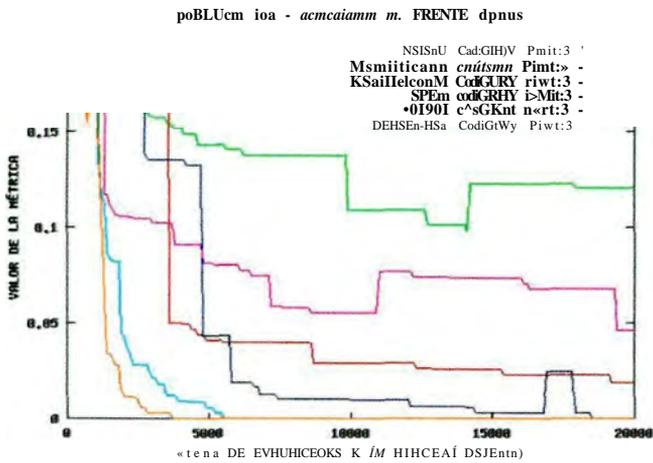


Figura 4.158. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 3.0% y tamaño de población 100.

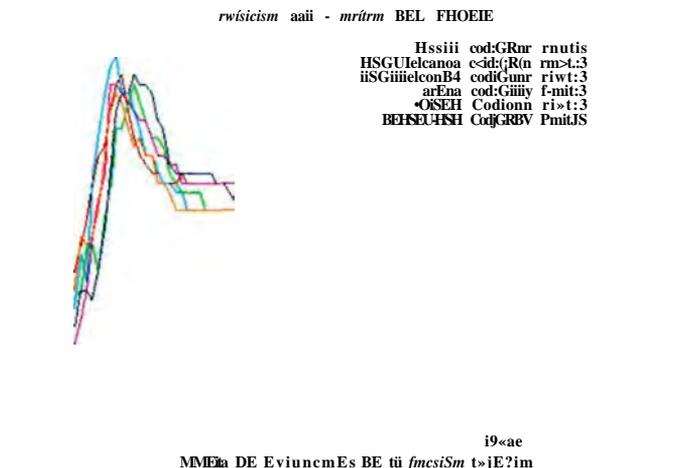
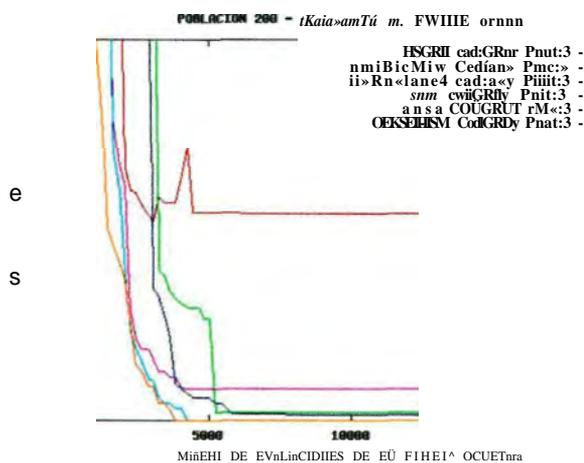


Figura 4.159. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento Generacional. Probabilidad de mutación 3.0% y tamaño de población 200.

Caso X) Diferentes algoritmos: Probabilidad Mutación 0.8%; Estrategia Steady-State (4.163 a 4.165)

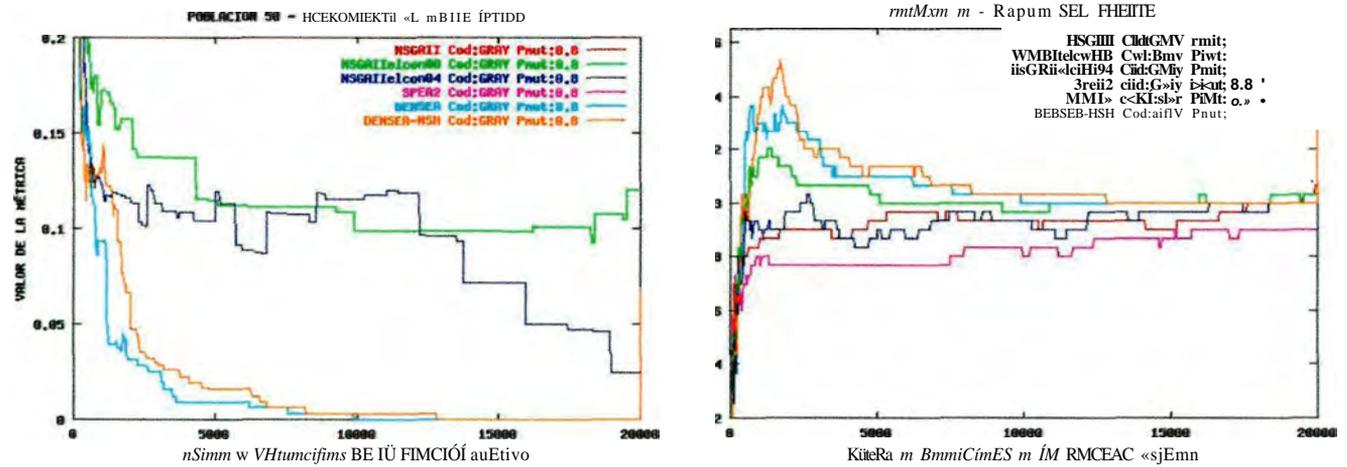


Figura 4.163. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 0.8% y tamaño de población 50.

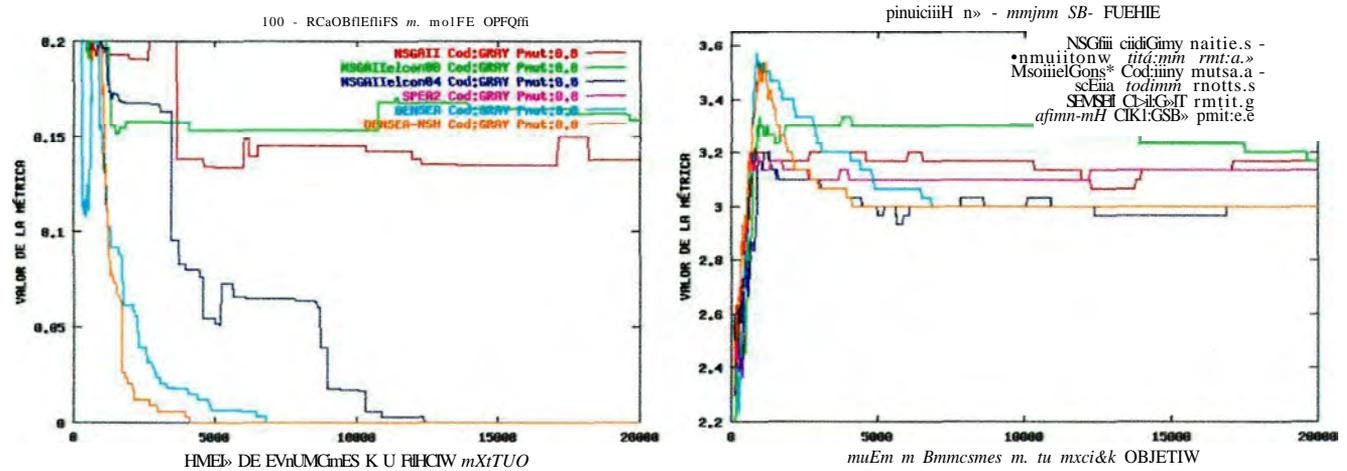


Figura 4.164. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 0.8% y tamaño de población 100.

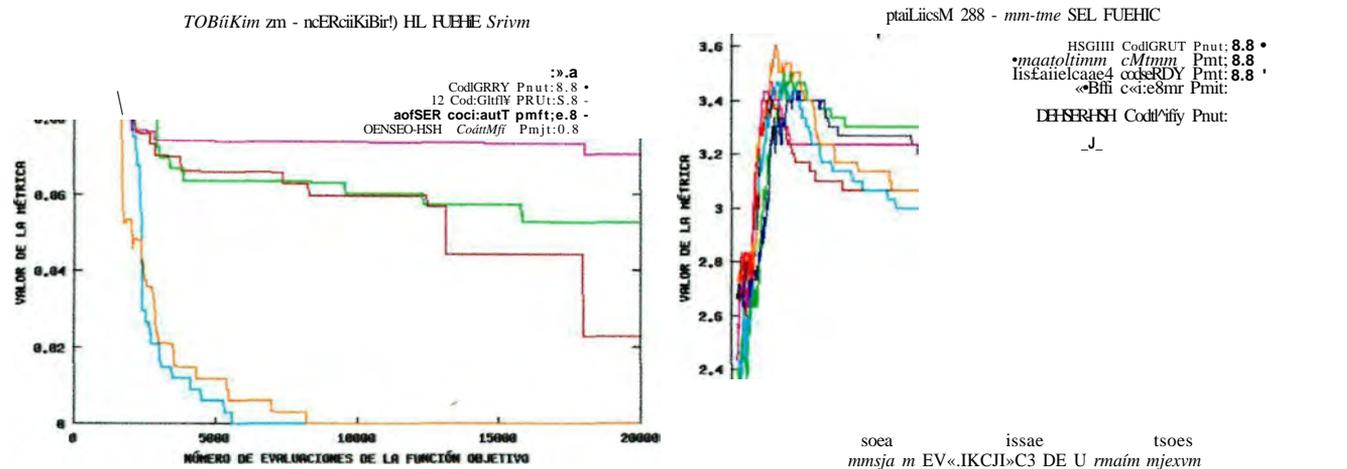


Figura 4.165. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 0.8% y tamaño de población 200.

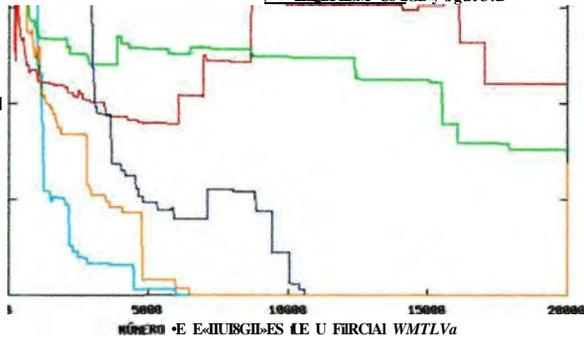
Caso X) Diferentes algoritmos: Probabilidad Mutación 1.5%; Estrategia Steady-State (4.166 a 4.168)

poemciai 98 - RcetawzacTa SL FICHIE ÜPOMO

```

ICSntl Cod:0)V Pnuitü
HsMJiiloanw CKI:!!>» nwtm.s
Kii>iietcomM codsemIT f>nit:i.s
a>Efi Codiamv Pmit:i:!! -
    
```

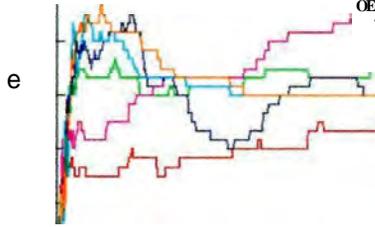
HEBLIISM Co:GBw JgutU.B



fmkJHiiim 38 - mFLiira ge. HKHIE

```

RSeRII Cod:GHIV Pilot:!! !
teatxhym e<:ar>nr rnit:i:ii
mmaxlm)* tadatm muti:s
arena codGRHY fmit:i:!!
oiaen QxiSim? rnauc.s
OEHSEIISH Cod:Solny-
    
```



imEm BE EvnuRetraES se ui FIKICIÍM

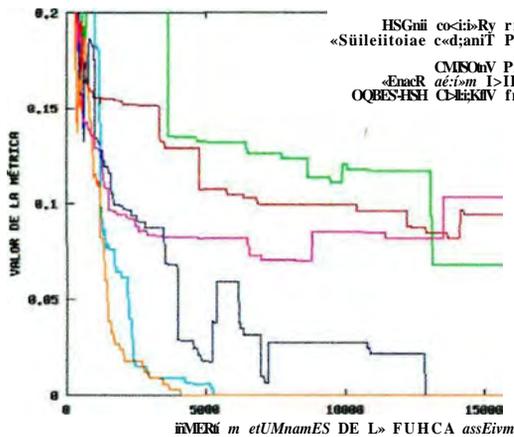
Figura 4.166. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 1.5% y tamaño de población 50.

FOUACOR ma - ocEsumEEw m. FIEITE áPTiite

```

HSGñi co:i>Ry rntu:s
«Süileitoiae c<:aniT Piwü.s
CMISOMV PnuitLS
«EncR ac:om I>IM:1.9 -
OQBESISH Co:EKIV fauta.S
    
```

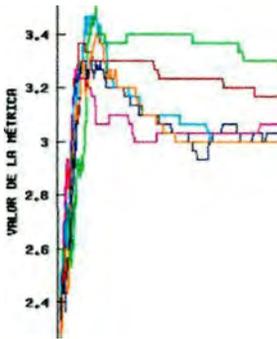
iñERi m etUMnamES DE L» FUHCA assEivm



raauiCiiim VI - «rtiisB na. FIEiite

```

HSGñi CodiatBY Pnut:1.9
iBCinislewin Mdiotin rmtu:s
«SanieiconiH Cod:OBV Pnuit.S
SPEK2 Cod:KBY Pmitü.S
véneta c<i:a>f matn.i
MUSEB-IISH Co>:diIW Pnut:1.5
    
```



X
=tr-

masm m enjumma K ÍM nwerfe OBJETIW

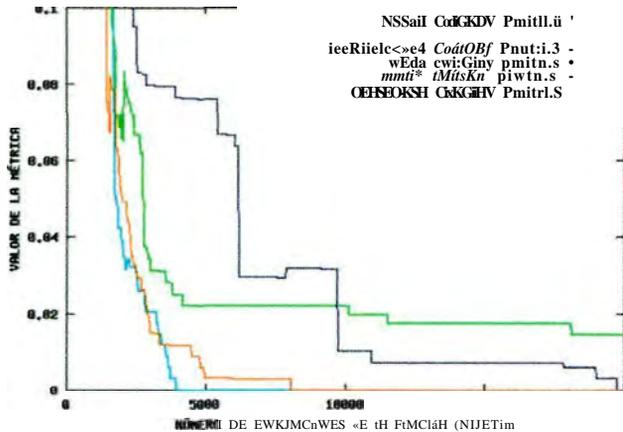
Figura 4.167. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 1.5% y tamaño de población 100.

iiiauen 288 - MXKñiEHr» n. FREÜTE OPTIMI

```

NSSaii Cod:KDV Pmitü.ü
ieeRiieic<»e4 Cod:OBf Pnut:i.3 -
wEda ewi:Giny pmitn.s
mmü* iMitsKn piwtn.s -
OEHSEIISH Co:GRHY PmitrL.S
    
```

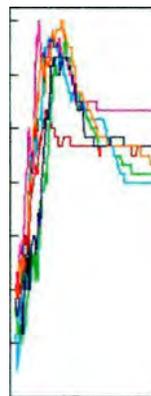
NÚMERO DE EJECUCIONES DE LA FICHA WMTLVa



pntAcin). aot - «nutn SEL FREKIE

```

RSSDII Cod:GRRY l-mit:!! !
eeneilcaoM CadiGn» rmitL.S -
sraa cod:GMY cmitü.s -
BEeER cmJiasBY rinrt:i.s
DEHSEIISH Co:GRHY Pmit:1.5 -
T -
    
```



Mué» HE EV(timErQHES DE Ur FMCISH «tiEro»

Figura 4.168. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 1.5% y tamaño de población 200.

Caso X) Diferentes algoritmos: Probabilidad Mutación 3%; Estrategia Steady-State (4.169 a 4.171)

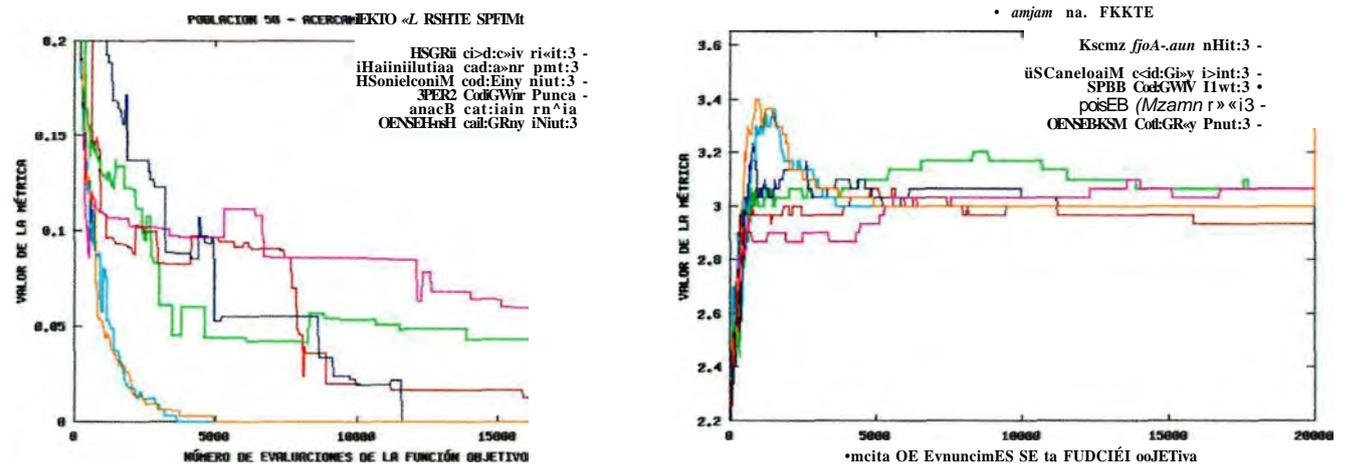


Figura 4.169. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 3.0% y tamaño de población 50.

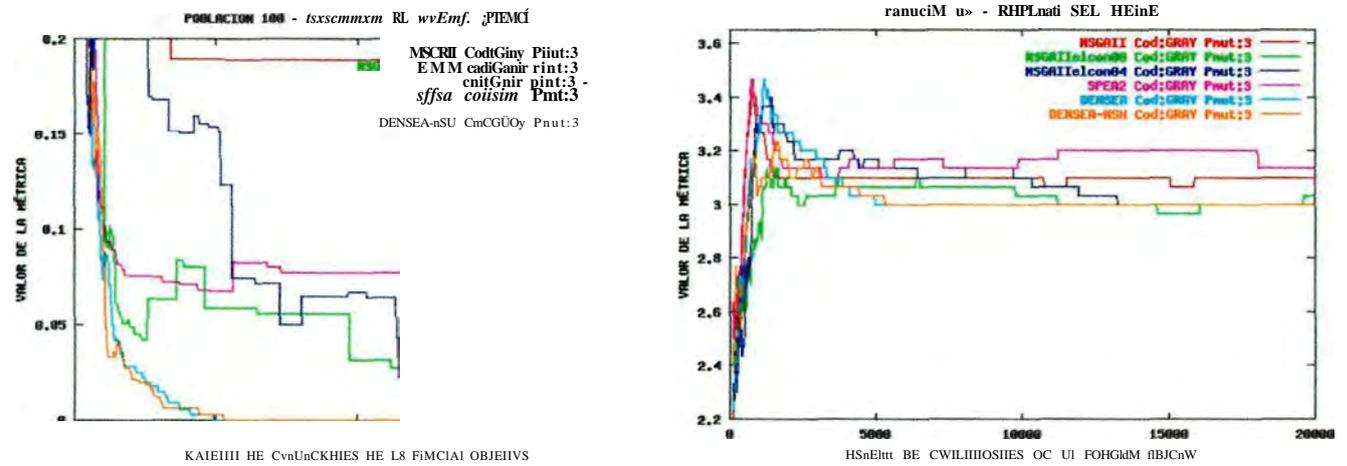


Figura 4.170. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 3.0% y tamaño de población 100.

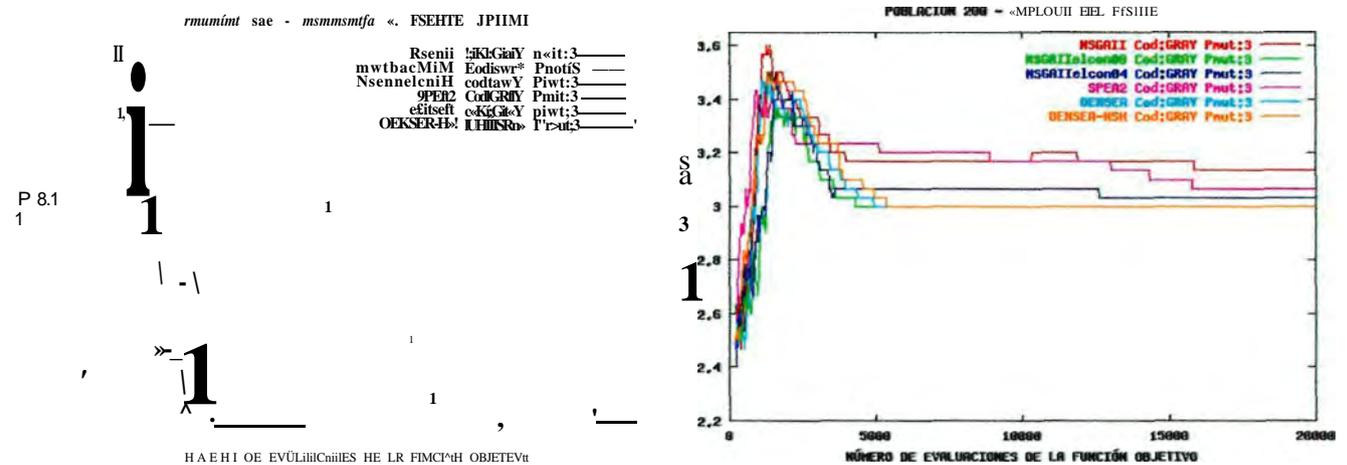


Figura 4.171. Métricas en comparativa de algoritmos en Caso X. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 3.0% y tamaño de población 200.

Caso Y) Diferentes algoritmos: Probabilidad Mutación 1.5%; Estrategia Generacional (4.181 a 4.183)

i

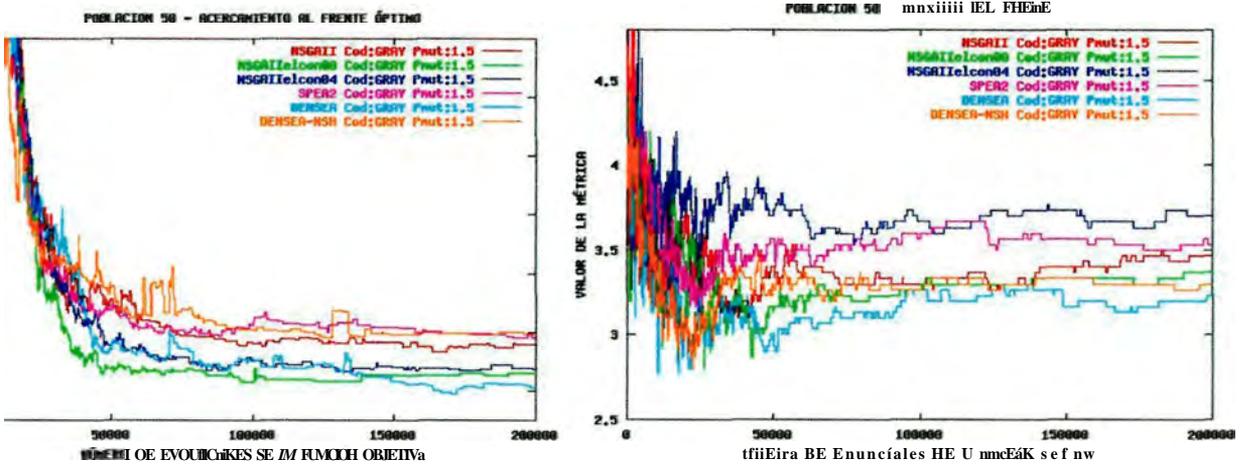


Figura 4.181. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 1.5% y tamaño de población 50.

S B.12

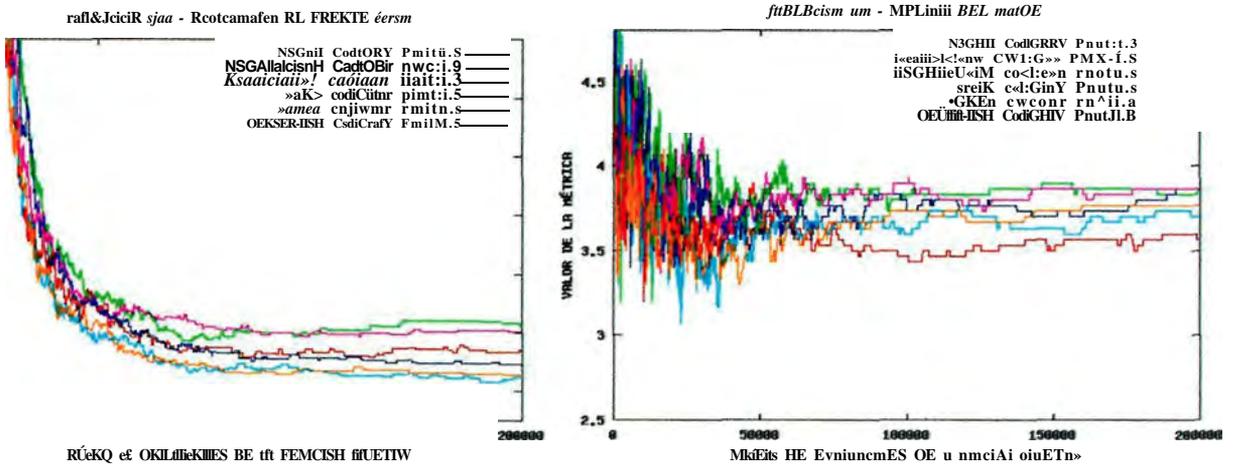


Figura 4.182. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 1.5% y tamaño de población 100.

e «.1»

S

S «.12

i

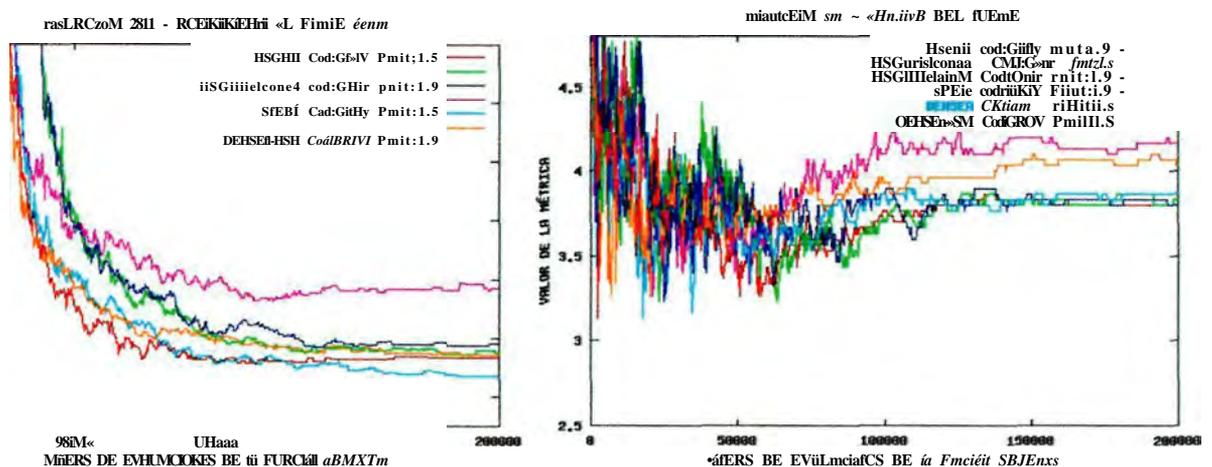


Figura 4.183. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 1.5% y tamaño de población 200.

Caso Y) Diferentes algoritmos: Probabilidad Mutación 3% ; Estrategia Generacional (4.184 a 4.186)

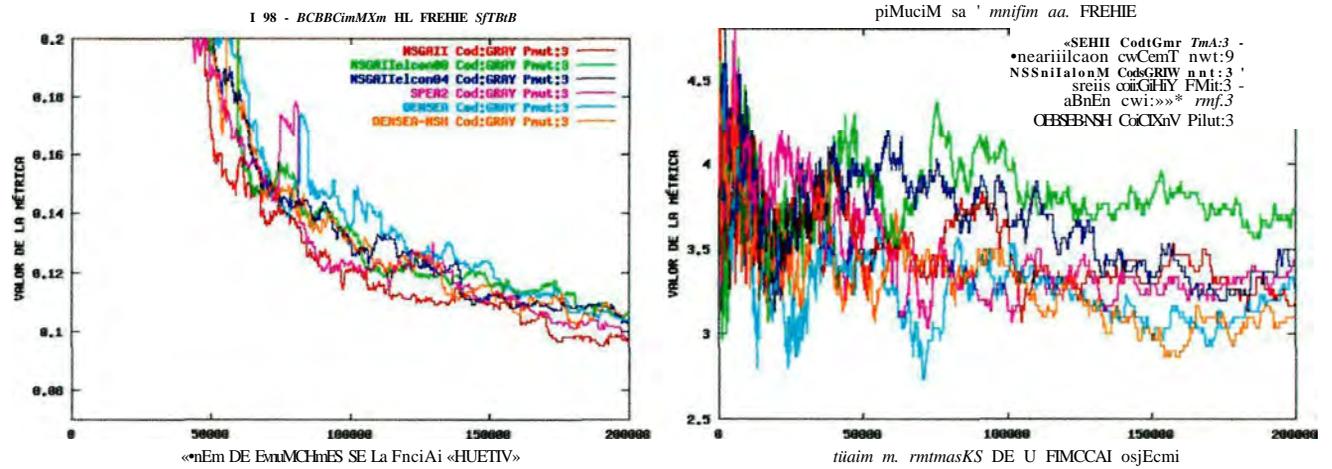


Figura 4.184. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 3.0% y tamaño de población 50.

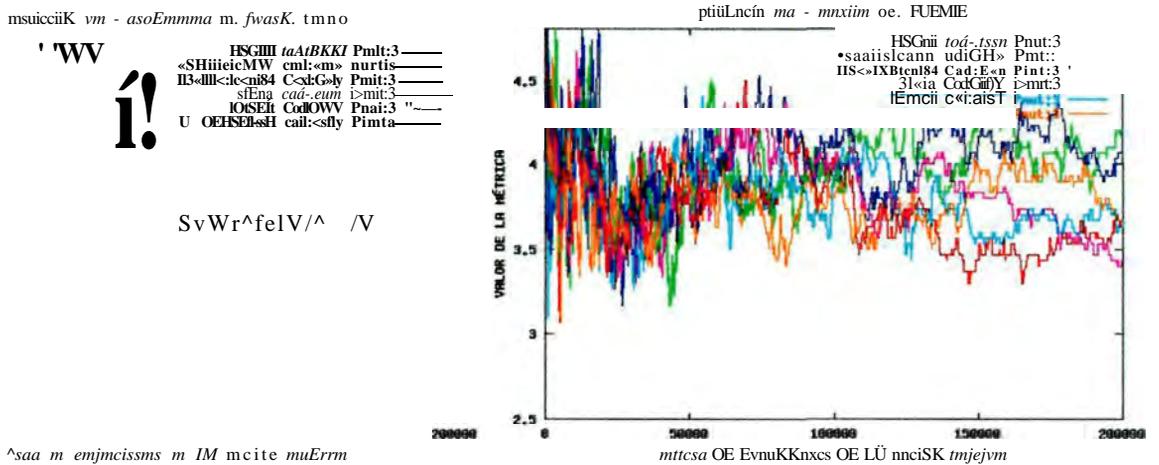


Figura 4.185. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 3.0% y tamaño de población 100.

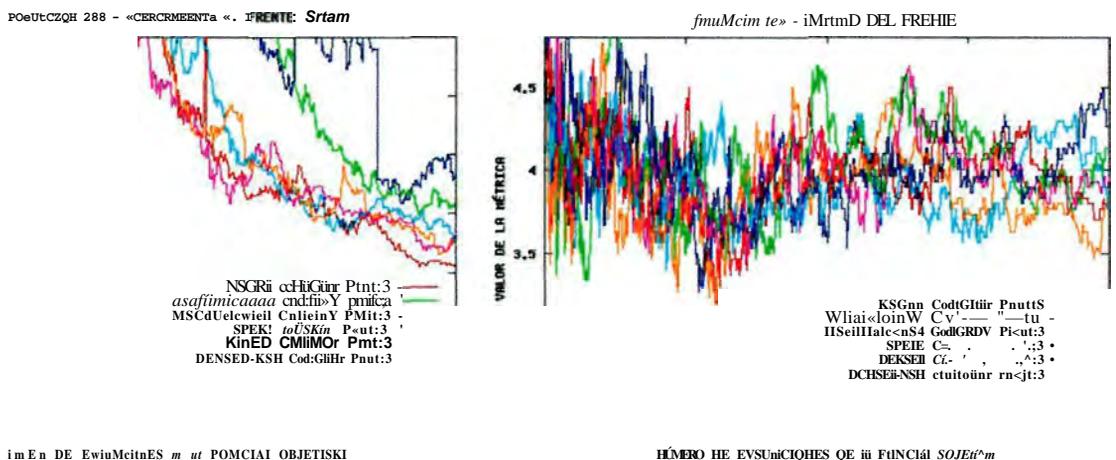


Figura 4.186. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento Generacional. Probabilidad de mutación 3.0% y tamaño de población 200.

Caso Y) Diferentes algoritmos: Probabilidad Mutación 1.5%; Estrategia Steady-State (4.193 a 4.195)

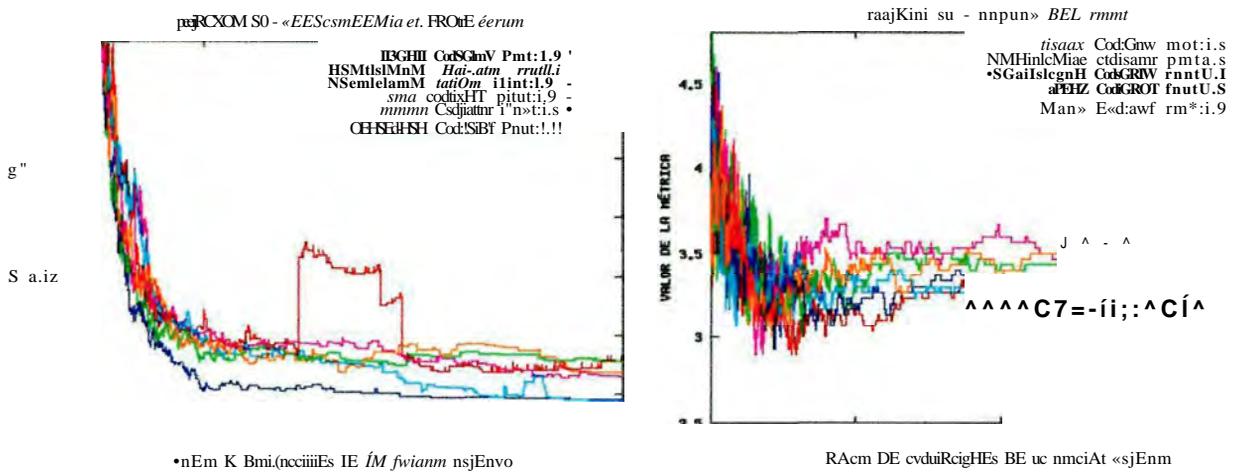


Figura 4.193. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 1.5% y tamaño de población 50.

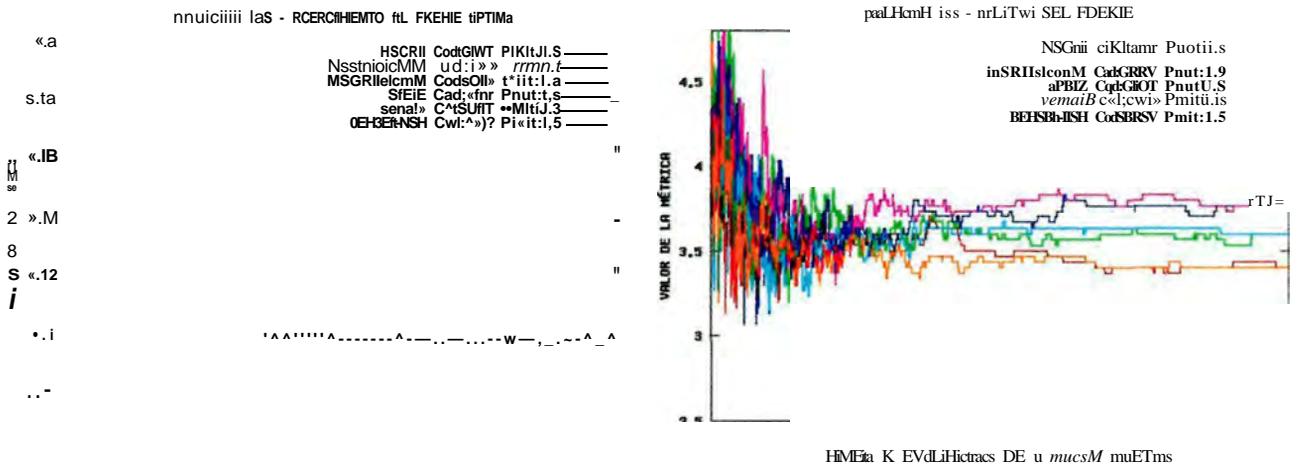


Figura 4.194. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 1.5% y tamaño de población 100.

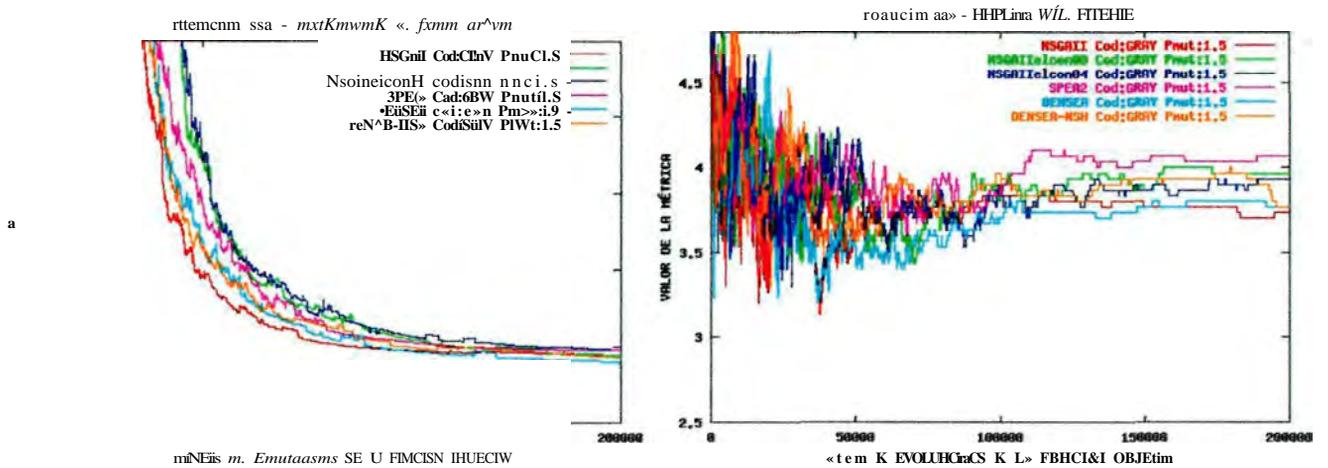


Figura 4.195. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 1.5% y tamaño de población 200.

Caso Y) Diferentes algoritmos: Probabilidad Mutación 3%; Estrategia Steady-State (4.196 a 4.199)

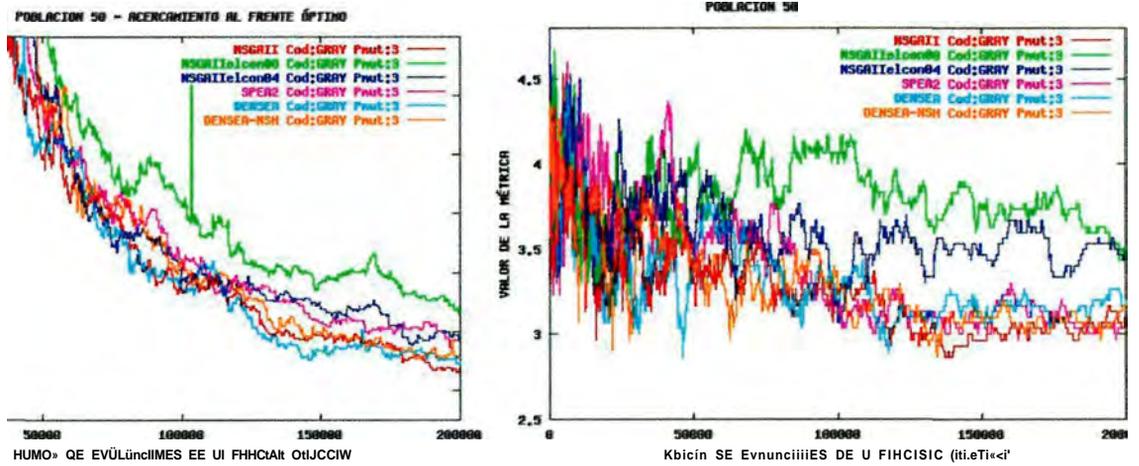


Figura 4.196. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 3.0% y tamaño de población 50.

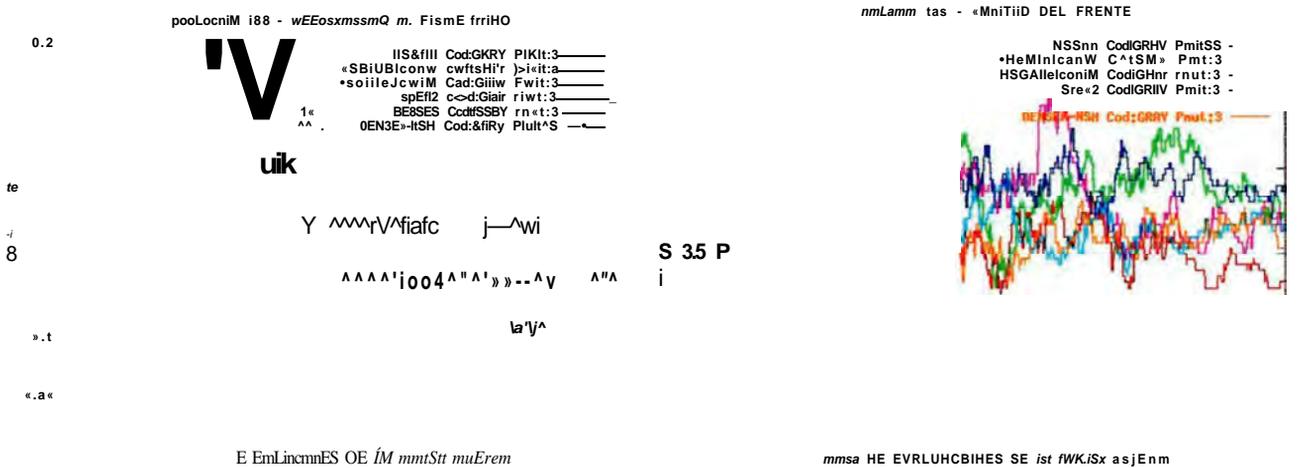


Figura 4.197. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 3.0% y tamaño de población 100.

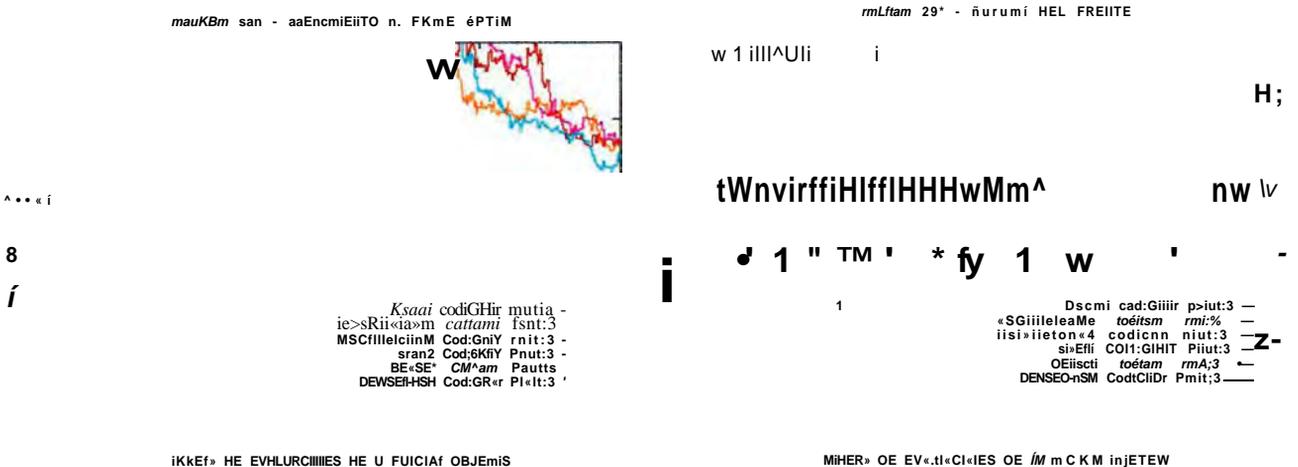


Figura 4.198. Métricas en comparativa de algoritmos en Caso Y. Reemplazamiento EstadoEstacionario. Probabilidad de mutación 3.0% y tamaño de población 200.

Caso X) Comparativa Generacional / SteadyState:

Métricas de acercamiento al frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 0.8%:

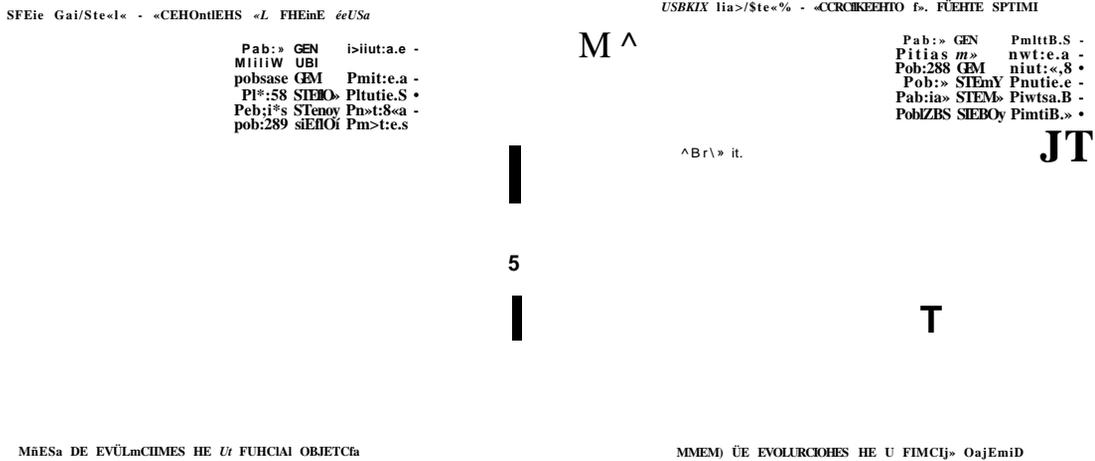


Figura 4.205. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en CasoX. Probabilidad de mutación 0.8%. Algoritmos SPEA2 y NSGAI1.

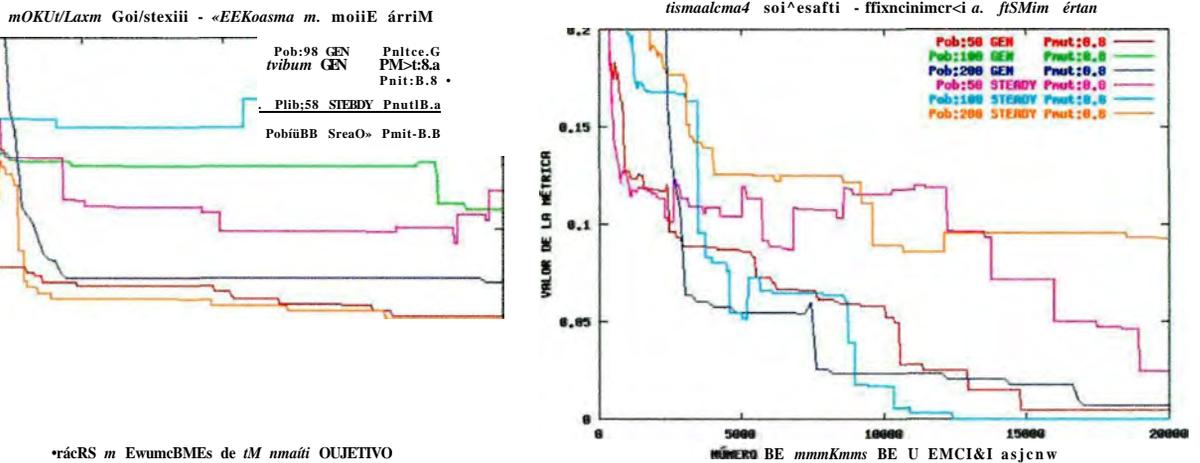


Figura 4.206. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 0.8%. Algoritmos NSGAI1 con elitismo controlado $r=0.0$ y $r=0.4$.

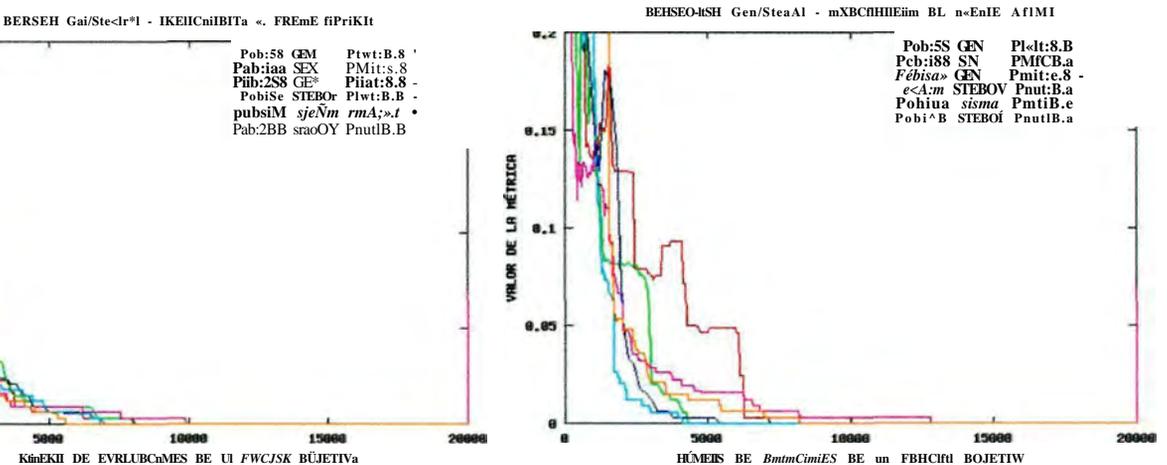


Figura 4.207. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 0.8%. Algoritmos DENSEA y DENSEA-NSH.

Métricas de acercamiento al frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 1.5%:

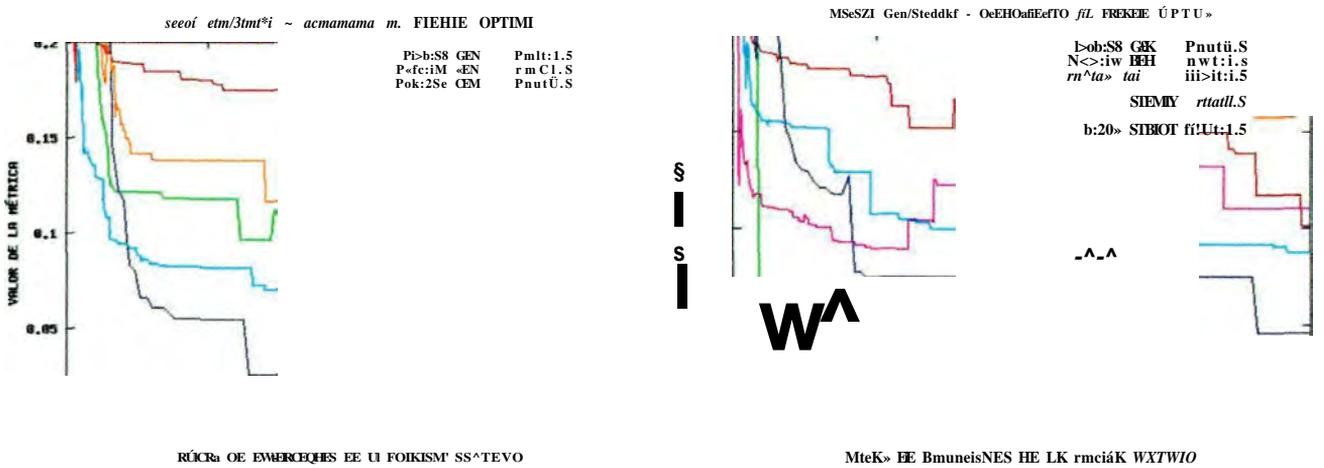


Figura 4.208. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 1.5%. Algoritmos SPEA2 y NSGAI2.

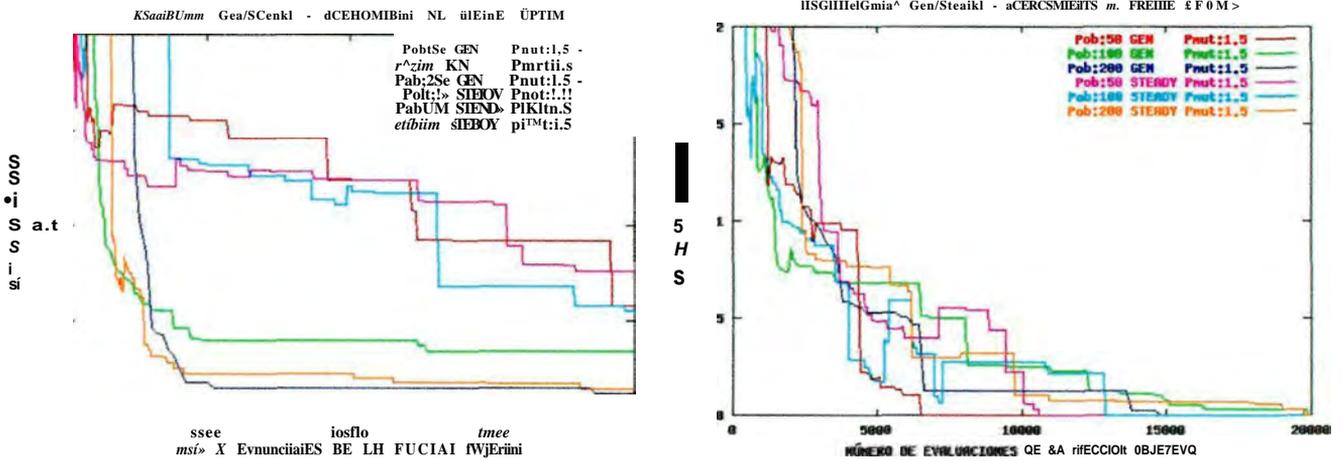


Figura 4.209. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 1.5%. Algoritmos NSGAI2 con elitismo controlado $r=0.0$ y $r=0.4$.

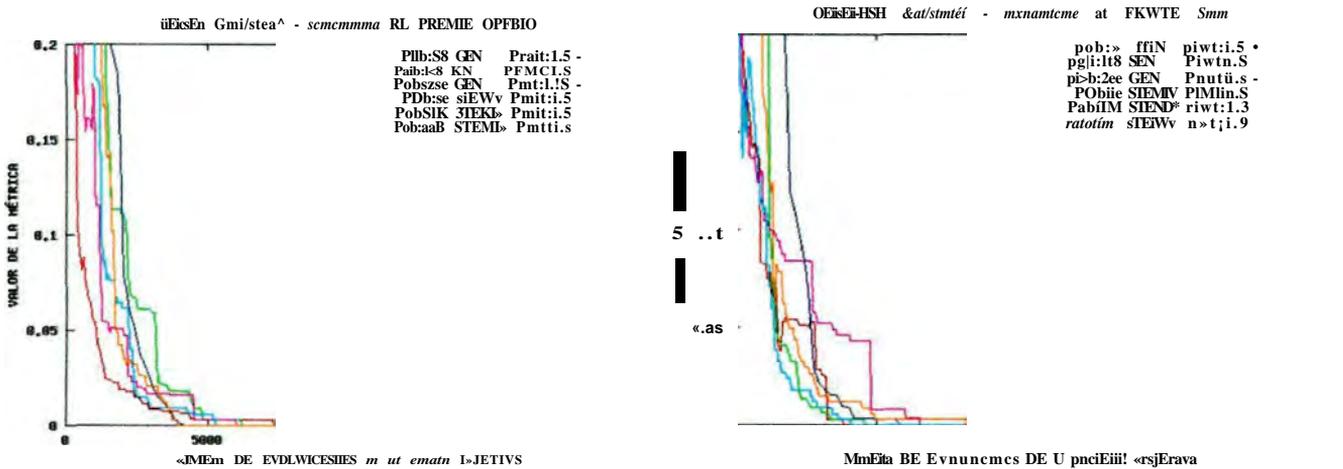


Figura 4.210. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 1.5%. Algoritmos DENSEA y DENSEA-NSH.

Métricas de acercamiento al frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 3%:

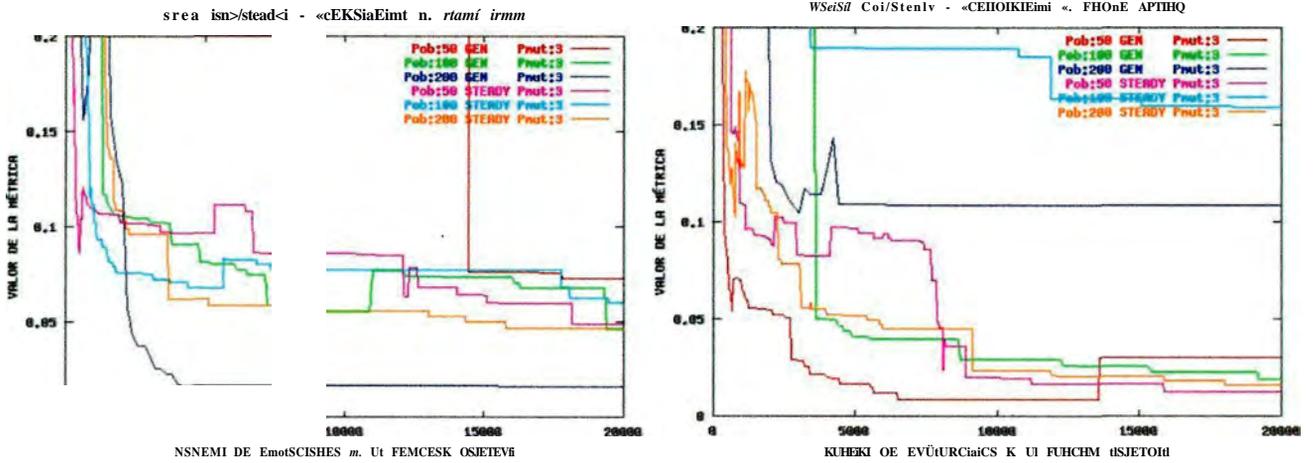


Figura 4.211. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 3.0%. Algoritmos SPEA2 y NSGAI1.

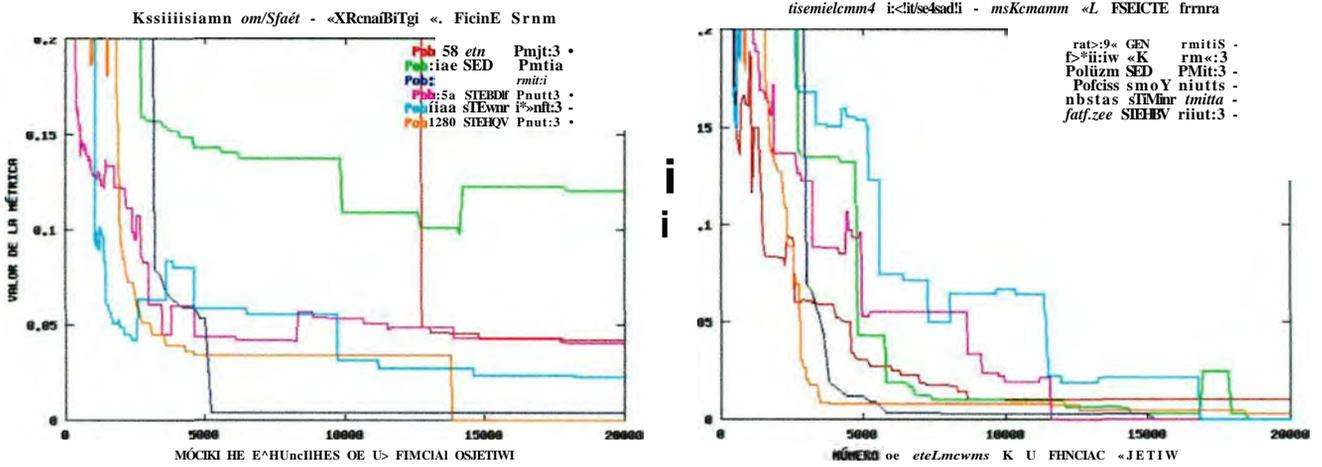


Figura 4.212. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 3.0%. Algoritmos NSGAI1 con elitismo controlado $r=0.0$ y $r=0.4$

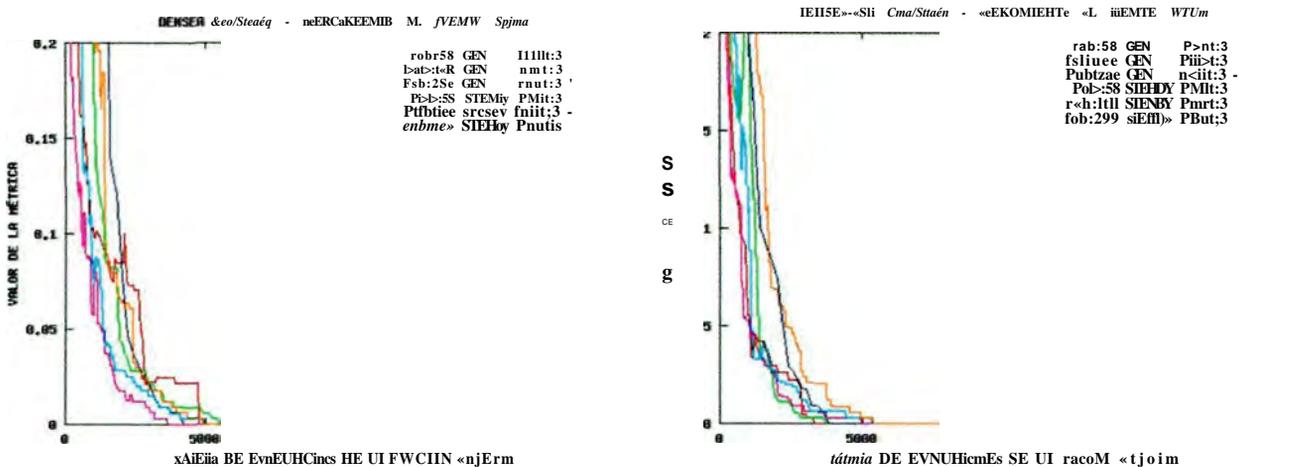


Figura 4.213. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 3.0%. Algoritmos DENSEA y DENSEA-NSH.

Capítulo 4. Contribuciones

Métricas de acercamiento al frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 6%:

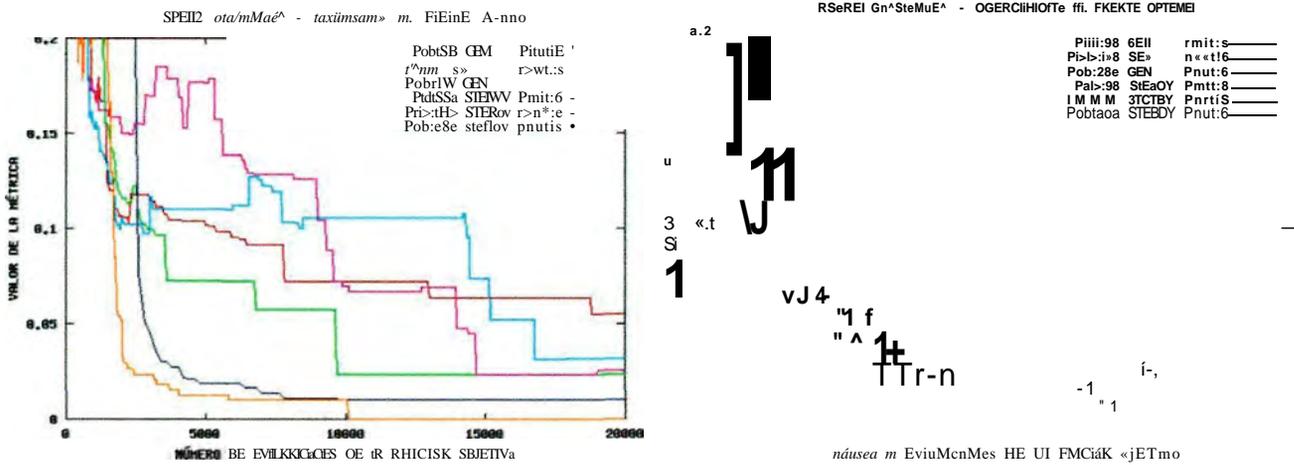


Figura 4.214. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 6.0%. Algoritmos SPEA2 y NSGAI1.

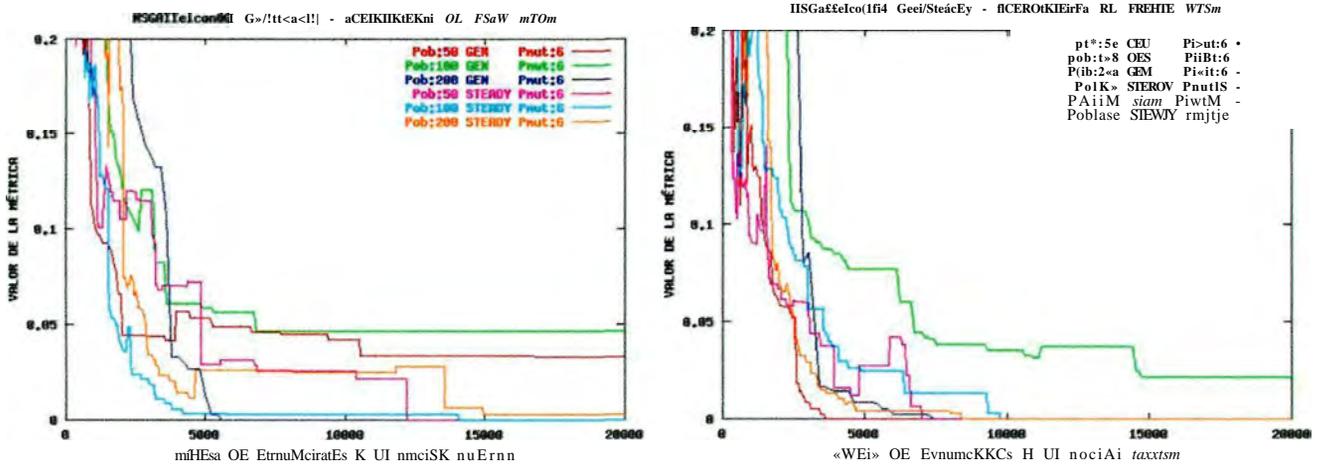


Figura 4.215. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 6.0%. Algoritmos NSGAI1 con elitismo controlado $r=0.0$ y $r=0.4$

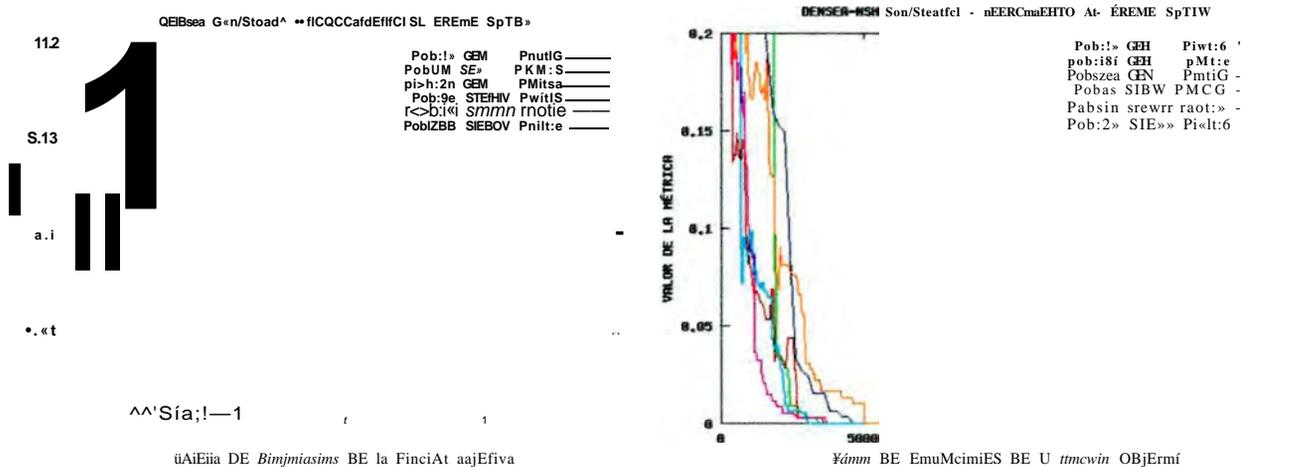


Figura 4.216. Métrica acercamiento al frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 6.0%. Algoritmos DENSEA y DENSEA-NSH.

Métricas de amplitud del frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 0.8%:

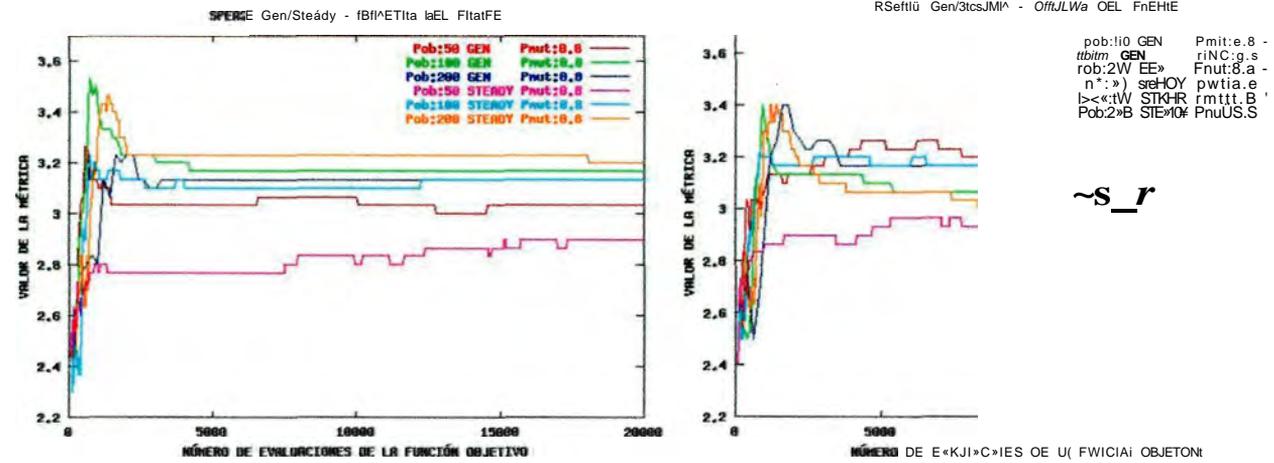


Figura 4.217. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 0.8%. Algoritmos SPEA2 y NSGAI.

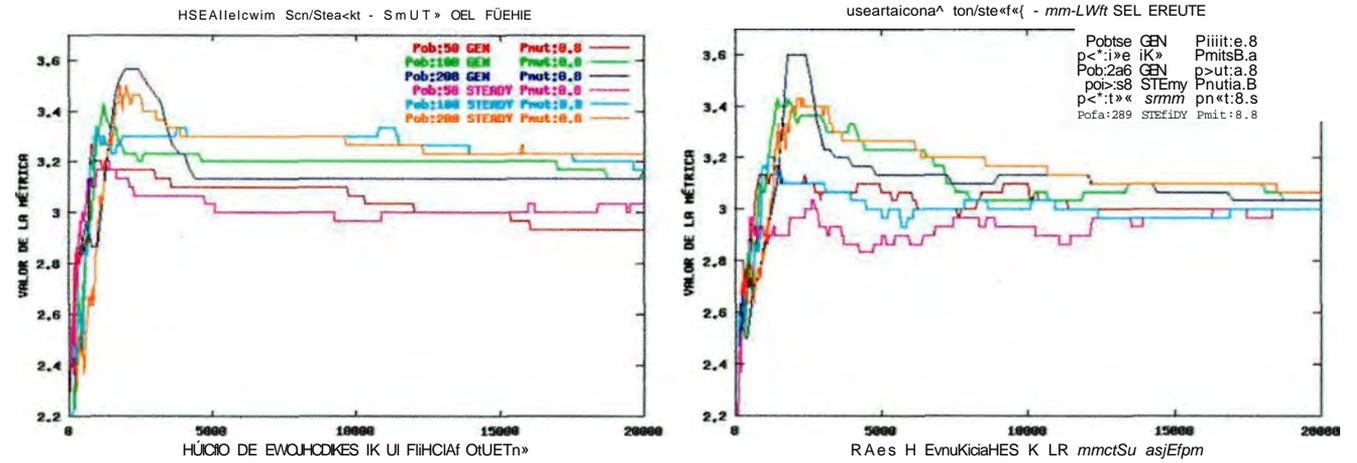


Figura 4.218. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 0.8%. Algoritmos NSGAI con elitismo controlado $r=0.0$ y $r=0.4$.

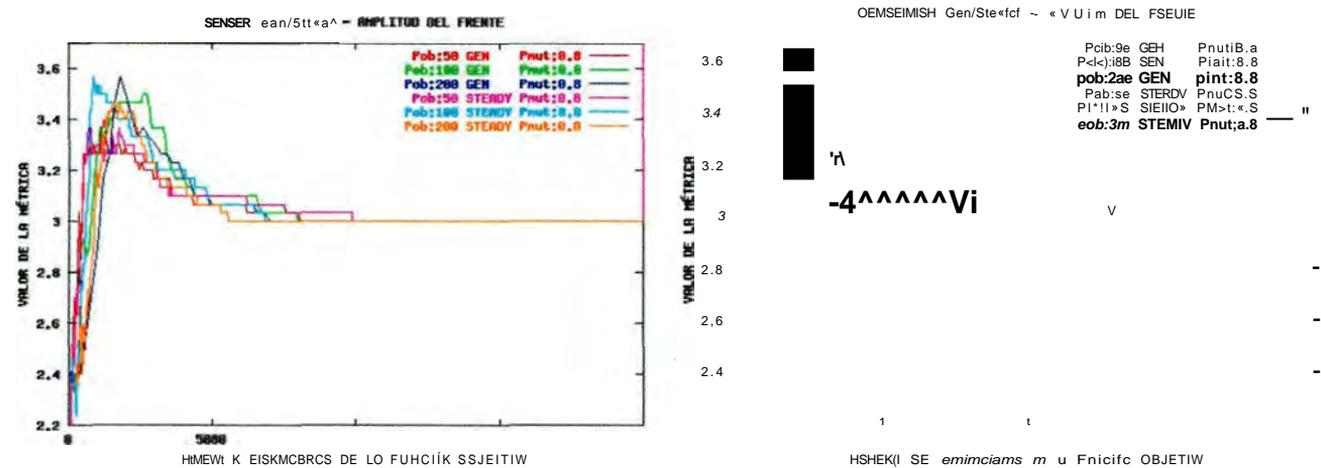


Figura 4.219. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 0.8%. Algoritmos DENSEA y DENSEA-NSH.

Métricas de amplitud del frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 1.5%:

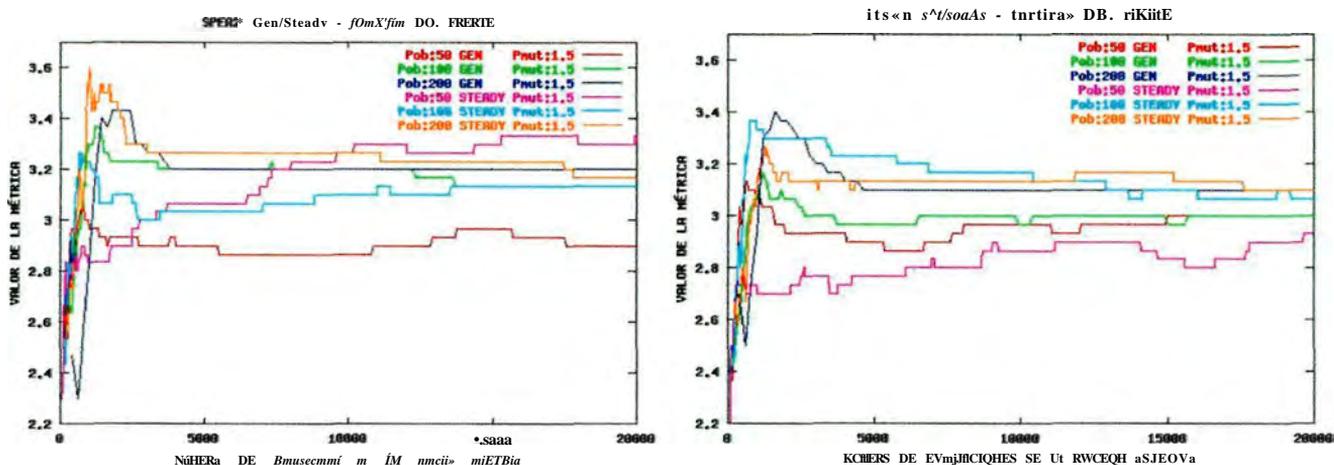


Figura 4.220. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 1.5%. Algoritmos SPEA2 y NSGAI2.

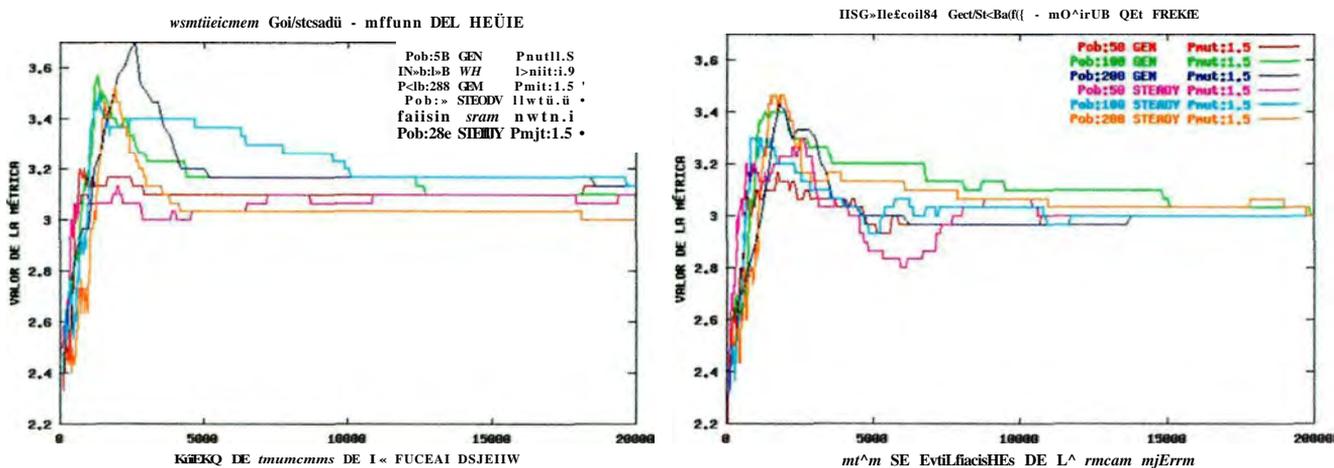


Figura 4.221. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 1.5%. Algoritmos NSGAI2 con elitismo controlado $r=0.0$ y $r=0.4$.

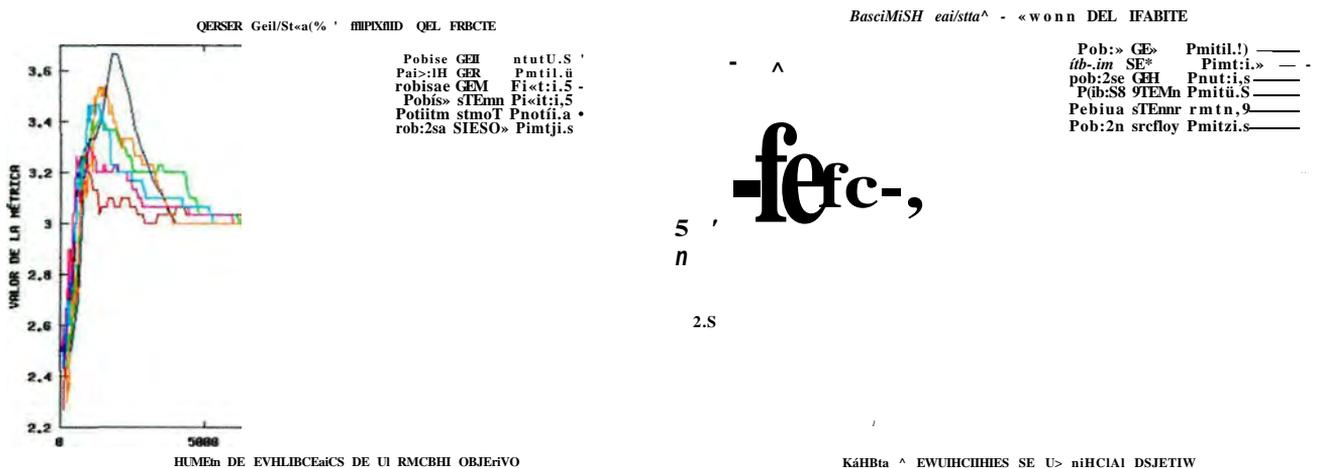


Figura 4.222. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 1.5%. Algoritmos DENSEA y DENSEA-NSH.

Capítulo 4. Contribuciones

Métricas de amplitud del frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 3%:

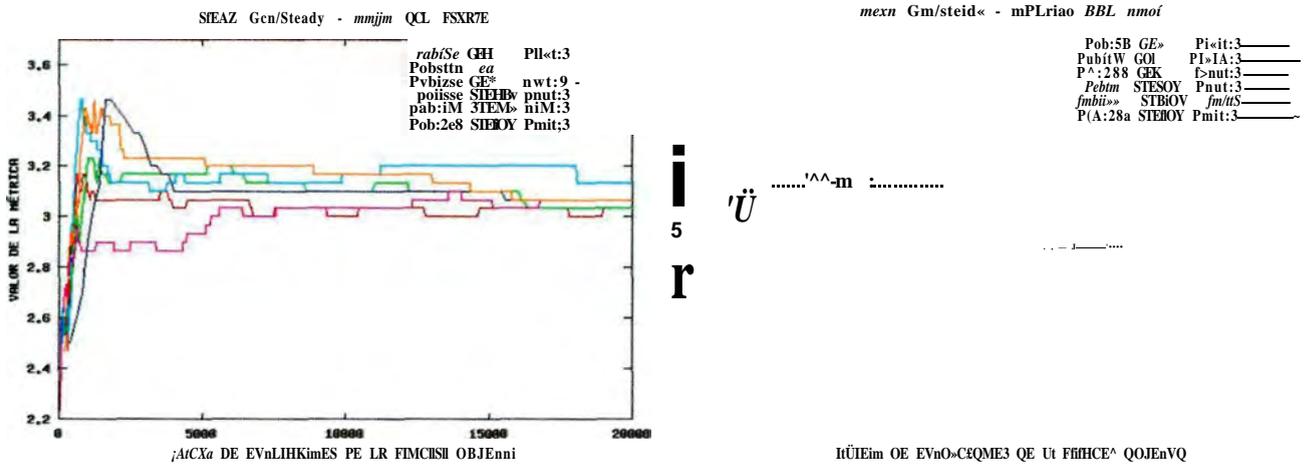


Figura 4.223. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 3.0%. Algoritmos SPEA2 y NSGAI2.

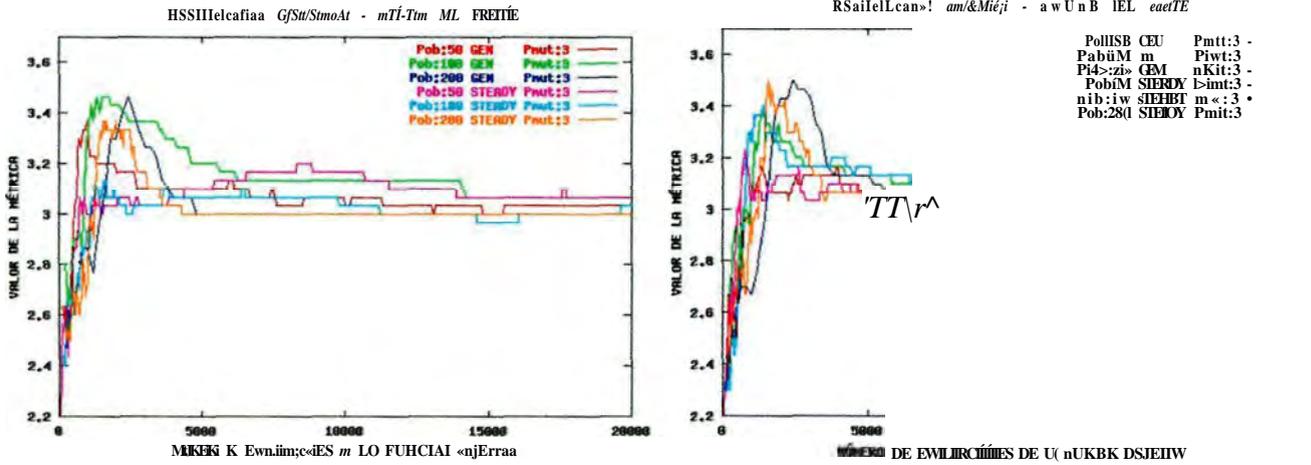


Figura 4.224. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 3.0%. Algoritmos NSGAI2 con elitismo controlado $r=0.0$ y $r=0.4$.

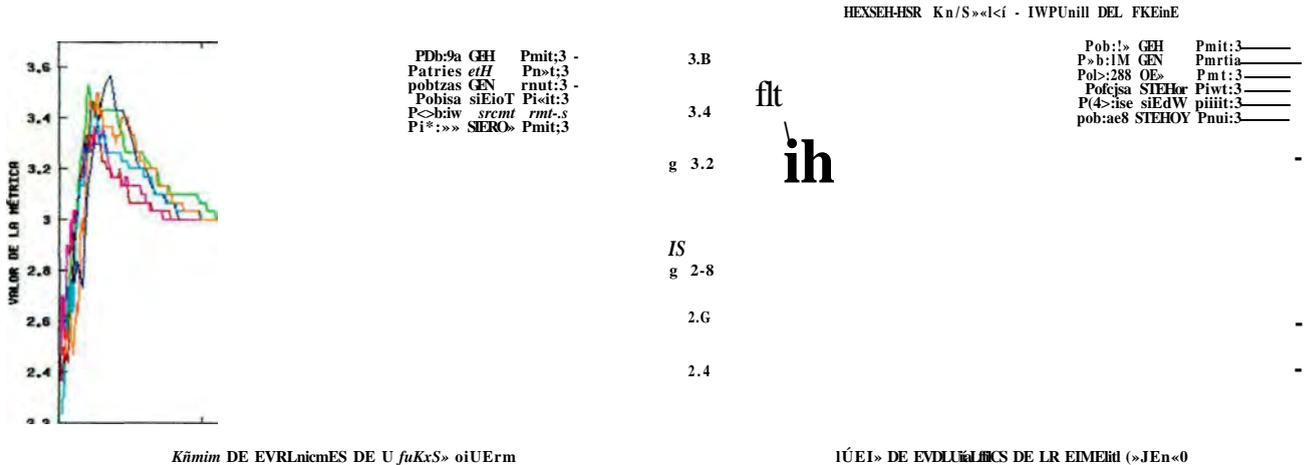


Figura 4.225. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 3.0%. Algoritmos DENSEA y DENSEA-NSH.

Métricas de amplitud del frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 6%:

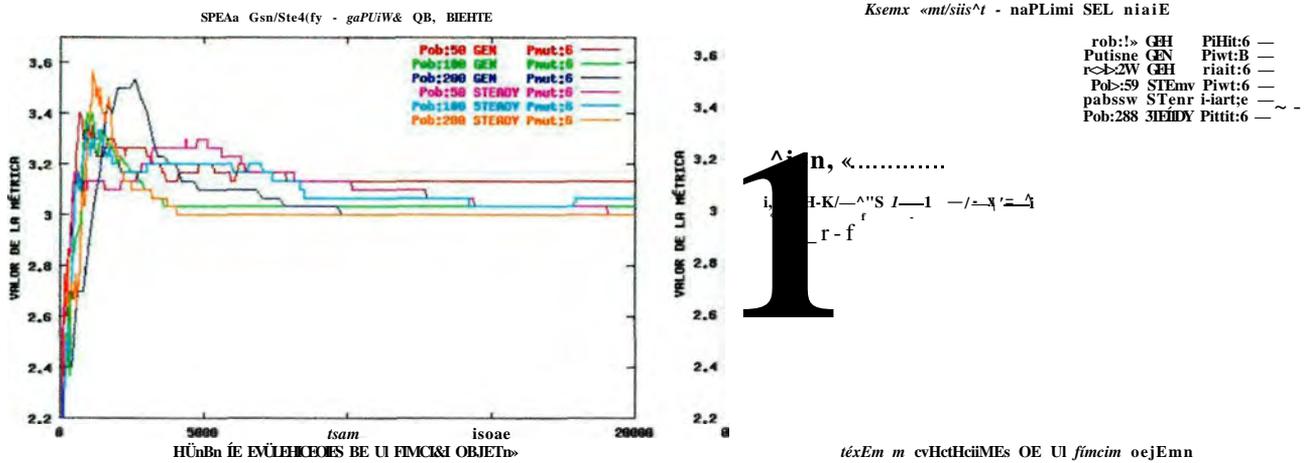


Figura 4.226. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 6.0%. Algoritmos SPEA2 y NSGAI.

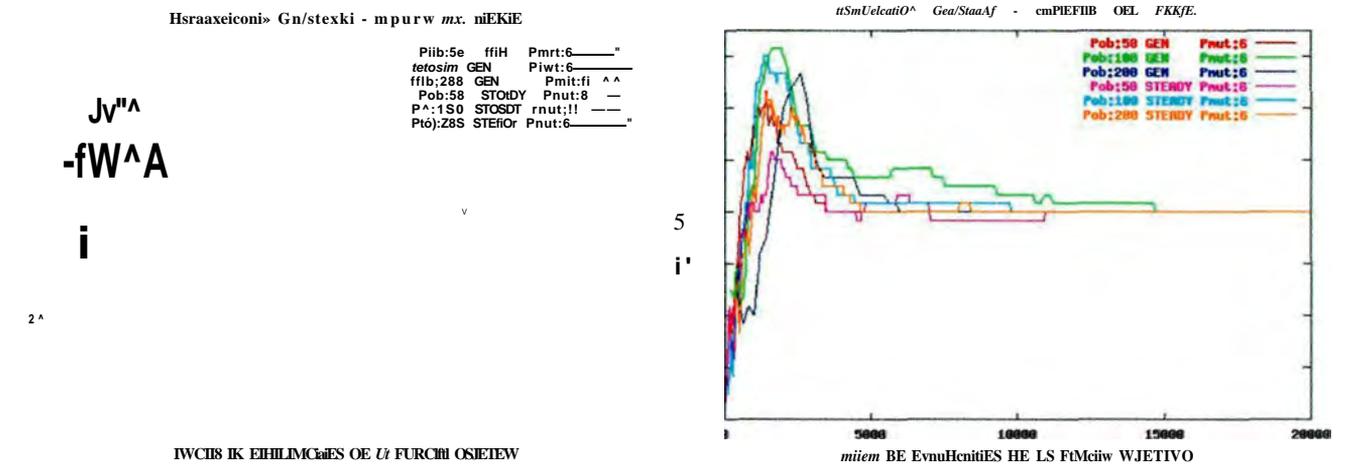


Figura 4.227. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 6.0%. Algoritmos NSGAI con elitismo controlado $r=0.0$ y $r=0.4$.

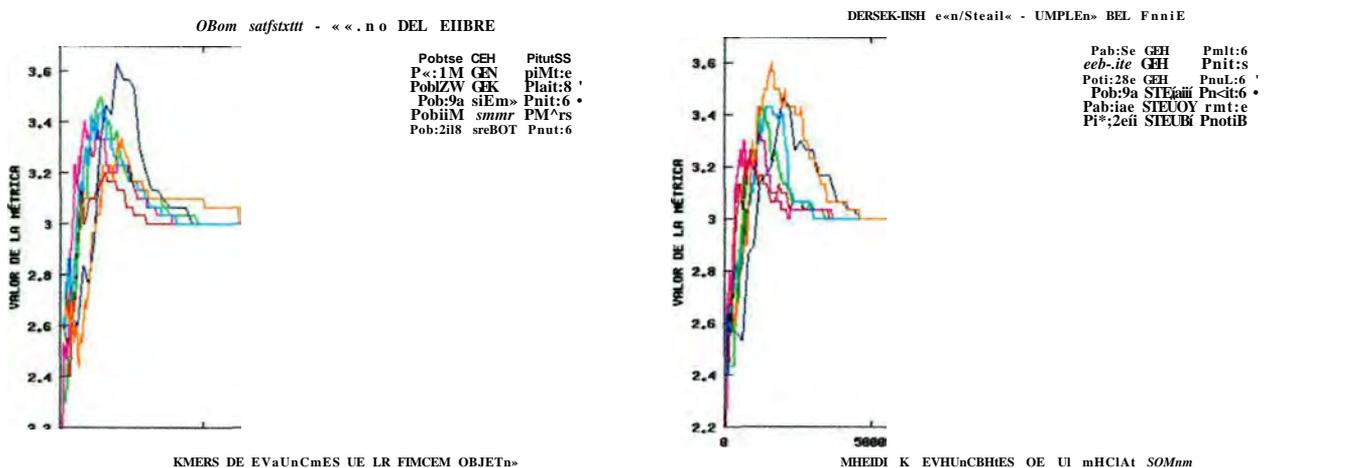


Figura 4.228. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Probabilidad de mutación 6.0%. Algoritmos DENSEA y DENSEA-NSH.

Número de evaluaciones en alcanzar frente óptimo, Pmutación 0.8%:

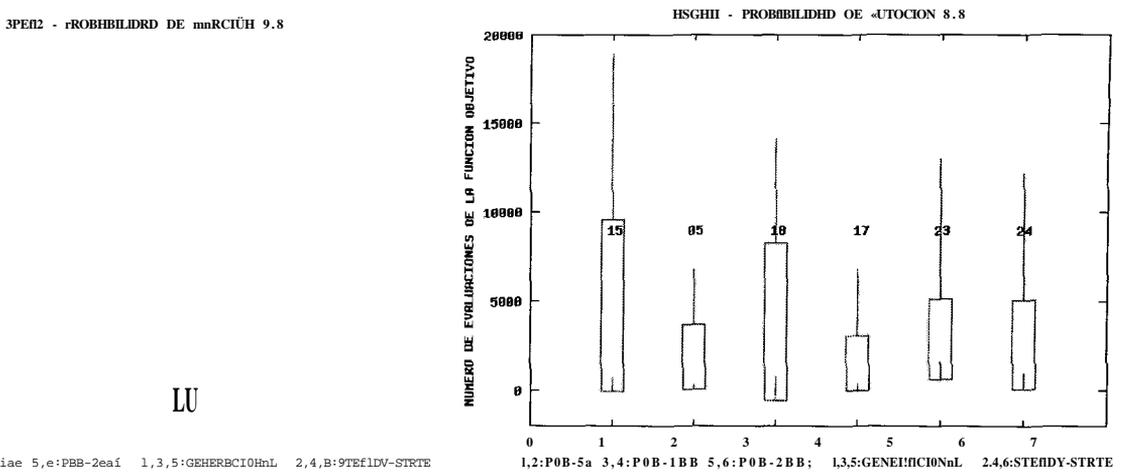


Figura 4.229. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Parete. Algoritmos SPEA2 y NSGAII. Probabilidad mutación 0.8%

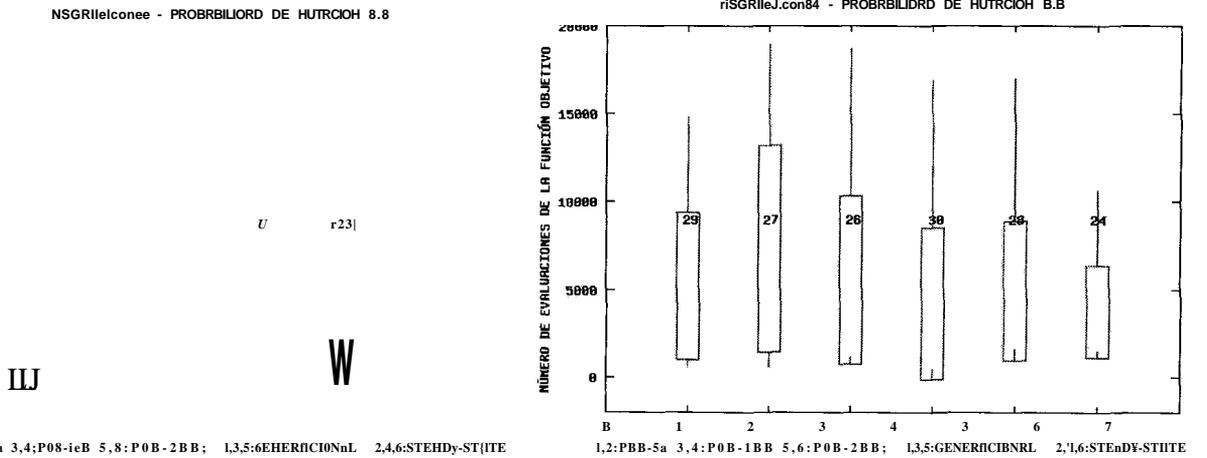


Figura 4.230. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Parete. Algoritmos NSGAII con elitismo controlado r=0.0 y r=0.4. Probabilidad mutación 0.8%

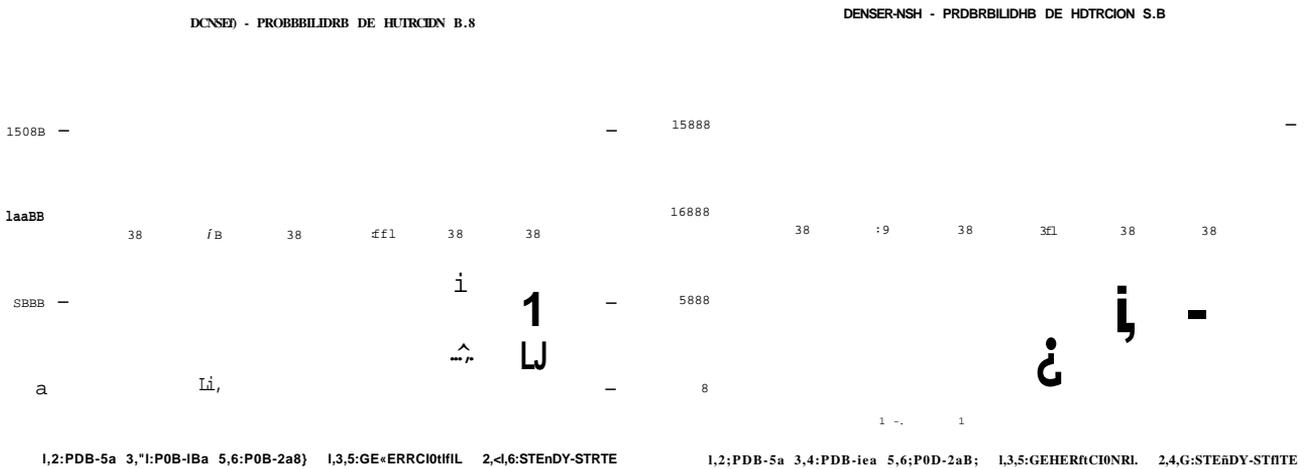


Figura 4.231. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Case X. Gráficas sobre Frente de Pareto. Algoritmos DENSEA y DENSEA-NSH. Probabilidad mutación 0.8%

Número de evaluaciones en alcanzar frente óptimo, Pmutación 1.5%:

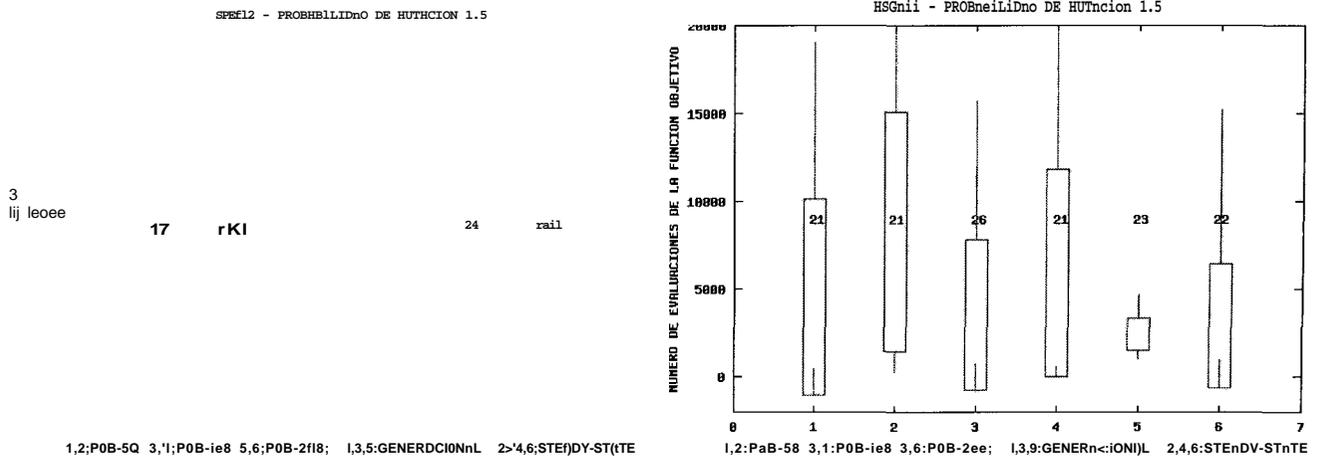


Figura 4.232. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Pareto. Algoritmos SPEA2 y NSGAI. Probabilidad mutación 1.5%

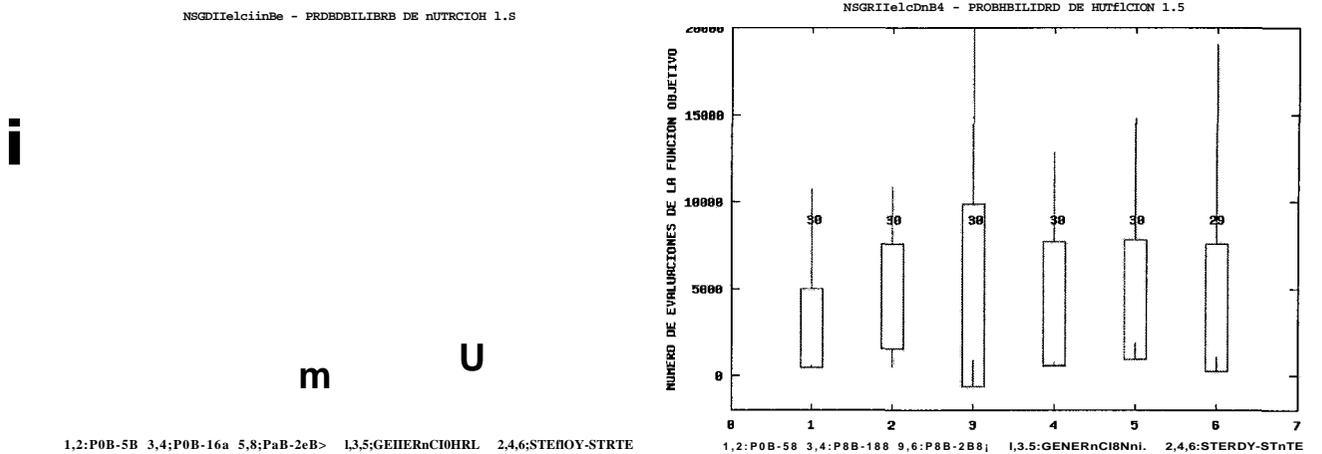


Figura 4.233. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Pareto. Algoritmos NSGAI con elitismo controlado $r=0.0$ y $r=0.4$. Probabilidad mutación 1.5%

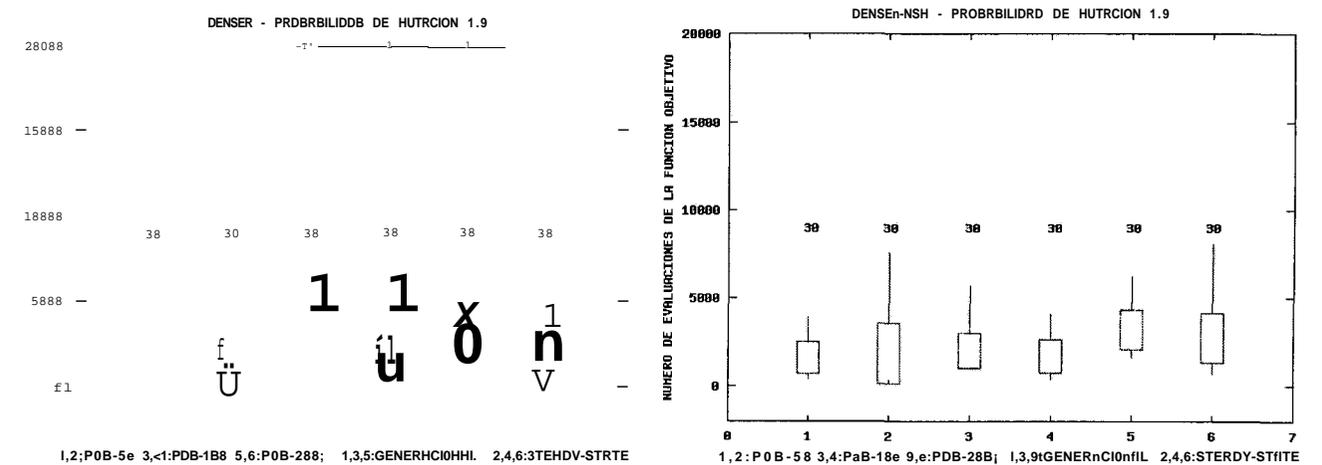


Figura 4.234. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Pareto. Algoritmos DEN SEA y DENSEA-NSH. Probabilidad mutación 1.5%

Número de evaluaciones en alcanzar frente óptimo, Pmutación 3%:

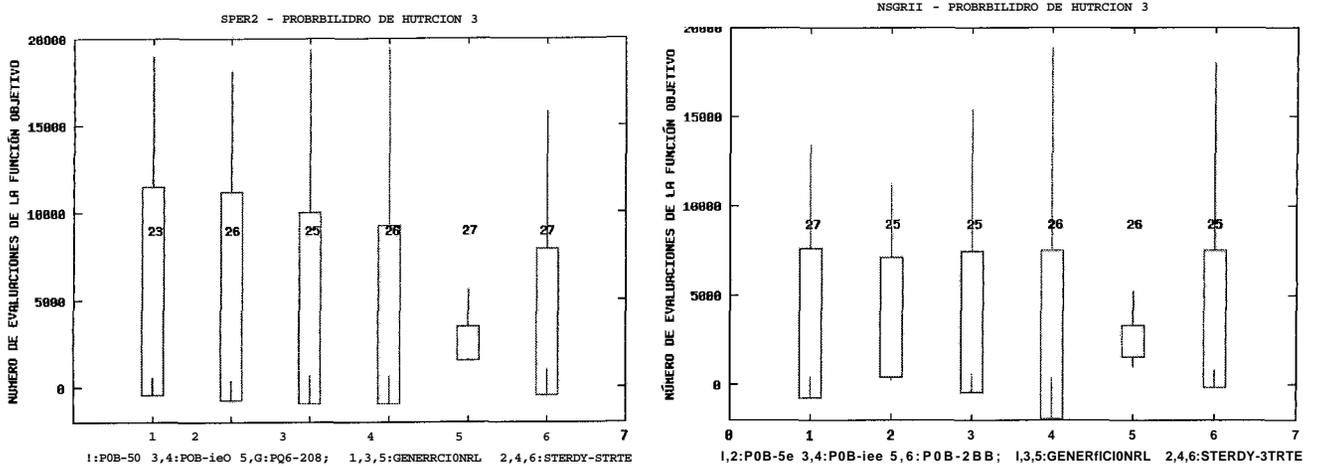


Figura 4.235. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Pareto. Algoritmos SPEA2 y NSGAI. Probabilidad mutación 3.0%

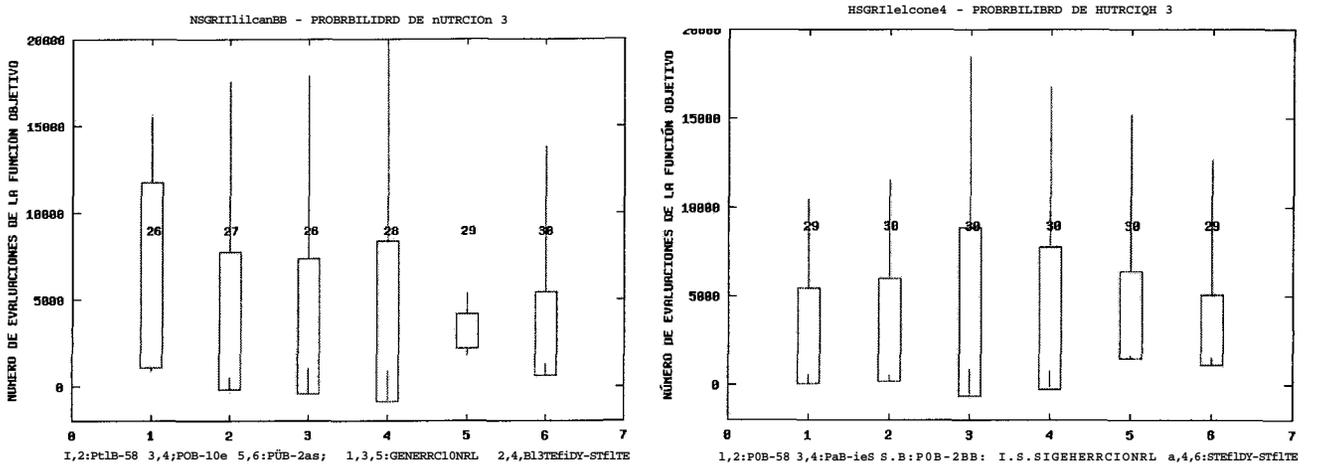


Figura 4.236. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Pareto. Algoritmos NSGAI con elitismo controlado $r=0.0$ y $r=0.4$. Probabilidad mutación 3.0%

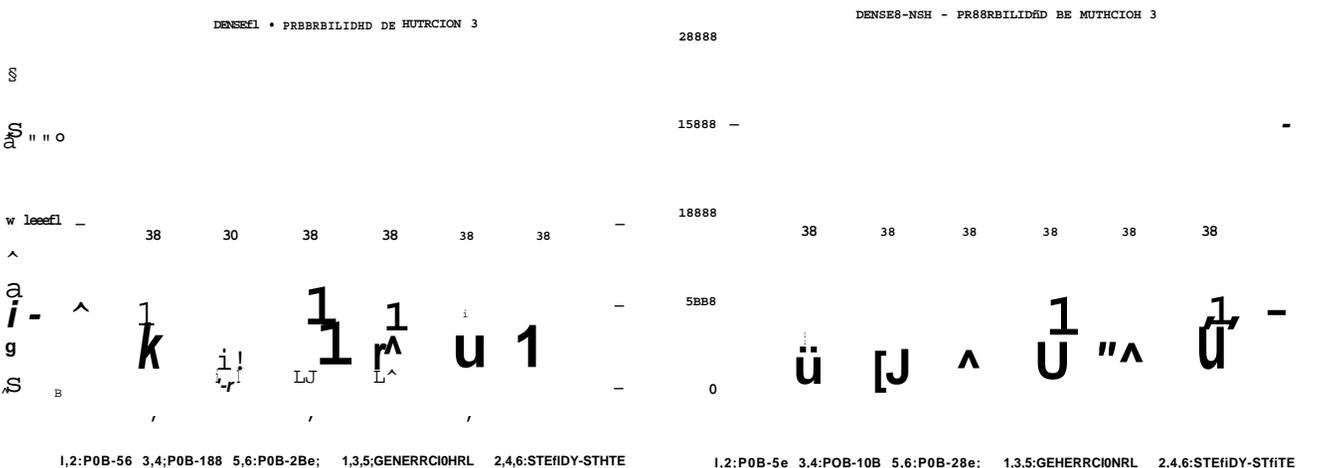


Figura 4.237. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Pareto. Algoritmos DENSE A y DENSE A-NSH. Probabilidad mutación 3.0%

Número de evaluaciones en alcanzar frente óptimo, Pmutación 6%:

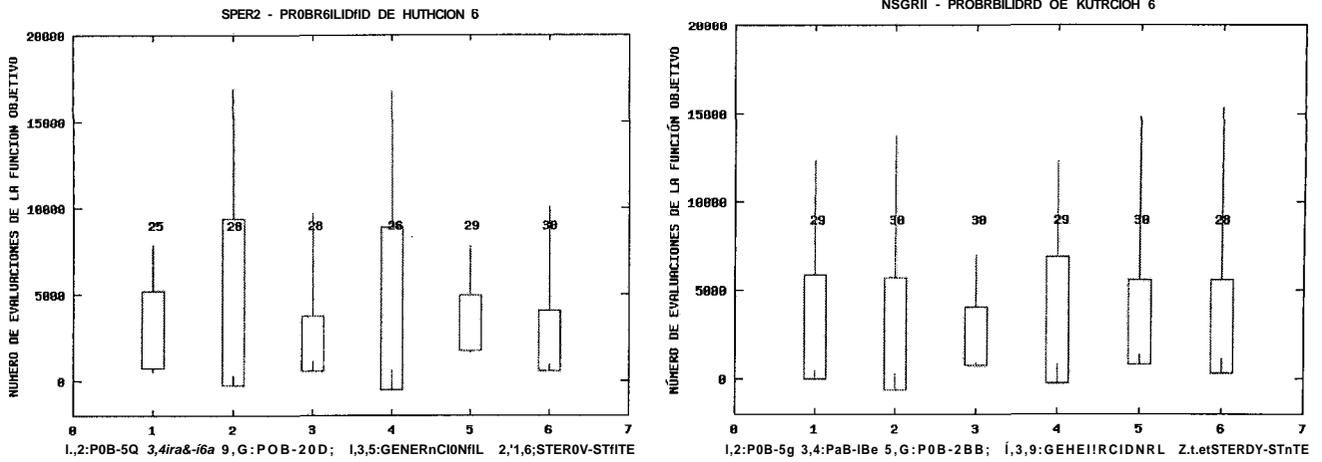


Figura 4.238. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Pareto. Algoritmos SPEA2 y NSGAI. Probabilidad mutación 6.0%

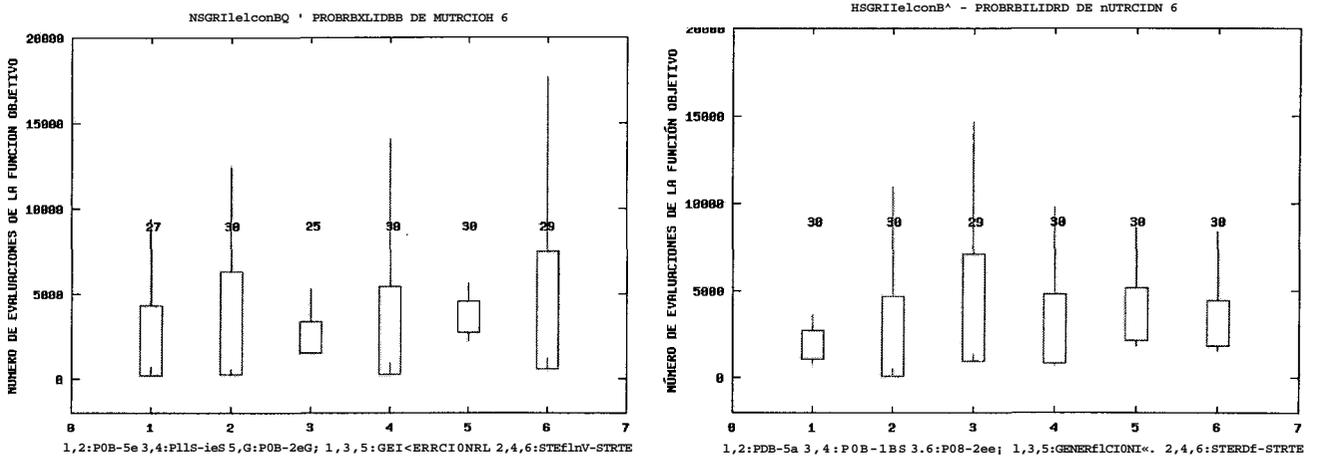


Figura 4.239. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Pareto. Algoritmos NSGAI con elitismo controlado $r=0.0$ y $r=0.4$. Probabilidad mutación 6.0%

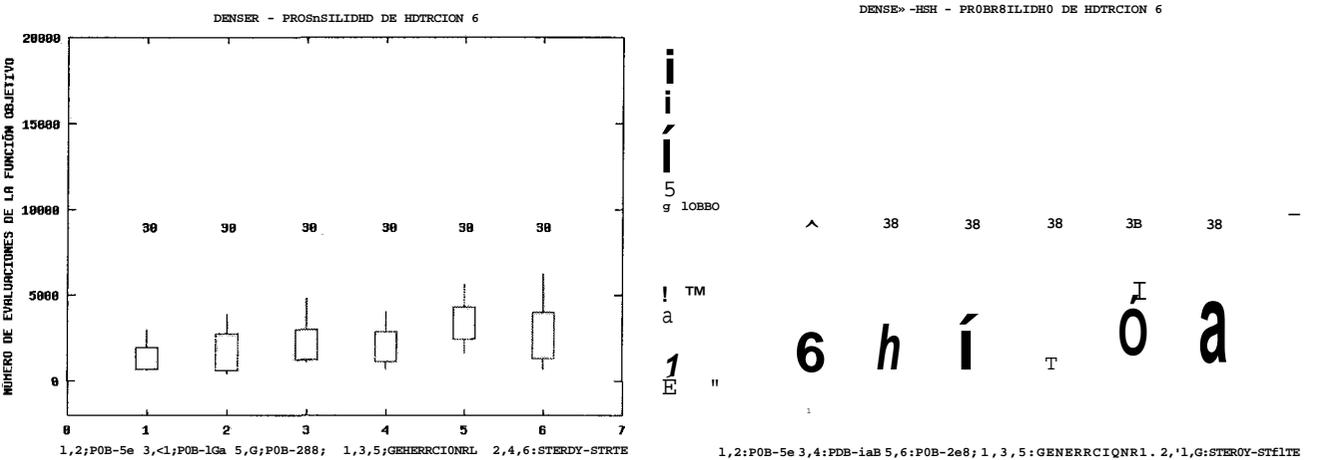


Figura 4.240. Comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso X. Gráficas sobre Frente de Pareto. Algoritmos DENSEA y DENSEA-NSH. Probabilidad mutación 6.0%

Capítulo 4. Contribuciones

Métricas de amplitud del frente, comparando 3 tamaños poblacs. de cada algoritmo, Pmutación 3%:

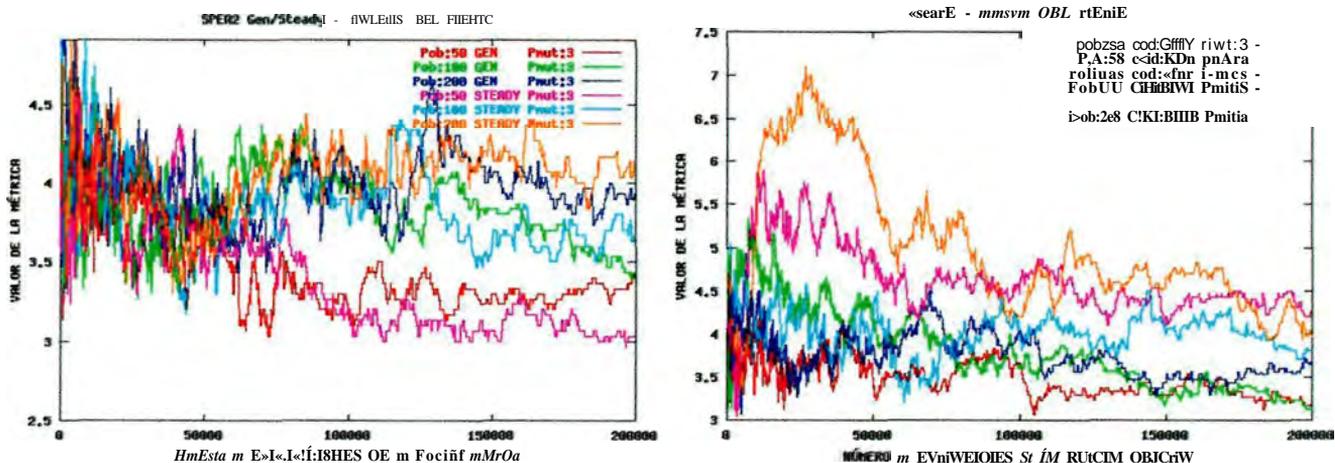


Figura 4.262. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 3.0%. Algoritmos SPEA2 y NSGAI.

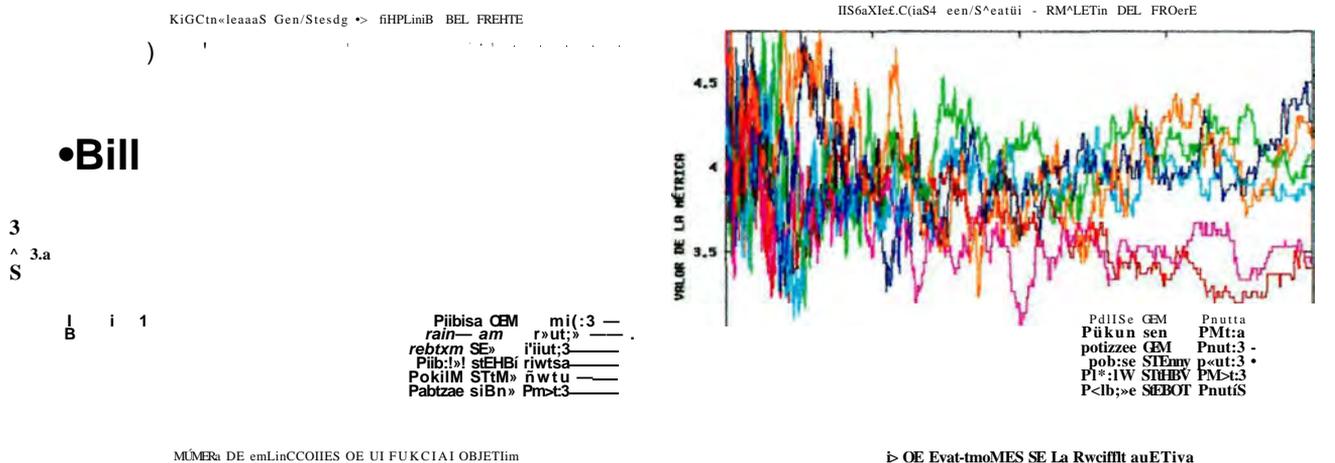


Figura 4.263. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 3.0%. Algoritmos NSGAI con elitismo controlado $r=0.0$ y $r=0.4$.

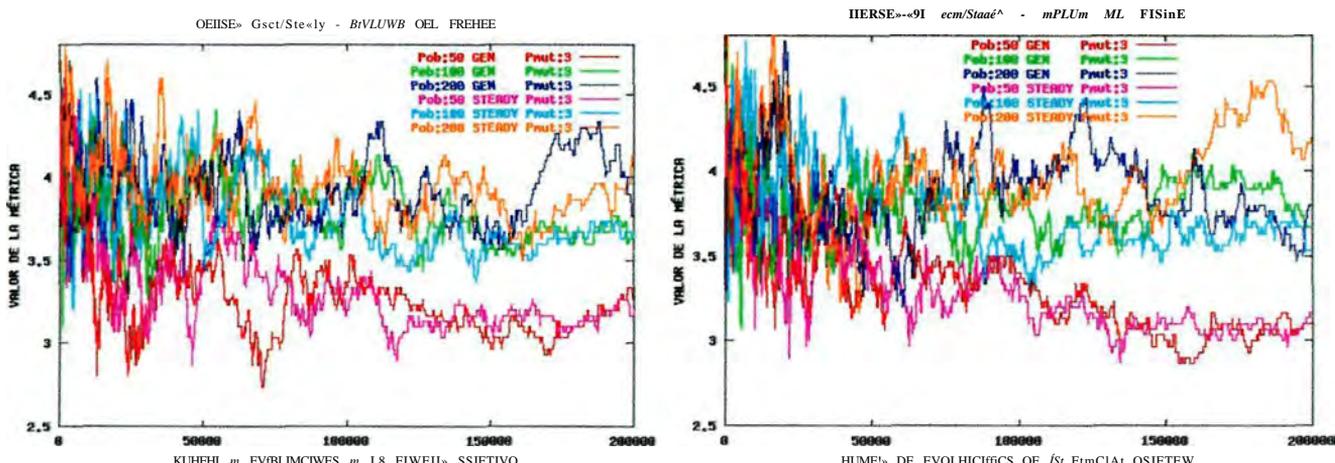


Figura 4.264. Métrica amplitud del frente en comparativa de reemplazamientos Generacional / EstadoEstacionario en Caso Y. Probabilidad de mutación 3.0%. Algoritmos DENSEA y DENSEA-NSH.

Caso X) DENSEA vs DENSEA-NSH: Pmutación del 0.8% y 3 tamaños de población:

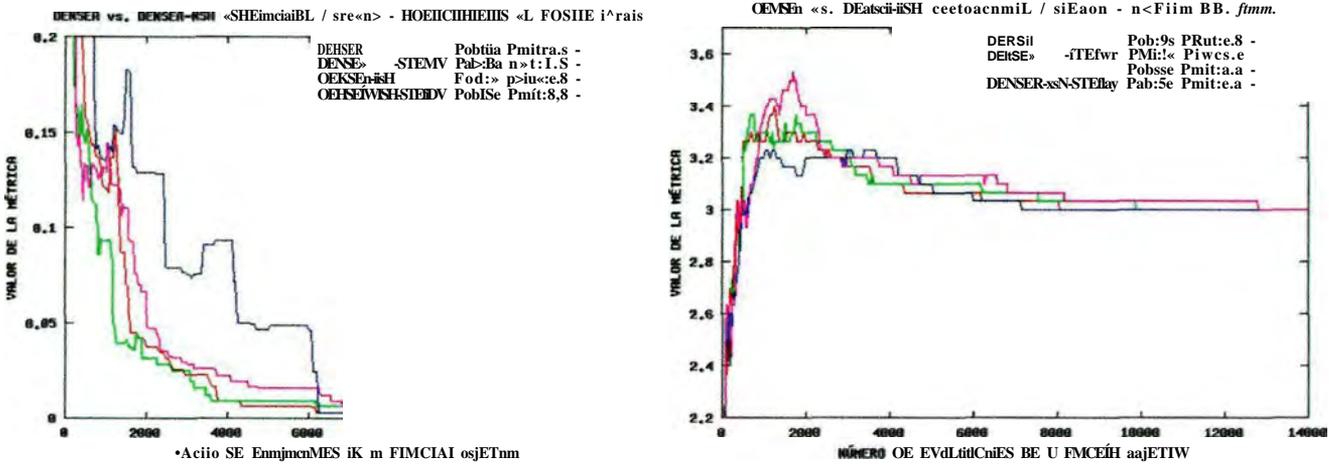


Figura 4.268. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso X. Probabilidad mutación 0.8%. Tamaño población 50.

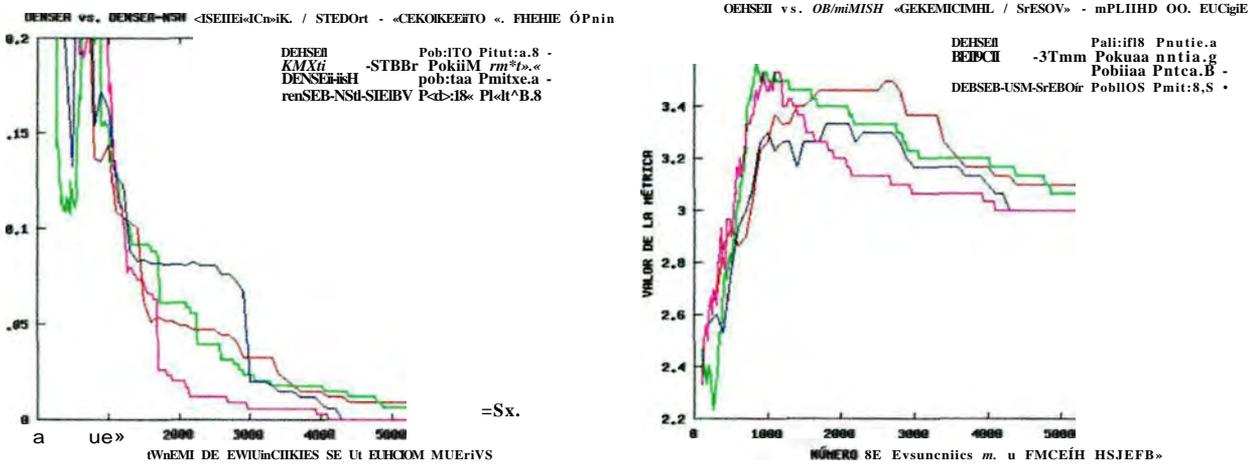


Figura 4.269. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso X. Probabilidad mutación 0.8%. Tamaño población 100.

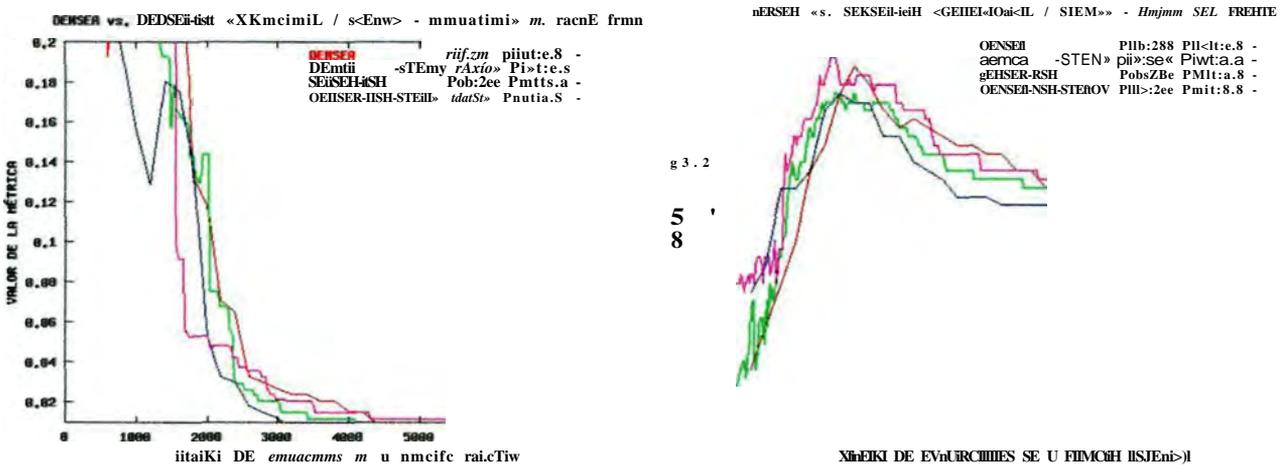


Figura 4.270. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso X. Probabilidad mutación 0.8%. Tamaño población 200.

Caso X) DENSEA vs DENSEA-NSH: Pmutación del 3% y 3 tamaños de población:

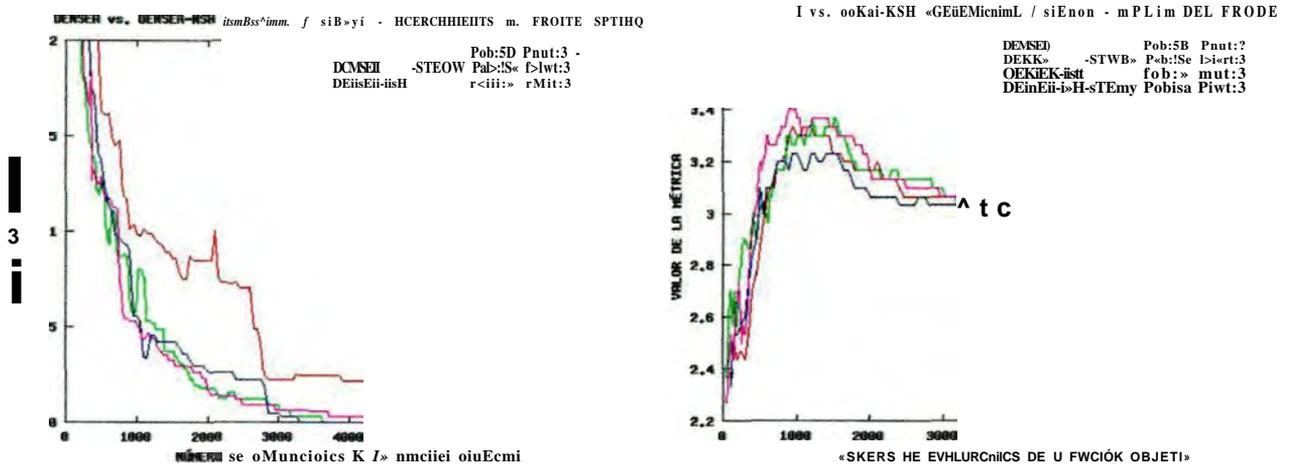


Figura 4.274. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso X. Probabilidad mutación 3.0%. Tamaño población 50.

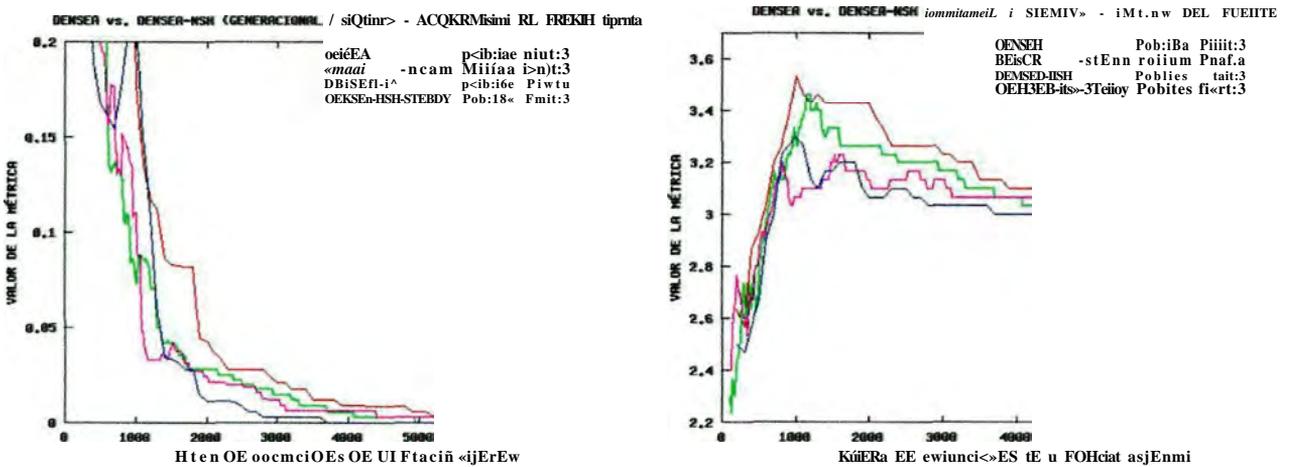


Figura 4.275. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso X. Probabilidad mutación 3.0%. Tamaño población 100.

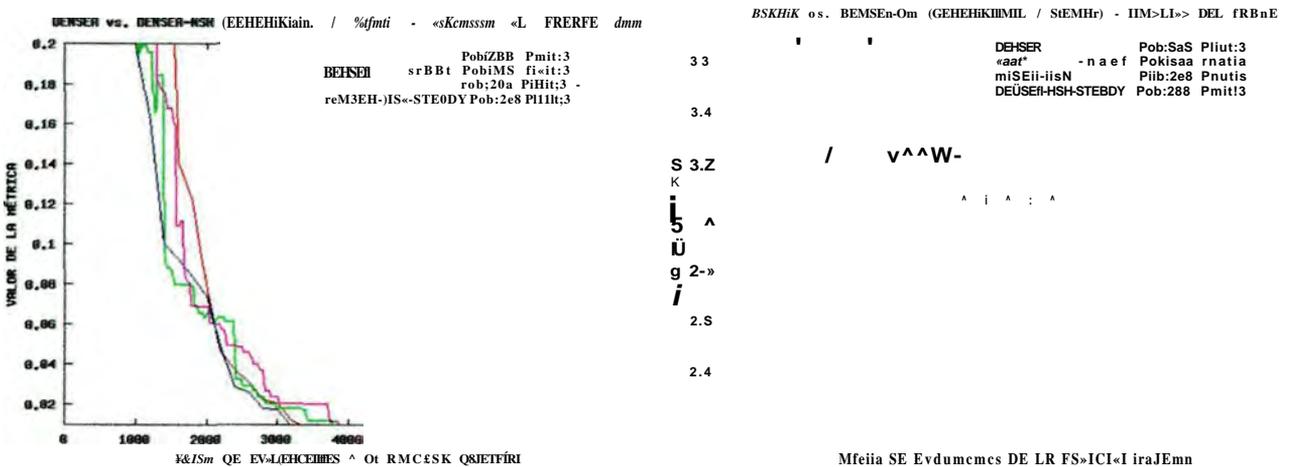


Figura 4.276. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso X. Probabilidad mutación 3.0%. Tamaño población 200.

Caso X) DENSEA vs DENSEA-NSH: Pmutación del 6% y 3 tamaños de población:

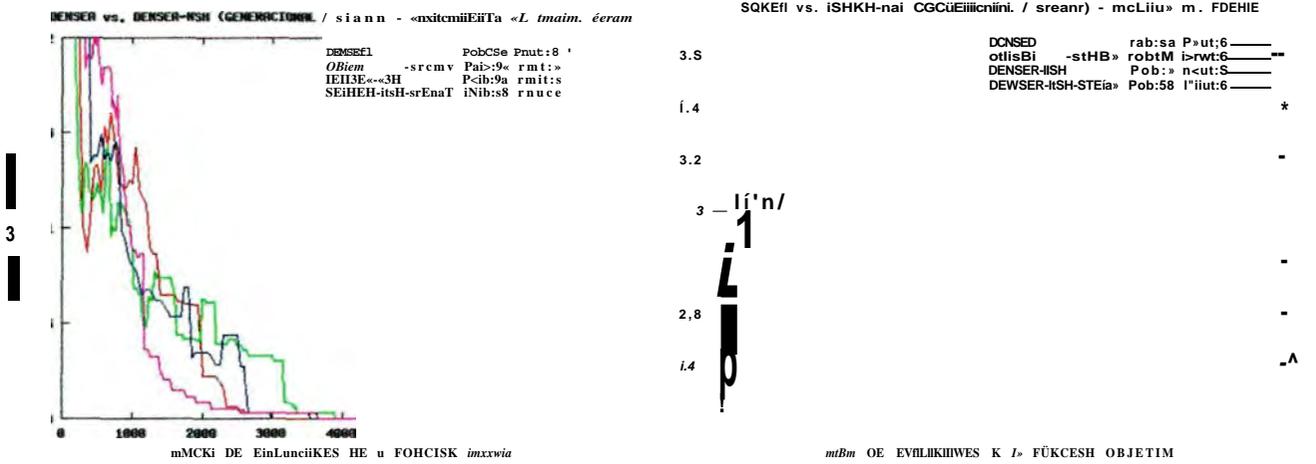


Figura 4.277. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso X. Probabilidad mutación 6.0%. Tamaño población 50.

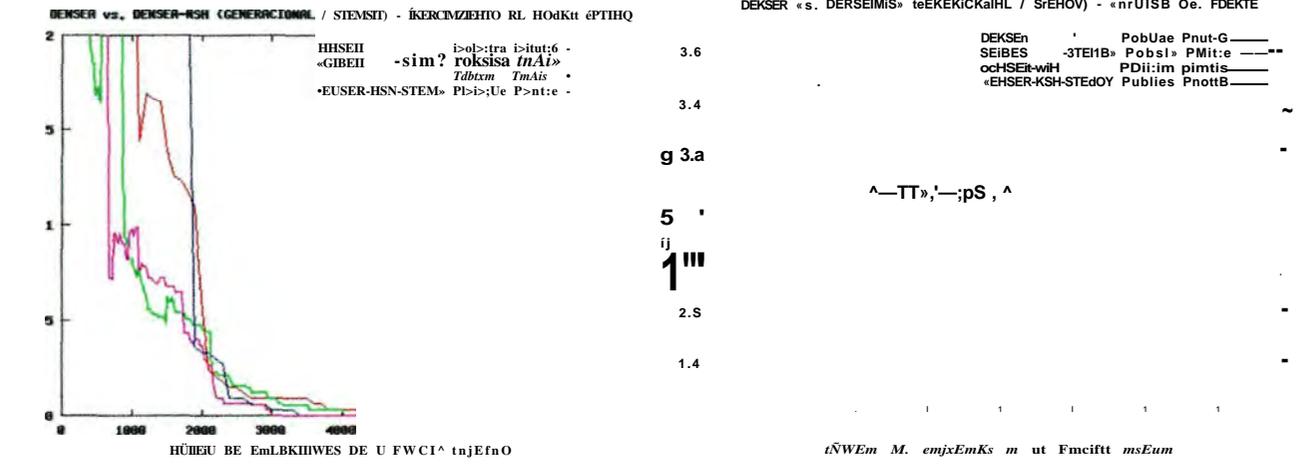


Figura 4.278. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso X. Probabilidad mutación 6.0%. Tamaño población 100.

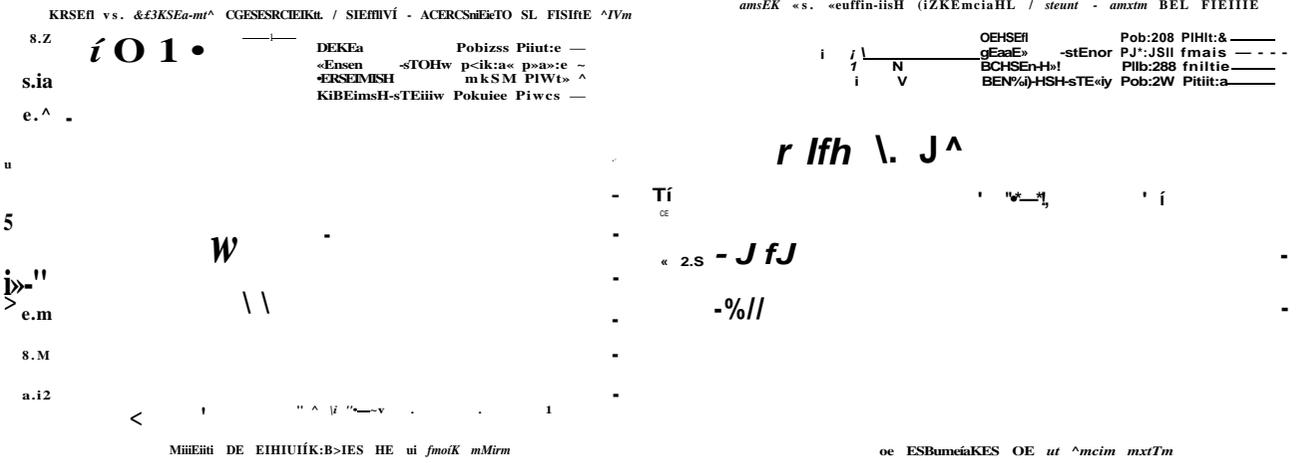


Figura 4.279. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso X. Probabilidad mutación 6.0%. Tamaño población 200.

Caso Y) DENSEA vs DENSEA-NSH: Pmutación del 1.5% y 3 tamaños de población:

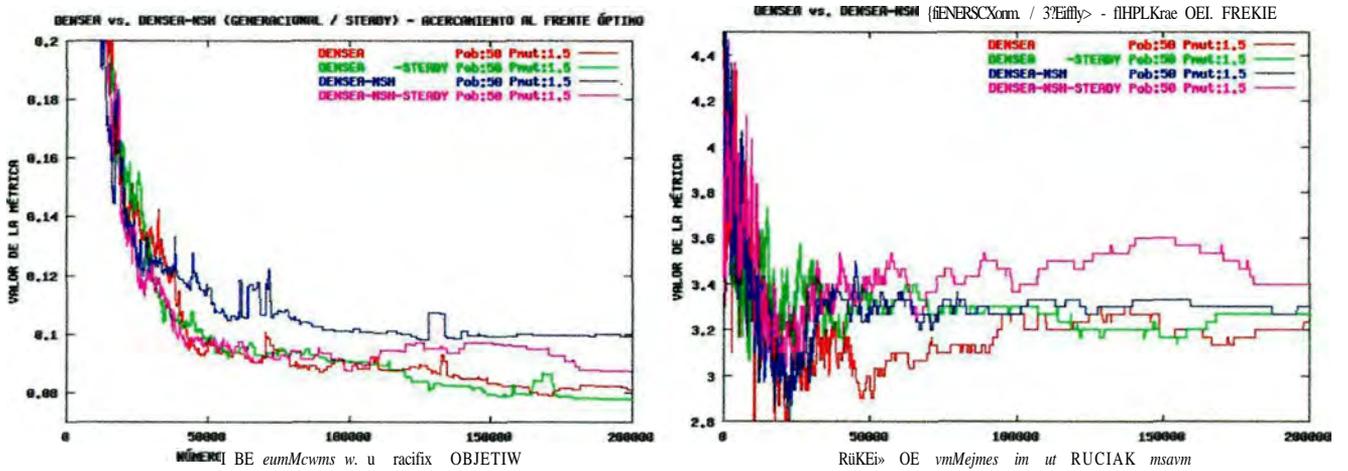


Figura 4.286 Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 1.5%. Tamaño población 50.

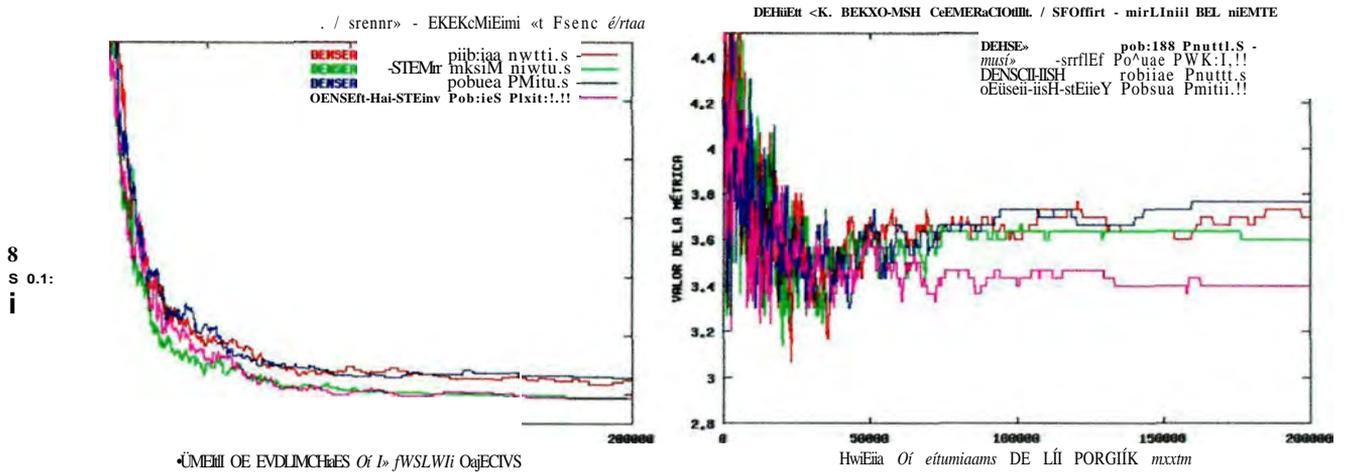


Figura 4.287. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 1.5%. Tamaño población 100.

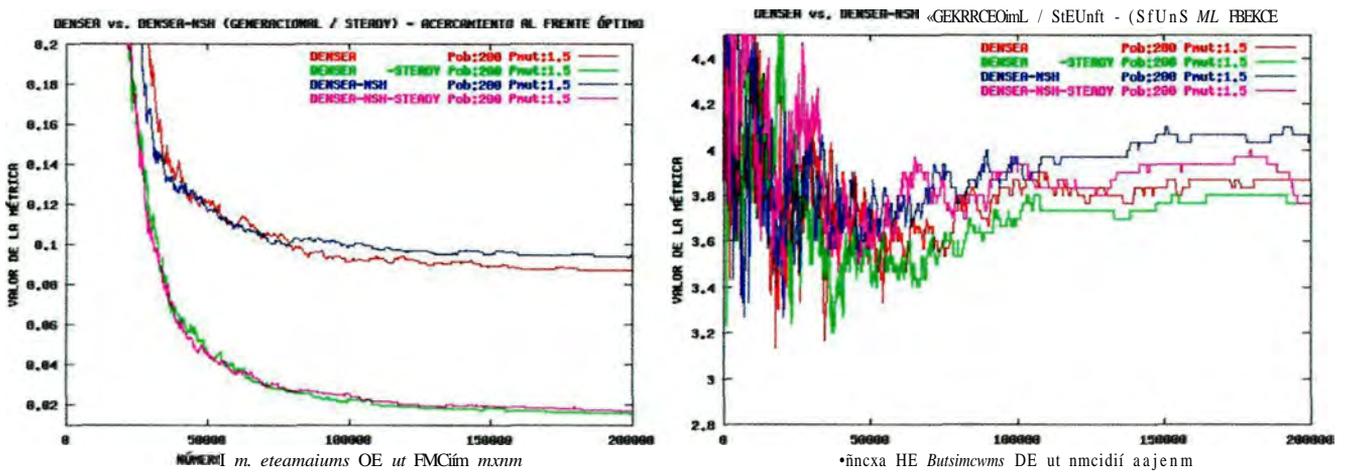


Figura 4.288. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 1.5%. Tamaño población 200.

Caso Y) DENSEA vs DENSEA-NSH: Pmutación del 3% y 3 tamaños de población:

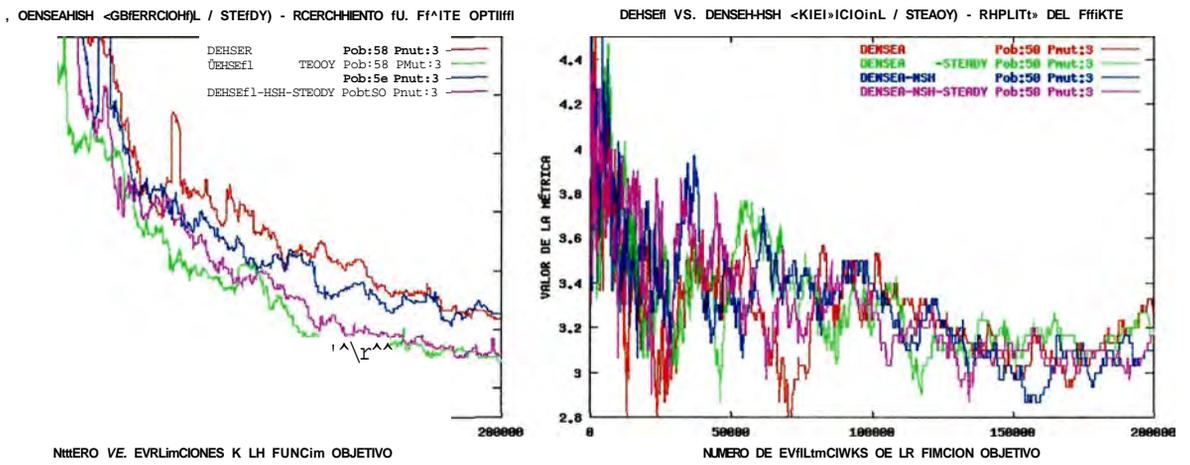


Figura 4.289. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 3.0%. Tamaño población 50.

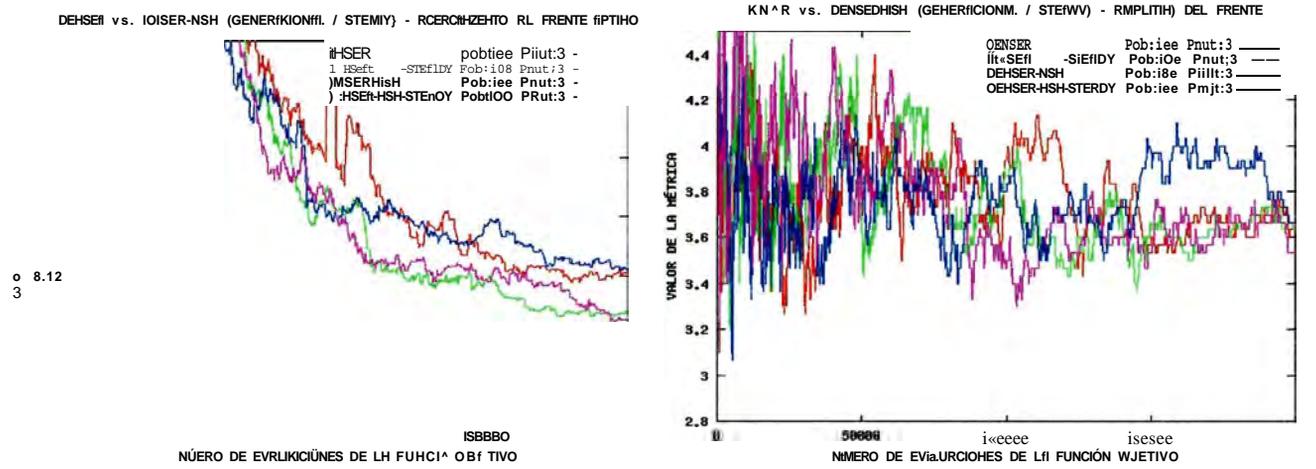


Figura 4.290. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 3.0%. Tamaño población 100.

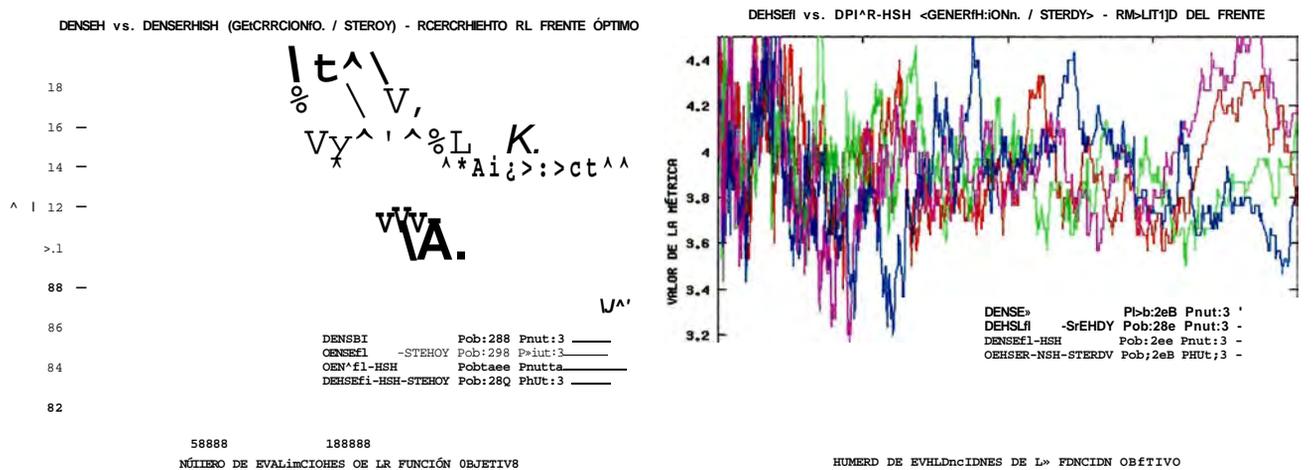


Figura 4.291. Métricas en comparativa de algoritmos DENSEA y DENSEA-NSH en Caso Y. Probabilidad mutación 3.0%. Tamaño población 200.

Bibliografía

- [4.1] H. Abbass, K. Deb, "Searching under Multi-evolutionary Pressures". Evolutionary Multi-Criterion Optimization 2003, *Lecture Notes in Computer Science*, pp. 391-404.
- [4.2] H. Aguirre, K. Tanaka, "Selection, Drift, Recombination, and Mutation in Multiobjective Evolutionary Algorithms on Scalable MNK-Landscapes", Third International Conference on Evolutionary Multi-Criterion Optimization 2005, *Lecture Notes in Computer Science* n° 3410, pp. 355-369.
- [4.3] R. Balling, "The Maximin Fitness Function: Multiobjective City and Regional Planning", Evolutionary Multi-Criterion Optimization 2003, *Lecture Notes in Computer Science*, Springer. pp. 1-15.
- [4.4] L. Barbulescu, J.P. Watson and D. Whitley, "Dynamic Representations and Escaping Local Óptima: Improving Genetic Algorithms and Local Search", *Seventeenth American Conference on Artificial Intelligence (AAAI) 2000*. pp: 879-884
- [4.5] S. Bleuer, M. Brack, L. Thiele, E. Zitzler, "Multiobjective Genetic Programming: Reducing Bloat using SPEA2", *Proceedings of CEC'2001*, pp. 536-543.
- [4.6] R. Caruana, J. Schaffer, "Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms", *Proceedings of the 5th Int. Conference on Machine Learning* 1988, Morgan Kaufmann.
- [4.7] U.K. Chakraborty, C.Z. Janikow, "An analysis of Gray versus binary encoding in genetic search", *Information Sciences: an International Journal; Special Issue: Evolutionary Computation*, 156 (2003) pp. 253-269.
- [4.8] S.Y. Chen, S.D. Rajan, 'Improving the efficiency of genetic algorithms for frame design', *Engineering Optimization*, 30 (3-4) 281-307, 1998.
- [4.9] C. Coello Coello, "A Short Tutorial on Evolutionary Multiobjective Optimization", pp. 21-40, in Evolutionary Multi-Criterion Optimization 2001, *Lecture Notes in Computer Science*, Springer.
- [4.10] C. Coello Coello, G.T. Pulido, "Multiobjective Optimization using a Micro-Genetic Algorithm". Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), San Francisco, Morgan Kaufmann.
- [4.11] D. Come, K. Deb, P. Fleming, J. Knowles, "The Good of the Many Outweighs the Good of the One: Evolutionary Multi-Objective Optimization", *IEEE coNNections*, 1 (1), pp. 9-13, 2003.
- [4.12] K. Deb. 'Evolutionary Algorithms for Multi-Criterion Optimisation in Engineering Design'. *Evolutionary Algorithms in Engineering and Computer Science (EUROGEN 1999)*, John Wiley & Sons.
- [4.13] K. Deb, Multiobjective Optimization using Evolutionary Algorithms. John Wiley & Sons - Series in Systems and Optimization- (2001).
- [4.14] L. Eshelman, 'The CHC Adaptive Search Algorithm : How to have Safe Search when engaging in nontraditional Genetic Recombination'. In *I Foundations Of Genetic Algorithms (FOGA 1990)* pp 265-284. Ed. G. Rawlings. Morgan Kaufmann Publishers.
- [4.15] L. Eshelman, J.D. Schaffer, Preventing premature convergence in genetic algorithms by preventing incest, in: R.K. Belew, L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, 1991, pp. 115-121.

- [4.16] M. Galante, Genetic algorithms as an approach to optimize real-world trusses, *Int. J. Numer. Meth. Engrg.* 39 (1996) 361-382.
- [4.17] D.E. Goldberg, M.P. Santani, Engineering optimization via genetic algorithm, in- *Proceedings of the Ninth Conference on Electronic Computation*, ASCE, New York, 1986, pp. 471-482.
- [4.18] D.E. Goldberg, '*Genetic algorithms for search, optimisation, and machine learning*' Reading, MA: Addison Wesley (1989).
- [4.19] D. Greiner, J.M. Emperador, G. Winter, "Single and Multiobjective Frame Optimization by Evolutionary Algorithms and the Auto-adaptive Rebirth Operator", in *Computer Methods in Applied Mechanics and Engineering*, Elsevier, 193 (2004) 3711-3743.
- [4.20] D. Greiner, J.M. Emperador, G. Winter, Multiobjective optimization of bar structures by Pareto GA, in: *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering*, Barcelona, CIMNE, 2000.
- [4.21] D. Greiner, G. Winter, J.M. Emperador, B. Galván, "Gray Coding in Evolutionary Multicriteria Optimization: Application in Frame Structural Optimum Design", *Lecture Notes in Computer Science* 3410, Evolutionary Multi-Criterion Optimization 2005, Springer, pp. 576-591.
- [4.22] D. Greiner, G. Winter, J.M. Emperador. 'Searching for an efficient method in multiobjective frame optimisation using evolutionary algorithms'. In *Computational Solid and Fluid Mechanics* Massachusetts Institute of Technology Conference on Computational Fluid and Solid Mechanics 2003, Elsevier Science, pp. 2285-2290.
- [4.23] D. Greiner, G. Winter y J.M. Emperador. 'Un estudio comparativo en optimización de estructural metálicas mediante los algoritmos NSGA-II y SPEA2 Algorithms'. In *Desarrollos recientes en métodos numéricos para ingeniería y ciencias aplicadas*, (2002), SVMNI, pp. 31-38.
- [4.24] D. Greiner, G. Winter, J.M. Emperador, "Optimising Frame Structures by different strategies of GA", *Finite Elem. in Anlys. and Design*, Elsevier, 37(5) pp.381-402, (2001).
- [4.25] D. Greiner, G. Winter, J.M. Emperador, "*Aplicación de los algoritmos genéticos en problemas de optimización de estructuras de pórticos planos y nudos rígidos*". P.F.C. E.T.S. Ingenieros Industriales. Universidad de Las Palmas de GC (2000).
- [4.26] R. Hinterding, H. Gielewski, T.C. Peachey, "The nature of mutation in genetic algorithms", *Proceedings of the Sixth International Conference on Genetic Algorithms* 1989, Morgan Kaufmann, pp. 70-79.
- [4.27] J.H. Holland, Genetic Algorithms, *Scientific American*, 1992, pp. 38-45.
- [4.28] M. Jensen, "Guiding Single-objective optimization using Multi-objective Methods", *Applications of Evolutionary Computation*, Lecture Notes in Computer Science, pp. 268-279, Best paper award EvoCOP 2003.
- [4.29] K. de Jong, J. Sarma, "Generation Gaps Revisited". In *II Foundations Of Genetic Algorithms (FOGA 1992)* pp. 19-28. Ed. D. Whitley. Morgan Kaufmann Publishers.
- [4.30] E. de Jong, R. Watson, J. Pollack, "Reducing bloat and promoting diversity using multi-objective methods", *Proceedings of the Genetic and Evolutionary Computation Conference* 2001, pp. 11-18.
- [4.31] J. Knowles, R. Watson, D. Come, "Reducing Local Óptima in Single-Objective Problems by Multiobjectivization". *Evolutionary Multi-Criterion Optimization* 2001, *Lecture Notes in Computer Science*, pp. 269-283.

- [4.32] K. Krishnakumar, Micro-genetic algorithms for stationary and non-stationary function optimization, SPIE: *Intell. Control Adapt. Syst.* 1196 (1989).
- [4.33] K.E. Mathias, D. Whitley, "Transforming the Search Space with Gray Coding", In D. Schaffer, ed. *IEEE Int. Conf. on Evolutionary Computation*, (1994) pp. 513-518.
- [4.34] G. Meneghetti, V. Pediroda, C. Poloni, "Application of a multiobjective Genetic Algorithm and a Neural Network to the Optimization of Foundry Processes", *Evolutionary Algorithms in Engineering and Computer Science*, pp. 457-469. Eds: K. Miettinen, P. Neittanmaki, M. Makela, J. Periaux. John Wiley & Sons, 1999.
- [4.35] D.J. Mikulin, *Using genetic algorithms to fit HLGGM data*, Ph.D. Thesis, University of Exeter, 1997.
- [4.36] Y. Nojima, K. Narukawa, S. Kaige, H. Ishibushi, "Effects of Removing Overlapping Solutions on the Performance of the NSGAI Algorithm", Third International Conference on Evolutionary Multi-Criterion Optimization 2005, *Lecture Notes in Computer Science* n° 3410, pp. 341-354.
- [4.37] N.J. Radcliffe, P.D. Surry, "Fundamental Limitations on Search Algorithms: Evolutionary Computing in Perspective", *Lecture Notes in Computer Science* 1000 (Computer Science Today: Recent Trends and Development), Springer, 1995.
- [4.38] S. Rajeev, C.S. Krishnamoorthy, Genetic algorithms-based methodologies for design optimization trusses, *J. Struct. Engrg. ASCE* 123-3 (1997) 350-358.
- [4.39] J. Rowe, D. Whitley, L. Barbulescu, P. Watson, "Properties of Gray and Binary Representations", *Evolutionary Computation*, 12-1 (2004) pp. 46-76.
- [4.40] G. Rudolph (1994). "Convergence Analysis of Canonical Genetic Algorithms", *IEEE Transactions on Neural Networks* 5-1 (1994) pp. 96-101.
- [4.41] G. Syswerda, "A Study of Reproduction in Generational and Steady-State Genetic Algorithms", *Foundations of Genetic Algorithms (FOGA 1991)*, Morgan Kaufmann, pp. 94-101.
- [4.42] G. Syswerda, "Uniform Crossover in Genetic Algorithms", *Proceedings of the Third International Conference on Genetic Algorithms* 1989, Morgan Kaufmann, pp. 2-9.
- [4.43] D. A. Van Veldhuizen, G. Lamont, "*Multiobjective Evolutionary Algorithm Test Suites*", Proceedings of the 1999 ACM Symposium on Applied Computing. pp. 351-357, San Antonio, Texas, ACM.
- [4.44] N. Weicker, G. Szabo, K. Weicker, P. Widmayer, "Evolutionary Multiobjective Optimization for Base Station Transmitter Placement with Frequency Assignment", *IEEE Transactions on Evolutionary Computation*, 7-2 (2003) pp. 189-203.
- [4.45] D. Whitley, "A free lunch proof for Gray versus Binary Codings", *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO 1999, Morgan Kaufmann.
- [4.46] D. Whitley, "The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best", *Proceedings of the Third International Conference on Genetic Algorithms* 1989, Morgan Kaufmann, pp. 116-123.
- [4.47] D. Whitley, J. Kauth, "Genitor: A Different Genetic Algorithm", Technical Report Cs-88-101, Colorado State University.
- [4.48] D. Whitley, K. Mathias, P. Fitzhom, Delta coding: an iterative search strategy for genetic algorithms, in: R.K. Belew, L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, 1991, pp. 77-84.

- [4.49] D. Whitley, K. Mathias, S. Rana, J. Dzubera, "Evaluating Evolutionary Algorithms", *Artificial Intelligence Journal* 85 (1996) pp. 245-276.
- [4.50] D. Whitley, S. Rana, R. Heckendorn, "Representation Issues in Neighborhood Search and Evolutionary Algorithms", in eds.: D. Quagliarella, J. Periaux, C. Poloni, G. Winter, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science* (1997), John Wiley & Sons.
- [4.51] D.H. Wolpert, W.G. MacReady, "No Free Lunch Theorems for Optimization", *IEEE Transactions on Evolutionary Computation*, 4 (1997) pp. 67-82.
- [4.52] D.H. Wolpert, W.G. MacReady, "No Free Lunch Theorems for Search", Technical Report SFI-TR-95-02-010, Santa Fe Institute, July 1995.
- [4.53] E. Zitzler, M. Laumanns, L. Thiele, "*SPEA2: Improving the Strength Pareto Evolutionary Algorithm*", TIK-Report 103, (May 2001) Swiss Federal Institute of Technology (ETH) Zurich.
- [4.54] E. Zitzler, M. Laumanns, L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization", In: *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Eds: KC. Giannakoglou, DT. Tsahalis, J. Periaux, KD. Papaihou, T. Fogarty, (EUROGEN 2001), Septiembre 2001, Athens, Greece.
- [4.55] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications", Swiss Federal Institute of Technology (ETH) Zürich, *PhD Thesis*, 1999.

Capítulo 5. Aplicaciones a otros problemas de optimización multiobjetivo en ingeniería industrial

La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica.

ARISTÓTELES (384- 322 A.C)

5.1 Introducción. Objetivos del estudio

La resolución de problemas de optimización reales de aplicación en otras aplicaciones de las ciencias e ingeniería mediante algoritmos evolutivos es muy numerosa. En el último lustro, el desarrollo de nuevos algoritmos evolutivos multiobjetivo eficientes ha extendido esta resolución a problemas multicriterio, siendo los campos donde han sido usados muy numerosos (ver sección 2.2).

En este capítulo se presenta la resolución mediante algoritmos evolutivos multiobjetivo de otros problemas de diseño óptimo en ingeniería que forman parte de las líneas de investigación seguidas en la división de Computación Evolutiva y Aplicaciones (CEANI) del Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI) de la Universidad de Las Palmas de Gran Canaria (ULPGC): Diseño Óptimo de Sistemas de Seguridad y Diseño Óptimo en Ingeniería Eléctrica.

Se hará especial énfasis en aquellos aspectos comunes o en concordancia con el problema estructural abordado con anterioridad en capítulos anteriores y en aportaciones aplicadas en él que pudieran ofrecer ventajas competitivas en la resolución de estos problemas.

5.2 Problema de Optimización Multiobjetivo en Diseño de Sistemas de Seguridad

5.2.1 Introducción

Los conceptos considerados en la disciplina de ingeniería conocida como diseño de sistemas han experimentado una considerable evolución en la última década, no sólo a causa de la creciente complejidad de los sistemas modernos, sino también debido al cambio de criterio que implica la necesidad de obtener diseños óptimos en lugar de diseños meramente adecuados. Los requerimientos de diseño son especialmente estrictos en los sistemas cuyo fallo implica daño a personas, medio ambiente o instalaciones de importancia social-técnica. Para estos sistemas, modernos requerimientos de seguridad obligan a considerar escenarios complejos con muchas variables, normalmente desarrollados bajo marcos de Asesoramiento Probabilístico de Riesgos. En estos sistemas debe ser prestada especial atención al diseño de Sistemas de Seguridad, considerando alternativas de diseño (en la etapa de diseño) y/o estrategias de mantenimiento (durante la operación del sistema), con el fin de desarrollar una optimización global. En ambos casos, diferentes objetivos en conflicto serán considerados simultáneamente por los tomadores de decisiones, de tal forma que los métodos multiobjetivo son especialmente adecuados de ser usados.

Puntos clave en la Optimización de Diseño de Sistemas de Seguridad (ODSS) son el uso de modelos matemáticos avanzados de los sistemas y potentes métodos multiobjetivo de optimización. El Análisis de Árboles de Fallo (AAF) es el modelo matemático más ampliamente usado, mientras que los Algoritmos Evolutivos han sido empleados satisfactoriamente en optimización. Aunque grandes avances se han introducido en los últimos años, problemas importantes requieren mayor desarrollo para facilitar la aplicación a una amplia variedad de sistemas técnicos complejos, especialmente el uso de métodos multiobjetivo más eficientes.

Sistemas de Optimización considerando Confiabilidad han sido estudiados por más de una década [5.35], pero ODSS usando optimizadores evolutivos han sido introducidos recientemente [5.3], y continúan en los últimos años [5.42] hacia sistemas más complejos. Las contribuciones mencionadas incluyen tanto el diseño del sistema como la estrategia de mantenimiento durante la operación del sistema. En ciertos problemas, sólo una de las dos es necesaria, por tanto ambas han sido estudiadas independientemente (estrategia de mantenimiento de sistemas existentes [5.38][5.39]; y diseño del sistema [5.21]). La no-disponibilidad y el coste han sido consideradas como las funciones objetivo principales a optimizar tanto en mono-objetivo (minimizar la no-disponibilidad sujeto a restricciones de coste) como en multi-objetivo (minimizar la no-disponibilidad y el coste simultáneamente) [5.24]. Los algoritmos evolutivos [5.34][5.25] han sido el método de optimización preferente por autores [5.23], principalmente por su capacidad de búsqueda de soluciones en espacios de búsqueda complejos y de gran tamaño.

Las opciones de diseño de sistemas han sido consideradas usando Árboles de Fallos con puertas lógicas apropiadas para permitir selección de componentes y de niveles de redundancia-diversidad [5.2]. En los últimos años nuevas puertas lógicas han sido desarrolladas, permitiendo modelar sistemas complejos, dinámicos [5.16] o tolerantes a fallos [5.15]. La principal tarea que atañe a los modelos de árboles de fallos, es la evaluación cuantitativa, que ha sido desarrollada usando la aproximación clásica de mínimos conjuntos de corte (MCC) [5.46], o los diagramas de decisión binarios (DDB) [5.7][5.10]. Ambas aproximaciones tienen conocidos inconvenientes: la primera presenta crecimiento exponencial con el número de eventos básicos o puertas lógicas del árbol, mientras que la última depende de la ordenación de eventos básicos sin fundamentos matemáticos, soportada sólo por unas pocas heurísticas existentes [5.4][5.5]. Experiencias con sistemas reales demuestran pobres resultados con MCC y coste computacional prohibitivo de la evaluación del árbol de fallos usando DDB. En consecuencia, nuevas aproximaciones están bajo desarrollo como el uso de diagramas de expresión booleana (DEB) [5.48], métodos directos basados en un criterio de orden intrínseco (COI) de los eventos básicos [5.26][5.27], o métodos de Monte Carlo basados en potentes técnicas de reducción de varianza [5.17].

En este trabajo, un estudio comparativo del desempeño de los últimos y más avanzados métodos multiobjetivo es presentado aplicado a problemas ODSS. Las características principales de la aproximación son:

1. La construcción de diagramas de decisión binarios es puenteada usando un método directo y eficiente ('método de los pesos' [5.17][5.26]) para evaluar árboles de fallos.
2. Una codificación apropiada del problema es usada integrando restricciones de diseño y permitiendo reducir el espacio de búsqueda y la aparición de soluciones no factibles.

3. Un análisis de exploración-explotación es realizado en búsqueda de un método evolutivo multiobjetivo eficiente, y considerando los algoritmos SPEA2 (con un operador truncamiento optimizado [5.31]), NSGAI, y dos algoritmos NSGAI con elitismo controlado.

En las siguientes secciones, las características principales de la metodología empleada son descritas. En la sección 2 el diseño de sistemas de seguridad es introducido. La sección 3 presenta los algoritmos evolutivos empleados, y la sección 4 describe los dos casos test de un sistema de seguridad de una planta nuclear de generación de energía eléctrica, además de consideraciones particulares. En la sección 5 los resultados son descritos y analizados, y el capítulo finaliza con la sección de conclusiones.

5.2.2 Diseño Óptimo de Sistemas de Seguridad

La misión de los sistemas de seguridad es operar cuando ciertas condiciones o eventos suceden, previniendo situaciones dañinas comprometidas o mitigando sus consecuencias. Cuando sea posible, el diseño de dichos sistemas debe asegurar el éxito de la misión incluso en el caso de un fallo único de sus componentes, o cuando sucesos externos puedan prevenir subsistemas de seguridad completos de funcionar. El diseñador debe optimizar el desempeño conjunto del sistema decidiendo el mejor compromiso entre: La selección de componentes, la localización de componentes/subsistemas y, en algunos casos intervalos temporales entre mantenimientos preventivos.

En problemas ODSS la medida de desempeño del sistema es normalmente la no-disponibilidad del sistema, siendo una variable secundaria el coste del sistema. La tarea de optimización completa está restringida, a causa de consideraciones prácticas que limitan los recursos físicos y económicos disponibles.

Los componentes del sistema son elegidos de entre alternativas posibles, cada una de las cuales de diferente naturaleza, características y manufactura. La localización de componentes/subsistemas implica decidir la disposición conjunta del sistema así como los niveles de redundancia, diversidad y separación física. La redundancia duplica ciertos componentes del sistema; la diversidad es incorporada alcanzando ciertas fiaciones por medio de dos o más componentes/subsistemas diferentes. La separación física entre subsistemas garantiza el éxito de la misión, incluso en el caso de causas críticas comunes de fallo.

En ODSS automatizados, un software se encarga de obtener el diseño óptimo considerando todas las alternativas y restricciones mencionadas con anterioridad. Existen dos componentes principales en esos programas: el modelo del sistema y el método de optimización.

5.2.2.1 El Modelo del Sistema

El modelo del sistema debe reflejar todas las alternativas de diseño, normalmente usando como punto de partida el conocimiento del equipo de diseño y las restricciones físicas de los recursos. Un conjunto de variables binarias llamadas 'variables de indicación' permiten elegir (interruptor 'encendido') o rechazar (interruptor 'apagado') las diferentes alternativas de configuración. Para cada configuración seleccionada se calcula la no-disponibilidad del sistema y el coste. El análisis de árboles de fallo (AAF) es la metodología más ampliamente aceptada para modelizar sistemas de seguridad con propósitos de diseño óptimo.

FaUo de SCIS

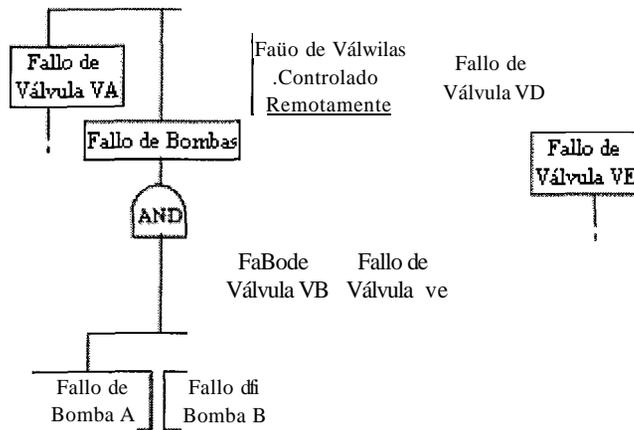


Figura 5.1. Árbol de Fallos Simplificado del Sistema de Contención de Inyección mediante Spray de una central nuclear

Un árbol de fallos es un diagrama lógico booleano usado para identificar las relaciones causales que conducen a un modo específico de fallo del sistema de seguridad [5.46]. El modo de fallo del sistema es denominado el 'evento cabecero' y el árbol de fallo es desarrollado usando puertas lógicas bajo este evento hasta que los eventos de fallo componentes, denominados 'eventos básicos' son localizados. H.A. Watson de los laboratorios de teléfonos Bell desarrolló el análisis de árboles de fallo en 1961-62, y los primeros artículos publicados fueron presentados en 1965 [5.33]. Desde entonces, el uso de la metodología ha sido muy difundido y avances importantes a nivel teórico [5.44][5.46], metodológico [5.37] y práctico [5.40] se han desarrollado. La figura 5.1 muestra un árbol de fallos simplificado del sistema representado en la figura 5.3 (primer caso test).

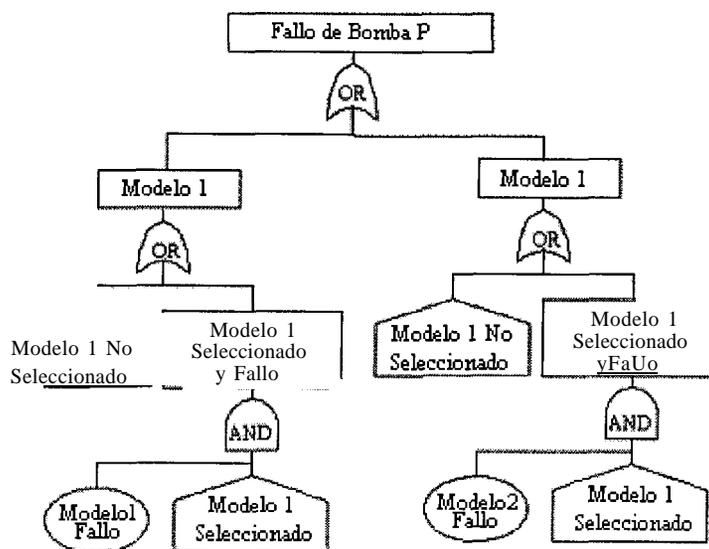


Figura 5.2. Ejemplo de Árbol de Fallos con Alternativas de Diseño

Una contribución importante para ODSS fue la posibilidad de introducir alternativas de diseño en el árbol de fallos [5.2], por medio de la asociación de 'eventos casa' usados en la lógica del árbol de

fallos con 'variables indicadoras' del diseño del sistema. La figura 5.2 muestra una sección del árbol de fallos, que permite seleccionar entre dos modelos de bombas hidráulicas y su nivel de redundancia (una o dos bombas). De entre otras medidas del sistema [5.46], el AAF permite calcular la no-disponibilidad del sistema para cada configuración de los 'eventos casa' (variables indicadoras). A lo largo del tiempo, muchos métodos han sido desarrollados para la evaluación cuantitativa de los árboles de fallos.

Durante los años 90 fue presentada y desarrollada la contribución más importante a esta tarea: el uso de la conversión de árboles de fallos a diagramas de decisión binarios (DDB) [5.7][5.10] que son adecuados para el cálculo sencillo de la medida del 'evento cabecero' (en nuestro caso la no-disponibilidad del sistema). A pesar del difundido desarrollo y uso de la metodología, un inconveniente importante permanece aún sin solución: la determinación de la ordenación de eventos básicos necesaria para obtener el más simplificado DDB adecuado para un fácil cálculo, de entre el conjunto de DDB asociados con un árbol de fallos. Algunas contribuciones han sido presentadas en el área [5.4][5.5], pero hasta hoy, sólo unas pocas heurísticas soportan la selección de la adecuada ordenación de eventos básicos. Ésa es la razón de evaluar árboles de fallo. Usamos en este trabajo un algoritmo directo eficiente, denominado el 'método de los pesos', que proporciona límites superior e inferior exactos de la no-disponibilidad del sistema para un error preespecificado máximo. Este algoritmo está basado en un teorema que establece un criterio de orden intrínseco (COI) para probabilidades de estado elementales. El COI permite seleccionar "a priori" los estados elementales más relevantes para computar la probabilidad del evento cabecera, con independencia de las probabilidades básicas del sistema. El COI está exclusivamente basado en las posiciones de Os y 1 s de las n-uplas binarias de los estados elementales. Este método ha sido exhaustivamente descrito en los últimos años [5.17][5.26].

5.2.2.2 El Método de Optimización

Los algoritmos evolutivos [5.23] [5.25] [5.34] han sido el método de optimización más empleado. Una consideración importante previa al uso de un algoritmo evolutivo es decidir la codificación de las soluciones candidatas. Tanto código binario como código real puede ser usado, siendo los operadores a emplear en cada caso distintos. En el caso de problemas de ODSS que consideren alternativas de diseño y estrategias de mantenimiento, el código más empleado ha sido el binario [5.3]. Aunque existen aplicaciones de algoritmos evolutivos mono y multiobjetivo con codificaciones avanzadas mixtas enteras, recientemente ciertos autores han proclamado problemas de convergencia en ODSS [5.42] a causa de la diferente naturaleza de las variables implicadas, por un lado variables binarias estrictas (alternativas de diseño) y por otra parte variables reales (estrategias de mantenimiento). En la sección siguiente se describe en profundidad el método de optimización considerado.

5.2.3 La Aproximación Evolutiva

Si la mejora en un criterio implica el empeoramiento en otro, como ocurre en el caso de la no-disponibilidad y el coste del sistema, (ambas funciones a minimizar), se requiere una optimización

multiobjetivo. Tratando de resolver de manera eficiente el problema tratado, un análisis de distintos métodos multiobjetivo y probabilidades de mutación ha sido considerado, concentrándonos especialmente en el balance explotación-exploración. Entre los algoritmos más recientes [5.9], aquellos que incluyen elitismo e independencia paramétrica son destacados [5.43]. Una primera comparación de métodos multiobjetivo evolutivos puede verse en [5.32]. Se han escogido para este caso, los algoritmos SPEA2 [5.58], NSGA-II [5.14], y NSGA-II con elitismo controlado (EC) [5.13]. Se han considerado dos métricas, definidas en el espacio de objetivos, referidas a la precisión y cobertura del fi-ente de soluciones no-dominadas encontrado. Se han promediado sobre treinta ejecuciones independientes de cada algoritmo y se han representado durante el proceso de convergencia del algoritmo en las figuras 5.8 a 5.16.

La primera métrica (Métrica 1) es la métrica MI* de Zitzler [5.57], representativa del acercamiento al fi-ente de Pareto óptimo. Para evaluar esta métrica, perteneciente al tipo de métricas escalables, el mejor fi-ente de Pareto debe ser conocido. Su expresión es [5.57]:

$$M^*(U) = \frac{\sum_{y \in Y^p} \min_{u \in U} \|u - y\|}{|Y^p|} \quad (5.1)$$

Donde $U = f(A) \subset Y$ (siendo A un conjunto de vectores de decisión, Y el espacio de objetivos y Y^p referido al conjunto de Pareto).

La segunda métrica (Métrica 2) mide la cobertura del fi*ente. Se adopta el valor escalado de la métrica M3* de Zitzler [5.57]:

$$M3^*(C) = \frac{\text{popsize}}{2} \max_{i=1} \{ \frac{|M_i - V_i|}{|M_i|} \} \quad (52)$$

La métrica 2 se evalúa como:

$$Metric\ 2 = \frac{\sum_{j=1}^n \sum_{r=1}^n (j^r - r^j)}{MaxM3}$$

Por tanto, significa que alcanzando un valor de uno, la máxima cobertura del frente es alcanzada. En el caso test 2 (figura 5.5), se observa que las soluciones más difíciles de localizar son las inferiores a la derecha, aquellas con coste mayor a 912, como puede observarse en la tabla 5.2 y en la zona derecha de la figura 5.6. Por tanto, la carencia de cobertura total del frente reflejada en las figuras de resultados, se debe a la dificultad de alcanzar las mencionadas soluciones.

5.2.4 Casos Test

5.2 A A Descripción

Un diagrama simplificado del modelo de un sistema de contención por inyección en spray (SCIS) de una central nuclear de generación eléctrica puede verse en la figura 5.3, donde V significa válvula, y B, bomba. La misión del SCIS es la inyección de suficiente cantidad de agua borada en el contenedor para limpiar la contaminación radiactiva liberada tras un accidente de pérdida de refrigerante. Los valores de coste y no-disponibilidad de los componentes del sistema son descritos en la tabla 5.1, siendo C el valor del coste del componente y P, la probabilidad de no-disponibilidad del componente. Para los válvulas uno y cuatro, tres modelos distintos son posibles; para el resto de válvulas y las bombas, dos modelos distintos son considerados. El conjunto de soluciones total y el conjunto de Pareto pueden ser observadas en la figura 5.4, donde el eje x representa el coste total del sistema y el eje y representa su no-disponibilidad.

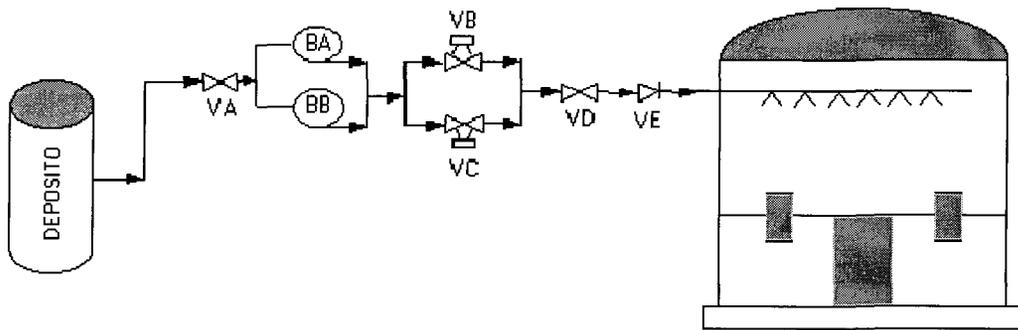


Figura. 5.3. Diagrama del caso test A

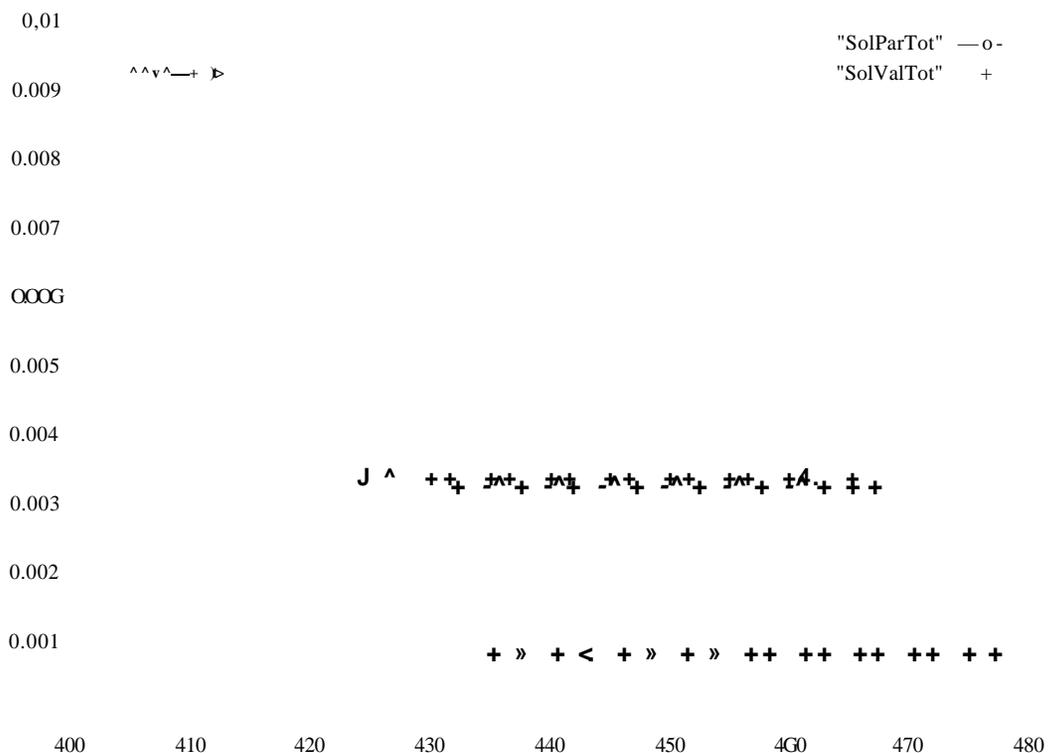


Figura 5.4. Soluciones y Conjuntos de Pareto para el caso test A

Tabla 5.1. Valores de parámetros del caso test A.

	VA, VD	VB, ve	VE	PA,PB
Modelo 1	P=2.9E-3 C=50	P=3.0E-3 C=65	P=5.0E-4 C=37	P=3.5E-3 C=90
Modelo 2	P=8.7E-3 C=35	P=1.0E-3 C=70	P=6.0E-4 C=35	P=3.8E-3 C=85
Modelo 3	P=4.0E-4 C=60	"	-	""

Si se duplica la línea, por motivos de seguridad, obtenemos el diagrama representado en la figura 5.5, donde los componentes de la segunda rama tienen los mismos parámetros que los primeros. El valor de las soluciones de Pareto es representado en la tabla 5.2. En este caso test B, el espacio de búsqueda se ha incrementado de forma notable por el incremento de posibles combinaciones de elección de los dispositivos producto de tener dos ramas en paralelo en lugar de una, como puede ser observado en la figura 5.6.

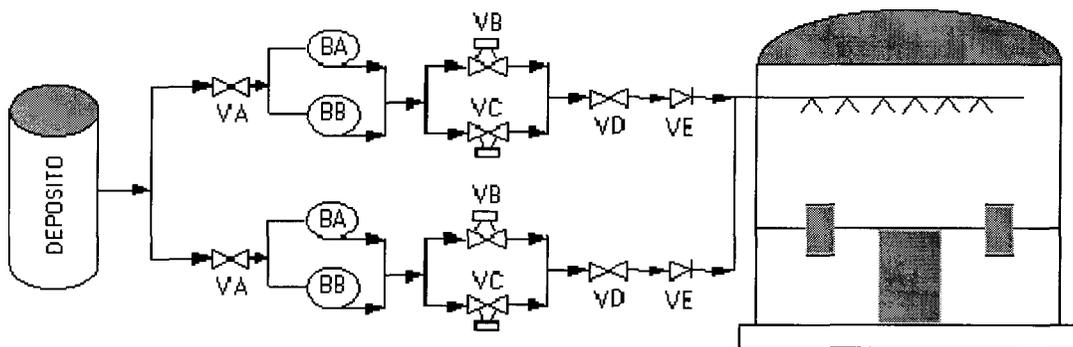


Figura 5.5. Diagrama del caso test B

Tabla 5.2. Conjunto de Valores de Pareto del caso test B

Cost	Unavailability	Cost	Unavailability
810	0.0003217533	877	0.0000161601
812	0.0003199908	879	0.0000160294
814	0.0003182379	884	0.0000159571
819	0.0003181330	885	0.0000138547
824	0.0003180280	887	0.0000128842
825	0.0002186917	889	0.0000127531
827	0.0002169189	890	0.0000091560
829	0.0002157307	892	0.0000085154
834	0.0002156252	894	0.0000083839
835	0.0001742693	899	0.0000083462
837	0.0001724920	900	0.0000056120
839	0.0001715472	902	0.0000052204
840	0.0001150281	904	0.0000050886
842	0.0001132449	909	0.0000050657
844	0.0001126247	910	0.0000020593
849	0.0001125186	912	0.0000019171
850	0.0000703462	914	0.0000017850
852	0.0000685586	919	0.0000017771
854	0.0000681832	924	0.0000017692
859	0.0000680768	929	0.0000017666
860	0.0000255526	934	0.0000017640
862	0.0000237605	939	0.0000017625
864	0.0000236305	944	0.0000017610
869	0.0000235239	949	0.0000017596
874	0.0000234884	954	0.0000017583
875	0.0000173781		

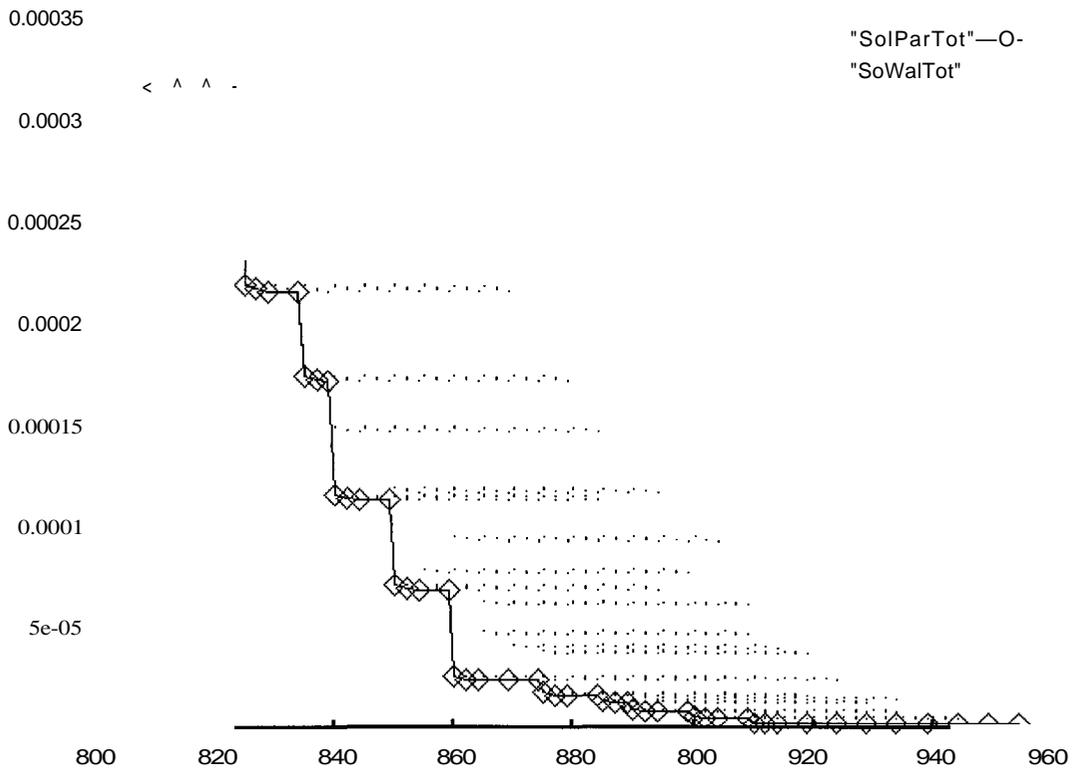


Figura 5.6. Soluciones y Conjuntos de Pareto para el caso test B

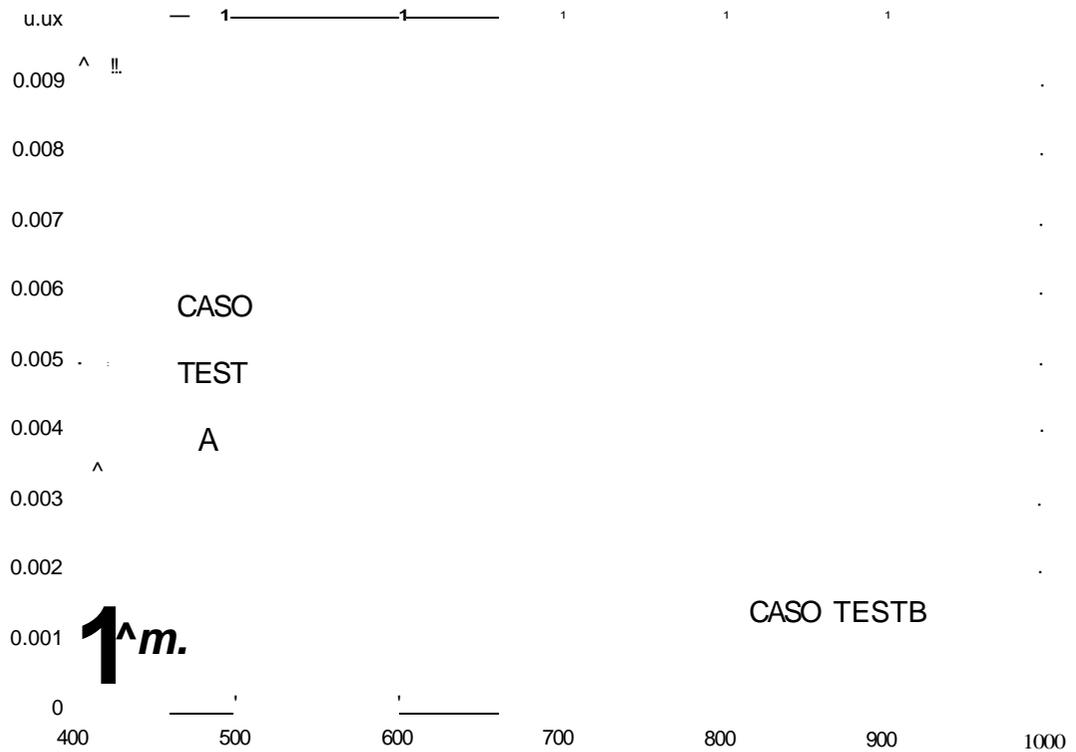


Figura 5.7. Frentes Óptimos de los casos Test A y B

Se puede observar, que el coste se ha incrementado con respecto al primer caso test, sin embargo la no-disponibilidad del sistema ha descendido de forma significativa. En la figura 5.7 se puede apreciar gráficamente el resultado de duplicar en paralelo la rama. El conjunto de soluciones óptimas posibles

del caso test B, ha conseguido soluciones mucho menores en no-disponibilidad (desplazamiento de las soluciones cuadradas hacia abajo) -por tanto soluciones más fiables-, si bien ello ha sido a costa de incrementar el coste de la instalación (desplazamiento hacia la derecha de las soluciones cuadradas) -por tanto soluciones económicamente más caras-. Por tanto, las soluciones de Pareto del caso B (cuadrados) aparecen en un fi-ente no-dominado desplazado hacia abajo (menor no-disponibilidad) y hacia la derecha (mayor coste), en comparación con el caso A, donde el fi^ente de soluciones no-dominadas de Pareto (triángulos) están situadas más a la izquierda (menor coste) y arriba (mayor no-disponibilidad).

5.2.4.2 Consideraciones Evolutivas

Con el fin de estudiar independientemente el efecto de los dos factores del elitismo controlado, se consideran dos valores del parámetro r , cero y 0.4. Se imponen cuatro frentes ($K=4$) y por tanto, se consideran cuatro algoritmos en nuestro análisis (SPEA2, NSGAI y dos algoritmos con elitismo controlado en NSGAI). Se consideran treinta ejecuciones independientes por algoritmo testado y tres probabilidades de mutación uniforme (0%, 0.4% and 3%). En total, todo ello supone doce combinaciones, balanceando exploración y explotación.

El tamaño de la población es de cien individuos (en el caso de NSGA-II con EC y $r=0.4$, este tamaño poblacional supone los siguientes tamaños de los frentes: 62, 25, 10 y 3). Un cruce uniforme con probabilidad de 1.0, y codificación binaria son empleados.

El tratamiento de las restricciones en este problema es de gran importancia. Se relaciona con respecto a la configuración del problema descrito, considerando las alternativas de diseño. La codificación binaria estándar representa en cada bit cada alternativa simple de componentes de diseño [5.42], considerando 1 su selección y 0 su rechazo. Se propone aquí la integración de las restricciones en la codificación del cromosoma, permitiendo el reducir el espacio de búsqueda y las posibles soluciones no factibles, cuyo árbol de fallos no será evaluable para la no-disponibilidad del sistema [5.28][5.32]. Su significación en los dos casos tests presentados se muestra a continuación:

Primer Caso Test: Soluciones Factibles: 288; Soluciones óptimas de Pareto: 17. Tamaño del espacio de búsqueda para la codificación binaria simple con 16 bits = 65536 soluciones. Tamaño del espacio de búsqueda para la codificación binaria integrante de las restricciones con 9 bits = 512 soluciones.

Segundo Caso Test: Soluciones Factibles: 82944; Soluciones óptimas de Pareto: 178. Tamaño del espacio de búsqueda para la codificación binaria simple con 32 bits = 4294967296 soluciones. Tamaño del espacio de búsqueda para la codificación binaria integrante de las restricciones con 18 bits = 262144 soluciones.

Debido al mayor tamaño del espacio de búsqueda, concentramos el análisis en el segundo caso test, siendo el primero demasiado pequeño con la codificación incorporando restricciones para ser analizado.

5.2.5 Análisis de Resultados

Los resultados promedios de las treinta ejecuciones para cada métrica son presentados. A la izquierda se representa la métrica 1 (precisión) y a la derecha la métrica 2 (cobertura). El eje x corresponde con

el número de generaciones y el eje y con el valor de la métrica referida. Las figuras se organizan en dos grupos. En la primera sección (figuras 5.8 a 5.11) se muestran los resultados comparativos agrupados por algoritmo y comparando las diferentes probabilidades de mutación. En estas figuras, los símbolos diamante, cruz, y cuadrado se corresponden con las probabilidades de mutación de 3%, 0.4% and 0% respectivamente. Una segunda sección (figuras 5.12 a 5.16) muestra los resultados comparados agrupados por probabilidad de mutación y comparando los distintos algoritmos. Las figuras 5.12 a 5.14 representan los promedios de las métricas y las figuras 5.15 y 5.16 las desviaciones típicas de las métricas. En estas figuras, los símbolos diamante, cruz, cuadrado y aspa se corresponden respectivamente con los algoritmos NSGA-II con EC y $r=0.4$; NSGA-II con elitismo controlado y $r=0.0$, NSGA-II y SPEA2.

5.2.5.1 Analizando los Algoritmos con Diferentes Probabilidades de Mutación

De la observación de la métrica 1 (parte izquierda de las figuras) de las figuras 5.8 a 5.11, puede observarse un comportamiento común en los cuatro algoritmos testeados. En las generaciones iniciales, la mayor probabilidad de mutación (diamante) es la que exhibe una convergencia más lenta hacia el frente óptimo de Pareto, y el algoritmo que carece de mutación es el más rápido (cuadrado). El algoritmo con probabilidad de mutación de 0.4% está entre ellos. La menor exploración permite una explotación más intensiva que incrementa la presión de selección sobre el mejor individuo encontrado hasta el momento. Sin embargo, se aprecia que la diferencia no es significativa, y alrededor de la generación 20 todos los casos alcanzan una aproximación similar al frente óptimo no-dominado.

La métrica 2 es indicativa de la cobertura total del frente. Aquí las diferencias entre probabilidades de mutación son más notorias. El comportamiento común se presenta en las figuras 5.8, 5.9 y 5.11 (parte derecha de las figuras) donde la requerida diversidad introducida en la población por una mayor probabilidad de mutación, es esencial para localizar las soluciones localizadas en el extremo inferior derecho del frente. La línea de diamantes es la más cercana al mejor valor de uno en todos los algoritmos estudiados. Observamos cómo una probabilidad de mutación del 3% conduce a una velocidad de convergencia hacia el frente no muy lejana de las otras, y permite una mayor cobertura del mismo.

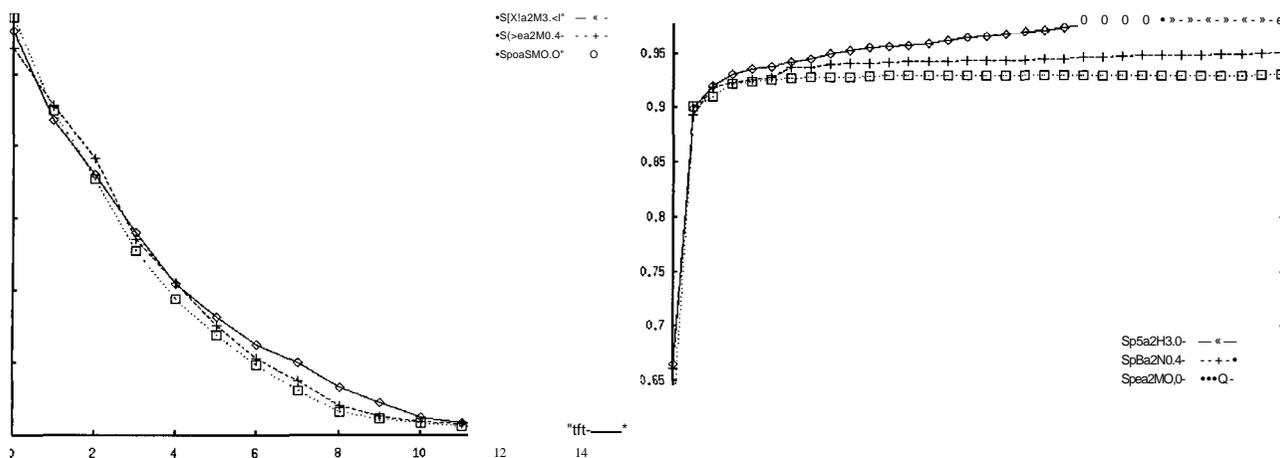


Figura 5.8. Promedios de las métricas 1 y 2, comparando SPEA2 con diferentes probabilidades de mutación

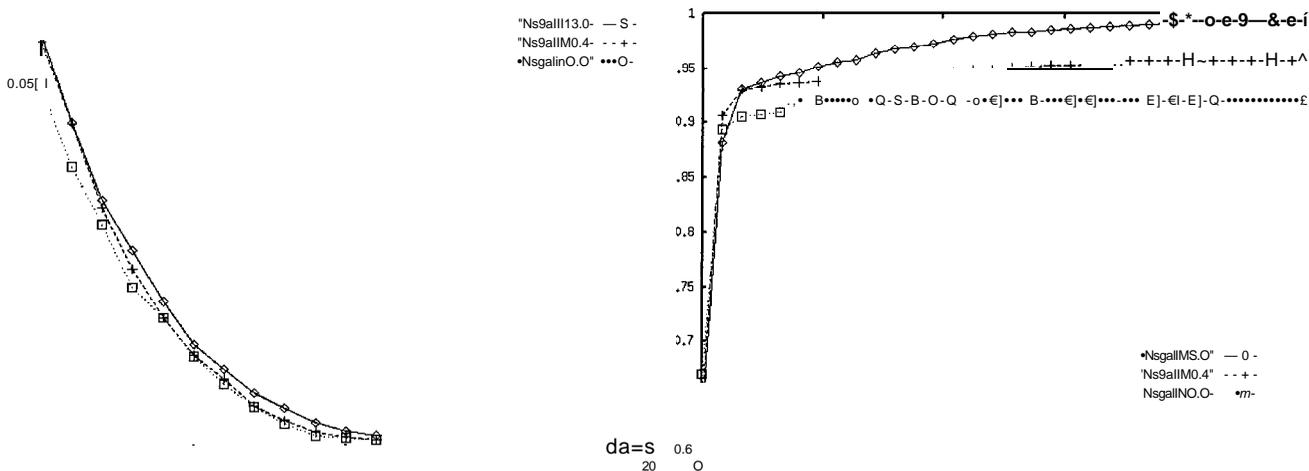


Figura 5.9. Promedios de las métricas 1 y 2, comparando NSGAI con diferentes probabilidades de mutación

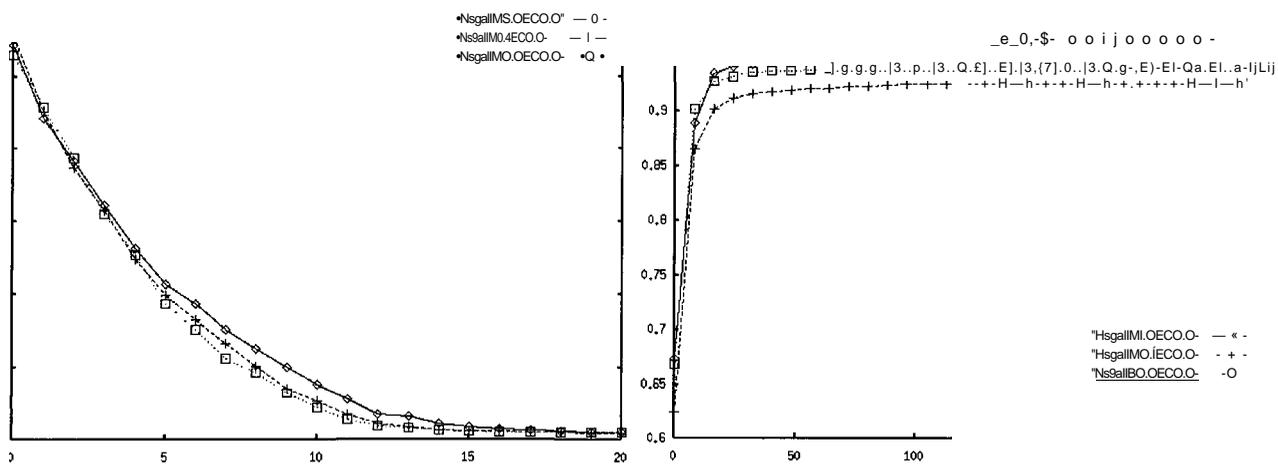


Figura 5.10. Promedios de las métricas 1 y 2, comparando NSGAI con elitismo controlado ($r=0.0$) con diferentes probabilidades de mutación

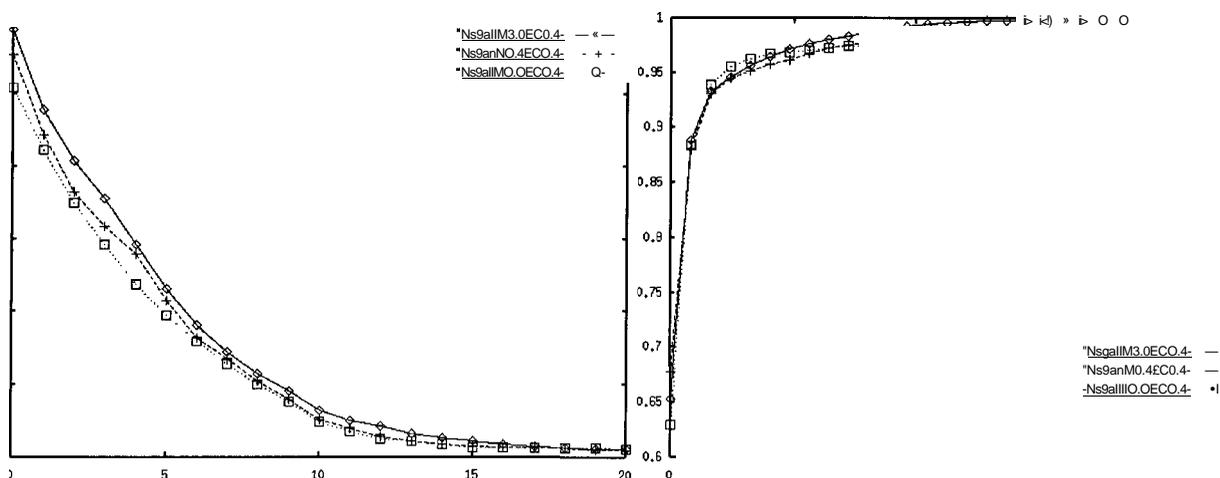


Figura 5.11. Promedios de las métricas 1 y 2, comparando NSGAI con elitismo controlado ($r=0.4$) con diferentes probabilidades de mutación

5.2.5.2 Analizando los Algoritmos con Probabilidad de Mutación Fija

De la observación de la métrica 1 (parte izquierda de las figuras) de las figuras 5.12 a 5.14, puede observarse un comportamiento común en los cuatro algoritmos ensayados. En las generaciones iniciales, los dos algoritmos con elitismo controlado (EC) (diamante y cruz) son los que experimentan una convergencia más lenta hacia el frente de Pareto, y el NSGAI, que tiene la mayor presión de selección (elitismo más selección por torneo), es el más rápido (cuadrado). Como ha sido deducido también en la sección anterior, aquí nuevamente las figuras muestran que incrementando el balance de explotación favorece una búsqueda más rápida en las etapas iniciales. Sin embargo, entonces la diversidad es decrementada, como se observa en la métrica 2. El algoritmo que introduce mayor diversidad en la población, el NSGAI con EC y $r=0.4$, alcanza los mayores valores de la métrica de cobertura. Es destacable la colaboración positiva entre altas probabilidades de mutación y el elitismo controlado, como puede observarse en la parte derecha de las figuras 5.14 y 5.16, mostrando que el elitismo controlado y la mutación pueden ser complementarios, interactuando simultáneamente de forma positiva. Otros estudios, como Laumanns et al. [5.36] aplicado al problema biobjetivo de knapsack, encuentran que un fuerte elitismo y altas probabilidades de mutación presentan el mejor desempeño global, y aquí hay una cierta combinación de ambos.

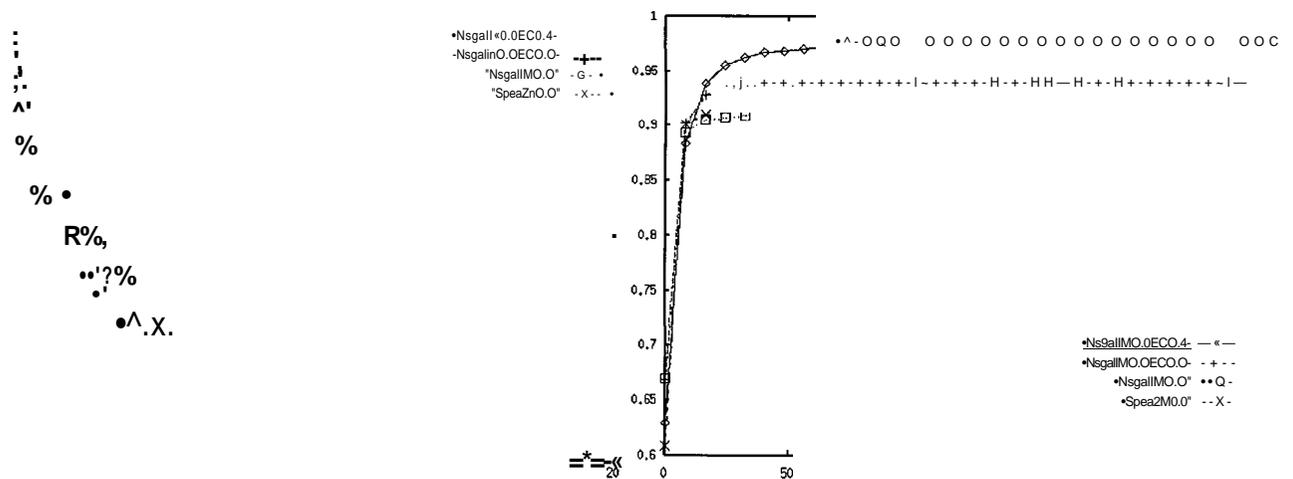


Figura 5.12. Promedios de las métricas 1 y 2, comparando los cuatro algoritmos sin mutación

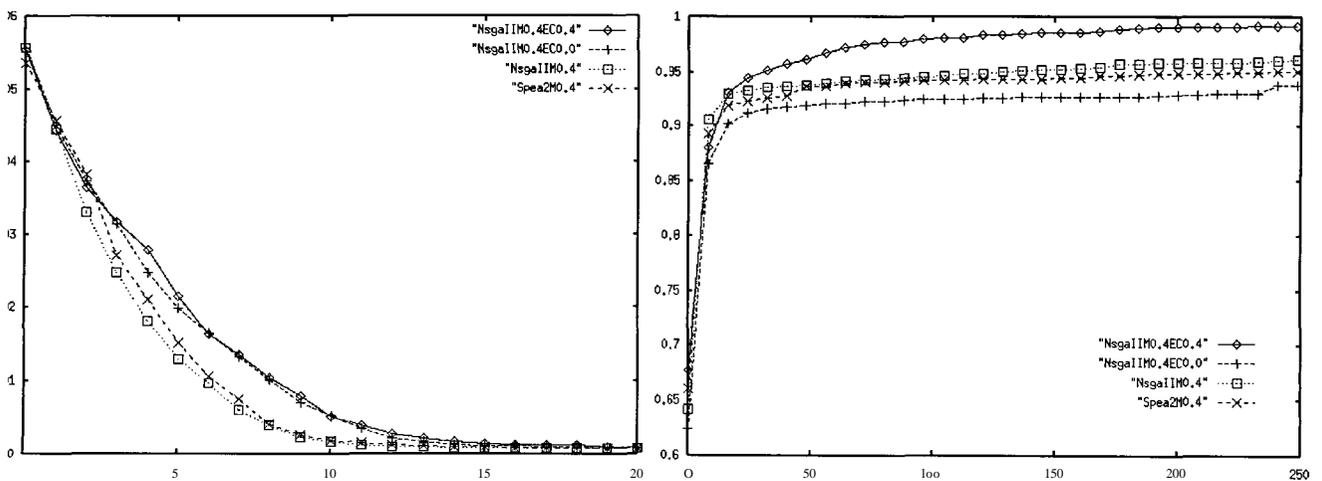


Figura 5.13. Promedios de las métricas 1 y 2, comparando los cuatro algoritmos con mutación de 0.4%.

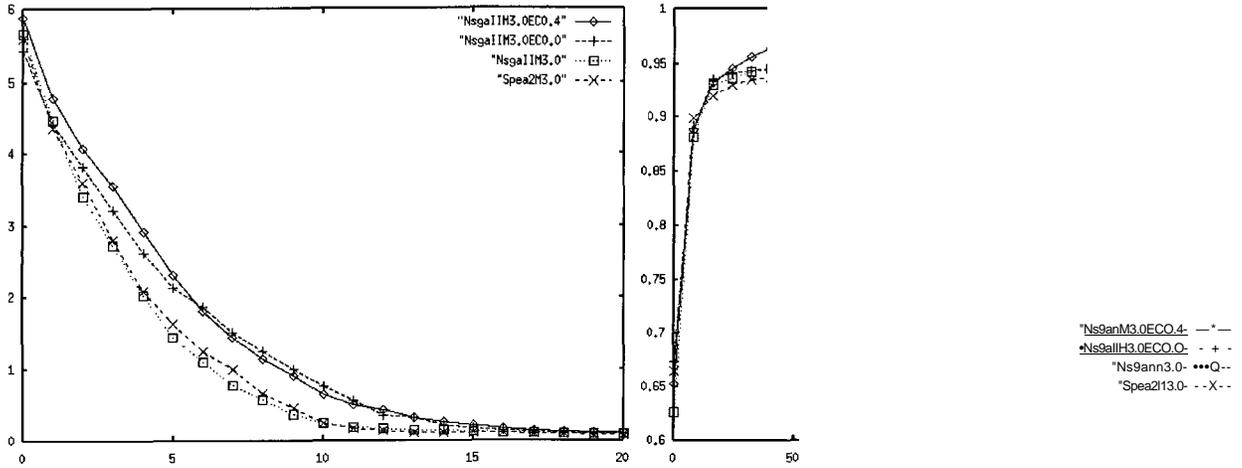


Figura 5.14. Promedios de las métricas 1 y 2, comparando los cuatro algoritmos con probabilidad de mutación del 3.0%.

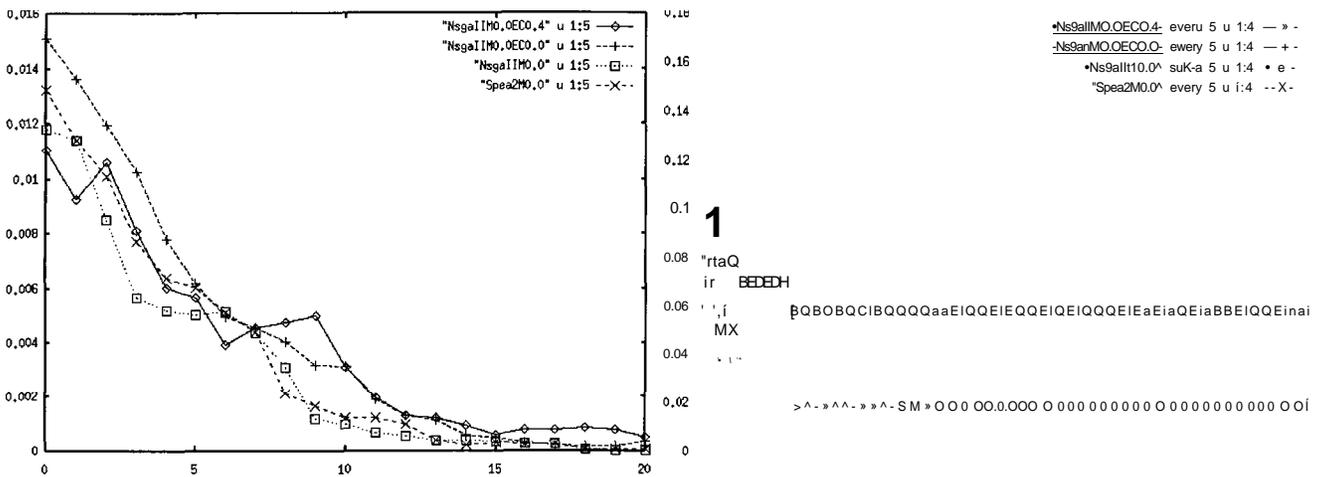


Figura 5.15. Desviaciones Típicas de las métricas 1 y 2, comparando los cuatro algoritmos sin mutación

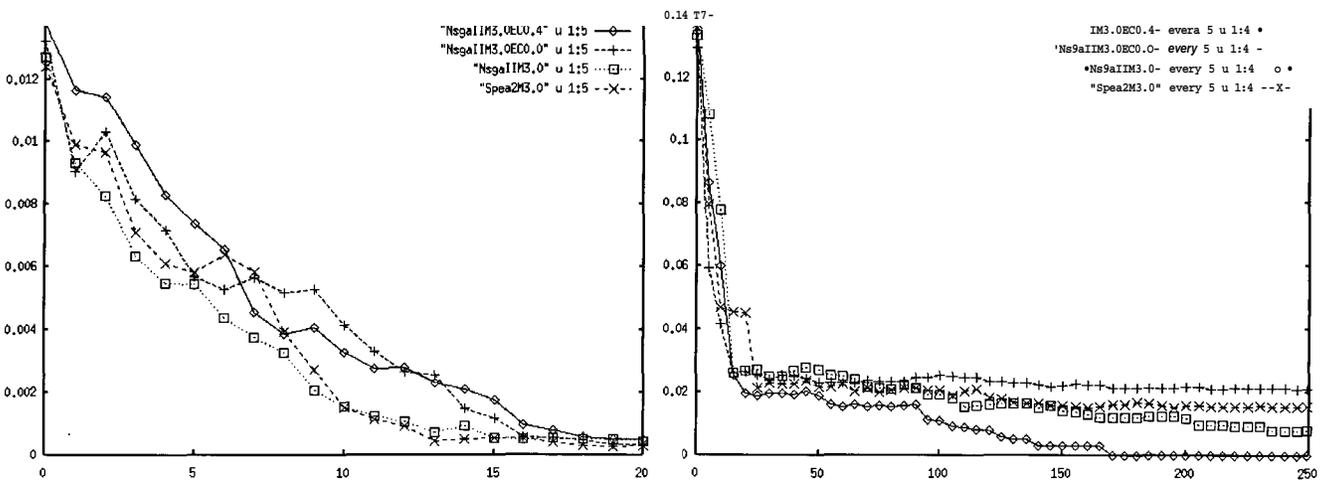


Figura 5.16. Desviaciones Típicas de las métricas 1 y 2, comparando los cuatro algoritmos con probabilidad de mutación del 3%.

5.2.6 Conclusiones

Una nueva, exitosa y eficiente metodología se ha introducido para la optimización multiobjetivo del diseño de sistemas de seguridad. Esta metodología implica: el puentado de diagramas de decisión binarios empleando el 'método de los pesos' [5.17][5.26]; una codificación binaria que integra restricciones de diseño evitando muchas soluciones inviables; y un algoritmo multiobjetivo eficiente.

Un estudio comparativo de algunos de los más recientes algoritmos evolutivos multicriterio se ha realizado, mostrando que el NSGAI con elitismo controlado y diversidad lateral ha obtenido los mejores resultados promedios conjuntos en el caso test de sistema de seguridad de una central nuclear de generación eléctrica.

Un análisis del equilibrio entre la exploración-explotación en problemas de diseño óptimo multicriterio de sistemas de seguridad ha sido discutido, considerando tanto los operadores de mutación como elitismo controlado en el NSGAI. Los resultados obtenidos muestran que el operador de 'elitismo controlado' proporciona ventajas en la convergencia del algoritmo, siendo capaz de cooperar con el operador de mutación con beneficios en el comportamiento de convergencia. Una conclusión similar a la obtenida en el capítulo anterior de esta tesis aplicada a la optimización multicriterio de estructuras de nudos rígidos.

Esta ventaja del elitismo controlado mostrada a través de los resultados descritos, coincide en característica del espacio funcional con el problema multicriterio estructural: una de las funciones objetivo condiciona un espacio funcional discreto. Ello provoca en ambos casos la acumulación de soluciones en el frente no-dominado de Pareto, y por tanto los operadores favorecedores de la diversidad poblacional redundan en un aumento de la eficacia del algoritmo. Todo ello induce a pensar que en esta aplicación, también el algoritmo DENSEA o DENSEA-NSH puede proporcionar una mejora de los resultados obtenidos. Como tarea futura se pretende la aplicación de este algoritmo al problema de diseño de sistemas de seguridad con el fin de corroborar experimentalmente esta ventaja teórica.

5.3 Problema de Optimización Multiobjetivo en Despacho de Cargas en Sistemas de Generación de Energía Eléctrica

5.3.1 Despacho de cargas Multiobjetivo

En la industria de producción de energía eléctrica es vital conseguir un funcionamiento eficiente de las Centrales de Producción, cubriendo en todo momento las necesidades de demanda de energía de la forma más económica posible. Sin embargo, y debido a la importancia cada vez mayor de reducir el impacto medio ambiental en todos los ámbitos, se hace necesario incorporar a su vez la minimización de las emisiones a la atmósfera de agentes químicos derivados de los procesos de combustión en las unidades de generación que queman combustibles fósiles (en ese propósito se enmarca el protocolo de Kyoto). El problema será resuelto así desde un punto de vista multiobjetivo, que contemple tanto la minimización total de los costes económicos asociados a la generación de potencia, como la minimización de la contaminación medio ambiental. Así, podremos tener diferentes soluciones del problema si nuestro objetivo es la obtención de un mínimo económico, una mínima emisión de NO_x, una mínima emisión de SO₂ o una mínima emisión de Partículas o cualquier combinación de ellas.

El problema que se resuelve es pues, un problema de programación óptima multicriterio, de características altamente combinatorio, con múltiples y variadas restricciones tanto de tipo desigualdad como de igualdad.

En [5.19][5.22][5.54] se describe y resuelve el problema de despacho de cargas monoobjetivo, que determina qué unidades generadoras deben ponerse en servicio, en qué orden y con qué carga a aportar al sistema, desde el punto de vista de lograr el funcionamiento más económico para un periodo de tiempo determinado. Es decir, con la finalidad de minimizar el coste total de consumo de combustible para un determinado sistema eléctrico y durante un período especificado. En este trabajo resolvemos el problema multiobjetivo con dos objetivos en conflicto: la minimización del coste anterior económico y simultáneamente, la minimización de emisiones de determinados contaminantes a la atmósfera debido a los procesos de combustión en las unidades generadoras de potencia.

Las *restricciones* consideradas en el problema multiobjetivo son las mismas que las consideradas en el problema monoobjetivo [5.54], pero además extendidas con las correspondientes de la emisión de contaminantes a la atmósfera.

5.3.2 Esquema de la codificación y funciones objetivo

En el planteamiento de resolución del problema con algoritmos evolutivos, se ha considerado una codificación real. Un esquema general del individuo o cromosoma considerado aparece en la figura 5.17. Como se puede comprobar cada gen del individuo es un número real comprendido entre cero y la potencia máxima disponible para cada una de las N unidades de generación de potencia consideradas, y durante un periodo de tiempo que abarca M horas o unidades de tiempo.

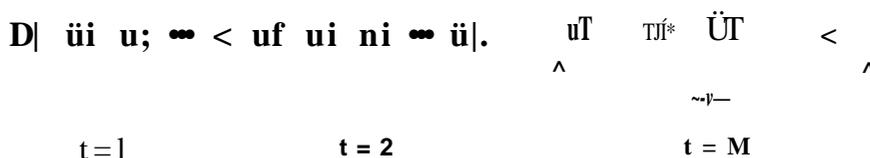


Figura 5.17. Individuo considerado para la aplicación de algoritmos evolutivos (codificación real)

5.3.3 Función Objetivo: Coste Económico

La función objetivo *fcoste* es la suma de los costes asociados al consumo de combustible de todas las unidades generadoras para todos las horas del intervalo de estudio. Es decir, se acumulan:

Los costes de generación de las unidades en cada hora del intervalo que son función de la carga a la que se encuentren dichas unidades.

El coste de arranque de cada una de las unidades cuando pase de un estado 'off' (parada) en la hora $t-1$ a un estado 'on' (en marcha) en la hora t .

El coste de parada de cada una de las unidades cuando pase de un estado 'on' (en marcha) en la hora $t-1$ a un estado 'off' (parada) en la hora t .

Evidentemente, para una hora determinada, sólo se puede dar uno de estos dos últimos sumandos o ninguno de ellos, es decir, o la unidad ha acoplado, o ha desacoplado, o está funcionando. La expresión de esta función es la dada por la ecuación 5.4:

$$f_{\text{coste}} = \sum_{t=1}^M \sum_{i=1}^N \left[\frac{G'_i(p_i)}{\Delta t} + S_i(x'_i) + D_i U_i \right] A \quad (5.4)$$

Donde:

$G'_i(p_i)$ es el coste de generación de la unidad i en la hora t , que es función de la potencia generada por la unidad i en la hora t , p_i .

$S_i(x'_i)$ es el coste de arranque de la unidad i , que es función del tiempo que ha estado la unidad parada hasta esa hora, x'_i .

D_i es el coste de parada de la unidad i , que es constante para cada unidad.

U_i es un factor cuyo valor es 0 ó 1 en función de que la unidad i esté parada o no, respectivamente.

5.3.4 Incorporación de Funciones Objetivo Medioambientales

Los contaminantes típicos que se suelen considerar en estudios de impacto medio ambiental o en procesos de vigilancia y control de la contaminación atmosférica en el entorno de Centrales Eléctricas

son: NO_x, SO₂ y Partículas. Así, tenemos como *funciones objetivo* adicionales, f_i , suma de las emisiones atmosféricas del contaminante i , de todas las unidades generadoras y en todas las horas del intervalo de estudio:

$$f_i = \sum_{i=1}^M \sum_{i=1}^N Z_i \quad (5.5)$$

Las *funciones de aptitud* resultan de adicionarle a las funciones objetivo una función de penalización sobre las restricciones. Dichas funciones de penalización tienen como fin empeorar el resultado obtenido por las funciones objetivo con el objeto de que en el proceso de selección tengan menos oportunidades de supervivencia los individuos que se alejen de cumplir las restricciones establecidas. Se han considerado como funciones de penalización la expresión de 'frecuencia y duración':

$$FF_j = Y_j \left[\frac{A_j}{N_{Vj}} + \frac{B_j}{S_{Vj}} \right] \quad (5.6)$$

$(N_{Vj})_j$ es el número de violaciones de la restricción j , $(S_{Vj})_j$ es la suma de las contribuciones de las violaciones de la restricción j

A_j y B_j son factores de homogeneización de unidades, i_{\max} es el número de restricciones consideradas para cada función objetivo j .

Así, para la función objetivo f_{coste} se consideran todas las restricciones del problema monoobjetivo [5.54], y para las restantes funciones objetivo f sólo se considera la restricción referente al límite máximo permitido para el contaminante i por unidad generadora de potencia.

El problema consiste, pues, en hallar el individuo caracterizado por el estado de las N unidades para el intervalo de M horas considerado, tal que minimice las distintas funciones objetivo consideradas y verifique todas las restricciones.

5.3.5 Algoritmo Evolutivo

Se han introducido conceptos incorporados en los denominados Agentes de Evolución Flexible [5.51][5.53] e integrados en el algoritmo multiobjetivo anterior descrito. Diversas experiencias previas, nos condujeron a un algoritmo donde la mutación convencional fue reemplazada por el motor de muestreo, siendo los muestreadores empleados (un total de cuarenta y uno), similares a los descritos en [5.52] y empleando además un cruce antitético estándar con probabilidad de uso 1.0. Se incorporan a través de una mutación de un 3% de probabilidad, los conceptos de Estructura Dinámica de Operadores (EDO), Código Genético Extendido (CGE) y Mecanismos de Control Central (MCC). El CGE permite almacenar en cada cromosoma el muestreador empleado en la mutación de cada variable real. Mediante MCC, en este caso consistentes en tomas de decisiones puramente probabilísticas, se determinan qué muestreadores se emplean para efectuar la mutación en cada individuo y sobre qué variables de la población. Todo ello proporciona la estructura dinámica de operadores EDO, que dota de un mecanismo de cooperación y competición entre operadores genéticos, a lo largo de todo el proceso evolutivo, en este caso concreto, referido al ámbito de la mutación.

5.3.6 Aplicación test

Hemos considerado como aplicación test el sistema descrito en la red temática europea INGENET de optimización con algoritmos evolutivos en aplicaciones de interés industrial (referencia e información en www.inRenet.ulpgc.es/functional/databases/ceani/index.html, caso test T54.5), basado en caso test descrito en [5.11]. Así, resolvemos la programación óptima de 10 unidades para un horizonte temporal de 24 horas. Las expresiones que describen los diferentes costes considerados y la emisión de contaminantes a la atmósfera aparecen reflejados en la Tabla 5.3. Los datos del problema están incorporadas en las Tablas 5.4 a 5.9. Como factor de seguridad, el caso test considera una reserva rodante, para cada hora, de un 15% de la demanda total. Los factores de homogeneización de las unidades que intervienen en las distintas funciones de penalización pueden verse en la Tabla 5.10.

Tabla 5.3. Expresiones que describen los diferentes costes considerados y la emisión de contaminantes a la atmósfera

Coste de generación	$G_i(P_i) = A_i P_i^2 + B_i P_i + C_i$
Coste de arranque	$S_i(x_i) = C_i (1 - \alpha_i)^{x_i} + F_i$
Coste de parada	$D_i = Cte$
Coste de contaminación	$Pollut_i = A_i \{p_i/J_i + B_i P_i + C_i\}$
Coste de emisión de contaminantes	$(Pollut_i)^2 = A_i P_i^2 + B_i P_i + C_i$
Coste de penalización	$\{p_i/J_i + B_i P_i + C_i\} + D_i$

Tabla 5.4. Datos del problema de despacho de carga

Unidad	P _{min} (MW)	P _{max} (MW)	M _{if} (h)	M _p (h)	E _i (i)
1	24	60	3	1	-1
2	32	80	3	1	-1
3	40	100	4	2	1
4	48	120	4	2	5
5	60	150	5	3	-7
6	112	280	5	2	3
7	208	520	8	4	-5
8	60	150	4	2	3
9	128	320	5	2	-6
10	80	200	5	2	-3

p_i es la potencia generada por la unidad i en la hora t y x_i es el número de horas que ha estado la unidad i parada hasta la hora t .

3

Para cada unidad aparecen la potencia mínima y máxima, el número mínimo de horas que debe estar funcionando y parada, y el estado inicial de la unidad (un número negativo indica el número de horas apagada y un número positivo indica el número de horas encendida)

Tabla 5.5. Demanda horaria de electricidad (en MW)

Hora	Demanda	Hora	Demanda	Hora	Demanda
1	1459	9	1271	17	1066
2	1372	10	1242	18	1037
3	1299	11	1197	19	993
4	1280	12	1182	20	978
5	1271	13	1154	21	963
6	1314	14	1138	22	1022
7	1372	15	1124	23	1081
8	1314	16	1095	24	1459

Tabla 5.6. Datos para el cálculo de los costes del despacho de cargas en unidades monetarias (u. m.)

Unidad i	C_r	F	a'	D	A	B	C
1	85	20.59	0,20	15	0.0454	9,9214	159,33
2	101	20.59	0,20	25	0.0356	10,375	22,16
3	114	22,57	0,20	40	0,0254	15,031	262,87
4	94	10,65	0,18	32	0,0212	15,031	315,44
5	113	18,64	0,18	29	0,0106	20,765	485,69
6	176	27.57	0,15	42	0,0253	18.995	1236,9
7	267	34.75	0,09	75	0.0085	19,566	4437,2
8	282	45.48	0,09	49	0,0116	22,251	300,86
9	187	38.62	0,13	70	0,0050	20,927	1044,2
10	227	26,64	0,11	62	0,0091	23,107	416,58

Tabla 5.7. Coeficientes y límites de las emisiones de NOx

Unidad i	A_i	B_i	Q_i	Límite (mg/N _{iii})
1	-2,286	235.92	-1890	2000
2	-1,793	246,71	-2636	2000
3	0,0193	1,7774	230,49	450
4	0,016	1,7774	276,59	450
5	-0,005	16,287	-321.4	750
6	-0,069	36,931	-1626	750
7	-0.245	154,16	-11546	300
8	0,1686	-20,0	1361,8	750
9	-0,002	16,414	-691,1	750
10	0,1313	-20,77	1885.6	750

Tabla 5.8. Coeficientes y límites de las emisiones de SO2

Unidad i	A_i	f_i	C	Límite {mg/Nm ³ }
1	-1.0753	20.742	461.67	400
2	-1.0753	20.742	461.67	400
3	-0.0035	0.32223	72.281	400
4	-0.0035	0.32223	72.281	400
5	-0.2316	22.548	832.69	1700
6	2.1697	-111.14	2885.5	1700
7	-0.6957	32.271	-276.39	400
8	-0.1504	13.47	1119.5	1700
9	-0.2316	22.548	932.69	1700
10	-0.1504	13.47	1119.5	1700

Tabla 5.9. Coeficientes y límites de las emisiones de partículas

Unidad j	A, \dots	$B,$	$C,$	A	Límite (mg/Nm ³)
1	0	0.8494	-24.1	179.31	50
2	0	0.8494	-24.1	179.31	50
3	-5. 10 ^{'''}	0.0921	-5.4989	126.22	50
4	-5. 10 ^{'''}	0.0291	-5.4989	126.22	50
5	0	-0.0261	2.3144	-17.201	50
6	0	0.4321	-19.522	298.97	75
7	0	-0.0859	3.6284	-26.344	50
8	0	0.1279	-8.434	189.4	75
9	0	-0.0261	2.3144	-17,201	50
10	0	0.1279	-8.434	189.4	75

Tabla 5.10. Factores de homogeneización de unidades que aparecen en las funciones de penalización

Función Objetivo	Restricción/ Objetivo	A_j	f_{ij}
J_{coste}	Demanda - Producción	100000	10000
	Tiempo mínimo de parada	100000	10000
	Tiempo mínimo de funcionamiento	100000	20000
f_i	Violación de la reserva rodante mínima	100000	1000
	Violación del límite máximo permitido para el contaminante;	1000	100

5.3.7 Resultados

Se representan resultados obtenidos mediante el método descrito en los apartados anteriores, en el caso concreto de minimización bicriterio de coste monetario y de concentración de contaminante emitidos de NOx. La solución representada se obtuvo para un total de 30000 generaciones y una población de 100 individuos, empleando cruce antitético estándar - probabilidad de 1.0 - y la mutación flexible descrita en el apartado 5.3.5 con una probabilidad del 3% uniforme para cada una de las variables. En la figura 5.18 se muestran otras soluciones encontradas del mismo problema frente a la presentada en este capítulo: en ella, las referencias 1 y 3 se corresponden con frentes finales obtenidos mediante algoritmo genético y MSPEA (consultar [5.6]). La referencia 2 se encuentra disponible en [5.18]. Se observa que el frente obtenido domina por completo a las soluciones referenciadas encontradas, además de permitir encontrar una mayor densidad del frente. Todas las soluciones representadas cumplen por completo las restricciones impuestas, tanto en términos de coste monetario, como en concentración de contaminantes.

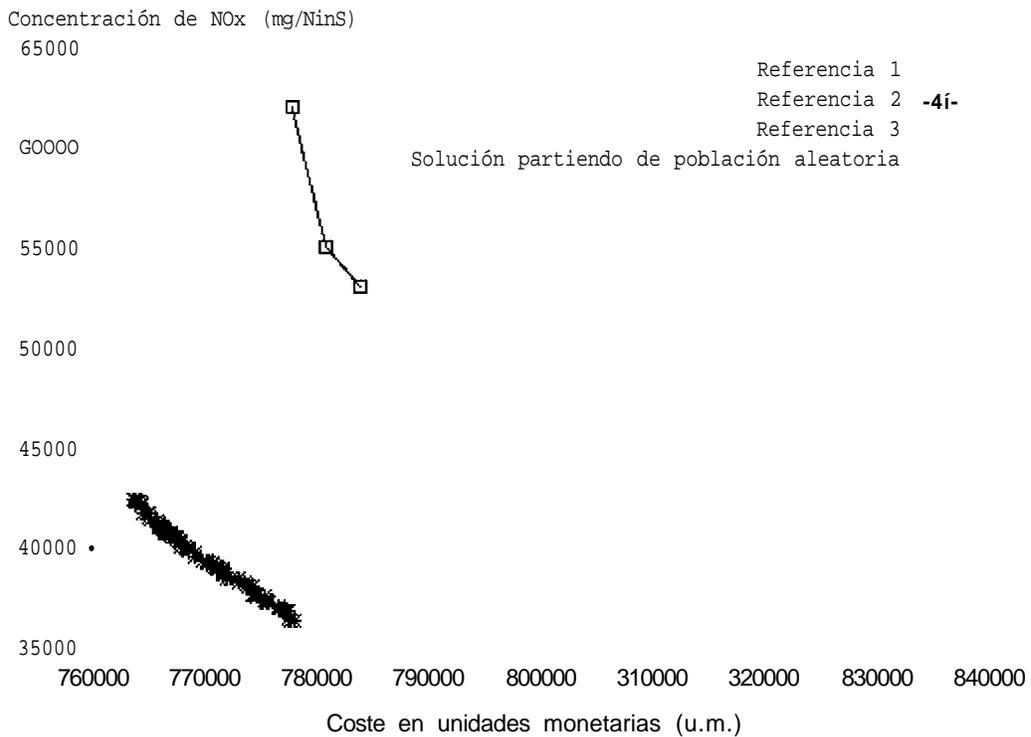


Figura 5.18. Frente Óptimo de Pareto partiendo de población inicial aleatoria frente a otros resultados referenciados.

Un método que puede permitir obtener frentes de Pareto mejorados, como se ha señalado en la última sección del capítulo anterior para aplicaciones en el cálculo estructural, correspondiente al renacimiento autoadaptativo, (consultar también [4.19] [4.25]), es la inclusión en la población inicial de una solución de alta calidad respecto a alguno de los objetivos tratados. Siguiendo este procedimiento, se incluye la mejor solución monocriterio en términos de coste (solución S), obtenida mediante el procedimiento descrito en [5.54], que resulta ser la solución mostrada en la Tabla 5.11. La solución insertada en la población inicial conjuntamente con el frente final de soluciones no-dominadas generado se representa en la figura 5.19.

Tabla 5.11. Detalle de los costes correspondientes a la Solución Monobjetivo Insertada S (Nox: 500361 mg/Nm3)

Coste de generación:	724193.53
Coste de arranques:	1228.01
Coste de paradas:	270.00
Coste total:	725691.53

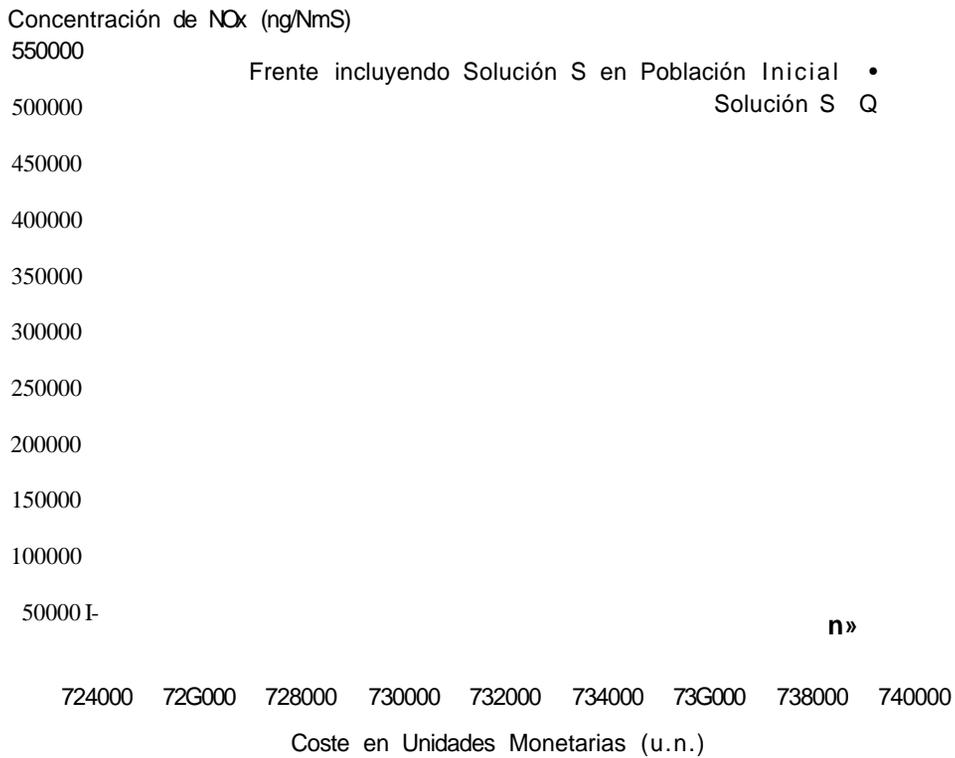


Figura 5.19. Frente Óptimo de Pareto partiendo de población inicial incluyendo a mejor solución monocriterio S en la población inicial y solución S.

La función objetivo en términos de la función de coste es excelente, no obstante, la concentración de contaminantes no satisface todas las restricciones impuestas (la concentración de NOx resulta en un valor de la segunda función objetivo de 500361 mg/Nm3). Sin embargo, el algoritmo es capaz de aprovechar la información contenida en la solución insertada y beneficiarse de ella, alcanzando un frente final de soluciones no dominadas excelente cuyo componente de coste mejora con creces el obtenido en la solución anterior (figura 5.18). Tras 40000 generaciones y una población de 850 individuos, el frente total no dominado obtenido a lo largo de toda la evolución del algoritmo es el mostrado en la figura 5.21, donde todas las soluciones mostradas (un total de 1595) satisfacen las restricciones tanto en coste como en concentración de NOx. Este nuevo frente obtenido, se representa conjuntamente con el mejor frente partiendo de una población inicial aleatoria en la figura 5.20.

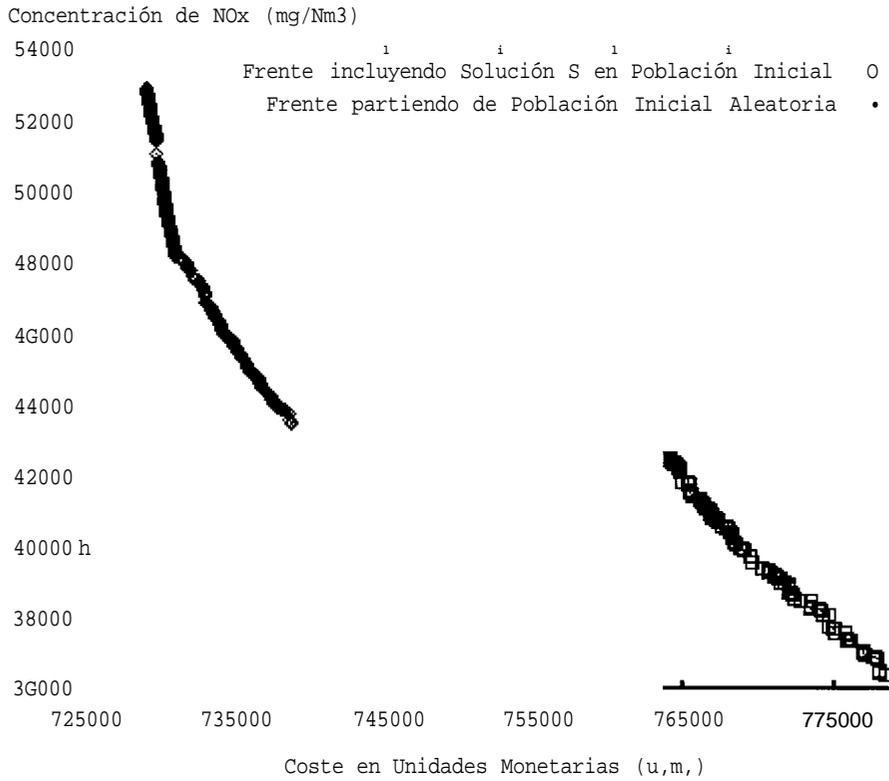


Figura 5.20. Frentes de soluciones no-dominadas partiendo de población inicial aleatoria e incluyendo solución S en la población inicial.

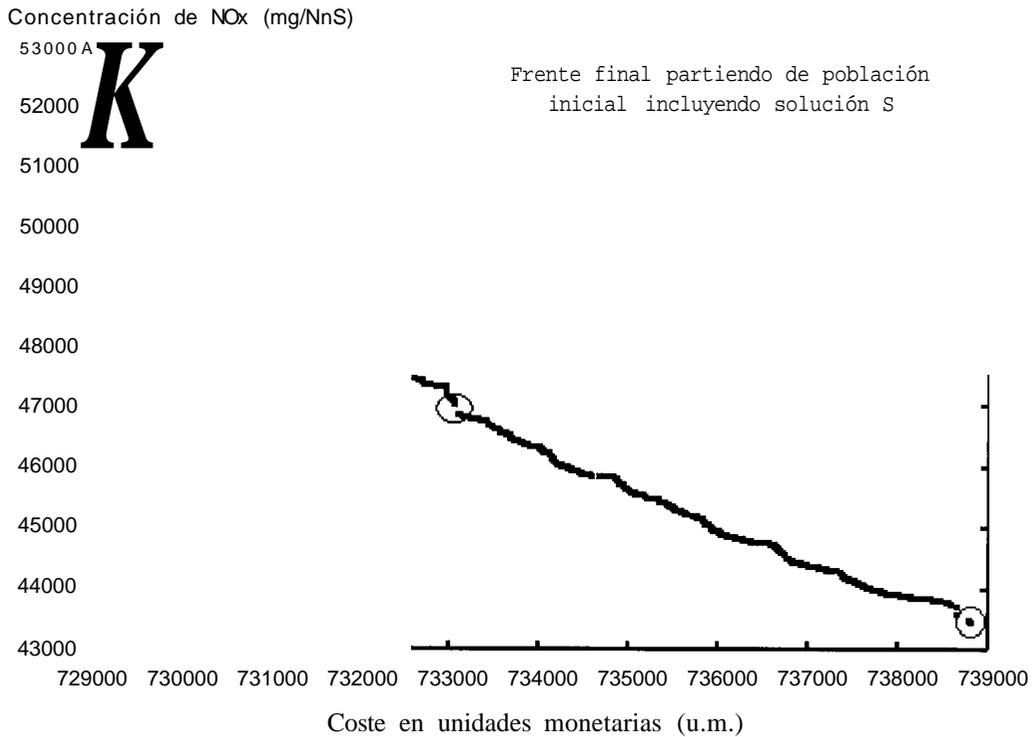


Figura 5.21. Frente Óptimo de Pareto partiendo de una solución de alta calidad en coste (solución S).

Se detallan a continuación, en las Tablas 5.12, 5.13 y 5.14, las soluciones extremas del frente, así como una solución intermedia del mismo, tal y como se describe en la figura 5.21 (los círculos destacan las soluciones seleccionadas).

Tabla 5.12. Solución Extrema 1. Coste: 729207.4 u.m. / Nox: 52895.9 mg/Nm³

	O.	U ₁	m	D ₄	v _i	U _s	U _r	U _s	U _s	U _s	U _s
i	60	80	100	120	125	117	474	81	299	0	
2	60	80	100	120	125	115	470	80	221	7	
3	60	80	100	120	122	49	469	89	258	5	
4	60	80	100	120	121	17	470	86	241	0	
5	60	80	100	120	111	50	473	79	246	28	
6	60	80	100	120	103	22	476	78	209	86	
7	60	80	100	120	144	28	472	79	229	86	
8	60	80	100	120	121	23	481	39	255	94	
9	60	80	100	120	125	29	468	13	227	89	
10	60	80	100	120	149	147	36	149	318	116	
11	60	80	100	120	147	131	24	110	319	127	
12	60	80	100	120	148	121	10	103	319	129	
13	60	80	100	120	149	118	44	99	319	108	
14	60	80	100	120	148	118	24	86	320	105	
15	60	80	100	120	146	119	46	77	316	104	
16	60	80	100	120	148	145	22	0	318	123	
17	60	80	100	120	149	135	67	0	318	102	
18	60	80	100	120	148	118	29	4	320	89	
19	60	80	100	120	140	21	3	79	319	93	
20	60	80	100	120	135	12	59	71	319	91	
21	60	80	100	120	134	10	25	70	311	86	
22	60	80	100	120	148	22	11	95	319	99	
23	60	80	100	120	150	141	29	111	319	1	
24	60	80	100	120	137	115	478	89	277	8	

Tabla 5.13. Detalle de solución Intermedia. Coste: 733203.5 u.m. / NOx: 46843.3 mg/Nm³

	\bar{u}_1	\bar{u}_2	u_1	t_1	u_3	u_6	≤ 7	$m, i: i: Mm$		
1	60	80	100	120	123	113	520	92	249	35
2	60	80	100	120	95	115	520	89	192	36
3	60	80	100	120	114	26	520	95	209	2
4	60	80	100	120	118	25	520	94	187	15
5	60	80	100	120	111	8	520	94	185	37
6	60	3	100	120	88	2	520	94	217	114
7	60	4	100	120	131	44	520	88	232	119
8	60	80	100	120	78	32	520	13	224	130
9	60	80	100	120	106	59	520	10	175	109
10	60	80	100	120	150	162	18	129	319	121
11	60	80	100	120	144	133	7	106	314	137
12	60	80	100	120	150	124	0	97	319	130
13	60	80	100	120	149	118	19	98	312	115
14	60	80	100	120	122	113	104	105	314	122
15	60	80	100	120	124	115	12	101	306	116
16	60	80	100	120	145	139	24	0	319	130
17	60	80	100	120	140	130	11	0	320	114
18	60	80	100	120	144	112	86	0	318	100
19	60	80	100	120	144	20	11	77	314	96
20	60	80	100	120	93	6	57	84	316	122
21	60	80	100	120	109	0	30	76	310	106
22	60	80	100	120	148	48	32	76	319	117
23	60	80	100	120	150	148	11	104	317	37
24	60	80	100	120	105	120	520	95	257	0

Tabla 5.14. Detalle de Solución Extrema 2. Coste: 738811.5 u.m. / NOx: 43470.7 mg/Nm³

	U ₁	U ₂	U ₃	U ₄	U ₅	U ₆	U ₇	U ₈	U ₉	U ₁₀
1	0	80	100	120	107	119	520	97	314	0
2	0	80	100	120	87	119	520	94	251	0
3	60	80	100	120	99	82	520	92	226	0
4	60	80	100	120	95	47	520	97	207	7
5	60	80	100	120	91	0	520	102	196	5
6	60	0	100	120	88	3	520	96	213	115
7	60	0	100	120	149	54	520	93	206	122
8	60	80	100	120	115	19	520	0	204	114
9	60	80	100	120	109	40	520	0	166	115
10	60	80	100	120	147	147	0	134	320	132
11	60	80	100	120	150	117	33	105	320	144
12	60	22	100	120	150	181	60	107	319	144
13	60	0	100	120	148	143	74	116	312	154
14	60	0	100	120	149	117	6	117	317	156
15	60	0	100	120	147	119	60	108	320	148
16	60	12	100	120	147	134	39	82	319	132
17	60	80	100	120	147	118	2	11	318	121
18	60	80	100	120	140	113	2	33	316	107
19	60	80	100	120	117	30	3	92	305	117
20	60	80	100	120	132	0	48	90	262	132
21	60	80	100	120	108	0	20	95	277	121
22	60	80	100	120	147	0	90	89	316	108
23	60	80	100	120	150	137	38	115	318	44
24	60	5	100	120	147	112	520	82	316	0

5.3.8 Conclusiones

Se presenta en este capítulo una resolución multiobjetivo vía computación evolutiva, que supera en robustez, eficiencia computacional y en calidad de soluciones, las técnicas multicriterio utilizadas en una aplicación test de la base de datos de la red europea INGenet.

Es de destacar la capacidad que incorpora la metodología descrita, de proporcionar una herramienta real que permite gestionar óptimamente el despacho de cargas en sistemas de generación de potencia, disponiendo para la toma final de decisiones de una frontera de soluciones no-dominadas de alta calidad considerando tanto la minimización del coste económico como las emisiones de contaminantes atmosféricos.

Se ha empleado una técnica demostrada como eficaz en el caso del problema multiobjetivo estructural, como es la inserción de soluciones de alta calidad aun para sólo una de las funciones objetivo consideradas en la población inicial como un medio de obtener fi-entes no-dominados finales mejores que partiendo de una población aleatoria.

5.4 Conclusiones

Se han presentado dos problemas de distintos ámbitos de la ingeniería: diseño de sistemas de seguridad y asignación de unidades de potencia en producción de energía eléctrica; ambos pertenecientes a líneas de investigación del Instituto de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería de la Universidad de Las Palmas de Gran Canaria. Mostrando la versatilidad de aplicación de los algoritmos evolutivos multicriterio, ambos han sido resueltos optimizando en cada uno las dos funciones objetivo en conflicto requeridas en el diseño (en el primer caso, coste y no-disponibilidad; en el segundo caso: coste y emisión de contaminantes atmosféricos). Además, las soluciones encontradas han sido obtenidas considerando algunas de las aportaciones satisfactorias presentadas en el problema multicriterio estructural, mejorando con ello resultados obtenidos con anterioridad en dichos problemas. Por consiguiente, se han extendido algunas de las aportaciones a otro tipo de problemas, mostrando su generalizabilidad y capacidad de mejora de resultados en otros ámbitos.

Así, en el caso del diseño de sistemas de seguridad de una central nuclear de generación eléctrica, se ha mostrado que el NSGAI con elitismo controlado y diversidad lateral ha obtenido los mejores resultados promedios conjuntos en el caso test considerado. El operador de 'elitismo controlado' proporciona ventajas en la convergencia del algoritmo, siendo capaz de cooperar con el operador de mutación con beneficios en el comportamiento de convergencia. Esta conclusión es similar a la obtenida en la sección de esta tesis aplicada a la optimización multicriterio de estructuras de nudos rígidos donde se comparaban los diferentes algoritmos evolutivos multicriterio. Ambos problemas comparten su caracterización del espacio funcional: una de las funciones objetivo forma un espacio funcional discreto. Ello provoca en ambos casos la acumulación de soluciones en el frente no-dominado de Pareto, y por tanto los operadores favorecedores de la diversidad poblacional redundan en un aumento de la eficacia del algoritmo. Por ello, se sugiere como línea futura la aplicación del algoritmo DENSEA a este problema.

Por otra parte, en el problema multicriterio eléctrico, se ha empleado la inclusión de soluciones de alta calidad en la población inicial como un medio de obtener frentes no-dominados finales mejores que partiendo de una población aleatoria. Si bien en el caso estructural el modo de obtener las soluciones era mediante el renacimiento autoadaptativo, en este caso se ha aplicado una técnica de doble lazo (dos algoritmos evolutivos anidados).

Se ha mostrado cómo algunas de las técnicas empleadas en el problema estructural multicriterio han permitido resolver más eficazmente otros problemas de optimización en ingeniería dentro de las líneas de investigación del instituto.

Bibliografía

- [5.1] S. Alonso, B. Galván, B. González, I. Jiménez y G. Winter. "Algoritmos Evolutivos Flexibles: Cooperación y competición entre operadores genéticos con codificación real". *Actas del Segundo Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados. (MAEB-2003)* Gijón. España.
- [5.2] J.D. Andrews, Optimal safety system design using fault tree analysis. In *Proc. Instn. Mech. Engrs.* Vol. 208, pp. 123-131. 1994.

- [5.3] J.D. Andrews, R.L. Patisson. Optimal Safety-system performance. In *Proceedings of the 1997 Reliability and Maintainability Symposium*, Philadelphia, Pennsylvania, January 1997, pp. 76-83.
- [5.4] L.M. Bartlett, J.D. Andrews. An ordering heuristic to develop the binary decision diagram based on structural importance. *Reliability Engineering & System Safety*, Vol. 72, issue 1, 2001, pp. 31-38.
- [5.5] M. Bouissou. An ordering Heuristic for building Binary Decision Diagrams from Fault Trees. *Proceedings RAMS 96*, Las Vegas, Jan. 96.
- [5.6] J. Bravo de Laguna, B. Galván, J. Medina. "Módulo para la Asignación Óptima Multicriterio de Grupos de Generación Térmica con Integración de Energía Eólica para un Sistema Aislado". PFC Escuela Técnica Superior de Ingenieros Industriales. Universidad de Las Palmas de Gran Canaria. Enero 2002.
- [5.7] R.E. Bryant. "Symbolic boolean manipulation with ordered binary-decision diagrams", *ACM Computing Surveys* 24(3): 1992, pp. 293-318.
- [5.8] V. Chankong, Y. Haimes, "Multiobjective Decision Making Theory and Methodology", North-Holland, New York, 1983.
- [5.9] C.A. Coello Coello, "A Short Tutorial on Evolutionary Multiobjective Optimization", pp. 21-40, *in Evolutionary Multi-Criterion Optimization*, Springer, 2001.
- [5.10] O. Coudert, J.C. Madre. "Fault Tree analysis: IO^* Prime Implicants and beyond", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 240-245, 1993.
- [5.11] D. Dasgupta. "Optimal Scheduling of thermal Power Generation using Evolutionary Algorithms in Engineering Applications". Editors: D. Dasgupta.
- [5.12] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multiobjective Optimization: NSGA-II", Sixth International Conference on Parallel Problem Solving from Nature (PPSN-VI), Paris (2000), pp. 849-858.
- [5.13] K. Deb, T. Goel, "Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence", pp. 67-81, *in Evolutionary Multi-Criterion Optimization*, Springer, 2001.
- [5.14] K. Deb, A. Pratap, S. Agrawal, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation* 6 (2), 182-197, 2002.
- [5.15] J.B. Dugan, S.J. Bavuso, M.A. Boyd. [1992]. "Dynamic Fault-Tree models for fault-tolerant computer systems", *IEEE Transactions on Reliability* 41(3): 363-377.
- [5.16] J.B. Dugan, B. Venkataraman, R. Gulati. [1997]. "A software package for the analysis of dynamics fault tree models", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 64-70.
- [5.17] B. Galván. "Contribuciones a la Evaluación Cuantitativa de Árboles de Fallos". *PhD Thesis*, Departamento de Física, Las Palmas de Gran Canaria University (Canary Islands-Spain), 1999.
- [5.18] B. Galván, M. Cruz, G. Winter. "Multiobjective Optimization in Load Dispatch of Electric Power System by Eas". INGENET 'Open Day' Workshop. Von Karman Institute, Bélgica, Junio 2001.
- [5.19] B. Galván, B. González, G. Winter, S. Alonso. Optimización del despacho de cargas en sistemas eléctricos de potencia con un Agente de Evolución Flexible. *Actas del Segundo Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB-2003)*, Gijón, España.
- [5.20] B. Galván, D. Greiner, J. Periaux, M. Sefrioui, G. Winter. 'Parallel Evolutionary Computation

- for Solving Complex CFD Optimization Problems : a review and some nozzle applications'. In *'Parallel Computational Fluid Dynamics. Trends and Applications'*. North-Holland, (2003) pp. 573-604.
- [5.21] B. Galván, D. Marín, E. Benitez, S. Alonso, J. Juvier, "Safety System Design Optimization using Genetic Algorithms with Incomplete Information", *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, CIMNE, Barcelona 2002.
- [5.22] B. Galván, G. Winter, D. Greiner, M. Cruz, S. Cabrera. "Generator Scheduling in Power Systems by Genetic Algorithm and Expert System", In: Miettinen et al. Eds. *Evolutionary Algorithms in Engineering and Computer Science*, pp. 411-424, John Wiley & Sons. 1999.
- [5.23] M. Gen, R. Cheng. *Genetic Algorithms & Engineering Design*. Wiley Interscience, John Wiley & Sons, USA, 1997.
- [5.24] B. Giuggioli Busacca, M. Marseguerra, E. Zio, Multiobjective optimization by genetic algorithms: application to safety systems, *Reliability Engineering & System Safety*, Vol. 72, issue 1, 2001 pp. 59-74
- [5.25] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [5.26] L. González, D. García, B. Galván. "An Intrinsic Order Criterion to Evaluate Large, Complex Fault Trees" *IEEE Transactions on Reliability*, Vol 53 (3) September 2004 pp. 297-305.
- [5.27] L. González, D. García, B. Galván. "Sobre el análisis computacional de funciones Booleanas estocásticas de muchas variables". *EA.CABS-Actas del primer encuentro de Algebra computacional y Aplicaciones - Santander (Spain)*, 1995 Sep, pp. 45-55.
- [5.28] D. Greiner, B. Galván, G. Winter. 'Safety Systems Optimum Design using Multicriteria Evolutionary Algorithms'. *Lecture Notes in Computer Science 2632*, Evolutionary Multi-Criterion Optimization 2003, Springer, pp. 722-736.
- [5.29] D. Greiner, JM Emperador, G. Winter. "Multiobjective Optimization of Bar Structures by Pareto-GA". *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2000*. Barcelona, España. CIMNE.
- [5.30] D. Greiner, G. Winter, J.M. Emperador, B. Galván. A Comparative Analysis of Controlled Elitism in the NSGA-II applied to Frame Optimization. In: *Proceedings of the IUTAM Symposium on Evolutionary Methods in Mechantes*. Krakow, Poland. Kluwer Academic Publishers (2004), pp. 101-110.
- [5.31] D. Greiner, G. Winter, J.M. Emperador, B. Galván. An efficient adaptation of the truncation operator in SPEA2. In: *Actas del Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados AEB-02*. Eds: Herrera et al. Mérida, Spain, February 2002.
- [5.32] D. Greiner, G. Winter, B. Galván. Optimización Multiobjetivo en Sistemas de Seguridad: Una Comparativa entre los algoritmos NSGA-II y SPEA2. En: *Actas del IV Congreso de Fiabilidad: Confiabilidad*. Eds: Galván, Winter, Cuesta and Aguasca. Spain, 2002.
- [5.33] D.F. Haasl. "Advanced concepts on Fault Tree Analysis". The Boeing company System Safety Symposium, 1965, USA.
- [5.34] J.H. HoUand. *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press., 1975.
- [5.35] W. Kuo W, V. Rajendra Prasad, An Annotated Overview of System-Reliability Optimization.

IEEE Transactions on Reliability, Vol. 49, No. 2, pp. 176-187, June 2000.

[5.36] M. Laumanns, E. Zitzler, L. Thiele. On the Effects of Archiving, Elitism, and Density Based Selection in Evolutionary Multi-objective Optimization. pp. 181-196, in *Evolutionary Multi-Criterion Optimization*, Springer, 2001.

[5.37] W.S. Lee, D.L. Grosh, F.A. Tillman, C.H. Lie. "Fault Tree Analysis, Methods and Applications - A review". *IEEE Transactions on Reliability*, R-34(3), 1985, pp. 194-203.

[5.38] S. Martorell, A. Muñoz, y V. Serradell, Age-dependent models for evaluating risks and costs of surveillance and maintenance of components. *IEEE Transactions on Reliability*, 1996, Vol. 45, No.3, pp. 433-442.

[5.39] S. Martorell, C. Sánchez, V. Serradell, Constrained optimization of test intervals using steady-state genetic algorithm. *Reliability Engng. & System Safety*, Vol. 67, 2000, pp. 215-232.

[5.40] NUREG-75/014: WASH-1400. "Reactor Safety Study". U.S. Nuclear Regulatory Commission, 1975.

[5.41] S. Obayashi, S. Takahashi, Y. Takeguchi, "Niching and elitist models for MOGAs". Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V), Amsterdam (1998), pp. 250-259.

[5.42] R.L. Patisson, J.D. Andrews. Genetic algorithms in optimal safety system design. In *Proc. Inst. Mech. Engrs.* Vol. 213, Part E, pp. 187-197. 1999.

[5.43] R. Purshouse, P. Fleming, "Why use Elitism and Sharing in a Multiobjective Genetic Algorithm?", *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2002*, pp. 520-527, New York, Morgan Kaufmann Publishers.

[5.44] W.G. Schneeweiss. Boolean Functions with Engineering Applications and Computer Programs. Springer-Verlag, 1989.

[5.45] N. Smivas, K. Deb (1995) '*Multiobjective optimisation using non-dominated sorting in genetic algorithms*' *Evolutionary Computation* 2(3) : 221-248.

[5.46] W.E. Vesely, F.F. Goldberg, N.H. Roberts, D.F. Haals. Fault Tree Handbook. Systems and Reliability research office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington D.C., January, 1981.

[5.47] D.A. Van Veldhuizen, G. Lamont, "Multiobjective Evolutionary Algorithm Test Suites", *Proceedings of the 1999 ACM Symposium on Applied Computing*. pp. 351-357, San Antonio, Texas, ACM.

[5.48] P.F. Williams, M. Nikolskaia, A. Rauzy. "Bypassing BDD construction for reliability analysis". *Information Processing Letters*. Elsevier, 75 (2000) 85-89.

[5.49] G. Winter, M. Cruz, B. Galván, D. Greiner. "Optimal Scheduling of Power Generation by Evolutionary Algorithms". European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2000. Barcelona, España. CIMNE.

[5.50] G. Winter, M. Galán, P. Cuesta, D. Greiner, 'Genetic Algorithms: A Stochastic Improvement Technique. Tools, Skills, Pitfalls and Examples.' *Genetic Algorithms in Engineering and Computer Science*. John Wiley & Sons.(EUROGEN 1995).

[5.51] G. Winter, B. Galván, S. Alonso, B. González. "Evolving from Genetic Algorithms to Flexible Evolution Agents ". *Late Breaking Papers GECCO-2002*. pp 466-473. New York.

[5.52] G. Winter, B. Galván, S. Alonso y B. González. "Una propuesta de evolución flexible en el

diseño de algoritmos evolutivos". *Actas del Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados. (AEBO-2002)* pp. 246-252. Mérida, España.

[5.53] G. Winter, B. Galván, S. Alonso, B. González, J.I. Jiménez, D. Greiner. "A Flexible Evolutionary Agent: cooperation and competition among real-coded evolutionary operators", *Soft Computing Journal*, Springer, June 2004.

[5.54] G. Winter, B. Galván, B. González, S. Alonso. 'Economic Dispatch Optimisation in Electric Power Systems by a Flexible Evolution Agent', *Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Application to Industrial and Societal Problems*. EUROGEN 2003. Ed: G. Bugeba, J.A. Desideri, J. Periaux, M. Schoenauer and G. Winter. CIMNE, Barcelona, España.

[5.55] G. Winter, B. Galván, D. Greiner, J. Bravo de Laguna. 'Métodos Evolutivos para Optimización Global Multiobjetivo y Aplicaciones en Ingeniería'. En *Desarrollos recientes en métodos numéricos para ingeniería y ciencias aplicadas* SVMNI, pp. 17-25.

[5.56] G. Winter, D. Greiner, B. Galván and B. González. 'Economical and Environmental Electric Power Dispatch Optimization', *Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Application to Industrial and Societal Problems*. EUROGEN 2003. Ed: G. Bugeba, J.A. Desideri, J. Periaux, M. Schoenauer and G. Winter. CIMNE, Barcelona, España.

[5.57] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD Thesis*. Swiss Federal Institute of Technology (ETH), Zurich 1999.

[5.58] E. Zitzler, M. Laumanns, L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization", *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, CIMNE, Barcelona 2002.

[5.59] E. Zitzler, M. Laumanns, L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", TIK-Report 103, (May 2001) Swiss Federal Institute of Technology, Zurich.

Capítulo 6. Conclusiones y Líneas Futuras

*Para un auténtico escritor,
cada libro debería ser un nuevo comienzo
en el que él intenta algo
que está más allá de su alcance.*

ERNEST HEMINGWAY (1899-1961)

6.1 Introducción

En la presente tesis se ha abordado el problema de cálculo óptimo multiobjetivo del diseño de estructuras de barras considerando simultáneamente la minimización de la masa restringida (factor correspondiente al coste material de la estructura) y la minimización del número de distintos tipos de perfiles (factor correspondiente al coste constructivo); dicho cálculo se realiza considerando variables discretas, que son los perfiles reales disponibles en el mercado y bajo las directrices de la normativa española de construcción.

En este contexto, el problema propuesto se caracteriza por requerir un método que permita resolver un problema de optimización de forma global, evitando los numerosos óptimos locales que se presentan en el problema abordado; que permita el operar con variables discretas y realizar la optimización en un espacio funcional también discreto; y, finalmente, que sea un método multiobjetivo o multicriterio que considere simultáneamente varios objetivos a optimizar.

En la última década, principalmente, han surgido métodos evolutivos multicriterio basados en el criterio de no-dominancia de Pareto, que reúnen la capacidad de optimizar según los requisitos demandados en el párrafo anterior: Optimización global, discreta y multicriterio (obteniendo un frente de soluciones no-dominadas en una única ejecución), de los cuales se ha efectuado un análisis en el capítulo 2.2.

Habiéndose aplicado estos métodos en el problema estructural satisfactoriamente, y realizado un estudio con los algoritmos más destacados del estado del arte considerando diferentes factores de la optimización para obtener mayor eficacia en la resolución, en este trabajo además se ha diseñado un nuevo algoritmo muy competitivo respecto de los existentes (DENSEA: Zxiplicate Elimination A'non-dominated Sorting Evolutionary Algorithm; Algoritmo de ordenación no-dominada con eliminación de duplicados, y su variante el DENSEA-NSH). Se exponen a continuación las conclusiones derivadas de este estudio del problema estructural (se trata: el algoritmo DENSEA, distintos aspectos de la optimización, la multiobjetivización y el operador renacimiento autoadaptativo), además de otras conclusiones obtenidas referidas a: una adaptación del operador truncamiento en el SPEA2 para problemas bifuncionales y la resolución de otros dos problemas de aplicación industrial multicriterio de optimización (el diseño de sistemas de seguridad y el despacho de carga en generación eléctrica).

6.2 Conclusiones. Consecuencias

1. DESARROLLO DE UN NUEVO ALGORITMO: DENSEA

Habiendo efectuado un análisis exhaustivo a través de dos casos test estructurales del desempeño de los algoritmos más destacados según el estado del arte de los algoritmos evolutivos multicriterio en la resolución del diseño óptimo de estructuras de barras considerando la minimización de la masa restringida y el número de distintos tipos de perfiles (apartados 4.4 a 4.7 y 4.9), se observa que, de entre los algoritmos existentes y escogidos de entre el estado del arte, el algoritmo que obtiene mejores resultados es el NSGAI con elitismo controlado. Este algoritmo destaca por incorporar mecanismos de garantía de la diversidad de la población entre fi-entes, conjugándola positivamente con el operador mutación.

Tras este análisis se destaca la importancia de garantizar una adecuada diversidad poblacional como constituyente de una adecuada exploración del algoritmo y un desempeño mejor del mismo, y en esa orientación se ha contribuido con un nuevo algoritmo, el algoritmo DENSEA (Z)uplicate jElimination Aⁿ-dominated Sorting ¿ⁿvolutionary Jlgorithm; Algoritmo de ordenación no-dominada con eliminación de duplicados) caracterizado por considerar especialmente la diversidad poblacional mediante tres mecanismos:

La eliminación de las soluciones duplicadas

El reemplazamiento de esas soluciones duplicadas mediante soluciones de la mitad inferior de la población, fomentando así el incremento de diversidad.

La selección de reemplazamiento de la población de la siguiente generación, que permite la renovación completa de la mitad de la población en cada generación.

Este algoritmo propuesto (así como su adaptación para el caso estructural, DENSEA-NSH) proporciona de forma clara, los mejores resultados en los dos casos test ensayados y en el conjunto de circunstancias, superando a todos los demás ensayados en este trabajo. De acuerdo con estos resultados, la característica diferenciadora del algoritmo DENSEA es la presencia de ciertos operadores que fomentan el mantenimiento y creación de diversidad en la población, proporcionando una ventaja competitiva respecto al resto de algoritmos.

Ello se puede justificar en el ámbito teórico, a causa de que el problema multicriterio estructural abordado se caracteriza por que una de las funciones a minimizar es el número de distintos tipos de perfiles. Ésta es una función de valores discretos, perfectamente definidos, limitados en su cuantía y de número reducido respecto del tamaño de población empleado en los algoritmos evolutivos. Normalmente, cuando las funciones a optimizar son funciones continuas, es el mecanismo de distribución de soluciones a lo largo del frente, el encargado de proteger la diversidad de las mismas al distribuirlas homogéneamente a lo largo de toda la superficie del frente de soluciones no-dominadas. Pero en este caso, cada frente tiene un número limitado de posiciones donde ubicar soluciones (condicionadas por la función objetivo número de tipos de perfiles distintos), y el mecanismo de distribución homogénea de soluciones no es eficaz, acumulándose soluciones sin posibilidad de discriminarlas - eliminarlas como en espacios funcionales continuos.

Este mejor desempeño en este problema estructural multicriterio es especialmente relevante si hacemos extensivo el algoritmo DENSEA a otro tipo de problemas similares donde este operador sea aprovechable: el resultado observado será presuntamente generalizable en el caso de caracterizarse alguna de las funciones objetivo por constituir un espacio discreto y de pequeño número respecto del tamaño poblacional. Una adecuada gestión de la diversidad poblacional, donde la eliminación de duplicados tiene un papel fundamental, parece configurarse como un ingrediente imprescindible de lo que podría ser una tercera generación de algoritmos evolutivos multicriterio, especialmente en el caso de problemas de espacio de búsqueda discretos.

2. ANALIZANDO LOS DISTINTOS ASPECTOS CONSIDERADOS EN EL ESTUDIO:

Considerando los diferentes aspectos analizados del problema, se pueden extraer las siguientes conclusiones relativas a cada uno de los siguientes:

- **Codificación empleada**

Se ha efectuado un análisis comparativo entre la codificación binaria estándar y el código Gray estándar binario reflejado (Standard Binary Reflected Gray code), obteniéndose a partir de los resultados, que las ventajas argumentadas que en el ámbito teórico el código Gray nos podía proporcionar, permitiendo una mayor homogeneidad en la correspondencia entre el espacio genotípico y el espacio fenotípico, y basadas en otros estudios de aplicación en casos de optimización monoobjetivo, quedan corroboradas en este trabajo a través del análisis mostrado de los resultados experimentales obtenidos, en optimización multiobjetivo. De estos resultados se concluye que, el uso de la codificación Gray, permite obtener en promedio menor coste computacional, mayor precisión, obteniendo más frecuentemente frentes más completos y simultáneamente con una mayor robustez en optimización multicriterio de estructuras, acentuándose su efecto ventajoso incrementalmente con el aumento del tamaño del problema.

Se destaca igualmente cómo la importancia de homogeneidad en la correspondencia fenotípica y genotípica es clave en un problema como el tratado, donde se emplean variables discretas en lugar de una equivalencia binaria - real, puesto que cada una de las variables (perfil real con su área, momento de inercia, etc.) presenta un cambio significativo al incrementarse en tan solo una unidad.

- **Elitismo**

Si bien, tal como ha sido expuesto en la revisión del estado del arte descrita en el apartado 2.2, el operador elitismo ha sido considerado como un factor fundamental en el correcto desempeño de los algoritmos evolutivos multicriterio, se ha comparado un algoritmo evolutivo multicriterio carente de él (NSGA) frente a otro perteneciente a la última generación (NSGAI) en el problema estructural.

Los resultados confirman que, para este problema, es clara la ventaja que proporciona el elitismo en el comportamiento de los algoritmos evolutivos multicriterio. Ello se muestra en el acercamiento al frente óptimo, así como en la amplitud del frente de Pareto alcanzado, durante toda la evolución del algoritmo para ambos casos test.

- **Probabilidad de Mutación**

A partir de los resultados obtenidos en ambos casos test, se pueden extraer ciertas directrices orientativas acerca de la probabilidad de mutación en el problema estructural abordado: una probabilidad de mutación en tomo a $1/n$, siendo n el número de bits del cromosoma, parece acertada en cromosomas de tamaño pequeño, siendo necesario un aumento de la probabilidad de mutación respecto de esa orientación a medida que aumenta el tamaño del cromosoma.

- **Estrategia poblacional**

En optimización estructural monocriterio se obtuvo en [PFC][JFEA2001] que la estrategia poblacional de estado estacionario ('steady-state') era claramente ventajosa en el problema de minimización de la masa restringida. Habiéndose implementado ésta en los algoritmos evolutivos multicriterio de última generación ensayados, a partir de los resultados obtenidos (ver sección 4.8) se puede extraer, que en optimización multiobjetivo la estrategia de estado estacionario no resulta tan claramente ventajosa. Es en el caso test correspondiente a la mayor estructura ensayada, y con un tamaño de población en tomo al número de bits del cromosoma (200 individuos en ese caso), donde se aprecia un drástico mejor comportamiento en el acercamiento al frente óptimo sin perjuicio de la amplitud del mismo. La calidad de las soluciones obtenidas es mejor, y el coste en obtenerla también si las comparaciones son en términos del número de evaluaciones de la función objetivo (en términos del tiempo de cálculo existe un ligero incremento).

Para el caso de un cálculo de diseño óptimo en un ordenador personal convencional, donde no está disponible la posibilidad de paralelización (caso de un estudio de ingeniería por ejemplo), es posible en un tiempo de cálculo razonable (unos pocos minutos) mediante la estrategia de estado estacionario recomendada, obtener soluciones óptimas con mayor acercamiento al frente óptimo sin detrimento en la amplitud del frente obtenido.

- **Los algoritmos DENSEA y DENSEA-NSH comparados**

Se han comparado los dos algoritmos propuestos en esta tesis específicamente: DENSEA y DENSEA-NSH (abordados en la sección 4.10). Si bien el algoritmo DENSEA-NSH parece tener un ligero mejor comportamiento en la métrica de amplitud del frente y el DENSEA en la métrica de acercamiento al frente, los resultados comparativos no permiten establecer diferencias notables entre ambos algoritmos, siendo su comportamiento cualitativamente de similar magnitud y mejor que el del resto de algoritmos comparados en las secciones anteriores para los casos test ensayados. Se destaca que el algoritmo DENSEA-NSH es totalmente competitivo frente al DENSEA en términos de calidad de las soluciones obtenidas (tanto en acercamiento al frente óptimo como en amplitud), habiéndose implementado un operador distribuidor de la diversidad a lo largo del fi-ente con independencia de parámetros, y además simplificado su tiempo de computación sin perjuicio del rendimiento (especialmente destacable en el caso de la estrategia de estado estacionario).

Resumiendo, el algoritmo DENSEA-NSH se configura como un algoritmo con un mecanismo de diversidad independiente de parámetros y de rápido y sencillo cálculo (específico del problema estructural debido a la naturaleza discreta natural de una de las funciones objetivo), que ha demostrado ser totalmente competitivo frente a mecanismos contrastados, implementados entre otros en el DENSEA, y sin perjuicio cualitativo de las soluciones proporcionadas por el mismo.

3. MULTIOBJETIVIZACIÓN

La introducción de 'criterios ayudantes' o multiobjetivización de un problema para resolver más eficazmente una optimización monocriterio (o una multicriterio con menor número de objetivos buscados) ha sido sugerida de modo teórico sólo muy recientemente y con demostración práctica en contados casos.

Se presenta en este trabajo un innovador enfoque que considera la introducción de un criterio que discretiza el espacio de búsqueda permitiendo mantener soluciones adicionales durante la misma: el número de distintos tipos de perfiles en el caso estructural. La combinación de este criterio ayudante con un algoritmo multicriterio eficaz (el propuesto en esta tesis: DENSEA y DENSEA-NSH) resulta en una mejora de los resultados obtenidos frente a la optimización monocriterio evolutiva para el problema de la minimización de la masa restringida.

Si comparamos los resultados obtenidos mediante los algoritmos DENSEA y DENSEA-NSH, tanto los valores de los promedios como los valores de la mejor solución obtenida, superan claramente a los conseguidos por el algoritmo monocriterio (seleccionado como el mejor en desempeño en una comparativa de algoritmos evolutivos monocriterio habiéndose considerado estrategias generacionales, de estado estacionario y el algoritmo CHC [4.24] [4.25]). En el menor caso test ante todos los distintos tamaños poblacionales y todas las probabilidades de mutación ensayadas y para cada una de las treinta ejecuciones independientes, los algoritmos DENSEA y DENSEA-NSH son capaces de localizar la solución óptima en un número de evaluaciones mucho menor al ensayado. En el mayor caso test las diferencias tanto en promedio como en mejores soluciones obtenidas son también notables frente al caso monocriterio, siendo no solamente mejores los valores finales, sino que esa ventaja se mantiene a lo largo de toda la convergencia.

Además otra ventaja de los algoritmos DENSEA y DENSEA-NSH es su mayor robustez ante los cambios en la probabilidad de mutación frente al algoritmo monocriterio. La variación de las probabilidades de mutación según se alejan de los valores óptimos origina peores resultados en la optimización monocriterio, mientras que su influencia es mucho menor en la optimización multicriterio siendo el algoritmo en este último caso mucho más robusto, observándose este hecho en todos los casos test.

Podemos por tanto concluir tras analizar los resultados obtenidos que la introducción del número de distintos tipos de perfiles como 'criterio ayudante' en el problema de la minimización restringida de la masa en estructuras de barras mediante los algoritmos DENSEA y DENSEA-NSH, permite obtener soluciones más precisas que la optimización monocriterio y también es un método más robusto ante la variación de la probabilidad de mutación en la optimización evolutiva.

Se propone la posible extensión de esta técnica en otros problemas de optimización en ingeniería, donde sea posible introducir un criterio ayudante que discretice el problema y suponga un criterio en conflicto con el perseguido.

4. RENACIMIENTO AUTOADAPTATIVO

El operador renacimiento autoadaptativo ha sido propuesto en este trabajo y comparado frente al renacimiento original y otros algoritmos similares, como el microalgoritmo genético. Se han obtenido resultados claramente superiores en el problema de optimización discreta estructural de minimización de la masa restringida, pudiéndose alcanzar más rápidamente soluciones de mayor calidad.

5. ADAPTACIÓN DEL OPERADOR TRUNCAMIENTO EN EL SPEA2 PARA PROBLEMAS BICRITERIO

En el contexto de búsqueda de eficiencia en los algoritmos evolutivos multicriterio, se ha propuesto una adaptación del operador de truncamiento del SPEA2 para problemas de optimización biñmcionales. Así, se ha reducido su tiempo de cálculo desde $O(L^{\wedge}logL)$ hasta $O(2LlogL)$, siendo L el número de soluciones no dominadas en la población.

6. OTROS PROBLEMAS EN INGENIERÍA INDUSTRIAL

• **Problema de Diseño de Sistemas de Seguridad**

Una nueva, exitosa y eficiente metodología se ha introducido para la optimización multiobjetivo del diseño de sistemas de seguridad contemplando la minimización del coste y la minimización de la no-disponibilidad. Esta metodología implica: el puentado de diagramas de decisión binarios empleando el 'método de los pesos' [5.17][5.26]; una codificación binaria que integra restricciones de diseño evitando muchas soluciones inviables; y un algoritmo multiobjetivo eficiente.

De entre los algoritmos ensayados, el NSGAI con elitismo controlado ha obtenido los mejores resultados promedios conjuntos en el caso test de sistema de seguridad de una central nuclear de generación eléctrica. El operador de 'elitismo controlado' proporciona ventajas en la convergencia del algoritmo, siendo capaz de cooperar con el operador de mutación con beneficios en el comportamiento de convergencia. Esta ventaja coincide en característica del espacio fíncional con el problema multicriterio estructural: una de las fianciones objetivo condiciona un espacio fíncional discreto y por tanto los operadores favorecedores de la diversidad poblacional redundan en un aumento de la eficacia del algoritmo. Todo ello induce a pensar que en esta aplicación, también el algoritmo DENSEA o DENSEA-NSH puede proporcionar una mejora de los resultados obtenidos.

• **Problema Eléctrico**

Se presenta una optimización evolutiva multiobjetivo, que supera en robustez, eficiencia computacional y en calidad de soluciones a las técnicas multicriterio utilizadas en una aplicación test de la base de datos de la red europea INGenet. Proporciona una herramienta real que permite gestionar óptimamente el despacho de cargas en sistemas de generación de potencia, disponiendo para la toma final de decisiones de una frontera de soluciones no-dominadas de alta calidad considerando tanto la minimización del coste económico como las emisiones de contaminantes atmosféricos.

Se ha empleado una técnica eficaz en el caso del problema multiobjetivo estructural, que es la inserción en la población inicial de soluciones de alta calidad aún para sólo una de las funciones

objetivo consideradas. De este modo, se consiguen obtener frentes no-dominados finales mejores que partiendo de una población aleatoria.

6.3 Líneas Futuras de Investigación

Como futuras líneas de investigación surgidas como consecuencia de la continuidad en el trabajo descrito, pueden citarse las descritas a continuación:

Relativas al problema estructural:

- Se propone la implementación de la paralelización en la resolución del mismo; buscando una técnica eficiente que combine la paralelización con la optimización estructural multicriterio evolutiva.
- Se propone la extensión de este estudio con estructuras tridimensionales de barras, para lo que se requiere exclusivamente la ampliación del evaluador estructural al caso tridimensional.
- Se propone la incorporación en la optimización estructural de criterios de diseño que incluyan consideraciones constructivas reales dadas en el proceso de fabricación, como pueden ser: la inclusión de diferentes precios según el tipo de perfil empleado; el considerar el grado de aprovechamiento de los perfiles en la fabricación estructural según su dimensionalidad debido a la disponibilidad limitada de longitud de los perfiles en el archipiélago canario; o incluir otros factores influyentes en el coste de fabricación como el tiempo de soldadura empleado.
- Consideración de la inclusión de elementos de incertidumbre en mecánica estructural, desde la perspectiva de la fiabilidad estructural y de la mecánica computacional estructural estocástica.

Relativas a la optimización multicriterio evolutiva:

Con la intención de mejorar el rendimiento (bien reduciendo el tiempo de cálculo necesario o bien obteniendo soluciones de mayor calidad) se hacen las siguientes sugerencias de aplicación en otros problemas de optimización multicriterio evolutivos:

- Se propone la extensión del uso de los algoritmos propuestos: DENSE A y DENSEA-NSH a otros problemas de optimización, especialmente de dominio discreto. Se sugiere para el caso del diseño de sistemas de seguridad contemplado en este trabajo, donde parece clara la ventaja que estos algoritmos pueden proporcionar.
- Se propone la extensión del uso del operador renacimiento autoadaptativo a otros problemas de optimización.
- Se propone la extensión de la multiobjetivización mediante un criterio ajoidante basado en la discretización del problema a otros problemas de optimización.
- Se propone la extensión del uso de la técnica de introducción de soluciones de alta calidad directamente en la población inicial en otros problemas de optimización.

Bibliografía

1. [3.1] R. Abascal García. '*Cálculo Matricial de Estructuras*'. Apuntes E.T.S.I.I.
2. [4.1] H. Abbass, K. Deb, "Searching under Multi-evolutionary Pressures". Evolutionary Multi-Criterion Optimization 2003, *Lecture Notes in Computer Science*, pp. 391-404.
3. [3.2] H. Adeli, N.T. Cheng, "Augmented Lagrangian Genetic Algorithms for Structural Optimization", *Journal of Aerospace Engineering*, 7-1 (1994) pp 104-118.
4. [3.3] H. Adeli, N.T. Cheng, "Integrated genetic algorithm for optimization of space structures", *Journal of Aerospace Engineering*, 6-4 (1993) pp 315-328.
5. [3.4] H. Adeli, S. Kumar, "Distributed genetic algorithm for structural optimization", *Journal of Aerospace Engineering*, 8-3 (1995) 156-163.
6. [4.2] H. Aguirre, K. Tanaka, "Selection, Drift, Recombination, and Mutation in Multiobjective Evolutionary Algorithms on Scalable MNK-Landscapes", Third International Conference on Evolutionary Multi-Criterion Optimization 2005, *Lecture Notes in Computer Science* n° 3410, pp. 355-369.
7. [3.5] E. Alarcón Álvarez, R. Álvarez Cabal, M.S. Gómez Lera. '*Cálculo Matricial de Estructuras*' Ed. Reverte (1986).
8. [3.6] N. Ali, K. Behdinin, Z. Fawaz, "Applicability and viability of a GA based finite element analysis architecture for structural design optimization", *Computers & Structures* 81 (2003) 2259-2271.
9. [3.7] P. Almeida, G. Winter, G. Montero, R. Montenegro. 'Contribución numérica en la resolución directa de sistemas de ecuaciones asociados al método de elementos finitos con utilización de teoría de grafos' I Congreso de Métodos Numéricos en Ingeniería SEMNI 1990.
10. [5.1] S. Alonso, B. Galván, B. González, I. Jiménez y G. Winter. "Algoritmos Evolutivos Flexibles: Cooperación y competición entre operadores genéticos con codificación real". *Actas del Segundo Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados. (MAEB-2003)* Gijón. España.
11. [5.2] J.D. Andrews, Optimal safety system design using fault tree analysis. In Proc. *Instn. Mech. Engrs.* Vol. 208, pp. 123-131. 1994.
12. [5.3] J.D. Andrews, R.L. Patisson. Optimal Safety-system performance. In *Proceedings of the 1997 Reliability and Maintainability Symposium*, Philadelphia, Pennsylvania, January 1997, pp. 76-83.
13. [3.8] R. Arguelles Álvarez, R. Arguelles Bustillo. '*Análisis de Estructuras: Teoría, problemas y programas*'. E.T.S.I. Montes Fundación Conde del Valle de Salazar (1996) Madrid.
14. [2.1] T.W. Athan, P.Y. Papalambros, "A note on weighted criteria methods for compromise Solutions in multi-objective optimization", *Engineering Optimization*, 27 (1996) pp. 155-176.
15. [2.2] T. Back. '*Lectures on Evolution Strategies : Algorithms, Theory and Applications*'. Ingenet Lecture Series (1997).
16. [3.9] H. Baier, "Über Algorithmen zur Ermittlung und Charakterisierung Pareto-optimaler Lösungen bei Entwurfsaufgaben Elastischer Tragwerke", *Z Angew. Math. Mech.* 57 (1977) pp. 318-320.

17. [4.3] R. Balling, "The Maximin Fitness Function: Multiobjective City and Regional Planning", *Evolutionary Multi-Criterion Optimization 2003, Lecture Notes in Computer Science*, Springer. pp. 1-15.
18. [3.10] R.J. Balling, J.T. Taber, M.R. Brown, K. Day, "Multiobjective Urban Planning using a Genetic Algorithm", *ASCE Journal of Urban Planning and Development* 125-2 (1999) pp. 86-99.
19. [4.4] L. Barbulescu, J.P. Watson and D. Whitley, "Dynamic Representations and Escaping Local Óptima: Improving Genetic Algorithms and Local Search", *Seventeenth American Conference on Artificial Intelligence (AAAI) 2000*. pp: 879-884
20. [5.4] L.M. Bartlett, J.D. Andrews. An ordering heuristic to develop the binary decision diagram based on structural importance. *Reliability Engineering & System Safety*, Vol. 72, issue 1, 2001, pp. 31-38.
21. [3.11] A.D. Belegundu, D.V. Murthy, R.R. Salagame, E.W. Constants, "Multiobjective Optimization of Laminated Ceramic Composites using Genetic Algorithms", *Fifth AIAA/USAF/NASA Symposium on Multidisciplinary Analysis and Optimization*, (1994) pp. 1015-1022.
22. [2.3] E. Benini, A. Toffolo, A. Lazareto, "Evolutionary Algorithms for multiobjective design optimization of combined-cycle power plants". *Evolutionary Methods for Design, Optimization and Control*. Eds: Gianakoglou, Tsahalis, Periaux, Papailiou, Fogarty. (EUROGEN 2001. Greece). CIMNEpp. 337-342
23. [3.12] J.R. Beveridge, K. Balasubramaniam, D. Whitley, "Matching horizon features using a Messy Genetic Algorithm", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 499-516.
24. [3.13] H-G. Beyer, "Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 239-267.
25. [2.4] V. Bhaskar, K. Gupta, K. Ray, "Multiobjective Optimisation of an industrial wiped film poly(ethylene terephthalate) reactor: some further insights", *J. Computers and Chemical Engineering*, Elsevier, 25 (2001), pp. 391-407.
26. [4.5] S. Bleuer, M. Brack, L. Thiele, E. Zitzler, "Multiobjective Genetic Programming: Reducing Bloat using SPEA2", *Proceedings ofCEC 2001*, pp. 536-543.
27. [3.14] S. Botello, J.L. Marroquin, E. Ofiate, J.V. Horebeek, "Solving structural optimization problems with genetic algorithms and simulated annealing", *International Journal for Numerical Methods in Engineering*, 45 (1999) 1069-1084.
28. [5.5] M. Bouissou. An ordering Heuristic for building Binary Decision Diagrams from Fault Trees. *Proceedings RAMS 96*, Las Vegas, Jan. 96.
29. [3.15] J.-P. Brans, B. Mareschal, "How to select and how to rank projects: the PROMETHEE method for MCDM", *European Journal ofOperations Research*, 24 (1986) pp. 228-238.
30. [5.6] J. Bravo de Laguna, B. Galván, J. Medina. "Módulo para la Asignación Óptima Multicriterio de Grupos de Generación Térmica con Integración de Energía Eólica para un Sistema Aislado". PFC Escuela Técnica Superior de Ingenieros Industriales. Universidad de Las Palmas de Gran Canaria. Enero 2002.
31. [5.7] R.E. Bryant. "Symbolic boolean manipulation with ordered binary-decision diagrams", *ACM Computing Surveys* 24(3): 1992, pp. 293-318.

32. [2.5] J.T. Buchanan, "A Naïve Approach for Solving MCDM Problems: The GUESS Method", *Journal of the Operational Research Society*, 48 (1997) pp. 202-206.
33. [3.16] S.A. Bums, *Recent Advances in Optimal Structural Design*, Institute of American Society of ASCE-SEI (2002).
34. [3.17] C.V. Camp, S. Pezeshk, G. Cao, "Optimized Design of Two-Dimensional Structures using a Genetic Algorithm", *ASCE Journal of Structural Engineering*, 124-5 (1998).
35. [3.18] E. Cantú-Paz, D.E. Goldberg, "Efficient parallel genetic algorithms: theory and practice", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 221-238.
36. [3.19] D.G. Carmichael, "Computation of Pareto óptima in structural design", *International Journal of Numérica! Methods in Engineering*, 15 (1980) pp. 925-952.
37. [4.6] R. Caruana, J. Schaffer, "Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms", *Proceedings of the 5th Int. Conference on Machine Learning* 1988, Morgan Kaufmann.
38. [4.7] U.K. Chakraborty, C.Z. Janikow, "An analysis of Gray versus binary encoding in genetic search", *Information Sciences: an International Journal; Special Issue: Evolutionary Computation*, 156 (2003) pp. 253-269.
39. [5.8] V. Chankong, Y. Haimes, "Multiobjective Decisión Making Theory and Methodology", North-Holland, New York, 1983.
40. [2.6] A. Chames, W.W. Cooper, R.O. Ferguson, "Optimal estimation of executive compensation by linear programming", *Ma/jage. Sci.*, 1 (1955) pp. 138-151.
41. [3.20] T-Y. Chen, C-J. Chen, "Improvements of simple genetic algorithm in structural design", *International Journal for Numerical Methods in Engineering*, 40 (1997) 1323-1334.
42. [3.21] S.Y. Chen, S.D. Rajan, "A robust genetic algorithm for structural optimization", *Structural Engineering and Mechantes*, 10-4 (2000) 313-336.
43. [3.22][4.8] S.Y. Chen, S.D. Rajan, "Improving the efficiency of genetic algorithms for frame designs", *Engineering Optimization*, 30-3,4 (1998) 281-307.
44. [3.23] T.Y. Chen, C.C. Shieh, "Fuzzy multiobjective topology optimization" , *Computers & Structures* 78 (2000) 459-466.
45. [3.24] F.Y. Cheng y D. Li, "Genetic Algorithm Development for Multi-Objective Optimization of Structures", *AIAA Journal*, 36-5 (1998) 1105-1112.
46. [3.25] F.Y. Cheng, D. Li, "Multiobjective optimization design with Pareto genetic algorithm", *Journal of Structural Engineering*, 123-9(1997) 1252-1261.
47. [3.26] F.Y. Cheng, D. Li, J. Ger, "Multiobjective optimization of Seismic Structures", *Advanced Technology in Structural Engineering*, (2000), Section 6, chapter 4.
48. [3.27] F.Y. Cheng, Z.Q. Wang, "Development of seismic structural optimization and future trends", *Structural Engineering and Mechanics* (1999) 101-111.
49. [3.28] J-H. Chou, J. Ghaboussi, "Genetic algorithm in structural damage detection" , *Computers & Structures* 79 (2001) 1335-1353.
50. [3.29] R.F. Coelho, H. Bersini, P. Bouillard, "Parametrical mechanical design with constraints and preferences: application to a purge valve", *Comput Methods Appl. Mech Engrg*, 192 (2003) 4355-4378.

51. [3.30] C.A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art", *Comput. Methods Appl. Mech Engrg*, 191 (2002) 1245-1287.
52. [2.7][4.9][5.9] C.A. Coello Coello, "A Short Tutorial on Evolutionary Multiobjective Optimization", *Lecture Notes in Computer Science*, Evolutionary Multi-criterion Optimization 2001, pp. 21-40.
53. [2.8] C.A. Coello Coello, "Multiobjective Optimization using a Micro-Genetic Algorithm" *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, San Francisco, Morgan Kaufmann.
54. [3.31] C.A. Coello Coello, "An updated survey of GA-based multiobjective optimization *XQchmqQs*", *ACM Computing Surveys*, 32-2 (2000) 109-143.
55. [2.9] C.A. Coello Coello, "A comprehensive survey of evolutionary-based multiobjective optimization." *Knowledge and Information Systems* 1-3 (1999) pp. 269-308.
56. [3.32] C.A. Coello Coello, "An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design", *PhD Thesis*, Tulane University, LA, U.S.A. (1996).
57. [3.33] C.A. Coello Coello, A.D. Christiansen, "Multiobjective optimization of trusses using genetic algorithms", *Computers & Structures* 75 (2000) 647-660.
58. [3.34] C.A. Coello Coello, E. Mezura-Montes. Handling constraints in genetic algorithms using dominance-based tournaments. In *Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002)*, vol. 5, University of Exeter, Devon, UK, Parmee IC (ed.). Springer: Berlin, April 2002; 273-284.
59. [3.35] C.A. Coello Coello, G.T. Pulido, A.H. Aguirre, "Multiobjective evolutionary algorithms for structural optimization", *Computational Fluid and Solid Mechanics* (2003) 2244-2248.
60. [4.10] C. Coello Coello, G.T. Pulido, "Multiobjective Optimization using a Micro-Genetic Algorithm". *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, San Francisco, Morgan Kaufmann.
61. [2.10] C.A. Coello Coello, D. Van Veldhuizen, G. Lamont, '*Evolutionary Algorithms for solving multi-objective problems*'. Kluwer Academic Publishers - GENA Series (2002).
62. [3.36] M.Z. Cohn, Z. Lounis, "Optimal Design of Structural Concrete Bridge Systems", *Journal of Structural Engineering*, 120-9 (1994) 2653-2674.
63. [3.37] T.H. Gormen, CE. Leiserson, R.L. Rivest. '*Introduction to Algorithms*'. The MIT Press. McGrawHill. (1990).
64. [4.11] D. Come, K. Deb, P. Fleming, J. Knowles, "The Good of the Many Outweighs the Good of the One: Evolutionary Multi-Objective Optimization", *IEEE coNNections*, 1 (1), pp. 9-13, 2003.
65. [2.11] D. Come, N. Jerram, J. Knowles, M. Oates, "PESA-11: Region-based Selection in Evolutionary Multiobjective Optimization". In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, San Francisco, Morgan Kaufmann.
66. [2.12] D. Come, J. Knowles, M. Oates, "The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization", *Sixth International Conference on Parallel Problem Solving from Nature (PPSN-VI)*, Paris (2000), pp. 839-848.
67. [5.10] O. Coudert, J.C. Madre. "Fault Tree analysis: IO[^] Prime Implicants and beyond", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 240-245, 1993.

68. [3.38] V. Coverstone-Carrou, J.W. Hartmann, W.J. Masón, "Optimal multi-objective low-thrust spacecraft trajectories", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 387-402.
69. [3.39] W.A. Crossley, A.M. Cook, D.W. Fanjoy, V.B. Venkayya, "Using Two-Brach Tournament Genetic Algorithm for Multiobjective Optimization", *AISS / ASME / ASCE / AHS / ASC Structures, Structural Dynamics & Materials Conference*, Vol 2, pp. 1752-1762 AIAA, USA (1998).
70. [3.40] E. Cuthill, J. McKee, 'Reducing the bandwidth of sparse symmetric matrices'. *Proceedings Association Computing Machinery, National Conference* New York. (1969) pp. 157 - 172.
71. [2.13] I. Das, J.E. Dennis, "Normal-boundary Intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems", *SIAM Journal of Optimization*, 8 (1998) pp. 631-657.
72. [5.11] D. Dasgupta. *"Optimal Scheduling of thermal Power Generation using Evolutionary Algorithms in Engineering Applications"*. Editors: D. Dasgupta.
73. [1.1] L. Davis, "Handbook of Genetic Algorithms", Van Nostrand Reinhold, New York, (1991).
74. [2.14][4.13] K. Deb, *Multiobjective Optimization using Evolutionary Algorithms*. John Wiley & Sons -Series in Systems and Optimization- (2001).
75. [2.15] K. Deb, "Multi-Objective Genetic Algorithms: Problem Difficulties and Constructions of Test Problems", *Evolutionary Computation* 7-3 (1999) pp. 205-230. MIT Press.
76. [3.41] K. Deb, "An efficient constraint handling method for genetic algorithms" , *Comput. Methods Appl. Mech Engrg*, 186 (2000) 311-338.
77. [2.16][4.12] K. Deb. 'Evolutionary Algorithms for Multi-Criterion Optimisation in Engineering Design'. *Evolutionary Algorithms in Engineering and Computer Science (EUROGEN 1999)* John Wiley & Sons.
78. [2.17][5.12] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multiobjective Optimization: NSGA-II", *Sixth International Conference on Parallel Problem Solving from Nature (PPSN-VI)*, Paris (2000), pp. 849-858.
79. [2.18][5.13] K. Deb, T. Goel, "Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence", pp. 67-81, in *Lecture Notes in Computer Science*, Evolutionary Multi-Criterion Optimization 2001, Springer.
80. [3.42] K. Deb, S. Gulati, "Design of truss-structures for minimum weight using genetic algorithms", *Finite Elements in Analysis and Design*, 37-5 (2001) 447-465.
81. [2.19] K. Deb, S. Jain, "Running performance metrics for evolutionary multi-objective optimization, *Technical Report 2002/2004*, Indian Institute of Technology Kanpur, Mayo 2002.
82. [2.20] K. Deb, M. Mohán, S. Mishra, "Towards a quick computation of well-spread Pareto-Optimal Solutions", *Lecture Notes in Computer Science*, Evolutionary Multi-Criterion Optimization 2003, Springer, pp. 222-236.
83. [2.21][5.14] K. Deb, A. Pratap, S. Agrawal, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation* 6 (2), 182-197, 2002.
84. [1.2] K.A. De Jong, "An Analysis of the Behaviour of a Class of Genetic Adaptive Systems", *PhD Thesis dissertation*. (1975) Univ. of Michigan, Ann Arbor, MI.

85. [3.43] A.K. Dhingra, B.H. Lee, "Multiobjective Design of actively controlled structures using a hybrid optimization method", *International Journal for Numerical Methods in Engineering*, 38-20 (1995) pp. 3383-3401.
86. [3.44] A.K. Dhingra, B.H. Lee, "A Genetic Algorithm Approach to Single and Multiobjective Structural Optimization with Discrete-Continuous Variables", *International Journal for Numerical Methods in Engineering*, 37 (1994) pp. 4059-4080.
87. [3.45] A.K. Dhingra, B.H. Lee, "Optimal placement of actuators in actively controlled structures", *Engineering Optimization*, 23-2 (1994) pp. 99-118.
88. [3.46] L. Duckstein, "Multiobjective Optimization in structural design: The model choice problem", in: *New Directions in Optimum Structural Design*, John Wiley & Sons (1984) pp. 459-481.
89. [5.15] J.B. Dugan, S.J. Bavuso, M.A. Boyd. [1992]. "Dynamic Fault-Tree models for fault-tolerant Computer systems", *IEEE Transactions on Reliability* 41(3): 363-377.
90. [5.16] J.B. Dugan, B. Venkataraman, R. Gulati. [1997]. "A software package for the analysis of dynamics fault tree models", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 64-70.
91. [3.47] F. Erbatur, O. Hasancebi, I. Tütüncü, H. Kihc, "Optimal design of planar and space structures with genetic algorithms" , *Computers & Structures* 75 (2000) 209-224.
92. [4.14] L. Eshelman, 'The CHC Adaptive Search Algorithm: How to have Safe Search when engaging in nontraditional Genetic Recombination'. In / *Foundations Of Genetic Algorithms (FOGA 1990)* pp 265-284. Ed. G. Rawlings. Morgan Kaufmann Publishers.
93. [4.15] L. Eshelman, J.D. Schaffer, Preventing premature convergence in genetic algorithms by preventing incest, in: R.K. Belew, L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, 1991, pp. 115-121.
94. [3.48] E.M.R. Fairbairn, M. Silvano, R. Toledo Filho, J. Alves, N. Ebecken, "Optimization of mass concrete construction using genetic algorithms" , *Computers & Structures* 82 (2004) 281-299.
95. [3.49] D.W. Fanjoy, W.A. Crossley, "Topology design of planar cross-sections with a genetic algorithm: Part 1 - Overcoming the obstacles", *Engineering Optimization*, 34-1 (2002) 30-49.
96. [3.50] D.W. Fanjoy, W.A. Crossley, "Topology design of planar cross-sections with a genetic algorithm: Part 2 - Bending, torsion and combined loading applications", *Engineering Optimization*, 34-1 (2002) 49-64.
97. [3.51] C.A. Felippa, "A historical outline of matrix structural analysis: a play in three acts", *Computers & Structures* 19 -14 (2001) 1313-1324.
98. [3.52] C.W Feng, L. Liu, S.A. Bums, "Using Genetic Algorithms to Solve Construction Time-Cost Trade-Off Problems", *Journal of Computing in Civil Engineering*, 10-3 (1999) pp. 184-189.
99. [3.53] X-T. Feng, C. Yang, "Coupling recognition of the structure and parameters of non-linear constitutive material models using hybrid evolutionary algorithms", *International Journal for Numerical Methods in Engineering*, 59 (2004) 1227-1250.
100. [2.22] C. Fonseca, P. Fleming. On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. In H. M. Voigt, W. Ebeling, I. Redchenberg and H. P. Schwefel (Eds.). *Parallel Problem Solving from Nature - PPSNIV(1996)*. Lecture Notes in Computer Science, pp. 584-593. Berlín, Germany: Springer-Verlag.

101. [2.23] C. Fonseca, P. Fleming. "Multiobjective Optimization and Múltiple Constraint Handling with Evolutionary Algorithms I: A Unified Formulation". In *IEEE Transactions on Systems, Man and Cybernetics*, 28 - 1 , (1998) pp. 26-37.
102. [2.24] C. Fonseca, P. Fleming, Multiobjective Optimization and Múltiple Constraint Handling with Evolutionary Algorithms II: Application Example. In *IEEE Transactions on Systems, Man and Cybernetics*, 28 - 1, (1998) pp. 38-47.
103. [2.25] C. Fonseca, P. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* 5-1 (1995), pp. 1-16.
104. [2.26] C. Fonseca, P. Fleming, An overview of evolutionary algorithms in multiobjective optimization. *Technical Report*, Department of Automatic Control and Systems Engineering, University of Sheffield, UK (1994).
105. [2.27] C. Fonseca, P. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization." In S. Forrest (Ed.). *Proceedings of the Fifth International Conference on Genetic Algorithms* (San Mateo, California, 1993), pp. 416-423. University of Illinois at Urbana-Champaign: Morgan Kaufman Publishers.
106. [3.54] D.M. Frangopol (Ed.), "Optimal performance of Civil infrastructure systems", Proceedings of International workshop on Optimal Performance of Civil Infrastructure Systems, ASCE, 1998 222 p.
107. [3.55] T. Furukawa, "Parameter Identification with weightless regularization", *International Journal for Numerical Methods in Engineering*, 52 (2001) 219-238.
108. [3.56] T. Furukawa, G. Yagawa, "Inelastic constitutive parameter Identification using an evolutionary algorithm with continuous individuáis", *International Journal for Numerical Methods in Engineering*, 40 (1997) 1071-1090.
109. [1.3][2.28][3.57][4.16] M. Galante, "Genetic Algorithms as an approach to optimize real-world trusses", *International Journal Numerical Methods Engineering*, 39 (1996) pp. 361-382.
110. [1.4][3.58] M. Galante, "Un algoritmo genético simple para la optimización de estructuras planas articuladas". *Métodos numéricos para cálculo y diseño en ingeniería*, 9-2 (1993) pp 179-200.
111. [1.5][3.59] M. Galante, M. Cerrolaza, "Un algoritmo genético para optimizar estructuras esbeltas reales", *II Congreso de Métodos Numéricos en Ingeniería*, La Corufia, (1993).
112. [5.17] B. Galván. "Contribuciones a la Evaluación Cuantitativa de Árboles de Fallos". *PhD Thesis*, Departamento de Física, Las Palmas de Gran Canaria University (Canary Islands-Spain), 1999.
113. [5.18] B. Galván, M. Cruz, G. Winter. "Multiobjective Optimization in Load Dispatch of Electric Power System by Eas". INGENET 'Open Day' Workshop. Von Karman Institute, Bélgica, Junio 2001.
114. [5.19] B. Galván, B. González, G. Winter, S. Alonso. Optimización del despacho de cargas en sistemas eléctricos de potencia con un Agente de Evolución Flexible. *Actas del Segundo Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB-2003)*, Gijón, España.
115. [5.20] B. Galván, D. Greiner, J. Periaux, M. Sefrioui, G. Winter. 'Parallel Evolutionary Computation for Solving Complex CFD Optimization Problems : a review and some nozzle

- applications'. In *'Parallel Computational Fluid Dynamics. Trends and Applications'*. North-Holland, (2003) pp. 573-604.
116. [5.21] B. Galván, D. Marín, E. Benitez, S. Alonso, J. Juvier, "Safety System Design Optimization using Genetic Algorithms with Incomplete Information", *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, CIMNE, Barcelona 2002.
117. [5.22] B. Galván, G. Winter, D. Greiner, M. Cruz, S. Cabrera. "Generator Scheduling in Power Systems by Genetic Algorithm and Expert System", In: Miettinen et al. Eds. *Evolutionary Algorithms in Engineering and Computer Science*, pp. 411-424, John Wiley & Sons. (EUROGEN 1999).
118. [5.23] M. Gen, R. Cheng. *Genetic Algorithms & Engineering Design*. Wiley Interscience, John Wiley & Sons, USA, 1997.
119. [2.29] A.M. Geoffrion, J.S. Dyer, A. Feinberg, "An Interactive approach for multicriterion optimization with an application to the operation of an academic department", *Management Science*, 19 (1972) pp. 357-368.
120. [3.60] A. George, J. W. Liu, *Computer Solution for Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
121. [3.61] J.S. Gero, S.J. Louis, "Improving Pareto Optimal Designs using Genetic Algorithms", *Microcomputers in Civil Engineering*, 10-4 (1995) pp. 241-249.
122. [3.62] M.R. Ghasemi, E. Hinton, R.D. Wood, "Optimization of trusses using genetic algorithms for discrete and continuous variables", *Engineering Computations*, \ 6-2,3 (1999) 272-301.
123. [3.63] L. Gil, A. Andreu, "Shape and cross-section optimization of a truss structure", *Computers & Structures* 79 (2001) 681-689.
124. [5.24] B. Giuggioli Busacca, M. Marseguerra, E. Zio, Multiobjective optimization by genetic algorithms: application to safety systems, *Reliability Engineering & System Safety*, Vol. 72, issue 1, 2001 pp. 59-74
125. [1.6][2.30][4.18][5.25] D.E. Goldberg, *"Genetic Algorithms in Search, optimization and Machine Learning"* Reading, Massachusetts, (1989), Addison-Wesley.
126. [2.31] D.E. Goldberg, K. Deb, 'A Comparative Analysis of Selection Schemes Used in Genetic Algorithms'. In *IFOGA* (1990) pp 69-93. Ed. G. Rawlings. Morgan Kaufmann Publishers
127. [1.7][3.64][4.17] D.E. Goldberg, M.P. Samtani, "Engineering Optimization via genetic algorithm". *Proceedings Ninth Conference on Electronic Computation*, ASCE, New York, NY, (1986), pp. 471-482.
128. [5.26] L. González, D. García, B. Galván. "An Intrinsic Order Criterion to Evaluate Large, Complex Fault Trees" *IEEE Transactions on Reliability*, Vol 53 (3) September 2004 pp. 297-305.
129. [5.27] L. González, D. García, B. Galván. "Sobre el análisis computacional de funciones Booleanas estocásticas de muchas variables". *EACA95-Actas del primer encuentro de Álgebra computacional y Aplicaciones* - Santander (Spain), 1995 Sep, pp. 45-55.
130. [3.65] R.V. Grandhi, G. Bharatram, V.B. Venkayya, "Multiobjective Optimization of large-scale structures", *Z4 Journal*, 31-7 (1993) 1329-1337.

131. [4.19] D. Greiner, J.M. Emperador, G. Winter, "Single and Multiobjective Frame Optimization by Evolutionary Algorithms and the Auto-adaptive Rebirth Operator", in *Computer Methods in Applied Mechanics and Engineering*, Elsevier, 193 (2004) 3711-3743.
132. [4.20][5.29] D. Greiner, J.M. Emperador, G. Winter, Multiobjective optimization of bar structures by Pareto GA, in: *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2000)*, Barcelona, CIMNE, 2000.
133. [5.28] D. Greiner, B. Galván, G. Winter. 'Safety Systems Optimum Design using Multicriteria Evolutionary Algorithms'. *Lecture Notes in Computer Science 2632*, Evolutionary Multi-Criterion Optimization 2003, Springer, pp. 722-736.
134. [4.21] D. Greiner, G. Winter, J.M. Emperador, B. Galván, "Gray Coding in Evolutionary Multicriteria Optimization: Application in Frame Structural Optimum Design", *Lecture Notes in Computer Science 3410*, Evolutionary Multi-Criterion Optimization 2005, Springer, pp. 576-591.
135. [4.22] D. Greiner, G. Winter, J.M. Emperador. 'Searching for an efficient method in multiobjective frame optimisation using evolutionary algorithms'. In *Computational Solid and Fluid Mechanics*, Massachusetts Institute of Technology Conference on Computational Fluid and Solid Mechanics 2003, Elsevier Science, pp. 2285-2290.
136. [3.66][5.30] D. Greiner, G. Winter, J.M. Emperador, B. Galván. 'A Comparative Analysis of Controlled Elitism in the NSGA-II applied to Frame Optimization'. In *Solid Mechanics and its applications* n° 116, IUTAM Symposium on Evolutionary Methods in Mechanics 2002, Kluwer Academic Publishers, (2004) pp. 101-110.
137. [4.23] D. Greiner, G. Winter y J.M. Emperador. 'Un estudio comparativo en optimización de estructural metálicas mediante los algoritmos NSGA-II y SPEA2 Algorithms'. En *Desarrollos recientes en métodos numéricos para ingeniería y ciencias aplicadas*, (2002), SVMNI, pp. 31-38.
138. [5.31] D. Greiner, G. Winter, J.M. Emperador, B. Galván. An efficient adaptation of the truncation operator in SPEA2. En: *Actas del Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados AEB-02*. Eds: Herrera et al. Mérida, España, Febrero 2002.
139. [2.32][4.24] D. Greiner, G. Winter, JM. Emperador. "Optimising Frame Structures by different strategies of genetic algorithms". *Finite Elements in Analysis and Design*, Elsevier 2001; 37(5):381-402.
140. [3.67][4.25] D. Greiner, G. Winter, J.M. Emperador, "Aplicación de los Algoritmos Genéticos en Problemas de Optimización de Estructuras de Pórticos Planos y Nudos Rígidos", Proyecto Fin de Carrera, ETSI Industriales, ULPGC (2000).
141. [5.32] D. Greiner, G. Winter, B. Galván. Optimización Multiobjetivo en Sistemas de Seguridad: Una Comparativa entre los algoritmos NSGA-II y SPEA2. En: *Actas del IV Congreso de Fiabilidad: Confiabilidad*. Eds: Galván, Winter, Cuesta and Aguasca. Spain, 2002.
142. [3.68] D.E. Grierson, W.H. Pak, "Optimal sizing, geometrical and topological design using a genetic algorithm", *Structural Optimization*, 6-3 (1993) 151-159.
143. [3.69] A.A. Groenwold, N. Stander, J.A. Snyman, "A regional genetic algorithm for the discrete optimal design of truss structures", *International Journal for Numerical Methods in Engineering*, 44 (1999) 749-766.

144. [3.70] P. Gurfil, N.J. Kasdin, "Niching genetic algorithms-based characterization of geocentric orbits in the 3D elliptic restricted three-body problem", *Comput. Methods Appl. Mech Engrg*, 191 (2002) 5683-5706.
145. [3.71] W. Gutkowski, Z. Iwanow, J. Bauer, "ControUed mutation in evolutionary structural optimization", *Structural and Multidisciplinary Optimization*, 21-5 (2001) 355-360.
146. [5.33] D.F. Haasl. "Advanced concepts on Fault Tree Analysis". The Boeing company System Safety Symposium, 1965, USA.
147. [3.72] M.N.S. Hadi, Y. Arfiadi, "Optimum rigid pavement design by genetic algorithms", *Computers & Structures* 79 (2001) 1617-1624.
148. [3.73] P. Hajela, E. Lee, H. Cho, "Genetic Algorithms in Topologic Design of Grillage Structures", *Computer-Aided Civil and Infrastructure Engineering*, Vol 13, pp. 13-22 (1998).
149. [2.33][3.74] P. Hajela, C.Y. Lin, "Genetic search strategies in multicriterion optimal design", *Structural Optimization*, 4 (1992) 99-107.
150. [3.75] P. Hajela, C.J. Shih, "Multiobjective optimum design in mixed integer and discrete design variable problems", *AIAA Journal*, 28-4 (1990) pp. 670-675.
151. [2.34] G. Harik. 'Finding múltiple solutions in problems of bounded difficulty'. *Illigal Report N" 94002* (1994).
152. [3.76] G.R. Harik, D.E. Goldberg, "Linkage learning through probabilistic expresión", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 295-310.
153. [3.77] S.D. Harris, L. Elliott, D.B. Ingham, M. Pourkashanian, C.W. Wilson, "The optimisation of reaction rate parameters for chemical kinetic modelling of combustion using genetic algorithms", *Comput. Methods Appl. Mech Engrg*, 190 (2000) 1065-1090.
154. [3.78] O. Hasancebi, F. Erbatur, "Evaluation of crossover techniques in genetic algorithm based optimum structural design", *Computers & Structures* 78 (2000) 435-448.
155. [3.79] O. Hasancebi, F. Erbatur, "Constraint handling in genetic algorithm integrated structural optimization". *Acta Mechanica*, 139-1-4, (2000) pp. 15-31.
156. [3.80] M.S. Hayalioglu, "Optimum design of geometrically non-linear elastic-plastic steel frames via genetic algorithm", *Computers & Structures* 11 (2000) 527-538.
157. [3.81] S. Hernández, "Post-optimal procedures for structural optimization", *Advances in Engineering Software*, 31-6 (2000) 401-409.
158. [3.82] S. Hernández, "Post-optimal procedures for structural optimization", *Computer Aided Optimum Design of Structures V*, (1997) 85-100.
159. [1.8][3.83] S. Hernández, "Del diseño convencional al diseño óptimo. Posibilidades y variantes. Parte I. Análisis de sensibilidad y optimización local y global", *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, Vol 9-1, (1993) pp. 91-110.
160. [3.84] S. Hernández. 'Métodos de diseño óptimo de estructuras'. Colección Señor. Colegio de Ingenieros de Caminos, Canales y Puertos (Madrid). (1990).
161. [3.85] A. Hernández Aguirre, S. Botello, C. Coello, G. Lizárraga, E. Mezura-Montes, "Handling constraints using multiobjective optimization concepts", *International Journal for Numerical Methods in Engineering*, 59 (2004) pp. 1989-2017.

162. [4.26] R. Hinterding, H. Gielewski, T.C. Peachey, "The nature of mutation in genetic algorithms", Proceedings of the Sixth International Conference on Genetic Algorithms 1989, Morgan Kaufmann, pp. 70-79.
163. [1.9][5.34] J.H. HoUand. *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press., 1975.
164. [2.35][4.27] J.H. HoUand. 'Algoritmos Genéticos'. En "*Investigación y Ciencia*" (Septiembre 1992) pp 38-45.
165. [2.36] J. Hom, "The Nature of Niching: Genetic Algorithms and the Evolution of Optimal, Cooperative Populations", *Thesis*, University of Illinois at Urbana Champaign, EEUU (1997).
166. [2.37] J. Hom, N. Nafpliotis, "Multiobjective Optimization using the Niche Pareto Genetic Algorithm", *Technical Report Illigal 93005*, Universtity of Illinois at Urbana Champaign, EEUU (1993).
167. [3.86] O. Hrstka, A. Kucerová, M. Leps, J. Zeman, "A competitive comparison of different types of evolutionary algorithms", *Computers & Structures* 81 (2003) 1979-1990.
168. [3.87] M-W. Huang, J.S. Arora, "Optimal design with discrete variables: some numerical experiments", *International Journal for Numerical Methods in Engineering*, 40 (1997) 165-188.
169. [2.38] C.L. Hwang, A.S.M. Masud, "*Múltiple Objective Decisión Making - Methods and Applications: A State-of-the-Art Survey*", Springer-Verlag (1979).
170. [3.88] M. J. Jakiela, C. Chapman, J. Duda, A. Adewuya, K. Saitou, "Continuum structural topology design with genetic algorithms", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 339-356.
171. [3.89] K. Jarmai, J.A. Snyman, J. Farkas, G. Gondos, "Optimal design of a welded I-section frame using four conceptually different optimization algorithms", *Structural and Multidisciplinary Optimization*, 25-1 (2003) 54-61.
172. [2.39] A. Jaskiewicz, R. Slowinski, "The 'Light Beam Search' Approach - An Overview of Methodology and Applications." *European Journal of Operational Research*, 113 (1999) pp. 300-314.
173. [3.90] W.M. Jenkins, "A decimal-coded evolutionary algorithm for constrained optimization" , *Computers & Structures* 80 (2002) 471-480.
174. [3.91] W.M. Jenkins, "On the Application of Natural Algorithms to Structural Design Optimisation", *Engineering Structures*, 19-4 (1997) pp 302-208.
175. [3.92] W.M. Jenkins, "Towards structural optimization via the genetic algorithm" , *Computers & Structures* 40 (1991) 1321-1327.
176. [4.28] M. Jensen, "Guiding Single-objective optimization using Multi-objective Methods", *Applications of Evolutionary Computation, Lecture Notes in Computer Science*, pp. 268-279, Best paper award EvoCOP 2003.
177. [3.93] R.J. Jiang, F.T.K. Au, Y.K. Cheung, "Identification of masses moving on multi-span beams based on a genetic algorithm", *Computers & Structures* 81 (2003) 2137-2148.
178. [2.40] J. Jiménez, D. Greiner, J. Abderramán, P. Cuesta y G. Winter. 'Un análisis comparativo de algoritmos multiobjetivo: NSGA-II y MRSMO', *Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados*, AEB-02. Marida, España.

179. [4.29] K. de Jong, J. Sarma, "Generation Gaps Revisited". In // *Foundations Of Genetic Algorithms (FOGA 1992)* pp. 19-28. Ed. D. Whitley. Morgan Kaufmann Publishers.
180. [4.30] E. de Jong, R. Watson, J. PoUack, "Reducing bloat and promoting diversity using multi-objective methods", *Proceedings of the Genetic and Evolutionary Computation Conference 2001*, pp. 11-18.
181. [3.94] E.S. Kameshki, M.P. Saka, "Optimum design of nonlinear steel frames with semi-rigid connections using a genetic algorithm", *Computers & Structures* 79 (2001) 1593-1604.
182. [3.95] H. Kargupta, S. Bandyopadhyay, "A perspective on the foundation and evolution of the linkage learning genetic algorithms", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 269-294.
183. [3.96] C.L. Karr, D.A. Stanley, B. McWhorter, "Optimization of hydroclone operation using a genetic algorithm", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 517-530.
184. [3.97] A. Kaveh, V. Kalatjari, "Topology optimization of trusses using genetic algorithm, forced method and graph theory", *International Journal for Numerical Methods in Engineering*, 58 (2003) 771-791.
185. [3.98] A. Kaveh, V. Kalatjari, "Genetic algorithm for discrete-sizing optimal design of trusses using the forced method", *International Journal for Numerical Methods in Engineering*, 55 (2002) 55-72.
186. [3.99] H. Kawamura, H. Ohmori, N. Kito, "Truss topology optimization by a modified genetic algorithm", *Structural and Multidisciplinary Optimization*, 23-6 (2002) 467-472.
187. [3.100] M. Keser, S. Stupp, "Genetic algorithms in computational materials science and engineering: simulation and design of self-assembling materials", *Comput. Methods Appl. Mech Engrg*, 186(2000)373-385.
188. [3.101] S. Khajepour, D. Grierson, "Conceptual Design using Adaptive Computing", *Proceedings of the Genetic and Evolutionary Computation Conference* (2001), pp. 62-67.
189. [3.102] N.S. Khot, "Multicriteria optimization for design of structures with active control", *Journal of Aerospace Engineering*, 11-2 (1998) 45-51.
190. [3.103] H. Kim, H. Adeli, "Discrete cost optimization of composite floors using a floating-point genetic algorithm", *Engineering Optimization*, 33-4 (2001) 485-501.
191. [3.104] Y.J. Kim, J. Ghaboussi, "A New Method of Reduced Order Feedback Control using Genetic Algorithms", *Earthquake Engineering and Structural Dynamics*, 28-2 (1999) pp. 193-212.
192. [3.105] J.E. Kim, N. Sarigul-Klijn, "Elastic-dynamic rotor blade design with multiobjective optimization", *^Z4^ Journal*, 39-9 (2001) 1652-1661.
193. [2.41] J. Knowles, D. Come, "One metrics for computing non-dominated set. Proceedings of the World Congress on Computational Intelligence, (2002) pp. 711-716.
194. [2.42] J. Knowles, D. Come, "Approximating the Nondominated Front Using the Pareto Archive Evolution Strategy", *Evolutionary Computation*, 8-2 (2000) pp. 149-172.
195. [4.31] J. Knowles, R. Watson, D. Come, "Reducing Local Optima in Single-Objective Problems by Multiobjectivization". Evolutionary Multi-Criterion Optimization 2001, *Lecture Notes in Computer Science*, pp. 269-283.
196. [3.106] D.E. Ktiuth. 'El Arte de programar ordenadores' Vil volúmenes. (1980).
197. [3.107] J. Koski, "Multicriteria optimization in structural design: state of the art", in: *Proceedings of the 19' Design Automation Conferences ASME*, (1993) pp. 621-629.

198. [3.108] J. Koski, "Truss Optimization with Vector Criterion" Report n° 6, Tampere University of Technology, Tampere, Finland (1979).
199. [3.109] J. Koski, "Truss Optimization with Vector Criterion. Examples" Report n° 7, Tampere University of Technology, Tampere, Finland (1980).
200. [3.110] V.K. Koumousis, P.G. Georgiou, "Genetic Algorithms in Discrete Optimization of Steel Truss Roofs", *Journal of Computing in Civil Engineering*, 8-3 pp. 309-325.
201. [3.111] J.R. Koza, F.H. Bennett 111, D. Andre, M.A. Keane, "Synthesis of topology and sizing of analog electrical circuits by means of genetic programming", *Comput. Methods Appl. Mech* i:«rg, 186(2000)459-482.
202. [4.32] K. Krishnakumar, Micro-genetic algorithms for stationary and non-stationary function optimization, SPIE: *Intell. Control Adapt. Syst.* 1196 (1989).
203. [3.112] e s. Krishnamoorthy, "Structural optimization in practice: Potential applications of genetic algorithms", *Structural Engineering and Mechanics*, 11-2 (2001) 151-170.
204. [3.113] E.M. Krokosky, "The ideal multifunctional constructural material", *Journal of the Structural División ASCE* 94 (1968) 958-981.
205. [3.114] S. Kundu, "A Multicriteria Genetic Algorithm to Solve Optimization Problems in Structural Engineering Design", *Information Processing in Civil and Structural Engineering Design*, (1996), Civil-Comp Press, pp. 225-233.
206. [3.115] A. Kurapati, S. Azarm, "Immune Network Simulation with Multiobjective Genetic Algorithms for Multidisciplinary Design Optimization", *Engineering Optimization*, 33 (2000) pp. 245-260.
207. [2.43] F. Kursawe, "A variant of evolution strategies for vector optimization", *Lecture Notes in Computer Science* 496, Parallel Problem Solving from Nature, pp. 193-197. Springer (1991).
208. [5.35] W. Kuo W, V. Rajendra Prasad, An Annotated Overview of System-Reliability Optimization. *IEEE Transactions on Reliability*, Vol. 49, No. 2, pp. 176-187, June 2000.
209. [3.116] N. Lagaros, M. Papadrakakis, G. Kokossalakis, "Structural optimization using evolutionary algorithms", *Computers & Structures* 80 (2002) 571-589.
210. [2.44] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization", *Evolutionary Computation*, 10-3, (2002) pp. 263-282.
211. [2.45] M. Laumanns, G. Rudolph, H.P. Schwefel, "A spatial predator-prey approach to multiobjective optimization: a preliminary study", *Parallel Problem Solving from Nature, Lecture Notes in Computer Science*, (1998) pp. 241-249.
212. [5.36] M. Laumanns, E. Zitzler, L. Thiele. On the Effects of Archiving, Elitism, and Density Based Selection in Evolutionary Multi-objective Optimization. pp. 181-196, in *Evolutionary Multi-Criterion Optimization*, Springer, 2001.
213. [3.117] D. Lee, "Multiobjective design of a marine vehicle with aid of design knowledge" , *International Journal for Numerical Methods in Engineering*, 40 (1997) 2665-2677.
214. [5.37] W.S. Lee, D.L. Grosh, F.A. Tillman, C.H. Lie. "Fault Tree Analysis, Methods and Applications - A review". *IEEE Transactions on Reliability*, R-34(3), 1985, pp. 194-203.
215. [3.118] J.P. Leite, B.H.V. Topping, "Improved Genetic Operators for Structural Engineering Optimization", *Advances in Engineering Software*, 29-7/9 (1998) 529-562.

216. [3.119] G. Leitmann, "Some problems of scalar and vector-valued optimization in linear viscoelasticity", *Journal Optim. TheoryAppl.* 23 (1977) pp. 93-99.
217. [3.120] A.C.C. Lemonge, H.J.C. Barbosa, "An adaptive penalty scheme for genetic algorithms in structural optimization", *International Journal for Numerical Methods in Engineering*, 59 (2004) 703-736.
218. [3.121] X.S. Li, G.D. Cheng, D.Y. Changsong, "A central path-following algorithm for multi-objective optimization of large scale structures", Proceedings of Structural and Mechanical systems, Xidian University Press, (1999) 453-460.
219. [3.122] Q.S. Li, D.K. Liu, A.Y.T. Leung, N. Zhang, Q.Z. Luo, "A multilevel genetic algorithm for the optimum design of structural control systems", *International Journal for Numerical Methods in Engineering*, 55 (2002) 817-834.
220. [3.123] Q. Li, G.P. Steven, Y.M. Xie, "Evolutionary thickness design with stiffness maximization and stress minimization criteria", *International Journal for Numerical Methods in Engineering*, 52 (2001) 979-995.
221. [3.124] X. Liu, D.W. Begg, "Second-order sensitivity of Smart Structures", *Journal of Aerospace Engineering*, 12-1 (1999) 15-22.
222. [3.125] X. Liu, D.W. Begg, R.J. Fishwick, "Genetic approach to optimal topology/control design of adaptive structures", *International Journal for Numerical Methods in Engineering*, 41 (1998) pp. 815-830.
223. [3.126] X. Liu, D.W. Begg, D.R. Matravers, "Optimal Topology/Actuator Placement Design of Structures using GA", *Journal of Aerospace Engineering*, 10-3 (1997) 119-125.
224. [3.127] M. Liu, S.A. Bums, Y.K. Wen, "Optimal seismic design of steel frame buildings based on life cycle cost considerations", *Earthquake Engrg. Struct.* 32-9 (2003) 1313-1332.
225. [3.128] G.R. Liu, S.C. Chen, "Flaw detection in sandwich plates based on time-harmonic response using genetic algorithm", *Comput. Methods Appl. Mech Engrg*, 190 (2001) 5505-5514.
226. [3.129] B. Liu, R.T. Haftka, M.A. Akgün, A. Todoroki, "Permutation genetic algorithm for stacking sequence design of composite laminates", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 357-372.
227. [3.130] G.R. Liu, X. Han, K.Y. Lam, "A combined genetic algorithm and nonlinear least squares method for material characterization using elastic waves", *Comput. Methods Appl. Mech Engrg*, 191 (2002) 1909-1921.
228. [3.131] W.-H. Liu and A. H. Sherman. Comparative analysis of the Cuthill-McKee and the reverse CuthillMcKee ordering algorithms for sparse matrices. *SIAM Journal on Numerical Analysis*, 13(2):198-213, Apr. 1976.
229. [3.132] D.K. Liu, Y.L. Yang, Q.S. Li, "Optimum positioning of actuators in tall buildings using genetic algorithm", *Computers & Structures* 81 (2003) 2823-2827.
230. [3.133] Z. Lounis, M.Z. Cohn, "Multiobjective Optimization of Prestressed Concrete Structures", *Journal of Structural Engineering*, 119-3 (1993) 794-808.
231. [3.134] T.A. Lovell, "Application of genetic algorithms to state estimation of tethered systems", *Comput. Methods Appl Mech Engrg*, 192 (2003) 1799-1819.
232. [2.47] W. Mahfoud. 'Niching methods for Genetic Algorithms' *Illigal Report n" 95001* (1995).

233. [3.135] S. Marburg, "Developments in structural-acoustic optimization for passive noise control", *Archives of Computational Methods in Engineering*, 9-4 (2002) 291-370.
234. [3.136] J.L. Marcelin, "Genetic optimization of stiffened plates without the FE mesh support" , *International Journal for Numerical Methods in Engineering*, 54 (2002) 685-694.
235. [3.137] J.L. Marcelin, "Genetic optimization of stiffened plates and shells", *International Journal for Numerical Methods in Engineering*, 51 (2001) 1079-1088.
236. [2.46][3.138] R.T. Marler, J.S. Arora, "Survey of multi-objective optimization methods for engineering", *Structural and Multidisciplinary Optimization*, 26 (2004) pp. 369-395.
237. [3.139] H.C. Martin, G.C. Carey, "Introduction to finite element analysis: theory and application", New York: McGraw Hill; 1973.
238. [5.38] S. Martorell, A. Muñoz, y V. Serradell, Age-dependent models for evaluating risks and costs of surveillance and maintenance of components. *IEEE Transactions on Reliability*, 1996, Vol. 45, No.3, pp. 433-442.
239. [5.39] S. Martorell, C. Sánchez, V. Serradell, Constrained optimization of test intervals using steady-state genetic algorithm. *Reí. Engng. & System Safety*, Vol. 67, 2000, pp. 215-232.
240. [4.33] K.E. Mathias, D. Whitley, "Transforming the Search Space with Gray Coding", In D. Schaffer, ed. *IEEE Int. Conf. on Evolutionary Computation*, (1994) pp. 513-518.
241. [3.140] K. Matous, M. Leps, J. Zeman, M. Sejnoha, "Applying genetic algorithms to selected topics commonly encountered in engineering practice", *Comput. Methods Appl. Mech Engrg*, 190 (2000) 1629-1650.
242. [3.141] M. Matsumoto, J. Abe, M. Yoshimura, "A multiobjective optimization strategy with priority ranking of the design objectives", *Journal of Mechanical Design*, 115-4 (1993) 784-792.
243. [3.142] D.S. McCorkle, K.M. Bryden, C.G. Carmichael, "A new methodology for evolutionary optimization of energy systems", *Comput Methods Appl. Mech Engrg*, 192 (2003) 5021-5036.
244. [4.34] G. Meneghetti, V. Pediroda, C. Poloni, "Application of al muhiobjective Genetic Algorithm and a Neural Network to the Optimization of Foundry Processes", *Evolutionary Algorithms in Engineering and Computer Science*, pp. 457-469. Eds: K. Miettinen, P. Neittarmiaki, M. Mákela, J. Periaux. John Wiley & Sons, 1999.
245. [2.48] A. Messac, A. Ismail-Yahaya, C.A. Mattson, "The normalized normal constraint method for generating the Pareto Frontier", *Structural and Multidisciplinary Optimization*, 25 (2003) pp. 86-98.
246. [2.49] A. Messac, "Physical Programming: effective optimization for computational design", *AIAA Journal*, 34 (1996) pp. 149-158.
247. [3.143] M. Metcalf 'FORTRAN optimization'. Academic Press (1985).
248. [3.144] Z. Michalewicz, 'Genetic Algorithms + Data Structures: Evolution Programs', Springer Verlag, (1994).
249. [3.145] Z. Michalewicz, M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation* 1996; 4(1):1-32.
250. [2.50] K. Miettinen, "Some Methods for Nonlinear Multi-objective Optimization", *Lecture Notes in Computer Science*, Evolutionary Multi-criterion Optimization 2001, pp. 1-20.
251. [2.51] K. Miettinen, "Nonlinear Multi-objective Optimization", Kluwer Academic Publishers, 1999.

252. [3.146] K. Miettinen, M. Makela, J. Toivanen, "Numerical comparison of some penalty-based constraint handling techniques in genetic algorithms", *Journal of Global Optimization*, 27-4 (2003) pp. 427-446
253. [4.35] D.J. Mikulin, Using genetic algorithms to fit HLGGM data, *PhD Thesis*, University of Exeter, 1997.
254. [3.147] J.C. Miles, G.M. Sisk, C.J. Moore, "The conceptual design of commercial buildings using a genetic algorithm", *Computers & Structures* 79 (2001) 1583-1592.
255. [3.148] P. Miltiadou, L.C. Wrobel, "A BEM-based genetic algorithm for Identification of polarization curves in cathodic protection systems", *International Journal for Numerical Methods in Engineering*, 54 (2002) 159-174.
256. [3.149] S. Min, S. Nishiwaki, N. Kikuchi, "Unified topology design of static and vibrating structures using multiobjective optimization", *Computers & Structures* 75 (2000) 93-116.
257. [3.150] F. Moran Cabré. 'Análisis Matricial de estructuras en ordenadores personales compatibles'. Editorial Rueda. (1990).
258. [3.151] L. Moya Ferrer. 'Análisis Matricial de estructuras de barras' Ediciones UPC (1995).Barcelona.
259. [3.152] P.B. Nair, A.J. Keane, R.P. Shimpi, "Combining Approximation Concepts with Genetic Algorithm-Based Optimization", *AISS / ASME / ASCE / AHS / ASC Structures, Structural Dynamics & Materials Conference*, Vol 2, pp. 1741-1751 AIAA, USA (1998).
260. [2.52] H. Nakayama, "Aspiration level approach to Interactive multi-objective programming and its applications". In: *Advances in Multicriteria Analysis*. Kluwer Academic Publishers (1995) pp. 147-174.
261. [3.153] P. Nanakom, K. Meesomklin, "An adaptive penalty fimction in genetic algorithms for structural design optimization", *Computers & Structures* 79 (2001) 2527-2539.
262. [3.154] S. Narayanan, S. Azarm, "On Improving Multiobjective Genetic Algorithms for Design Optimization", *Structural Optimization*, 18 (1999) pp. 146-155.
263. [2.53] J.F. Nash, Equilibrium points in n-person games. *Proc. Nat. Acad. Sel U.S.A.*, 36 (1950) 46-49.
264. [2.54] J.F. Nash, Noncooperative games. *Annals of Mathematics*, 54 (1951) 289.
265. [3.155] E. Nimwegen, J. Crutchfield, "Optimizing epochal evolutionary search: population-size independent theory", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 171-194.
266. [1.10] A.E. Nix, M. D. Vose, "Modelling genetic algorithms with Markov chains" *Annals of Mathematics and Artificial Intelligence* 5, (1992), pp. 79-88.
267. [4.36] Y. Nojima, K. Narukawa, S. Kaige, H. Ishibushi, "Effects of Removing Overlapping Solutions on the Performance of the NSGAll Algorithm", Third International Conference on Evolutionary Multi-Criterion Optimization 2005, *Lecture Notes in Computer Science* n° 3410, pp. 341-354.
268. [3.156] Normativa española básica de edificación EA-95.
269. [3.157] Normativa española básica de edificación AE-88.
270. [5.40] NUREG-75/014: WASH-1400. "Reactor Safety Study". U.S. Nuclear Regulatory Commission, 1975.

271. [2.55][5.41] S. Obayashi, S. Takahashi, Y. Takeguchi, Niching and elitist models for MOGAs. *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, Amsterdam (1998), pp. 250-259.
272. [3.158] I. Ono, S. Kobayashi, K. Yoshida, "Optimal lens design by real-coded genetic algorithms using UNDX", *Comput Methods Appl. Mech Engrg*, 186 (2000) 483-497.
273. [3.159] E. Oñate. 'Cálculo de estructuras por el método de elementos finitos', CIMNE, (1992).
274. [3.160] A. Osyczka, "Multicriteria optimization for engineering design", in: *Design Optimization*, Academic Press (1985) 193-227.
275. [3.161] A. Osyczka, S. Krenich, "Evolutionary Algorithms for Multicriteria Optimization with Selecting a Representative Subset of Pareto Optimal Solutions", *First International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science* 1993, (2001) pp. 141-153.
276. [3.162] A. Osyczka, S. Kundu, "New Method to solve generalized multicriteria optimization problem using the simple genetic algorithm", *Structural Optimization*, 10-2 (1995) pp. 94-99.
277. [2.56] V. Pareto, "*Cours D' Economie Politique*", Volume I y II, F. Rouge, Lausaime, 1896.
278. [3.163] M. Papadrakakis, N. Lagaros, Y. Fragakis, "Parallel computational strategies for structural optimization", *International Journal for Numerical Methods in Engineering*, 58 (2003) 1347-1380.
279. [3.164] M. Papadrakakis, N. Lagaros, V. Plevris, "Multi-objective optimization of skeletal structures under static and seismic loading conditions", *Engineering Optimization*, 34-6 (2002) 645-669.
280. [3.165] M. Papadrakakis, N. Lagaros, Y. Tsompanakis, V. Plevris, "Large scale structural optimization: Computational methods and optimization algorithms". *Archives of computational methods in engineering*, 8-3 (2001) 239-301.
281. [5.42] R.L. Pattisson, J.D. Andrews. Genetic algorithms in optimal safety system design. In *Proc. Inst. Mech. Engrs*. Vol. 213, Part E, pp. 187-197. 1999.
282. [1.11] J. Periaux, B. Galván, B. González, A. Blesa, JI Jiménez, G. Winter, "Red Europea INGENet y Base de Datos de Casos Test en Aplicaciones de interés Industrial", *Actas del II Congreso español sobre Metaheurísticas, Algoritmos evolutivos y bioinspirados*, (2003) Gijón, pp. 62-69.
283. [2.57] J. Periaux, M. Sefi-ioui, "Nash Genetic Algorithms: Examples and Applications", In- *Innovative Tools for Scientific Computation in Aeronautical Engineering*, Ed: J. Periaux, P. Joly, O. Pironneau, E. Oñate. CIMNE, (2001) pp. 391-404.
284. [3.166] B. Pichler, R. Lackner, H.A. Mang, "Back analysis of model parameters in geotechnical engineering by means of soft computing", *International Journal for Numerical Methods in Engineering*, 57 (2003) 1943-1978.
285. [3.167] R. Picón. 'Resistencia de materiales' E.T.S.I.I. U.L.P.G.C. (1989)
286. [3.168] C. Poloni, A. Giurgevich, L. Onesti, V. Pediroda, "Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 403-420.

287. [3.169] P. Ponterosso, R. J. Fishwick, D. St. J. Fox, X.L. Liu, D.W. Begg, "Masonry arch collapse loads and mechanisms by heuristically ceded genetic algorithm", *Comput. Methods Appl. MechEngrg*, 190 (2000) 1233-1243.
288. [3.170] H.N. Press, B.P. Flannery, S.A. Teukolsly, W.T. Vetterling. 'Numerical Recipes in C. Cambridge University Press. (1989).
289. [2.58][5.43] R. Purshouse, P. Fleming, "Why use Elitism and Sharing in a Multiobjective Genetic Algorithm?", *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2002*, pp. 520-527, New York, Morgan Kaufmann Publishers.
290. [4.37] N.J. Radcliffe, P.D. Surry, "Fundamental Limitations on Search Algorithms: Evolutionary Computing in Perspective", *Lecture Notes in Computer Science* 1000 (Computer Science Today: Recent Trends and Development), Springer, 1995.
291. [3.171] S.D. Rajan, "Sizing, shape and topology design optimization of trusses using genetic algorithms", *ASCE Journal ofStructural Engineering*, 121-10 (1995) 1480-1487.
292. [3.172][4.38] S. Rajeev, C.S. Krishnamoorthy, "Genetic algorithms-based methodologies for design optimization of trusses", *Journal ofStructural Engineering-ASCE*, 123-3 (1997) 350-358.
293. [3.173] S. Rajeev, C.S. Krishnamoorthy, "Discrete Optimization of Structures using Genetic Algorithms". *Journal ofStructural Engineering*, v.1 18, n° 5, (1992) pp 1233-1250.
294. [3.174] I. Rajendran, S. Vijayarangan, "Optimal design of a composite leaf spring using genetic algorithms", *Computers & Structures* 79 (2001) 1121-1129.
295. [3.175] S.S. Rao, "Genetic Algorithmic Approach for Multiobjective Optimization of Structures", *ASME Annual Winter Meeting, Structures and Controls Optimization*, vol. AD-38, pp. 29-38 (1993).
296. [3.176] S.S. Rao, "Game theory approach for multiobjective structural optimization", *Computers & Structures* 25 (1987) 119-127.
297. [3.177] S.S. Rao, K. Sundararaju, B.G. Prakash, C. Balakrishna, "Multiobjective fuzzy optimization techniques for engineering design", *Computers & Structures* 42 (1992) 37-44.
298. [3.178] S.S. Rao, K. Sundararaju, B.G. Prakash, C. Balakrishna, "Fuzzy goal programming approach for structural optimization", *AIAA Journal*, 30-5 (1992) 1425-1432.
299. [3.179] I. Rechenberg, "Case studies in evolutionary experimentation and computation", *Comput. Methods Appl. MechEngrg*, 186 (2000) 125-140.
300. [3.180] J.T. Richardson, M.R. Palmer, G. Liepins, M. Hilliard. Some guidelines for genetic algorithms with penalty functions. In *Proceedings of the Third International Conference on Genetic Algorithms (ICGA-89)*, San Mateo, CA, Schaffer JD (ed.). George Masón University, Morgan Kaufmann: Los Altos, CA, June 1989; 191-197.
301. [2.59] O. Roudenko, "Application des Algorithmes Evolutionnaires aux Problemes d'Optimisation Multi-Objectif avec Contraintes", *7/7D Thesis*, Marzo 2004, École Polytechnique.
302. [4.39] J. Rowe, D. Whitley, L. Barbulescu, P. Watson, "Properties of Gray and Binary Representations", *Evolutionary Computation*, 12-1 (2004) pp. 46-76.
303. [4.40] G. Rudolph (1994). "Convergence Analysis of Canonical Genetic Algorithms", *IEEE Transactions on Neural Networks* 5-1 (1994) pp. 96-101.

304. [3.181] Y. Saad. 'Preconditioned Krylov Subspace Methods'. In "Algorithms for large scale systems. State of the art and applications in Science and Engineering". Kluwer. Editors G. Winter and E. Spedicato. (1997)
305. [3.182] M.P. Saka, "Optimum design of pitched roof steel frames with haunched rafters by genetic algorithm", *Computers & Structures* 81 (2003) 1967-1978.
306. [3.183] M.P. Saka, "Optimum Design of Grillage Systems using Genetic Algorithms", *Computer Aided Civil and Infrastructure Engineering*, Vol 13, pp. 297-302 (1998).
307. [3.184] J. Sakamoto, J. Oda, "Technique for determination of optimal truss layout using genetic algorithm", *Nippon Kikai Gakkai Ronbunshu*, a Hen v 59-562 (1993), pp. 1568-1573.
308. [3.185] S.S.S. Sakla, E. Elbeltagi, "Design of steel roofs subjected to driñed snow using genetic optimization", *Computers & Structures* 81 (2003) 339-348.
309. [2.60] G. Sánchez Carpena, "Diseño y Evaluación de algoritmos evolutivos multiobjetivo en optimización y modelización difusa", *Tesis Doctoral*, 2002, Universidad de Murcia, Departamento de Ingeniería de la Información y las Comunicaciones.
310. [3.186] E. Sandgren, "Multicriteria Design Optimization by Goal Programming", *Advances in Design Optimization*, (1994) pp 225-265.
311. [3.187] E. Sandgren, T.M. Cameron, "Robust design optimization of structures through consideration of variation", *Computers & Structures* 80 (2002) 1605-1613.
312. [3.188] K.C. Sarma, H. Adeli, "Life-cycle cost optimization of steel structures", *International Journal for Numerical Methods in Engineering*, 55 (2002) 1451-1462.
313. [3.189] K.C. Sarma, H. Adeli, "Bilevel parallel genetic algorithms for optimization of large steel structures", *Computer-Aided civil and infrastructure engineering*, 16-5 (2001) 295-304.
314. [2.61] D. Schaffer. Múltiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pp. 93-100. (1985) Lawrence Erlbaum.
315. [3.190] Herbert Schildt. 'Lenguaje C. Programación Avanzada'. Me Graw Hill. (1988).
316. [5.44] W.G. Schneeweiss. Boolean Functions with Engineering Applications and Computer Programs. Springer-Verlag, 1989.
317. [3.191] M. Schütz, H.P. Schwefel, "Evolutionary approaches to solve three challenging engineering tasks", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 141-170.
318. [2.62] M. Sefrioui, J. Periaux, "Nash Genetic Algorithms: Examples and Applications", *Proceedings of the Congress on Evolutionary Computation 2000*, IEEE Press, pp. 509-516.
319. [3.192] C.J. Shih, T.K. Lai, "Fuzzy weighting optimization with several objective functions in structural design", *Computers & Structures* 52 (1994) 917-924.
320. [3.193] A.E. Smith, D.W. Coit. Constraint handling techniques—^penalty functions. In: *Handbook of Evolutionary Computation*, Back T, Fogel DB, Michalewicz Z (eds). Chapter C 5.2. Oxford University Press and Institute of Physics Publishing: Bristol, UK, 1997.
321. [3.194] R.E. Smith, B.A. Dike, R.K. Mehra, B. Ravichandran, A. El-Fallah, "Classifier systems in combat: two-sided learning of maneuvers for advanced fighter aircraft", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 421-437.
322. [3.195] C.K. Soh, J. Yang, "Optimal Layout of Bridge Trusses by Genetic Algorithms", *Computer-Aided Civil and Infrastructure Engineering*, Vol 13, pp. 247-254 (1998).

323. [3.196] G. Soremekun, Z. Gürdal, R.T. Haftka, L.T. Watson, "Composite laminate design optimization by genetic algorithm with generalized elitist selection", *Computers & Structures* 79 (2001) 131-143.
324. [3.197] R. Spallino, G. Thierauf, "Thermal buckling optimization of composite laminates by evolution strategies", *Computers & Structures* 78 (2000) 691-697.
325. [2.63][5.45] N. Smivas, K. Deb, 'Multiobjective optimisation using non-dominated sorting in genetic algorithms' *Evolutionary Computation* 2-3 (1995) pp. 221-248.
326. [2.64] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms. *Technical report*, Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, India. (1993).
327. [3.198] W. Stadler, "Multicriteria optimization in mechanics (a survey)", *Applied Mechanics Review*, 37-3 (1984) pp. 277-286.
328. [3.199] W. Stadler, "Natural Structural shapes of shallow arches", *J. Appl. Mech.* 44 (1977) pp. 291-298.
329. [3.200] W. Stadler, "Natural Structural shapes (the static case)", *Q. J. Mech. Appl. Math*, 31 (1978) pp. 169-217.
330. [2.65] R.E. Steuer, "*Múltiple Critería Optimization: Theory, Computation and Applications*", John Wiley & Sons, (1986).
331. [3.201] G.P. Steven, Q. Li, Y.M. Xie, "Multicriteria optimization that minimizes maximum stress and maximizes stiffness", *Computers & Structures* 80 (2002) 2433-2448.
332. [3.202] G. Sved, L.J. Schmid, A.R. Simpson, "Minimum Weight Structures designed by Genetic Algorithms", *Computational Mechanics*, Eds: Cheung, Lee & Leung, University of Hong-Kong (1991) pp. 317-322.
333. [4.41] G. Syswerda, "A Study of Reproduction in Generational and Steady-State Genetic Algorithms", *Foundations of Genetic Algorithms (FOGA 1991)*, Morgan Kaufmann, pp. 94-101.
334. [4.42] G. Syswerda, "Uniform Crossover in Genetic Algorithms", *Proceedings of the Third International Conference on Genetic Algorithms* 1989, Morgan Kaufmann, pp. 2-9.
335. [3.203] J. Sziveri, C.F. Scale, B.H.V. Topping, "An enhanced parallel sub-domain generation method for mesh partitioning in parallel finite element analysis", *International Journal for Numerical Methods in Engineering*, 47 (2000) pp. 1773-1800.
336. [3.204] F. Tin-Loi, N.S. Que, "Identification of cohesive crack fracture parameters by evolutionary search", *Comput. Methods Appl. Mech. Engrg.* 191 (2002) 5741-5760.
337. [3.205] P. Tiwari, K. Parikh, J. Parikh, "Structural design considerations in house builders model: Optimization approach", *Journal of Infrastructure Systems*, 5-3 (1999) 102-110.
338. [3.206] B.H.V. Topping, J.P.B. Leite, "Parallel genetic models for structural optimization", *Engineering Optimization*, 31-1 (1998) 65-99.
339. [3.207] M.J. Tumer, H.C. Martin, R.C. Weikel, "Further development and applications of the stiffness method AGARD structures and materials panel", Paris, France, July 1962. In: *AGARDograph 72: Matrix Methods of Structural Analysis*, Pergamon Press, Oxford 1964. pp. 203-266.
340. [3.208] G.N. Vanderplaats, "CONMIN - A FORTRAN program for constrained function minimization. User's Manual", NASA Technical Memorandum TM X-62,282.

341. [3.209] P. Van der Linden. 'Expert C Programming. Deep C Secrets'. Prentice Hall (1994).
342. [2.66] D. Van Veldhuizen, G. Lamont, "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art", *Evolutionary Computation*, 8-2 (2000) pp. 125-147.
343. [2.67][4.43][5.47] D. Van Veldhuizen and Gary B. Lamont. "Multiobjective Evolutionary Algorithm Test Suites", *Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 351-357, San Antonio, Texas,. ACM . (1999).
344. [2.68] D. Van Vedhuizen, G. Lamont, "Multiobjective Evolutionary Algorithm Research: A History and Analysis". Technical Report TR-98-03, Department of Electrical and Computer Engineering, Air Forcé Institute of Technology, Wright-Patterson AFB, Ohio (1998).
345. [5.46] W.E. Vesely, F.F. Goldberg, N.H. Roberts, D.F. Haals. Fault Tree Handbook. Systems and Reliability research office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington D.C., January, 1981.
346. [3.210] M. Vose, J.E. Rowe, "Random heuristic search: applications to Gas and functions of unitation", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 195-220.
347. [3.211] D.R. Wallace, M.J. Jakiela, W. Flowers, "Design Search under Probabilistic Specifications using Genetic Algorithms", *Computer-Aided Design*, 28-5 pp. 405-421 (1996).
348. [4.44] N. Weicker, G. Szabo, K. Weicker, P. Widmayer, "Evolutionary Muhiobjective Optimization for Base Station Transmitter Placement with Frequency Assignment", *IEEE Transactions on Evolutionary Computation*, 7-2 (2003) pp. 189-203.
349. [3.212] D. S. Weile, E. Michielssen, "The use of domain decomposition genetic algorithms exploiting model reduction for the design of frequency selective surfaces", *Comput. Methods Appl. Mech Engrg*, 186 (2000) 439-458.
350. [4.45] D. Whitley, "A free lunch proof for Gray versus Binary Codings", *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO 1999, Morgan Kaufmann.
351. [2.70][4.46] D. Whitley, "The GENITOR Algorithm and Selection Pressure: Why Rank-Based AUocation of Reproductive Triáis is Best", *Proceedings of the Third International Conference on Genetic Algorithms* 1989, Morgan Kaufmann, pp. 116-123.
352. [4.47] D. Whitley, J. Kauth, "Genitor: A Different Genetic Algorithm", Technical Report Cs-88-101, Colorado State University.
353. [4.48] D. Whitley, K. Mathias, P. Fitzhom, Delta coding: an iterative search strategy for genetic algorithms, in: R.K. Belew, L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, 1991, pp. 77-84.
354. [4.49] D. Whitley, K. Mathias, S. Rana, J. Dzuberá, "Evaluating Evolutionary Algorithms", *Artificial Intelligence Journal* 85 (1996) pp. 245-276.
355. [2.69][4.50] D. Whitley, S. Rana, R. Heckendom, "Representation Issues in Neighborhood Search and Evolutionary Algorithms", in eds.: D. Quagliarella, J. Periaux, C. Poloni, G. Winter, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science* (EUROGEN 1997), John Wiley & Sons.
356. [2.71] A.P. Wierzbicki, "A Mathematical Basis for Satisficing Decisi3n Making", *Mathematical Modelling*, 3 (1982) pp. 391-405.
357. [3.213] E.L. Wilson, "Automation of the finite element method - a historical view", *Finite Element in Analysis and Design*, 13 (1993) pp. 91-104.

358. [5.48] P.F. Williams, M. Nikolskaia, A. Rauzy. "Bypassing BDD construction for reliability analysis". *Information Processing Letters*. Elsevier, 75 (2000) 85-89.
359. [2.72] G. Winter, "Recent Advances and Trends in Multiobjective Evolutionary Algorithms for Optimization. Applications in Dependability", In: *Proceedings of the IV Congress on Reliability: Dependability*, Eds: Galván, Winter, Cuesta and Aguasca. September 2002, Spain.
360. [3.214] G. Winter, C. Conde. '*Métodos y algoritmos básicos del álgebra numérica*' Reverte. Barcelona (1990).
361. [5.49] G. Winter, M. Cruz, B. Galván, D. Greiner. "Optimal Scheduling of Power Generation by Evolutionary Algorithms". European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2000. Barcelona, España. CIMNE.
362. [3.215][5.50] G. Winter, M. Galán, P. Cuesta and D. Greiner, 'Genetic Algorithms: A Stochastic Improvement Technique. Tools, Skills, Pitfalls and Examples' In '*Genetic Algorithms in Engineering and Computer Science*', John Wiley & Sons, pp. 217-249. (EUROGEN 1995).
363. [5.51] G. Winter, B. Galván, S. Alonso, B. González. "Evolving from Genetic Algorithms to Flexible Evolution Agents". *Late Breaking Papers GECCO-2002*. pp 466-473. New York.
364. [5.52] G. Winter, B. Galván, S. Alonso y B. González. "Una propuesta de evolución flexible en el diseño de algoritmos evolutivos". *Actas del Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados. (AEBO-2002)* pp. 246-252. Mérida España.
365. [5.53] G. Winter, B. Galván, S. Alonso, B. González, J.I. Jiménez, D. Greiner. "A Flexible Evolutionary Agent: cooperation and competition among real-coded evolutionary operators", *Soft Computing Journal*, Springer, June 2004.
366. [5.54] G. Winter, B. Galván, B. González, S. Alonso. 'Economic Dispatch Optimisation in Electric Power Systems by a Flexible Evolution Agent', *Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Application to Industrial and Societal Problems*. EUROGEN 2003. Ed: G. Bugada, J.A. Desideri, J. Periaux, M. Schoenauer and G. Winter. CIMNE, Barcelona, España.
367. [5.55] G. Winter, B. Galván, D. Greiner, J. Bravo de Laguna. 'Métodos Evolutivos para Optimización Global Multiobjetivo y Aplicaciones en Ingeniería'. En '*Desarrollos recientes en métodos numéricos para ingeniería y ciencias aplicadas*', SVMNI, pp. 17-25.
368. [5.56] G. Winter, D. Greiner, B. Galván and B. González. 'Economical and Environmental Electric Power Dispatch Optimization', *Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Application to Industrial and Societal Problems*. EUROGEN 2003. Ed: G. Bugada, J.A. Desideri, J. Periaux, M. Schoenauer and G. Winter. CIMNE, Barcelona, España.
369. [1.12] G. Winter, J. Periaux, M. Galán, P. Cuesta (Eds.) '*Genetic Algorithms in Engineering and Computer Science*', John Wiley & Sons, (1995).
370. [4.51] D.H. Wolpert, W.G. MacReady, "No Free Lunch Theorems for Optimization", *IEEE Transactions on Evolutionary Computation*, 4 (1997) pp. 67-82.
371. [4.52] D.H. Wolpert, W.G. MacReady, "No Free Lunch Theorems for Search", Technical Report SFI-TR-95-02-010, Santa Fe Institute, July 1995.

372. [3.216] J. Wu, S. Azarm, "On a new Constraint Handling Technique for Multi-Objective Genetic Algorithms", *Proceedings of the Genetic and Evolutionary Computation Conference* (2001), pp. 47-50.
373. [3.217] S-J. Wu, P-T. Chow, "Steady-state genetic algorithms for discrete optimization of trusses", *Computers & Structures* 56 (1995) 979-991.
374. [3.218] Y.M. Xie, G.P. Steven, "A simple evolutionary procedure for structural optimization", *Computers & Structures* 49 (1993) 885-896.
375. [3.219] Y.G. Xu, G.R. Liu, "Detection of flaws in composites from scattered elastic-wave field using an improved nGA and a local optimizer", *Comput. Methods Appl. Mech Engrg*, 191 (2002) 3929-3946.
376. [3.220] Y-P. Yang, Y-A. Chen, "Multiobjective optimization of hard disk suspension assemblies: Part II - integrated structure and control design", *Computers & Structures* 59 (1996) 771-782.
377. [3.221] Y-P. Yang, C-C. Kuo, "Passive and active design of hard disk suspension assemblies using multiobjective optimization techniques", *Comput. Methods Appl. Mech Engrg*, 145 (1997) 147-166.
378. [3.222] Y. Yang, C.K. Soh, "Automated optimum design of structures using genetic programming", *Computers & Structures* 80 (2002) 1537-1546.
379. [3.223] J.P. Yang, C.K. Soh, "Structural optimization by genetic algorithms with tournament selection", *Journal of Computing in Civil Engineering*, 11-3 (1997) 195-200.
380. [3.224] I.C. Yeh, "Hybrid Genetic Algorithms for Optimization of Truss Structures", *Microcomputers in Civil Engineering*, 14-3 (1999) pp. 199-206.
381. [3.225] C-H. Yeh, J.W. Labadie, "Multiobjective Watershed-Level Planning of Storm-Water Detention Systems", *Journal of Water Resources Planning and Management*, 123-6 (1997) 336-343.
382. [3.226] W.H. Zhang, "A compromise programming method using multibounds formulation and dual approach for multicriteria structural optimization", *International Journal for Numerical Methods in Engineering*, 58 (2003) 661-678.
383. [3.227] W.H. Zhang, "On the Pareto optimum sensitivity analysis in multicriteria optimization", *International Journal for Numerical Methods in Engineering*, 58 (2003) 955-977.
384. [3.228] W.H. Zhang, M. Domaszewski, C. Fleury, "An improving weighting method with multibounds formulation and convex programming for multicriteria structural optimization", *International Journal for Numerical Methods in Engineering*, 52 (2001) 882-902.
385. [3.229] W. Zhang, H. Yang, "A study of the weighting method for a certain type of multicriteria optimization problem", *Computers & Structures* 79 (2001) 2741-2749.
386. [2.73] M. Zeleny, "*Linear Multiobjective Programming*", Springer (1974) pp. 197-220.
387. [2.74][4.55][5.57] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications", *phD Tesis*, Institut für Technische Informatik und Kommunikationsnetze, Swiss Federal Institute of Technology, Zürich (1999).
388. [2.75] E. Zitzler, K. Deb, L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", *Evolutionary Computation*, 8-2 (2000) pp. 173-195.

389. [1.13][2.76][4.54][5.58] E. Zitzler, M. Laumanns, L. Thiele "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization", *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN 2001)* Eds: KC. Giannakoglou, DT. Tsahalis, J. Periaux, KD. Papailiou, T. Fogarty . John Wiley & Sons.
390. [2.77][4.53][5.59] E. Zitzler, M. Laumanns, L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", *TIK-Report 103*, (May 2001) Swiss Federal Institute of Technology (ETH) Zurich.
391. [2.78] E. Zitzler, M. Laumanns, L. Thiele, C. Fonseca, V. Grunert da Fonseca, "Why quality assessment of multiobjective optimizers is difficult", *Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann (2002) pp. 666-673.
392. [2.79] E. Zitzler, L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the SPEA. In *IEEE Transactions on Evolutionary Computation*, 3-4 (1999) pp. 257-271.
393. [2.80] E. Zitzler, L. Thiele. "An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach". *TIK-Report 43*, Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology, Switzerland (1998).
394. [2.81] E. Zitzler, L. Thiele. "Multiobjective Optimization using evolutionary algorithms - a comparative case study". *Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, Springer (1998) pp. 292-301.
395. [1.14] <http://ceani.ulpgc.es/ingenetcd> (consultado en fecha de Febrero 2005)