UNIVERSIDAD DE LAS PALMAS DE
GRAN CANARIA

PROGRAMA DE DOCTORADO
EN TECNOLOGÍAS DE TELECOMUNICACIÓN E
INGENIERÍA COMPUTACIONAL

TESIS DOCTORAL

CONTRIBUTIONS ON RELIABILITY TO RECONDITION THE MAINTENANCE PLAN AND THE DESIGN OF SYSTEMS BY USING EVOLUTIONARY ALGORITHMS

Andrés Cacereño Ibáñez
2022

ULPGC
Universidad de
Las Palmas de
Gran Canaria | Escuela de
Doctorado

D. JUAN JOSÉ AZNÁREZ GONZÁLEZ, COORDINADOR DEL PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE TELECOMUNICACIÓN E INGENIERÍA COMPUTACIONAL DE LA UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

INFORMA,

De que la Comisión Académica del Programa de Doctorado, en su sesión de fecha .... tomó el acuerdo de dar el consentimiento para su tramitación, a la tesis doctoral titulada "CONTRIBUTIONS ON RELIABILITY TO RECONDITION THE MAINTENANCE PLAN AND THE DESIGN OF SYSTEMS BY USING EVOLUTIONARY ALGORITHMS" presentada por el doctorando D. Andrés Cacereño Ibáñez y dirigida por los Doctores D. David Juan Greiner Sánchez y D. Blas José Galván González.

Y para que así conste, y a efectos de lo previsto en el Artº 11 del Reglamento de Estudios de Doctorado (BOULPGC 04/03/2019) de la Universidad de Las Palmas de Gran Canaria, firmo la presente en Las Palmas de Gran Canaria, a ……. de...de dos mil veintidós.

ULPGC Universidad de Las Palmas de Gran Canaria | Escuela de Doctorado

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

ESCUELA DE DOCTORADO

Programa de doctorado en Tecnologías de Telecomunicación e Ingeniería Computacional

Título de la Tesis: CONTRIBUTIONS ON RELIABILITY TO RECONDITION THE MAINTENANCE PLAN AND THE DESIGN OF SYSTEMS BY USING EVOLUTIONARY ALGORITHMS

Tesis Doctoral presentada por D. Andrés Cacereño Ibáñez

Dirigida por el Dr. D. David Juan Greiner Sánchez

Codirigida por el Dr. D. Blas José Galván González

Las Palmas de Gran Canaria, a……..de……..……..de 2022

El Director,	El Codirector,	El Doctorando,
GREINER SANCHEZ DAVID JUAN - 42877798W Firmado digitalmente por GREINER SANCHEZ DAVID JUAN - 42877798W Fecha: 2022.06.22 12:13:20 +01'00'	43256315P BLAS JOSE GALVAN (C:Q3570001B) Firmado digitalmente por 43256315P BLAS JOSE GALVAN (C:Q3570001B) Fecha: 2022.06.23 11:12:19 +02'00'	CACEREÑO IBAÑEZ ANDRES - 44321817N Firmado digitalmente por CACEREÑO IBAÑEZ ANDRES - 44321817N Fecha: 2022.06.22 12:06:25 +01'00'
(firma)	(firma)	(firma)

AGRADECIMIENTOS

Tiempo atrás llamé a la puerta del despacho del por entonces profesor titular de universidad Don Blas José Galván González, con quien había concertado una cita con el propósito de realizar el proyecto de fin de carrera que me permitiera adquirir el título de Ingeniero Técnico en Telecomunicaciones e iniciar mi carrera profesional fuera del ámbito universitario. Veinte años después, con la experiencia que aporta el haber superado incontables retos tanto en lo profesional, como en lo académico, y siempre en el ámbito de la Universidad de Las Palmas de Gran Canaria, me encuentro escribiendo estas líneas, en aras de depositar (para su posterior defensa) el presente documento de tesis doctoral. Podría sonar a tópico, pero la realidad es que hay personas cuya influencia cambia la vida, y este es el caso de Blas.

Han sido veinte años en los que he compartido multitud de experiencias, empezando por mis propios compañeros y compañeras en el ámbito de la universidad, y terminando por las personas que han formado y forman parte de multitud de empresas y administraciones públicas, tanto en el ámbito nacional como internacional. Todas y cada una de estas personas, en mayor o menor medida, han puesto su grano de arena en la consecución de esta tesis doctoral, si bien quisiera mencionar en particular al catedrático de universidad Don Gabriel Winter Althaus, al profesor titular de universidad Don Ricardo Aguasca Colomo y a mi director de tesis y profesor titular de universidad Don David Juan Greiner Sánchez.

Si importante ha sido el soporte profesional en el que me he encontrado integrado bajo el paraguas de la División de Computación Evolutiva y Aplicaciones (CEANI) del Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI) de la Universidad de Las Palmas de Gran Canaria, no lo es menos el de la familia y amigos. Mis padres soñaron con la mejor educación posible para sus hijos, y si bien trabajamos duro entre todos para hacer realidad tal sueño, considero que con esta tesis doctoral culminamos tal anhelo, al lograr este miembro de la familia el más alto grado académico alcanzable. Esto no habría sido posible sin el apoyo de la Universidad de Las Palmas de Gran Canaria, a través del contrato predoctoral en formación del que he disfrutado (Julio 2018 – Junio 2022).

RESUMEN.

Mediante la presente investigación se presenta una metodología para la optimización simultánea del diseño estructural de sistemas técnicos y de su estrategia de mantenimiento, para la cual se considera tanto el mantenimiento correctivo como el preventivo. En el ámbito de la investigación, el mantenimiento preventivo consiste en determinar el periodo óptimo para llevar a cabo las correspondientes tareas de mantenimiento, considerando los dispositivos previamente seleccionados para formar parte del diseño estructural del sistema.

La citada metodología se basa en el acople de Algoritmos Evolutivos Multiobjetivo y la Simulación por Eventos Discretos. Por tanto, se requiere de un concienzudo estudio en relación con la resolución de problemas complejos en el ámbito de la ingeniería. Además, la Simulación por Eventos Discretos implica el uso de un enfoque en base a simulación, por lo que tal técnica debe ser analizada en detalle.

Un minucioso estudio sobre la convergencia de las soluciones que el uso de la metodología aporta debe ser elaborado. Con tal objetivo, la metodología es aplicada a un caso de estudio a lo largo de toda la investigación. Diversos Algoritmos Evolutivos Multiobjetivo pertenecientes al estado del arte son aplicados, de modo que sus parámetros son estudiados siguiendo un método en cascada, de modo que:

- En primer lugar, un conjunto de Algoritmos Evolutivos Multiobjetivo es escogido atendiendo al estado del arte, clasificados según el mecanismo empleado para selección de individuos. Tal criterio de clasificación es comúnmente empleado para diferenciar tales algoritmos. Además, sus parámetros son meticulosamente explorados con el propósito de identificar las configuraciones que presentan un mejor desempeño.
- A continuación, deben considerarse diversas codificaciones y niveles de precisión en relación con los tiempos hasta iniciar una tarea de mantenimiento preventivo. La codificación presenta un impacto directo sobre el tamaño del cromosoma, pudiendo tener efecto en el desempeño sobre los algoritmos empleados. Por otra parte, el impacto de los niveles de precisión en relación con los tiempos hasta iniciar una tarea de

mantenimiento preventivo puede aportar información interesante desde el punto de vista práctico. En concreto, horas, días y semanas son consideradas como unidades temporales para la planificación del mantenimiento preventivo. La flexibilidad en relación con la precisión puede resultar de interés para los tomadores de decisiones.

- Finalmente, se estudia la multiobjetivización como técnica para el manejo de las funciones objetivo, con el propósito de determinar la idoneidad del uso de la citada técnica.

Con el propósito de testar la escalabilidad y generalización de la metodología, esta es aplicada a sistemas de mayor complejidad en lo que al número de alternativas de diseño se refiere. El caso de estudio es un sistema hidráulico formado por un máximo de 7 dispositivos. El uso de la metodología se extiende con el propósito de estudiar, en primer lugar, un sistema hidráulico que dobla el número de dispositivos al considerar una segunda rama en paralelo, y, en segundo lugar, un sistema aún mayor en el que el número de dispositivos a tomar en cuanta varía entre 6 y 36 en función de las alternativas de diseño. Además, el uso de la metodología se extiende al aplicarla y testarla en otro campo de la ingeniería, con el propósito de aportar soluciones fiables para redes de comunicación para subestaciones eléctricas.

Con el propósito de analizar los resultados obtenidos, el desempeño de las diferentes configuraciones es concienzudamente comparado al usar una métrica específica (el hipervolumen) y test estadísticos de significancia. Por tanto, como objetivo secundario de la presente investigación subyace el uso e interpretación de los resultados obtenidos.

A continuación, se exponen las conclusiones alcanzadas tras el desarrollo de la investigación, en la que, como se indicó anteriormente, se acoplan Algoritmos Evolutivos Multiobjetivo y Simulación por Eventos Discretos, con el propósito de la optimización simultánea del diseño estructural de sistemas técnicos (en base a la selección automática del diseño y dispositivos redundantes) y de la estrategia de mantenimiento (basada en la determinación del periodo óptimo para realizar tareas de mantenimiento preventivo). Disponibilidad y Coste son los objetivos en conflicto a considerar. El acople de ambas técnicas fue previamente considerado por diversos autores para la resolución de ambos problemas por separado, pero nunca considerándolos conjuntamente, con el propósito de determinar el periodo óptimo para el inicio de las correspondientes tareas de mantenimiento preventivo. El Algoritmo Evolutivo Multiobjetivo aporta una población de individuos, cada uno de los mismos codificando una alternativa de diseño y su correspondiente estrategia de mantenimiento.

Cada individuo representa una posible solución al problema, en la cual coexisten tanto las variables de decisión con relación al diseño estructural como las variables de decisión con respecto a la estrategia de mantenimiento. Tal idea implica la consideración a lo largo del proceso evolutivo de variables de decisión de diferente naturaleza, por lo que diversas transformaciones deben ser consideradas. Por una parte, se empleó codificación real cuando coexistieron variables de decisión de naturaleza binaria y entera para el diseño, y variables de decisión de naturaleza entera para la estrategia de mantenimiento. Por otra parte, se empleó codificación binaria cuando coexistieron variables de decisión de naturaleza binaria para el diseño, y variables de decisión de naturaleza entera para la estrategia de mantenimiento. En todos los casos, las transformaciones precisas cumplieron su función perfectamente.

Una vez el Algoritmo Evolutivo Multiobjetivo suministra una población de individuos, estos son usados para modificar y evaluar el Perfil de Funcionabilidad del Sistema mediante Simulación por Eventos Discretos. Los individuos evolucionan generación tras generación hasta alcanzar el criterio de parada.

Este proceso fue en primer lugar aplicado a un sistema técnico tomado como caso de estudio, para el que se aplicaron cinco Algoritmos Evolutivos Multiobjetivo pertenecientes al estado del arte (SMS-EMOA, MOEA/D, MOEA/D-DE, NSGA-II y GDE3) y su desempeño fue rigurosamente comparado. Finalmente, se obtuvo un conjunto de soluciones óptimas no dominadas.

Como conclusión a este estudio, cabe poner de manifiesto que el uso de Algoritmos Evolutivos Multiobjetivo acoplados a Simulación por Eventos Discretos para la optimización conjunta del diseño de sistemas y su estrategia de mantenimiento aporta soluciones de equilibrio Disponibilidad-Coste a problemas del mundo real, en los que se empleó información basada en experiencia de campo. Además, en relación con los Algoritmos Evolutivos Multiobjetivo empleados para la resolución del caso de estudio, los métodos que emplean como criterio para guiar la búsqueda de soluciones el indicador Hipervolumen (SMS-EMOA) y la dominancia de Pareto (NSGA-II y GDE3) presentaron mejor desempeño que los métodos basados en la descomposición del espacio de búsqueda (MOEA/D y MOEA/DDE). Sin embargo, el operador empleado para la creación de nuevos individuos no parece tener un efecto relevante, dado que los métodos que emplean Cruce Binario Simulado (SMS-EMOA y NSGA-II) presentaron un desempeño similar al método que emplea Evolución Diferencial (GDE3).

Una vez resuelto el caso de estudio mediante el uso de los diversos Algoritmos Evolutivos Multiobjetivo, las configuraciones que mostraron mejor desempeño se seleccionaron con el propósito de profundizar en el análisis. Por una parte, se analizó el efecto del tamaño de la muestra y su dirección extrema mínima, lo cual pone de manifiesto los beneficios del empleo de la metodología propuesta. Se demuestra la positiva sinergia del acople de Algoritmos Evolutivos Multiobjetivo y Simulación por Eventos Discretos cuando una única simulación por individuo de la población es empleada para la evaluación de las funciones objetivo, al aportar resultados competitivos. Tales resultados se confirman al contrastar con un análisis en base a valores promedio obtenidos atendiendo a ambas funciones objetivo para cada solución no dominada aportada por las diversas configuraciones. La metodología propuesta resulta ser un enfoque computacionalmente eficiente y robusto (no dependiente de los parámetros en relación con el número de muestras o la dirección de búsqueda para la minimización) en comparación con el uso de Simulación Monte Carlo, cuando se pretende resolver un problema de optimización multiobjetivo en el campo de la fiabilidad. Por otra parte, para el caso de estudio, el beneficio económico a partir del empleo de la metodología propuesta para la optimización conjunta del diseño estructural del sistema y su estrategia de mantenimiento se cuantifica en el intervalo de entre el 4 y el 10 por ciento.

Con el propósito de demostrar la escalabilidad y generalización de la metodología propuesta, esta fue aplicada a dos sistemas de mayor complejidad.

Ambos problemas se resolvieron de manera satisfactoria. Además, se realizó un estudio sobre el impacto en la modificación del tamaño del cromosoma para el sistema de mayor complejidad, para el que el conjunto de soluciones no dominadas aportadas cuando las variables de decisión del diseño presentan naturaleza binaria (cromosoma largo) presentó un mayor valor de hipervolumen que cuando las variables de decisión del diseño presentan naturaleza entera (cromosoma corto).

Una vez definida, desarrollada, implementada y testada la metodología tanto para el caso de estudio como para sistemas de mayor complejidad, un estudio más profundo fue llevado a cabo. Tal estudio abarcó, en primer lugar, un experimento en relación con la codificación del problema para la comparación del desempeño de siete tipos de codificación (real, binaria con cruce de un punto, binaria con cruce de dos puntos, binaria con cruce uniforme, código Gray con cruce de un punto, código Gray con cruce de dos puntos y código Gray con cruce uniforme). En segundo lugar, el estudio consistió en un experimento en relación con la precisión, mediante la comparación del desempeño al usar codificación binaria con tres niveles de precisión para un rango de unidades temporales (la hora, el día y la semana), lo cual presenta un impacto en el tamaño del cromosoma (cuanto menor la unidad temporal, mayor el tamaño del cromosoma). En este caso, se empleó como Algoritmo Evolutivo Multiobjetivo el NSGA-II, ya que tal método resultó muy competitivo en el estudio elaborado previamente. Un conjunto de soluciones óptimas no dominadas fue obtenido en ambos experimentos, empleando el caso de estudio definido previamente como base para el desarrollo de este estudio.

Con respecto al experimento sobre codificaciones, la codificación binaria con cruce de dos puntos resultó mejor ordenada cuando se aplicó la prueba de Friedman (en base a la distribución final aportada por el indicador Hipervolumen), aunque no se apreciaron diferencias estadísticamente significativas. En cuanto al experimento sobre precisión, la codificación binaria con cruce de dos puntos y la hora como unidad temporal resultó mejor ordenada cuando se aplicó la prueba de Friedman, aunque no se apreciaron diferencias estadísticamente significativas. Una interesante conclusión emerge a partir de este segundo experimento, la cual incide en la flexibilidad sobre la unidad temporal a emplear para planificar las tareas de mantenimiento preventivo. El uso de la hora, el día o la semana como unidad temporal no presenta efecto significativo en el desempeño, por lo que, bajo las condiciones de estudio, las tareas de mantenimiento preventivo pueden ser planificadas usando la semana como unidad temporal. Ello permite un mejor rango de tiempo para planificar en comparación al uso del día o la hora como unidad temporal.

Una vez estudiados los diversos Algoritmos Evolutivos Multiobjetivo, las codificaciones y los niveles de precisión, el caso de estudio previamente estudiado vuelve a ser explorado, pero en esta ocasión se consideraron dos enfoques desde el punto de vista de la optimización multiobjetivo. Por una parte, un enfoque que atiende a dos objetivos, Disponibilidad y Coste. Por otra parte, un enfoque que atiende a tres objetivos, donde nuevamente, Disponibilidad y Coste son los objetivos a evaluar. Sin embargo, en este caso, el Coste es descompuesto en Coste de Adquisición y Coste de Operación, atendiendo a un enfoque de multiobjetivización.

Con el propósito de identificar el enfoque de mejor desempeño, se lleva a cabo una concienzuda prueba de hipótesis. Esta consiste en comparar el desempeño de dos Algoritmos Evolutivos Multiobjetivo (SMS-EMOA y NSGA-II), cuando se consideran diversas configuraciones de los mismos. Se emplean en este caso codificación binaria y real. Cuando el enfoque de dos objetivos fue estudiado, no se encontraron diferencias significativas entre las configuraciones empleadas. Sin embargo, la configuración que obtuvo un mejor orden cuando se aplicó la prueba de Friedman se obtuvo cuando se empleó SMS-EMOA con codificación binaria y un gen por cromosoma como probabilidad de mutación. Cuando el enfoque de tres objetivos fue estudiado, tampoco se encontraron diferencias significativas entre las configuraciones empleadas. Sin embargo, la configuración que obtuvo un mejor orden cuando se aplicó la prueba de Friedman se obtuvo cuando se empleó NSGA-II con codificación real y 1.5 genes por cromosoma como probabilidad de mutación.

A continuación, fueron comparadas las configuraciones que obtuvieron un mejor orden cuando se aplicó la prueba de Friedman bajo ambos enfoques. En esta ocasión, se encontraron diferencias significativas, por lo que es posible concluir que se obtuvo un mejor desempeño cuando se aplicó la multiobjetivización. Además, NSGA-II con codificación real y 1.5 genes por cromosoma como probabilidad de mutación resultó mejor ordenado cuando se aplicó la prueba de Friedman, aportando a su vez el valor más alto para el indicador hipervolumen en cuanto al promedio, mediano y máximo. Por tanto, tal configuración y enfoque puede ser recomendado para resolver el problema de la optimización conjunta del diseño y la estrategia de mantenimiento, la cual aporta el mejor balance Disponibilidad-Coste.

Finalmente, la metodología propuesta se aplicó a un campo diferente de la ingeniería, con el propósito de suministrar diseños estructurales fiables para redes de comunicación para subestaciones eléctricas. Un caso de estudio específico fue explorado, para el cual se emplearon las conclusiones alcanzadas previamente como parámetros de configuración de los Algoritmos Evolutivos Multiobjetivo. Nuevamente, se atendió a dos enfoques en relación con la optimización multiobjetivo. Por una parte, un enfoque bajo dos objetivos, donde se consideraron Disponibilidad y Coste. Por otra parte, un enfoque bajo tres objetivos, donde se consideraron Disponibilidad y Coste, si bien este último fue descompuesto en Coste de Adquisición y de Operación (multiobjetivización). La aplicación de la metodología permite obtener soluciones de compromiso entre objetivos. Con el propósito de localizar la configuración de mejor desempeño, se realizó una concienzuda prueba de hipótesis.

El proceso se aplicó a una sección de un subsistema que sigue el estándar IEC61850. Dicho subsistema es una bahía de línea de bus sencillo, de una pequeña subestación para la transformación de energía de 220 kV a 132 kV. De nuevo, se comparó el desempeño de dos Algoritmos Evolutivos Multiobjetivo (SMS-EMOA y NSGA-II), para lo cual se emplearon diversas configuraciones de tales métodos.

En este caso, no se encontraron diferencias estadísticamente significativas cuando se empleó el enfoque de dos objetivos. Sin embargo, la configuración mejor ordenada según la prueba de Friedman se obtuvo cuando se empleó NSGA-II con codificación real y un gen por cromosoma como probabilidad de mutación. Cuando se empleó el enfoque de tres objetivos, se encontraron diferencias estadísticamente significativas. En este caso, el mejor desempeño se obtuvo cuando se empleó SMS-EMOA con codificación real y un gen por cromosoma como probabilidad de mutación. Las configuraciones mejor ordenadas se escogieron para su comparación, no encontrándose diferencias estadísticamente significativas en el desempeño en este caso. Sin embargo, la configuración mejor ordenada según la prueba de Friedman se obtuvo cuando se empleó la multiobjetivización (enfoque bajo tres objetivos). Por tanto, cabe concluir que la metodología aplicada es robusta al no darse diferencia estadísticamente significativa entre ambos enfoques. Sin embargo, un efecto positivo se pone de manifiesto al emplear la multiobjetivización al resultar mejor ordenado desde el punto de vista de la Prueba de Friedman. En este caso, SMS-EMOA con codificación real y un gen por cromosoma como probabilidad de mutación fue la configuración que obtuvo un mejor orden desde el punto de vista de la prueba de Friedman. Además, obtuvo el mejor valor promedio, mediano, máximo y mínimo en cuanto al hipervolumen. Por tanto, tal configuración podría ser recomendada para la resolución de un problema de este tipo.

Como se ha demostrado, la metodología propuesta es extensible a otros campos de la ingeniería para la resolución de problemas en los que se optimiza de manera conjunta el diseño y la estrategia de mantenimiento de sistemas.

En relación con el futuro de la presente investigación, diversas líneas quedan abiertas:

- La casuística estudiada a lo largo de la presente investigación en relación con la Fiabilidad del sistema está claramente definida y delimitada. Esta considera los siguientes aspectos, los cuales podrían ser extendidos tal y como se comenta a continuación.
  - Se consideran dos estados para el sistema, siendo estos el de operación o fallo. Mediante la atención a estados de deterioro, podrían ser considerados sistemas multi-estado.
  - Tras las reparaciones o actividades de mantenimiento correctivo, todos los dispositivos recuperan su estado tan-bueno-como-nuevo. Podrían considerarse reparaciones imperfectas.
  - Se han considerado redundancias activas, por lo que el sistema satisface la función requerida mientras las oportunas redundancias

    funcionen correctamente. Podrían considerarse otras redundancias tales como frías, cálidas o en estado de espera.
    - Cada dispositivo se considera como una unidad simple desde el punto de vista del mantenimiento. Un dispositivo de este tipo no puede ser descompuesto en menores niveles mantenibles. Podrían considerarse dispositivos de unidades múltiples en futuras investigaciones, así como múltiples modos de fallo.
    - Las actividades de mantenimiento preventivo se planifican en función del tiempo. Tales actividades podrían planificarse en función de la edad, uso o condición.
    - Las tareas de mantenimiento preventivo comienzan inmediatamente, una vez un dispositivo no satisface la función requerida, por lo que se considera la monitorización continua. Tal circunstancia podría no ser considerada por lo que podría atenderse al testeo del estado del dispositivo con el propósito de iniciar las correspondientes tareas de mantenimiento.
    - No se consideran dependencias entre dispositivos, por lo que estos operan de manera aislada. Podrían considerarse dependencias entre dispositivos.
- En lo que a los Algoritmos Evolutivos Multiobjetivo hace referencia, se emplearon diversos métodos pertenecientes al estado del arte a lo largo de la presente investigación. Estos se consideran métodos estándar. Métodos más modernos podrían ser empleados con el propósito de analizar y comparar su desempeño. Además, cuando se exploró el enfoque bajo multiobjetivización, cuanto más complejo el sistema, se observó mayor diferencia estadísticamente significativa en el desempeño. Esta circunstancia podría ser explorada más profundamente al aplicar la metodología a diseños estructurales más complejos.

Estos son algunos ejemplos sobre la dirección que podrían tomar futuras investigaciones. Como se observa, quedan abiertas diversas líneas de estudio.

INDEX.

FIGURES

TABLES

0. CHAPTER 0: INTRODUCTION AND OBJECTIVES.

0.1. Presentation of the research.

In the reliability field, several studies have been developed at *Centro de Aplicaciones Numéricas en Ingeniería* (CEANI) of *Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería* (SIANI) of *Universidad de Las Palmas de Gran Canaria* (ULPGC) since Professor Blas José Galván González presented his doctoral thesis in 1999 [1]. This was the main stone in a reliability research line which has been kept till recent days [2 - 4].

Furthermore, optimisation in the reliability field, and in special by using Evolutionary Algorithms in order to solve engineering problems, has been widely faced in our research institute [5 - 10]. For the present research, has been considered such a trajectory and research line in order to continue with the development of the knowledge.

From the Reliability of technical systems' point of view, Doctor K. Misra defined such a concept (Reliability) as "the probability of failure-free operation under specified conditions over an intended period of time" [11]. This definition leads to interest in the time taken for a system to fail (Time to Failure), which is a continuous random variable that can be represented by a continuous probability distribution. Therefore, the reliability concept does not allow studying events that occur after failure, so this is a concept regarding non-reparable elements. If the behaviour of reparable elements wants to be modelled, it will be necessary to refer to the Availability concept. The Doctors J. Andrews and T. Moss defined such a concept in three different senses [12]:

- In relation to standby systems which are required to function on demand (e.g., safety protection systems),
- In relation to the probability that systems are working at a particular time (e.g., continuously operating systems whose failure is revealed once it occurs and the repair process starts immediately),

- In relation to the fraction of the total time in which systems are available to perform their required function.

The first definition is suitable for standby systems, whose function is required on demand. An example of this kind of systems are the protection systems. Trying to prevent dangerous situations, the protection system monitors if an undesirable event takes place. If such a protection system is in the failure state when its function is demanded, this will be unavailable. Since the demand can occur at any time, the more time the system operates, the better both its probability of working and availability. Another example lies in redundant systems. This kind of systems keeps on the standby state while the main system is operating. When the main system fails, the redundant system operation is demanded. In case of failure of the redundant system, such a failure is revealed when its operation is demanded. Hence, the repair procedure starts when the failure is detected.

The second definition is appropriate for continuous operating systems, whose failure is disclosed at the moment. In this case, the repair procedure starts immediately.

The third definition is interesting to estimate the performance of a process because it is possible to compute the total production from the fraction of time in which the system is satisfactorily operating. This definition claims about the possibility of systems of being in the operating state at different time intervals all along its life cycle or mission time. This is the reason why it is possible to model not only the time to failure but also the time to repair by employing the Availability concept. The cycle failure-repair is repeated up to complete the mission time. The probability of non-operating for a system is represented by the concept Unavailability and it can be computed as Unavailability = 1 - Availability.

It is possible to establish a direct relation between Reliability and Availability. Using the Availability concept is possible to model the complete life cycle for a reparable element. However, using the Reliability concept, it is possible to model the life of a reparable element up to the first failure occurs [13]. Previously, the relationship between Reliability and Availability regarding time was claimed. The interest in the time to failure has already been mentioned regarding Reliability, however, when referring to Availability, the interest in the time encompasses not only the time to failure but also the time taken to repair the system (Time to Repair) and to recover the operating state.

Working with reparable elements implies the main objective of achieving maximum Availability. When a repairable system is not available, the system enters an unproductive phase. Throughout this, not only are resources not generated, but also, they are consumed until the system is brought back to the available state. There are many strategies to improve the availability of reparable elements. This research pays attention to two particular strategies: Including redundant devices and determining the time period to carry out preventive maintenance tasks regarding the devices including in the system design.

On the one hand, a redundancy is a component added to a subsystem from a series-parallel configuration in order to increase the number of alternative paths [14]. Including alternative paths by using redundant devices makes possible that the system keeps on working although the main device has failed. Moreover, including redundant devices in systems requires a modification of the design.

On the other hand, as it was explained above, when a reparable element is available it keeps on a productive phase. A reparable element may not be available because of a failure or a scheduled shutdown regarding a preventive maintenance activity. After a failure, the reparable element needs a time to repair and to recover the available state. However, after a scheduled shutdown in order to perform a preventive maintenance activity, the reparable element needs a time to develop such an activity and to recover the available state. Speaking in general, the time needed to perform a preventive maintenance activity is considered smaller than the time needed to repair an element (for reasons such as willing and trained human personnel, or available spare parts). Therefore, it is important to identify the optimum moment in which the system must be stopped, and a preventive maintenance activity must be conducted. This should ideally be done before the occurrence of the failure but as close as possible to such a failure in order to maximise the total system available time.

When projecting and building new industrial facilities, getting integrated design alternatives and maintenance strategies are of critical importance to achieve the physical assets optimal performance, which is needed to be competitive in the actual global markets.

The present research explores the problem of achieving simultaneously design alternatives and maintenance strategies regarding such design alternatives, which optimise the performance of the physical assets. This is done from a multi-objective optimisation point of view by employing Multi-objective Evolutionary Algorithms. Optimisation is particularly useful when the number of potential solutions is high and achieving the best solution is very difficult. Instead of the best solution, some sufficiently good solutions can be obtained [15]. By this way, interesting information is supplied to the decision makers, who will have to decide what the preferable design is, taking into account their requirements.

Furthermore, when the behaviour of a system wants to be emulated, the Discrete Event Simulation arises like a power tool to model complex systems, which can be analysed much more accurately due to a more realistic representation of their behaviour in practice.

0.2. Objectives.

For all above, the main objectives of the present research are summarising as follows:

- A methodology to the simultaneous optimisation of systems' structural design and their maintenance strategy pretends to be defined, implemented, and validated. Not only corrective maintenance but also preventive maintenance -by determining the periodic time to conduct the preventive maintenance tasks for the devices included in the system's design- must be considered for the maintenance strategy.

- Such a methodology will couple Multi-objective Evolutionary Algorithms and Discrete Event Simulation. Therefore, a thoroughly study about Multi-objective Evolutionary Algorithms to solve complex problems in the engineering field must be conducted. Moreover, since a simulation approach will be employed, such a technique must be analysed in detail.
- A thoroughly study regarding the convergence of the solutions that the methodology supplies must be conducted. To do that, the methodology will be applied to a case study, which will be explored along the research. Several Multi-objective Evolutionary Algorithms will be applied, and their parameters will be studied by following a cascade method where:
    - Firstly, attending to the state-of-the-art, a set of Multi-objective Evolutionary Algorithms will be chosen by considering their mechanism to create new individual. This is due to the fact that such a criterion is commonly used to classify such algorithms. Moreover, in order to find out the configurations that present a better performance, the main parameters of the algorithms must be meticulously explored.
    - Secondly, the encodings and accuracy levels regarding times to start a preventive maintenance task must be considered. The encoding has a direct impact in the size of the chromosome, and it may have an effect in the performance of the algorithms that must be explored. Moreover, the impact of the accuracy levels regarding times to start a preventive maintenance task can reveal information from a practical point of view. Concretely, the hour, the day and the week will be considered as time units for planning preventive maintenance tasks. The flexibility regarding the time interval to conduct a preventive maintenance task can be helpful to the decision makers.
    - Finally, a technique regarding how to deal with the objective functions is attended to achieve a better performance. It is the case of the multi-objectivisation technique, which is considered for the present research.

- In order to test the scalability and generalisation of the methodology, such a methodology will be applied to two more complex systems. The case study is a hydraulic system formed by a maximum of 7 devices. The methodology will be applied to consider a hydraulic system which doubles this number of devices on the and hand, and a bigger hydraulic system with a number of devices between 6 and 36 devices on the other hand. Moreover, extending the use of the methodology is followed, so this will be applied and tested in a different engineering field, in order to supply reliable architectural designs of Substation Communication Networks.
- To analyse the achieved results, in all cases above mentioned, the performances are compared in detail by using a specific metric (the hypervolume) and statistical significance tests (including post-hoc analysis). Therefore, achieving the knowledge regarding how to use and to interpretate the outcomes of applying such techniques is required as a secondary objective of the research.

0.3. Derived publications from the research.

As a result of the conducted research, several publications and presentations were developed. They are described as follow:

0.3.1. Indexed JCR contributions.

- Cacereño A, Greiner D, Galván BJ. Multi-Objective Optimum Design and Maintenance of Safety Systems: An In-Depth Comparison Study Including Encoding and Scheduling Aspects with NSGA-II. Mathematics. 2021; 9(15):1751. https://doi.org/10.3390/math9151751.

0.3.2. Book chapters.

- Cacereño A., Galván B., Greiner D. (2021) Solving Multi-objective Optimal Design and Maintenance for Systems Based on Calendar Times Using

NSGA-II. In: Gaspar-Cunha A., Periaux J., Giannakoglou K.C., Gauger N.R., Quagliarella D., Greiner D. (eds) Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences. Computational Methods in Applied Sciences, vol 55. Springer, Cham. https://doi.org/10.1007/978-3-030-57422-2_16.

- Cacereño A., Greiner D., Galván B. (2021) Multi-Objective Optimal Design and Maintenance for Systems Based on Calendar Times Using MOEA/D-DE. In: Vasile M., Quagliarella D. (eds) Advances in Uncertainty Quantification and Optimization Under Uncertainty with Aerospace Applications. UQOP 2020. Space Technology Proceedings, vol 8. Springer, Cham. https://doi.org/10.1007/978-3-030-80542-5_5.

0.3.3. Conferences.

- A. Cacereño, B. Galván, D. Greiner. Multiobjective Optimal Design and Maintenance for Systems based on Calendar Times. *13th International Conference on Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems*, EUROGEN 2019. Guimarães, Portugal (2019).

- A. Cacereño, B. Galván, D. Greiner. Multiobjective Optimal Design and Maintenance for Systems based on Calendar Times using MOEA/D-DE. *International Conference on Uncertainty Quantification & Optimisation*, UQOP 2020. Brussels, Belgium (2020).

- A. Cacereño, B. Galván, D. Greiner. Multiobjective Optimal Design and Maintenance for Systems based on Calendar Times using GDE3. *14th World Congress on Computational Mechanics (WCCM XIV) and 8th European Congress on Computational Methods in Applied Sciences and Engineering - ECCOMAS 2020*. Paris, France (2021).

- A. Cacereño, D. Greiner, B. Galván. Optimización del Diseño y la Estrategia de Mantenimiento de Sistemas atendiendo a 3 objetivos: Indisponibilidad y Costes de Adquisición y Operación. Congreso de Métodos Numéricos en Ingeniería, 12-14 septiembre 2022. Las Palmas de Gran Canaria, España (2022) (Accepted).

0.4. Structure of the document.

The structure of the thesis document is organised as follows: Chapter 1 introduces some basic concepts on Reliability. Chapter 2 deals with Multi-objective Optimisation by using Evolutionary Algorithms. In Chapter 3, such a Multi-objective Optimisation by using Evolutionary Algorithms is focused on the reliability problem to solve. Chapter 4 describes the methodology to apply in order to solve the problem handled. Chapter 5 shows applications and their experimental results. Finally, Chapter 6 explains the conclusions and future research.

1. CHAPTER 1: BASIC CONCEPTS ON RELIABILITY.

1.1. Background.

Although an element may be well designed and built, it will end up failing eventually. The more useful probabilistic concepts to describe the performance of an element are Reliability and Availability. When reparable elements are considered, they usually start their useful life working satisfactorily. All along their life cycle they commute among operating and failure states. After failing, a repair task is carried out to recover the operating state. The performance of reparable elements can be measured by using their Availability, which relates the operating time and the total mission time or life cycle. The process until to failure, the process up to repair and finally, the whole process is described as follow.

1.2. Time to failure.

When a non-reparable element starts operating at time $t = 0$, it is interesting to know some details about the time that such an element will be working. The time from $t = 0$ up to the failure takes place it is known as the *"time to failure"*. If multiple experiments are conducted regarding an element, which is operating until the failure occurrence, the *"time to failure"* will not be the same in all cases. The *"time to failure"* will suffer variations so it behaves as a random variable T. Concretely, the *"time to failure"* can take any value within a real interval so it behaves as a continue random variable.

As a continue random variable, the *"time to failure"* presents a cumulative function or distribution function $F(t) = P(T \leq t)$, when $t > 0$. Such a distribution function denotes the probability that a component fails within the interval $(0, t]$ or the probability that a component does not work beyond t. As a probability, it reaches a value of $0 \leq F(t) \leq 1$. The shape of a distribution function with value 0 at time $t = 0$ is shown in the Figure 1.1.

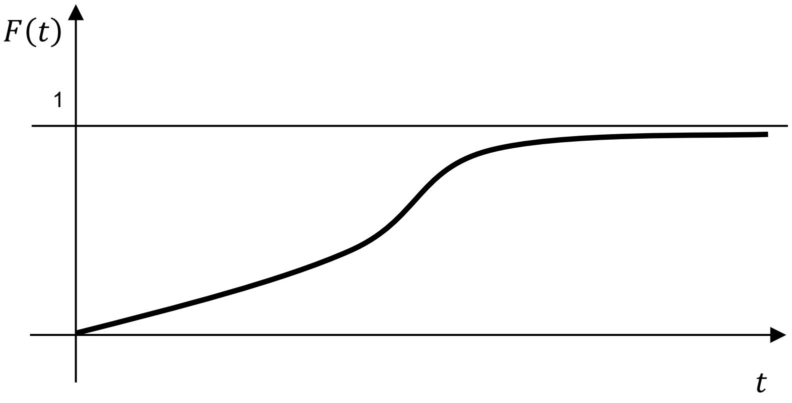


Figure 1.1: Distribution function.

Moreover, the *"time to failure"* has a density function $f(t)$ so the area contained under such a density function is the unit and its probability of being located between the times values t_1 and t_2 given $t_1 < t_2$ is equal to the area contained among these two values. The Equation 1.1 shows the way to compute such a probability when $t_1 = 0$ y $t_2 = t$ are considered.

$$F(t) = P(T \leq t) = \int_0^t f(u)du \tag{1.1}$$

It is possible to establish the relationship between $f(t)$ and $F(t)$ by using the Equation 1.2.

$$f(t) = \frac{d}{dt}F(t) \tag{1.2}$$

The shape of a density function is shown in the Figure 1.2.

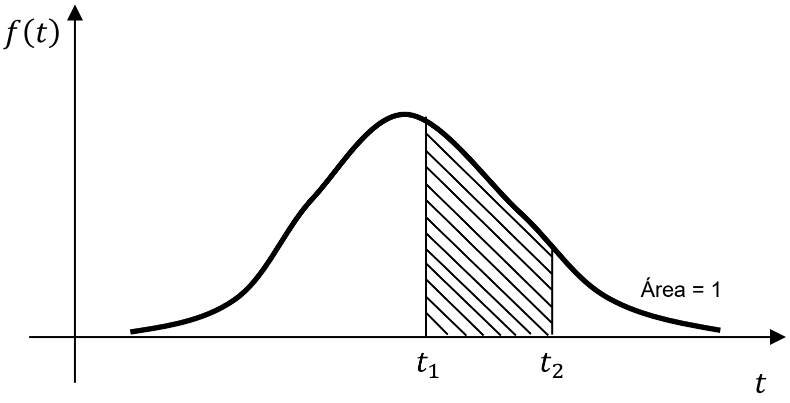


Figure 1.2: Density function.

The probability that an element fails within the period $[t\,, t + dt)$ is equivalent to $f(t)dt$. The transition to the failure state regarding such an element can be characterised by employing the failure rate $h(t)$. Thus, $h(t)dt$ represents the probability that an element fails within the interval $[t\,,\ t + dt)$ given that it was working within the interval $[0, t)$.

The shape of $h(t)$ is shown in the Figure 1.3, which is denominated the "*bath-tub*" curve due to its characteristic shape.

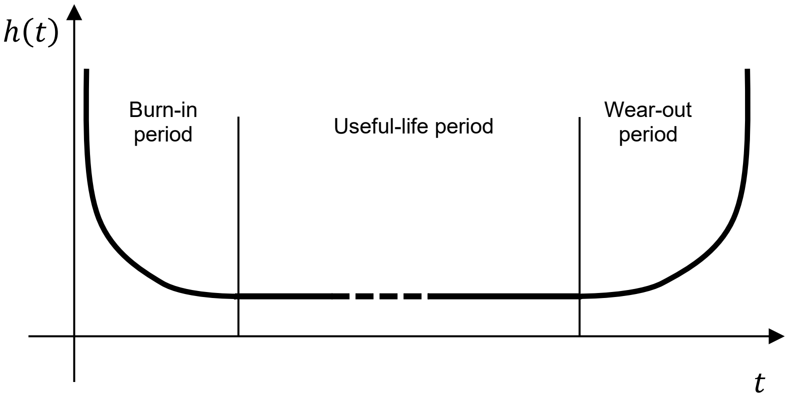


Figure 1.3: The "bath-tub" curve.

The failure rate usually is high during the initial phase. This is due to the appearance of hidden failures. Once the element survives such a phase, the failure rate trends to steady and to maintain at this level during a period of time. After that, an increment in the wear of the element is produced. Attending to the *“bath-tub”* curve, the lifetime of an element can be divided in three typical intervals: the *“burn-in”* period, the *“useful-life”* period and the *“wear-out”* period.

1.3. Time to repair.

Developing a similar process, it is possible to characterise the *“time to repair”*. As a continue random variable, it presents a cumulative or distribution function $G(t)$ which denotes the probability that a given failed component is repaired within the interval $(0, t]$. In this case, the transition from the failed state to the operating state is characterised by $m(t)$.

1.4. Unavailability.

The concepts *“time to failure”* and *“time to repair”* were previously introduced. Whereas the concept Reliability is related to the *“time to failure”*, the concept Maintainability is related to the *“time to repair”*. Under the Unavailability concept, both Reliability and Maintainability are related in order to define the way in which the system can fulfil the function for which it was designed, over a period of time. Andrews, J. and Moss, T. show how the availability can be calculated [12] by using the failure $(w(t))$ and repair $(v(t))$ intensities.

- A device, which is continuously subjected to the failure and repair process, presents a failure probability in the time interval $[t, t + dt)$, given it was working at $t = 0$, represented by $w(t)dt$. Two situations lead to failure in $[t, t + dt)$:
    - The device works continuously from 0 to t until the first failure in the time interval $[t, t + dt)$ (the probability of this is given by $f(t)dt$, where $f(t)$ is the failure density function).

- The device fails in $[t, t + dt)$ but this is not the first failure. In this second situation, the device has experienced one or more repairs prior to the failure and the last one was carried out in the interval $[u, u + du)$ (the probability of this is given by $w(u)du \times f(t - u)dt$.

The repair time u can occur at any point between 0 and t so adding all possibilities gives the Equation 1.3.

$$w(t)dt = f(t)dt + \int_0^t f(t-u)v(u)du\, dt \tag{1.3}$$

- A repair can only occur in the interval $[t, t + dt)$ in case of failure has occurred at some interval $[u, u + du)$ prior to t. The probability of this is $(t - u)dt \times w(u)du$, where $w(u)du$ is the probability of failing in $[u, u + du)$ given it was working at $t = 0$, $g(t - u)dt$ is the probability of repair in the interval $[t, t + dt)$ given it has been in failed state since last failure in the interval $[u, u + du)$ and it was working at $t = 0$ and knowing that $g(t)$ is the repair density function. Since u can vary between 0 and t, the Equation 1.4 can be obtained.

$$v(t)dt = \int_0^t g(t-u)w(u)du\, dt \tag{1.4}$$

- Cancelling dt from the Equations 1.3 and 1.4, the simultaneous integral equations defining the unconditional failure and repair intensities, which are shown in the Equations 1.5, are obtained.

$$\begin{gathered} w(t) = f(t) + \int_0^t f(t-u)v(u)du \\ v(t) = \int_0^t g(t-u)w(u)du \end{gathered} \tag{1.5}$$

The Unavailability ($Q(t)$) can be computed from the Equation 1.6.

$$Q(t) = \int_0^t [w(u) - v(u)]du \tag{1.6}$$

The opposite to the Unavailability $Q(t)$ is the Availability $A(t)$, so $Q(t) + A(t) = 1$.

When a device follows exponential failure and repair intensities (constant failure and repair rates), its Availability can be found through the solutions of the Equations 1.5 for $w(t)$ and $v(t)$, which can be carried out by using Laplace transforms. In this case, the Equation 1.7 is obtained so the Availability can be computed using the *Mean Time to Failure* $(MTTF)$ and the *Mean Time to Repair* $(MTTR)$.

$$A = \frac{MTTF}{MTTF + MTTR} \tag{1.7}$$

When a device does not follow exponential failure and/or repair intensities, finding the device Availability through the solutions of the Equations 1.5 for $w(t)$ and $v(t)$ might be complicated so a simulation approach can be suitable. Such a technique is explored along the present research.

1.5. Simulation.

When a new system is designed, the Discrete Event Simulation arises as a powerful modelling technique which allows complex systems to be analysed much more accurately due to a more realistic representation of their behaviour in practice.

As it was previously explained, when a device does not follow exponential failure and/or repair intensities, finding its Availability by using the solutions of the Equations 1.5 for $w(t)$ and $v(t)$ might be hard so a simulation approach can be suitable. The Availability in the present research will be computed by using the Equation 1.8.

$$A = \frac{\sum_{i=1}^{n} t_{f_i}}{\sum_{i=1}^{n} t_{f_i} + \sum_{j=1}^{m} t_{r_j}} \tag{1.8}$$

Where:

- n denotes the total number of operating times,
- t_{f_i} denotes the i-th operating time in hours (time to failure or time to start following a scheduled preventive maintenance activity),
- m denotes the total number of recovery times,
- t_{r_j} denotes the j-th recovery time in hours (due to repair or preventive maintenance activity).

Operating and recovery times are not previously known. They behave as random variables. If a historical record of both times is compiled and a statistical analysis is performed, operating and recovery times could be defined as probability density functions through their respective parameters. Those functions can arise from a specific typology (e.g., Exponential, Weibull, or Normal). The procedure to achieve random numbers, which follow a specific probability density function, is exposed in Ref. [12]. These numbers are used to compute the system's Availability by employing the Equation 1.8. Next, such a procedure is explained.

1.5.1. Exponential distribution.

The density function for the exponential distribution with mean μ satisfies the Equation 1.9.

$$f(t) = \frac{1}{\mu} e^{-t/\mu} \tag{1.9}$$

Such a distribution represents the variability of *"times to failure"* or *"times to repair"* with mean μ so random samples from this distribution can be obtained by first integrating to get the cumulative failure distribution $F(t)$, which is shown by the Equation 1.10.

$$F(t) = \int_0^t f(u)\,du = 1 - e^{-t/\mu} \tag{1.10}$$

The cumulative failure distribution presents the same properties and range as the distribution of random numbers. Hence, taking a random sample is possible when a random number X is generated and equated to $F(t)$ (with $0 \leq F(t) \leq 1)$, as shown in the Equation 1.11.

$$X = 1 - e^{-t/\mu} \tag{1.11}$$

Rearranging gives the *“time to failure”* as is shown in the Equation 1.12.

$$t = -\mu \ln(1 - X) \tag{1.12}$$

If X is uniform over $[0,1]$, then $1 - X$ so is. Therefore, the Equation 1.12 can be simplified as it is shown in the Equation 1.13.

$$t = -\mu \ln(X) \tag{1.13}$$

Next, an example is developed. The procedure to generate a random *“time to failure”* that follows an exponential distribution is shown. In this case a *“Mean Time to Failure”* of 1500 hours is considered. Firstly, a random number between 0 and 1 must be generated. Secondly, the Equation 1.13 is used to transform the sample. For instance, if the random number generated by the computer is 0.2508, the “*time to failure”* achieved will be $t = -1500 \ln(0.3508) \approx 1571$ hours.

1.5.2. Weibull distribution.

Random samples can be achieved from the Weibull distribution in a similar way as for the exponential distribution. A Weibull distribution with parameters β and η presents a density function as is shown in the Equation 1.14.

$$f(t) = \beta \frac{t^{\beta-1}}{\eta^{\beta}} e^{-(t/\eta)^{\beta}} \tag{1.14}$$

The cumulative distribution is shown in the Equation 1.15.

$$F(t) = 1 - e^{-(t/\eta)^{\beta}} \tag{1.15}$$

Random times can be achieved from using a random number X as is shown in the Equation 1.16:

$$\begin{gathered} X = 1 - e^{-(t/\eta)^{\beta}} \\ e^{-(t/\eta)^{\beta}} = 1 - X \\ (t/\eta)^{\beta} = -\ln(1 - X) \\ t = \eta[-\ln(1 - X)]^{1/\beta} \; o \; t = \eta[-\ln(X)]^{1/\beta} \end{gathered} \tag{1.16}$$

As an example, the procedure to generate a random *"time to failure"* is shown. Such a *"time to failure"* follows the Weibull distribution with parameters β and η with values 3 and 120, respectively. A random number is generated between 0 and 1, which is then transformed as is shown in the Equation 1.16. Considering that the random number supplied by the computer was 0.3508, the *"time to failure"* achieved will be $t = 120[-\ln(0.3508)]^{1/3} \approx 121$ hours.

1.5.3. Normal Distribution.

Achieving random samples from the normal distribution with media μ and standard deviation σ cannot be obtained by simple transposition of its formula. This is due to the fact that its density function, which is shown in the Equation 1.17, cannot be integrated to obtain a formula for $F(t)$.

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}[(t-\mu)/\sigma]^2} \tag{1.17}$$

Therefore, the central limit theorem can be used. Considering the independent random variables $X_1, X_2, \dots, X_n$, which are randomly distributed and have mean μ and variance σ^2. If $S_n = X_1 + X_2 + \cdots + X_n$, the random variable $(S_n - n\mu)/\sigma\sqrt{n}$ is asymptotically normally distributed with mean 0 and standard deviation 1. The random numbers $U(0,1)$ are identically distributed and may be used to form S_n. In practise, it is necessary to fix some finite number of n so that the resulting S_n will only be approximately normal. An interesting value of n is 12, since X_i presents $\mu = 0.5$ and $\sigma^2 = 1/12$, so S_n is $N(6,1)$. Using the Equation 1.18, 12 random numbers $U(0,1)$ were generated in order to achieve a random sample from the normal distribution.

$$X = \sum_{i=1}^{12} X_i \tag{1.18}$$

By the central limit theorem X is normally distributed with mean 6 and standard deviation 1. Time values can be achieved by using the Equation 1.19.

$$t = (X - 6)\sigma + \mu \tag{1.19}$$

As an example, a random time that follows the normal distribution with mean 10 and standard deviation 2 is generated. Next, 12 random numbers between 0 and 1 are generated and summed as in the Equation 1.18.

$$X = \sum_{i=1}^{12} X_i = 0.12 + 0.24 + 0.32 + 0.15 + 0.56 + 0.93 + 0.82 + 0.62 + 0.53 + 0.25 + 0.12 + 0.73 = 5.39$$

Next, the Equation 1.19 is used to estimate such a time.

$$t = (X - 6)\sigma + \mu = (5.39 - 6) \cdot 2 + 10 = 8.78 \text{ hours}$$

2. CHAPTER 2: MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS.

2.1. Background.

Solving complex problems with multiple feasible solutions is possible by employing optimisation. Optimisation allows finding one or more feasible solutions for a problem. Such solutions correspond to the extreme values of one or more objectives regarding the problem. When the problem has a single objective, it is called *“single-objective optimisation problem”*. Conversely, when the problem has more than one objective, it is called *“multi-objective optimisation problem”* (MOP). Most of real-world problems have different objectives that need to be optimised at the same time (often in conflict) and their solutions emerge from a set of solutions that represent the best compromise among objectives (Pareto optimal set) [16].

Assuming that the independent variable x has n dimensions and the MOP is a minimising problem, such a problem can be formulated as it is shown in the Equation 2.1:

$$\min_{x} f(x) = \min_{x}[f_1(x), f_2(x), \dots, f_m(x)] \tag{2.1}$$

It means that the vector of functions $f(x)$ is going to be simultaneously minimised for the set of m functions $f_i(x)$, where $i = 1,2, \dots, m$. MOP’s can present constraints of both equality (Equation 2.2) and inequality (Equation 2.3), as well as limited values for the decision variables (Equation 2.4). Therefore, if such constraints are not satisfied, the found solutions will not be feasible.

$$h_k(x) = 0, k = 1, 2, \dots, K \tag{2.2}$$

$$g_j(x) \geq 0, j = 1, 2, \dots, J \tag{2.3}$$

$$x_i^{inf} \leq x \leq x_i^{sup}, i = 1, 2, \dots, n \tag{2.4}$$

The main objectives in a MOP are [17]:

1. Achieving a solution set as close as possible to the Pareto optimal front.

2. Finding a set of previous solutions as varied as possible along the front.

When multi-objective optimisation is used, a solution set regarding a problem is looked for. Such a problem presents a minimum of two objectives in conflict. The achieved solution set is called Pareto front, due to the name of the economist Vilfredo Pareto. Vilfredo Pareto postulated the efficient mode of resources allocation, so such resources are efficiently allocated in the Pareto sense when it is unable to improve the welfare of any person without worsening the other. Commuting this sentence to the field of the multi-objective optimisation in engineering, it is possible to establish that the optimum solutions are the balanced solutions. From these solutions, it is not possible to improve one of the objectives without damaging another one. In Ref. [15], some concepts regarding the Pareto optimality are defined:

- Domination: The point x^* dominates the point x if the following conditions are maintained:
  - $f_i(x^*) \leq f_i(x) \; \forall \; i \in [1, k]$
  - $f_j(x^*) < f_j(x)$ for at least one $j \in [1, k]$.

  That is, x^* is as good as x for all objective function values and it is better than x for at least one objective function value. The notation $x^* \succ x$ is used to indecate that x^* dominates x from the mathematic point of view.

- Weak domination: The point x^* weakly dominates the point x if $f_i(x^*) \leq f_i(x) \; \forall \; i \in [1, k]$, that is, x^* is as good as x for all objective function values. To indicate that x^* weakly dominates x the mathematical expression $x^* \succcurlyeq x$ is used.

- Non-dominated: The point x^* is non-dominated if there is not a point x that dominates it.

- Pareto optimal points: A Pareto optimal point x^* is a point that is not dominated by any other point x in the search space. Mathematically, considering x^* a Pareto optimal point:

$$x^* \leftrightarrow \nexists x : (f_i(x) \leq f_i(x^*) \ \forall i \in [1,k] \wedge f_j(x) < f_j(x^*) \ \exists j \in [1,k]$$

- Pareto optimal set: The Pareto optimal set, which is denoted as P_s, is the set of all points x^* that are non-dominated.

$$P_s = \left\{x^* : \left[\nexists x : (f_i(x) \leq f_i(x^*) \ \forall i \in [1,k] \wedge f_j(x) < f_j(x^*) \ \exists j \in [1,k]\right]\right\}$$

- Pareto front: The Pareto front or non-dominated set, which is denoted as P_f, is the set of all function vectors $f(x)$ that corresponding to the Pareto set.

$$P_f = \{f(x^*) : x^* \in P_s\}$$

As an illustrative example, the Figure 2.1 shows a set of possible solutions for a two objectives problem, which are represented as crosses and points. In this case, crosses dominate points, due to the fact that a minimisation problem is being considered.

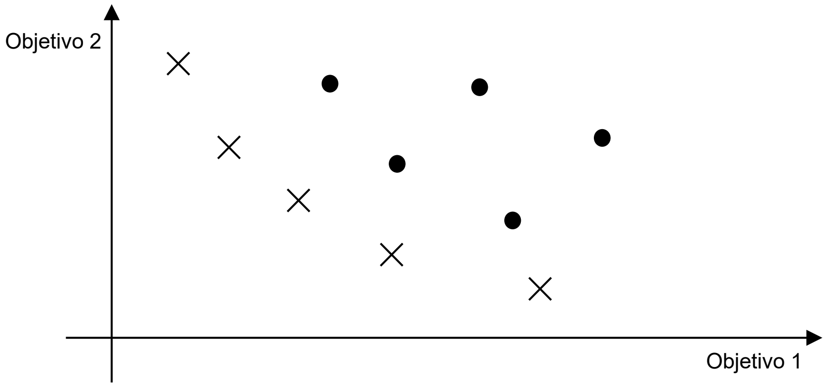


Figure 2.1: Non-dominated set (crosses).

2.2. Classic methods for Multi-objective Optimisation.

Various studies regarding classic methods for Multi-objective Optimisation were previously developed. The one conducted by K. Miettinen [18] is an example.

Attending to the moment when the decision makers supply their preferences, three main approaches are considered [19]:

1. A priori methods: In this case, the decision makers must stablish the order of preference among objectives before the process. Methods that follow to this kind of paradigm are the *Value Function Method* [20,21], the *Lexicographic Ordering Method* [22] or the *Goal Programming Method* [23].
2. Iterative methods: In this case, the decision makers must stablish and update their preferences among objectives at different moments along the process. Methods that follow to this paradigm are the *Geoffrion-Dyer-Feinberg Method* [24], the *Tchebicheff Method* [25], the *Reference Point Method* [26], the *GUESS Method* [27] or the *NIMBUS Method* [28].
3. A posteriori methods: Once achieved the set of solutions, the decision makers must stablish their preferences. Methods that follow to this kind of paradigm are the *Weighted Method* [29,30] or the ϵ*-Constraint Method* [31].

2.3. Multi-objective optimisation by employing Evolutionary Algorithms.

Different metaheuristics have been used to solve optimisation problems. The most popular are the Evolutionary Algorithms, which are based on the emulation of the natural selection mechanism [32]. Such Algorithms are considered along the present research.

2.3.1. Evolutionary Algorithms introduction.

Evolutionary Algorithms are based on the evolutionary theory of species, which was initially proposed by Darwin in 1859 [33]. Evolutionary Algorithms are the older Evolutionary Strategies, which present some characteristics based on the principles of the natural selection proposed by Darwin. Such principles claim that the better adapted individuals survive and transmit their features to the offspring. The principles of natural selection are resumed as follows [15]:

- A biologic system includes a population of individuals and many of them can reproduce.
- The individuals have a finite life.
- There are variations in the population.
- The more the ability to reproduce, the more the ability to survive.

Evolutionary Algorithms emulate such features. When an optimisation problem must be solved, a random population can be generated as a solution set. The possible solutions are called *“candidate solution”* or *“individual”*, and they can be generated as digit strings. Each digit is called *“allele”*. The sequence of digits that determines the characteristic of an *“individual”* is called *“gene”*. Specific *“genes”* are called “*genotypes*” and the parameters represented by “*genotypes*” are called “*phenotypes*”. The “*chromosome*” is the set of genes of an *“individual”*. Once the population has been created, the best individuals have the greatest chance of reproducing. Conversely, the worst individuals have the lowest probability to reproduce. In this way, an offspring is generated from a population. Generation after generation, better adapted individuals are achieved, which can be solutions to the optimisation problem.

Darwin proposed the basis of the species evolution theory; however, he did not explain how the genetic heredity takes place. Gregor Mendel was the first human being who understood such a mechanism so he could explain how natural selection works. Mendel presented his findings to the Natural History Society of Brünn in 1865 [34], but his postulates were not as relevant as the theories proposed by Darwin were. Nevertheless, the Mendel’s postulates would be recovered at the beginning of XX century [35].

Once the basis to invent computers were established (among other inventors, by von Neumann), the Italian mathematician Nils Barricelli developed the first Genetic Algorithm software in 1954 [36]. Other pioneers were the biologist Alexander Fraser [37] or George Box [38], who was more interested in solving engineering problems. Next, Ingo Rechenberg presented his first works in Evolution Strategies [39] in 1964, and Lawrence Fogel presented the Evolutionary Programming [40] in 1966. However, John Holland, who had started to study the Genetic Algorithms in the 60's, presented the mathematics of evolution in 1975 [41]. Due to the improvement of computing capabilities, research in the field of the Genetic Algorithms shoots up both in the 70's and in the 80's. A scientist consensus exists regarding David Shaffer designed the first Multi-objective Evolutionary Algorithm [42] in the middle of 80's, which was called *"Vector Evaluated Genetic Algorithm" (VEGA)*.

2.3.2. Basic evolutionary operators.

There are three basic operators, which are addressed to evolve a population of individuals. Their main objective consists of improving their fitness generation after generation. They are Selection, Crossover and Mutation.

2.3.2.1. Selection operator.

The Selection operator allows distinguishing between the best and worst adapted individuals from the population. Next, some methods to select individuals are shown:

- <u>Roulette-wheel selection or fitness-proportionate selection:</u> This method was proposed by De Jong [43]. Attending to the fitness regarding the individuals of the population, selection probabilities are set. In this way, choosing individuals with better fitness is more likely. Such probabilities are drawn in a roulette, as shown in the Figure 2.2. Selecting an individual is possible when the roulette is spun.

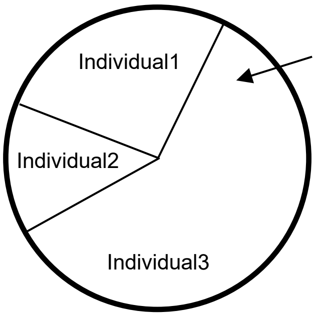


Figure 2.2: Roulette-wheel selection or fitness-proportionate selection.

- Stochastic Universal Sampling: Baker [44] proposed this method to solve a problem that the Roulette-wheel selection presents. Individuals with high fitness have high probability of not being selected. Since an individual has 40% probability of being selected, there is 60% probability of being selected that is shared by the rest of individuals. This implies that such an individual with high fitness has 60% probability of not being selected when the roulette is spun. Attending to the number of individuals to choose, uniformly spaced pointers are distributed, and the spinner is spun once. For instance, the Figure 2.3 shows how to select two individuals when the spinner is spun once.

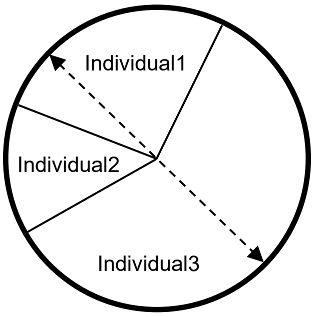


Figure 2.3: Stochastic Universal Sampling (selection of 2 individuals).

- Rank-Based Selection: In this case, the individuals are ordered by attending to their fitness. Next, a rank is given depending on such an order. The worst fitness individual achieves 1 as a rank, and so on up to reach the best achieved individual. Finally, the individuals are selected attending to the rank [45].
- Tournament Selection: In this case, several individuals from the population are chosen and the best of them is selected.

2.3.2.2. Crossover or recombination operator.

This operator guarantees that the selected individuals share a part of their genetic features in order to generate new individuals for the next generation. Some methods that are employed to implement the crossover operator are as follows:

- One-Point Crossover: It was proposed by Holland [41]. The chosen individuals exchange their genetic information from a specific point of the string. Considering the individuals x_1 and x_2:

$$x_1 = [x_1(1)\ x_1(2) \dots x_1(m-1)\ x_1(m) \dots\ x_1(n)]$$
$$x_2 = [x_2(1)\ x_2(2) \dots x_2(m-1)\ x_2(m) \dots\ x_2(n)]$$

  Their chromosomes are formed by n alleles or features. A point m is selected so the information is crossed from such a point. The offspring is formed by individuals y_1 and y_2:

$$y_1 = [x_1(1)\ x_1(2) \dots x_1(m-1)\ x_2(m) \dots\ x_2(n)]$$
$$y_2 = [x_2(1)\ x_2(2) \dots x_2(m-1)\ x_1(m) \dots\ x_1(n)]$$

  The crossover point m is randomly generated.

- Two-Point Crossover: It was proposed by DeJong [43]. In this case, two points m y k are selected to conduct the crossover. Therefore, the individuals x_1 and x_2:

$$x_1 = [x_1(1)\ x_1(2) \dots x_1(m-1)\ x_1(m) \dots x_1(k)\ x_1(k+1) \dots\ x_1(n)]$$
$$x_2 = [x_2(1)\ x_2(2) \dots x_2(m-1)\ x_2(m) \dots\ x_2(k)\ x_2(k+1) \dots x_2(n)]$$

  Such individuals are transformed as it is shown:

$$x_1 = [x_1(1)\ x_1(2) \dots x_1(m-1)\ x_2(m) \dots x_2(k)\ x_1(k+1) \dots\ x_1(n)]$$
$$x_2 = [x_2(1)\ x_2(2) \dots x_2(m-1)\ x_1(m) \dots\ x_1(k)\ x_2(k+1) \dots x_2(n)]$$

  The original individuals exchange the genetic information among alleles m and k.

- Uniform Crossover: It was proposed by Syswerda [46]. In this case, the exchange decision is taken regarding each allele with a 50% probability.

- Multi-Parent Crossover: It was proposed by Bremermann [47]. It consists of choosing randomly features from several individuals in order to build a new individual.
- Simulated Binary Crossover: The above Crossover types are used for both discrete and real encoding. The Simulated Binary Crossover is specific for problems in a continuous domain. The offspring are created from the original individuals x_1 and x_2 [48]:

  $$y_1(k) = (1/2)[(1-\beta_k)x_1(k) + (1+\beta_k)x_2(k)]$$
  $$y_2(k) = (1/2)[(1+\beta_k)x_1(k) + (1-\beta_k)x_2(k)]$$

  In this case, the location of the feature in the individual genetic string is represented by k and β_k is a random number from the density function:

  $$PDF = \begin{cases} {}^{1}\!/_{2}\,(\eta+1)\beta^{\eta}, 0 \le \beta \le 1 \\ {}^{1}\!/_{2}\,(\eta+1)\beta^{-(\eta-2)}, \beta > 1 \end{cases}$$

  Where η is a non-negative real number, whose recommended value by several authors [48] is between 0 and 5.

2.3.2.3. Mutation operator.

Although mutation is a biologic low likely process, it is important regarding the Evolutionary Algorithms. It allows exploring new solutions for the problem. Its implementation basically consists of considering a mutation probability (it is typically smaller than 2%). Such a mutation probability is applied to each allele of the chromosome achieved after crossing the original individuals. Therefore, variations in the chromosome have a probability of occurrence. Some examples are shown as follows [15].

- Uniform Mutation Centred at $x_i(k)$: This type of mutation can be written as follows:

  $$x_i(k) = \begin{cases} x_i(k) & si\ r \ge \rho \\ U[x_i(k) - \alpha_i(k), x_i(k) + \alpha_i(k)] & si\ r < \rho \end{cases}$$

  Where r is a random number from a uniform distribution between 0 and 1, ρ is the mutation rate and $\alpha_i(k)$ is the user-defined mutation magnitude.

Therefore, each feature $x_i(k)$ of the individual x_i mutes if $r < \rho$. In such a case, the feature $x_i(k)$ varies regarding the $\alpha_i(k)$ magnitude.

- Uniform Mutation Centred at the Middle of the Search Domain: This type of mutation can be written as follows:

$$x_i(k) = \begin{cases} x_i(k) & si\ r \geq \rho \\ U[x_{min}(k), x_{max}(k)] & si\ r < \rho \end{cases}$$

Where r is a random number from a uniform distribution between 0 and 1, and ρ denotes the mutation rate. Each feature $x_i(k)$ of the individual x_i mutes if $r < \rho$. In such a case, the feature $x_i(k)$ varies regarding the magnitude achieved when a random number is generated from a uniformly distributed distribution within the values of the domain.

2.3.2.4. Other interesting operators.

Although the basic operators regarding Evolutionary Algorithms were described above, there are other important operators that have allowed supplying excellent solutions to managed problems by Evolutionary Algorithms.

- Fitness sharing: It was introduced by Goldberg and Richardson [49]. This is an operator that allows devaluing the fitness of an individual in the population, when such an individual is located in a crowding area. This allows maintaining the diversity within the population.
- Elitism: It consists of injecting the best individuals achieved from a generation to the next. In this way, the knowledge achieved by the algorithm persists. Zitzler introduced formally such a concept [50] in multi-objective evolutionary algorithms. Normally, an external population is used in which the found non-dominated individuals are stored all along the evolutionary process.

2.3.3. Multi-objective Evolutionary Algorithms: The First Generation.

As explained above, the *"Vector Evaluated Genetic Algorithm" (VEGA)* [42] is considered as the first Multi-objective Evolutionary Algorithm. It was David Goldberg who included the Pareto Optimality concept to the Evolutionary Algorithms [42], so such a point brought the division to the first-generation Evolutionary Algorithms. On the one hand, the Evolutionary Algorithms non-based on Pareto dominance, and on the other hand, the Evolutionary Algorithms based on such a criterion.

Both Algorithms *VEGA* and *VOES (Vector‑Optimized Evolution Strategy)* [51] are examples of the first-generation Algorithms non-based on Pareto dominance. *VEGA* basically consists of partitioning, generation after generation, the population into as many different subpopulations as functions are optimised. Each sub-population is probabilistically generated by following the fitness of the individuals. Next, recombination and mutation are applied in order to create a new offspring. *VOES* is based on the diploid cell concept, which is the cell that presents two homolog chromosomes in its core. In this method, each individual is codified with a dominant solution and a recessive solution, so the evaluation of the individuals is achieved from the weighing of both solutions.

After publishing the Goldberg's ideas [32], almost all Multi-objective Evolutionary Algorithms have such influence. Basically, Goldberg suggested to locate a set of solutions from the population, which are non-dominates from the Pareto point of view. A superior rank is allocated to such solutions. The procedure is repeated for the rest of solutions until a rank is allocated to each one. This mechanism allows selecting solutions according to their rank. Some examples of Algorithms based on the non-dominance concept are:

- *Non-dominated Sorting Genetic Algorithm* (NSGA): This Algorithm was proposed by Srinivas y Deb [52]. The individuals of the population are classified based on the non-dominance criterion, so the non-dominated individuals fall into the same category. A fictional fitness value, which is proportional to the population size, is allocated to such individuals so they have a similar reproductive potential. In order to maintain the diversity along the Pareto front, a fitness sharing function is applied. Next, the individuals from this category are ignored and the process is repeated for the rest of individuals.

- *Niched-Pareto Genetic Algorithm* (NPGA): It was proposed by Horn, Nafploitis and Goldberg [53]. This Algorithm uses tournament selection based on the Pareto dominance. Two individuals are randomly chosen from the population and compared with a subset of such a population. When one individual is dominated by the subset and the other one is not, the non-dominated individual wins the tournament. Any other situation implies a draw, so the tournament is solved by using the fitness sharing.
- *Multi-Objective Genetic Algorithm* (MOGA): This Algorithm was proposed by Fonseca and Fleming [54]. In this case, the individual rank is computed by considering the number of individuals of the population that dominate such an individual. The highest fitness value is assigned to the non-dominated individuals, whereas the dominated individuals are penalised based on the population density relative to the region that they belong to.

2.3.4. Multi-objective Evolutionary Algorithms: The Second Generation.

The second generation of Multi-objective Evolutionary Algorithms starts when the elitism becomes the standard mechanism [55]. Next, some representative Evolutionary Algorithms of such a generation are shown:

- *Strength Pareto Evolutionary Algorithm* (SPEA): In this Algorithm proposed by Zitzler and Thiele [50], the non-dominated individuals from each generation are stored in a file. A strength value is computed regarding each individual in the file by dividing the number of individuals from the population that such an individual dominates, and the total number of individuals plus one. The fitness for each individual is computed regarding the strength value of the solutions which are dominated by such an individual. This process considers both the proximity to the true Pareto front and the distribution of the solutions. Furthermore, the size of the file is limited in order to avoid its excessive growth.
- *Pareto Archived Evolution Strategy* (PAES): It was introduced by Knowles and Corne [56]. It presents an original approach to maintain the diversity, which consists of dividing the search space by using a georeferenced grid.

In this way, it is possible to check the number of individuals that fall within each grid regarding their fitness. The best adapted solutions are stored in the external file, as in the previous case, so the ordination of individuals is based on both dominance and density in the grid.

- *Non-dominated Sorting Genetic Algorithm II* (NSGA-II): This Algorithm [57] is an improved version of NSGA. In this case, the elitism is implemented by employing a strategy that consists of including the best adapted individuals in the next generation. Besides ordering the individuals based on the non-dominance criterion, in order to maintain the diversity, the crowding is considered. In this way, the more the crowding distance, the more favoured the individuals which belong to the same front. This Algorithm is more efficient than the previous version and it is considered a referent method.

Other methods have been proposed, such as PESA (*Pareto Envelope-based Selection Algorithm*) [58], SPEA2 [59] or NSGA-II with controlled elitism [60].

2.3.5. Evolutionary Algorithms based on the selection criterion.

Nowadays, it exists a consensus from authors [61][62] which establishes a classification of the Multi-objective Evolutionary Algorithms based on their selection mechanism.

- Methods based on Pareto dominance: Such methods order the individuals attending to two levels. The first level is governed by the Pareto dominance and the second one by the contribution to diversity. The contribution from individuals to diversity is considered when such individuals belong to same dominance level. This criterion is used by methods such as NSGA-II [57] or SPEA2 [59].
- Methods based on indicators: These methods guide the search by using an indicator of the performance measure. The more commonly used indicator is the Hypervolume, which is employed by methods such as SMS-EMOA [63], FV-EMOA [64] or HypE [65].

- Methods based on decomposition of the search space: These methods decompose the problem in several sub-problems, which are simultaneously optimised. The convergence and diversity on solutions is achieved by employing several scalarising functions. This criterion is used by methods such as MOEA/D [66], NSGA-III [67] or MOGLS [68].

Attending to such a classification paradigm, five methods from the state-of-the-art are paid attention along the present research, which have been widely employed by the scientific community. Such methods are used in order to test and compare the performance from each one. The five methods are:

- Methods based on Pareto dominance: The methods NSGA-II [57] and GDE3 [69] are studied. In the case of GDE3, Differential Evolution [70] is used as a mechanism to create new individuals.
- Methods based on indicators: The SMS-EMOA [63] method is studied.
- Methods based on decomposition of the search space: The MOEA/D [66] is studied. Moreover, the MOEA/D-DE [71] is studied, which uses Differential Evolution as a mechanism to create new individuals.

2.3.5.1. Non-dominated Sorting Genetic Algorithm II (NSGA-II).

This method [57] carries out a double operation over the individuals of the population. On the one hand, the individuals are classified in relation to non-dominance levels or fronts in the sense of Pareto. On the other hand, the diversity is maintained when the crowding distance is computed in order to discriminate between individuals from the same front.

In order to classify the individuals of the population in non-dominance levels or fronts, two parameters are computed for each individual or solution p; the n_p parameter or number of solutions which dominate the solution p, and the S_p parameter or set of solutions which is dominated by the solution p. All solutions with parameter $n_p = 0$ belong to the first non-dominance front. Next, the set of solutions S_p is consulted for each solution from the first front and the dominance count n_p is reduced by a unit. In this way, the solutions which reaches the n_p value of zero are commuted to the second front. The process is repeated for the solutions included in the second front and so on till all solutions are integrated in different fronts.

In order to maintain the diversity, the first version of the method [50] had a function to measure the fitness sharing. Such a mechanism is replaced in NSGA-II, which has an approach to compare the crowding distance between solutions. This allows simplifying the procedure of the previous version. In order to estimate the crowding distance for the solutions around a specific solution of the population, it is computed the distance to the solutions located at both sides of such a specific solution and under the projection of each objective. Such a quantity is employed to estimate the formed perimeter of the cuboid when the closest neighbours are used as vertexes, and it is defined a crowding distance. This concept is illustrated in the Figure 2.4.

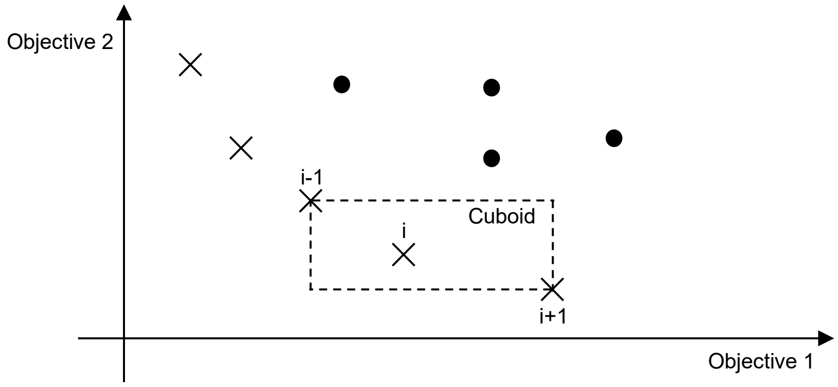


Figure 2.4: Crowding distance for the solution i.

Ordering the population by attending to the objective function value in ascend order is needed to compute the crowding distance. Next, an infinite value is assigned to the solutions with extreme values regarding each objective function. On the other hand, a distance value equal to the normalised absolute difference in the objective function values regarding the two adjacent solutions is assigned to the intermediate solutions. Such a calculation is repeated for all objectives and finally, the total crowding distance is computed as the sum of the individual distance values regarding each objective. Such values must be previously normalised.

Firstly, the selection mechanism looks at the ranks and non-dominance fronts. Secondly, in case of two individuals belong to the same front, the individual with the smaller crowding distance is preferred. The method starts generating a random population P_0. This is ordered based on the non-dominance criterion. A rank or front number is assigned to each solution which is equal to the non-dominance level. At first, the tournament selection, recombination and mutation are employed in order to create an offspring Q_0 with size N. Since the elitism is introduced when the actual population is compared to the best non-dominated solution from the previous population, the process is different after the initial generation. In order to create next generations, a population with a size of $2N$ is generated, which combines P_i and Q_i where i represents the i-th generation. If the size of the first front is smaller than N, all the solutions contained in the front F_1 will belong to the next generation. The solutions from the next fronts will be assigned in an ordered way until completing the population. In case of exceeding the population size, the individuals from the last front included in the population must be ordered regarding the crowding distance. Finally, the more crowded individuals are rejected till complete the population. The procedure is shown in the Figure 2.5.

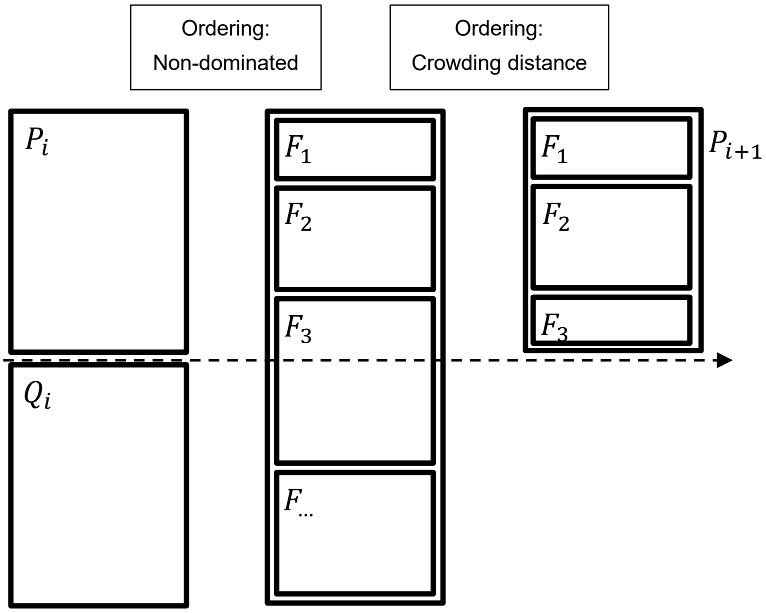


Figure 2.5: NSGA-II method procedure.

2.3.5.2. Third Evolution Step of Generalized Differential Evolution (GDE3).

This method [69] employed Differential Evolution (DE) as an operator [70] in order to create new individuals. The DE is based on the idea of taking the difference vector between two individuals and adding a scaled version of such a difference to a third individual. In this way, a new candidate solution is created. As an example, considering two individuals x_1 and x_2 from a population N, a scaled version of the difference between such individuals is added to a third individual x_3. Therefore, a mutation vector $v_i = x_3 + F(x_2 - x_1)$ is achieved, where F is a scale factor with typical value between 0.4 and 0.9 [15]. After creating such a mutation, this is crossed with any individual employed along the procedure in order to create a test vector u_i, where i represents the individual whom the test vector was created.

$$u_{ij} = \begin{cases} v_{ij} & if\ (r_{cj} < CR)\ or\ (j = \Im_r) \\ x_{ij} & otherwise \end{cases}$$

Where i denotes the individual from which the test vector was created, j denotes the j-th feature or dimension for the individual i, r_{cj} denotes a random number between 0 and 1 which is taken from a uniform distribution, CR is the constant crossing rate (typical values between 0.1 and 1 [15]) and $\Im_r$ is a random integer number which is taken from a uniform distribution with value between 0 and the individual maximum dimension. Therefore, the test vector u_i is a component-by-component combination from the individual x_i and the mutant vector v_i. $\Im_r$ guarantees that x_i and u_i will never be equals. Once the crossing is generated, x_i and u_i are compared so the best adapted goes to the next generation.

Regarding the GDE3 method, the first version extended the Differential Evolution (DE) for constrained multi-objective optimisation. It was achieved by modifying the DE's rule selection. It consisted of replacing, for the next generation, the old vector in favour of the test vector when such a test vector dominated weakly to the old vector regarding the constraints. The non-dominated vectors were not ordered during the optimisation process. There is not a mechanism in order to maintain the distribution and spreading of the solutions. Furthermore, there is not a repository for the non-dominated vectors. Although GDE supplied good solutions, it was sensitive to the control parameters selection.

Afterwards, GDE was modified in order to take decisions based on the crowding distance when both the vector test and the original vector were feasible and non-dominated one by the other. Such a modification improved the distribution and spreading of the solutions, but it slowed down the convergence of the population. It occurs since vectors far from the Pareto front are favoured instead of favouring the convergence of all vectors near to the Pareto front. Such a second version of GDE was still sensitive regarding the control parameters selection.

The third GDE version is extended to M objectives problems with K constraints. The GDE3 selection is based on the rules as follows [69]:

- In the case of infeasible vectors, the test vector is selected when it weakly dominates the original vector in the constraint violation space. Otherwise, the original vector is selected.
- In the case of feasible and infeasible vectors, the feasible vector is selected.
- When both vectors are feasible, the test vector is selected when weakly domain the original vector in the objective function space. When the original vector dominates the test vector, the original vector is selected. When neither vector dominates each other in the objective function space, both vectors are selected for the next generation.

After a generation, the population size might have been enlarged. In such a situation, the population must decrease to the original size by employing a similar selection approach to the one used in NSGA-II. In this case, the ordination process of non-dominated solutions is modified in order to consider the constraints, and the selection base on the crowding distance is improved to boost a better distribution of the solutions.

2.3.5.3. SMS-EMOA: Multi-objective Selection based on hypervolume.

Reaching the true Pareto front may be complicated. Therefore, some methods employ a variety of indicators in order to measure the approximation quality. One of the most relevant indicators is the Hypervolume [72]. The SMS-EMOA method [63] follows the Fleisher [73] idea. It claims that given a search space and a reference point which is dominated by the Pareto optimal solutions, maximising the Hypervolume is like finding the Pareto set.

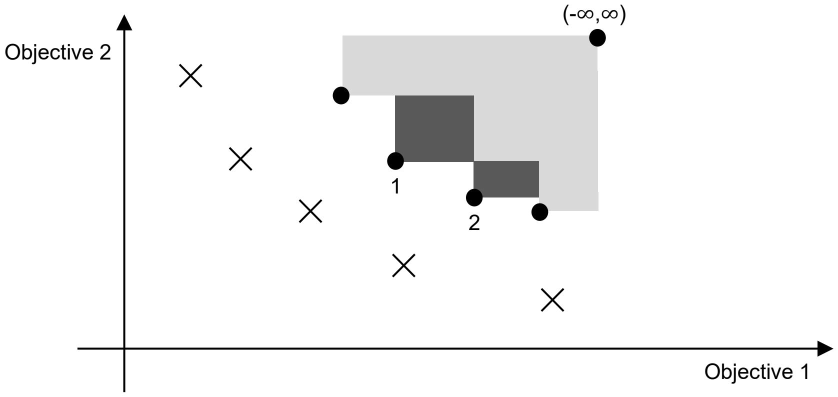


Figure 2.6: Hypervolume contributions from the individuals 1 and 2.

The algorithm classifies the solutions in fronts or levels based on the non-dominance criterion used by NSGA-II. Next, the Hypervolume is employed as a selection criterion in order to discard the individual with the worst level, which presents the lowest contribution to improve the Hypervolume.

The algorithm starts from an initial population and a new individual is created by applying the corresponding evolutionary operators. Such an individual will be part of the next generation whether it can replace another individual by supplying an improvement regarding the Hypervolume. The Figure 2.6 shows how the individuals of the front formed by crosses dominate the individuals of the front formed by points. The reference point is further located. The contribution to the Hypervolume due to the individual 2 is smaller than the contribution due to the individual 1. The individual 2 would be discarded from the population.

2.3.5.4. MOEA/D: Multi-objective Evolutionary Algorithm based on Decomposition.

Decomposition is a traditional strategy in Multi-objective Optimisation [66]. The MOEA/D algorithm decomposes the Multi-objective Optimisation Problem into N single objective optimisation sub-problems by using weighted vector and scalarising functions. Such typical scalarising functions used by MOEA/D present several approaches such as weighted sum, Tchebicheff or penalty-based boundary intersection (PBI).

Once the problem has been decomposed, MOEA/D solves simultaneously the sub-problems by evolving a population of individuals or solutions. Generation after generation, the population is formed by the best-found solutions regarding each sub-problem. The vicinity relationships between such sub-problems are defined based on the distances among their integration coefficient vectors. The optimum solutions of two neighbour sub-problems should be similar. Each sub-problem is optimised by using exclusively information from its neighbour sub-problems.

2.3.5.5. MOEA/D-DE: Multi-objective Evolutionary Algorithm based on Decomposition with Differential Evolution.

This method is a MOEA/D version, which is based on Differential Evolution [71]. This version is especially skilful to deal with complex forms regarding the Pareto set.

2.3.6. Comparative indicators.

The main objective in multi-objective optimisation consists of driving the search of solutions towards the Pareto frontier, achieving a diverse and width front with uniformity regarding such solutions. Once achieved the solutions front to the problem, the use of a quality indicator is needed. Two basic ways to classify the indicators are considered: Attending to the aspects that indicators measure when the approximation set or solutions front is evaluated and, attending to the number of approximation's sets or solutions' fronts which are evaluated [74]. In the first way, the indicators can be grouped as follows [75]:

- Cardinality indicators: Indicators which count the number of solutions contained by the approximation set.
- Precision indicators: Indicators which indicate the distance to the theoretical Pareto optimal front. A reference set is taken into account when the theoretical Pareto optimal front is not known.

- Diversity indicators: Indicators which refer to the distribution and extension of solutions along the front.

In the second way, indicators differ between two types:

- Unary indicators: Indicators which receive an approximation set as a parameter.
- Binary indicators: Indicators which receive two approximation sets to be compared.

Many indicators and quality indicators have been proposed in order to measure and compare the performance of solutions or approximations supplied by methods. The most used in the Multi-objective Evolutionary Optimisation [74] are the Hypervolume [72], the Generational Distance [76] and Epsilon (ϵ) [77]. In the case of the present research, the Hypervolume is used as an indicator. This unary indicator considers the aspects previously cited: Precision, diversity and cardinality. It is an indicator widely accepted and used because it offers the following features [78]:

- When an approximation set dominates to other one, the Hypervolume of the dominant set is bigger than the other set is.
- The approximation set that reaches the maximum Hypervolume value regarding a problem contains all the Pareto optimum solutions.

As an indicator capable of measuring the precision, the Hypervolume requires to select a reference point to compute the space covered by the solutions in the objective space.

3. CHAPTER 3: MULTI-OBJECTIVE OPTIMISATION BY USING EVOLUTIONARY ALGORITHMS IN THE RELIABILITY FIELD.

3.1. Generalities.

In a changing world regarding the technological development, the greater the possibility of implementing reliable systems, the greater the demand from consumers. This implies requesting high features in the contracted services and purchased products. Such a circumstance is not only a challenge but also an opportunity. Therefore, optimising the performance of physical assets is vital in a highly competitive market. Optimisation involves, in formal terms, defining the decision variables, the constraints and the objective function or functions, which describe the performance for the engineering problem. From such a definition, looking for the combination of the values of the decision variables is managed in order to achieve the wished objective.

From the structural design optimisation of complex systems point of view, Coit and Zio [79] classified the eras of the research evolution as follows:

- Era of Mathematical Programming,
- Era of Pragmatism,
- Era of Active Reliability Improvement.

They think that the Era of Mathematical Programming made possible to place the methodology bases to solve optimisation problems regarding the systems' Reliability. The viability of the methods and their applications were demonstrated. It is possible to emphasise, from such an era, the Dynamic Programming [80], the Linear Programming (or Entire Programming) [81], the Non-Linear Programming [82] and the Evolutionary Algorithms, whose bases were proposed by Holland. Such an author introduced the fundamental concepts regarding the Evolutionary Algorithms [83]. The methodologies developed throughout this era allowed solving reliability optimisation problems in a mathematically rigorous manner.

The Era of Pragmatism arose from the need to address more complex problems. More realistic behaviours were considered from the Reliability point of view. In this era, the Genetic Algorithms and other metaheuristics presented a dominant use. It is possible to emphasise as contributions of such an era the treatment of Multi-state Systems [84], Uncertainty [85] or several kinds of Redundancies [86].

Finally, the Era of Active Reliability Improvement was born from the new capabilities to compile and transmit information regarding the failure of devices in changing environments by using sensors in order to analyse and re-evaluate the system's Reliability. In this field, it is possible to emphasise the Dynamic Systems Reliability Models [87] and the System Reliability Optimisation customised for specific subset of users [88].

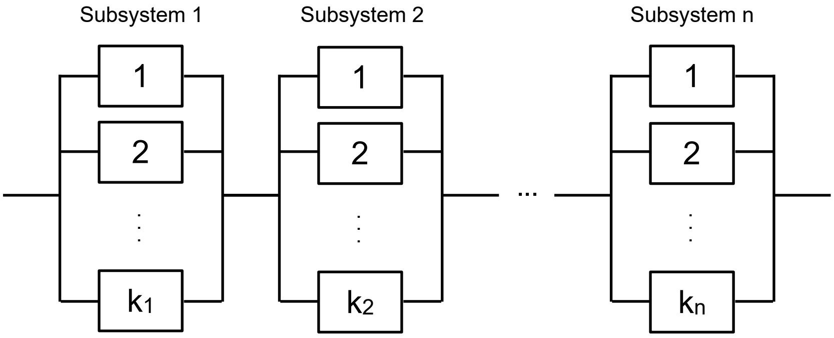


Figure 3.1: Series-parallel system.

From the design of complex systems point of view, the redundancy allocation problem is one of the more studied. This one is handled along the present research. The basic problem consists of optimising the structural design of a series-parallel system such as the one shown in the Figure 3.1. Deciding the number of redundant devices is needed for the several sub-systems in order to fulfil the objective. Such an objective may consist of maximising the Availability while operational Cost is minimising. In the present research, the problem is addressed in stages as it will be exposed later. At the beginning, a small dimension problem will be studied where the number of design alternatives is reduced. Next, a more complex problem will be solved where the number of redundant devices must be decided regarding several sub-systems. Therefore, the number of design alternatives will be bigger. In order to solve the problem, Evolutionary Algorithms are used as the basis of the methodology. Furthermore, attending to the proposed classification by Coit et al., the present study goes into the uncertainty dealing field. Distinguishing between random and epistemic uncertainty is common in the Reliability and Risk Analysis field [89]. The present research is closer to the first one due to the fact that it is referred to random phenomena, so the probabilistic model is more appropriate when such phenomena must be described. Such a randomness is employed when times to failure and times to repair are generated for the devices that form the system. These times follow their respective probability density functions. Such a functions are characterised from the parameterisation of their features, which are achieved from operational data statistical analysis. Some publications can supply such data [90] [91]. Finally, active redundancy is the type of redundancy that is considered for the present research. Such a redundancy involves that all devices are active along the mission time, so the system fulfil the required function while opportune redundancies work.

Up to now, the present section has attended to the Reliability optimisation from the structural design point of view. Speaking in general, devices suffer from wear due to internal (derived from operation) and external factors (derived from the environment where they work). Nowadays, companies are aware of maintenance activities have consequences in benefits. Such activities are the most relevant costs along the life cycle of the system. A better control of non-available times of systems are directly related to production and such control is possible by conducting preventive maintenance activities.

An interesting review regarding the maintenance optimisation was supplied by Jonge y Scarf [92], where a first differentiation between systems with single-unit and multi-units were done. A single-unit system is a system which cannot be decomposed in lower maintainable levels. Therefore, such a system is considered as a unit from the maintenance point of view. Conversely, a multi-unit system is a system which can be decomposed in lower-level maintainable units. In the present research, multi-unit systems are considered. De Jonge et al. continue the categorisation by attending to the deterioration states space, which are considered as follows:

- Deterioration process with two states: Operating and failed states.
- Deterioration process with three states: A third state is including between the operating and failed states. In this case, the states are called good, defective and failed (the delay-time model).
- Deterioration process with a discrete state space: This model presents a countable and finite number of states.
- Deterioration process with a continuous state space: In this case, the deterioration level can take values within a specific interval.

A deterioration process with two states is considered for the present research. The devices that form the system design can satisfy or not the required conditions. Moreover, both preventive and corrective maintenance activities are considered in relation to such devices. De Jonge et al. consider that maintenance activities should be planned based on time, age, use or condition. In the case of the present research, the preventive maintenance activities are planned based on calendar times. Prognostic, predictive and condition based are terms associated with preventive maintenance that is planned based on condition reports. In the case of the present research, the corrective maintenance activities are conducted based on condition because such activities are carried out depending on the device state. The preventive maintenance activity starts immediately once a device does not satisfy the required function so the continue monitorization of the system is considered. Attending to the repairs, these can be considered as perfect (the device recovers the as-good-as-new state) or imperfect. For the present research, repairs are considered as perfect, so the corrective maintenance activities restore the as-good-as-new state. Regarding maintenance, dependencies between devices are not considered, so each device works isolated. Therefore, several opened lines exist for future research.

3.2. Design optimisation: Redundancy allocation problem.

As explained before, regarding systems design optimisation on the Reliability field, the redundancy allocation problem has been widely studied. Solving such a problem follows increasing the systems' Reliability (or Availability) by adding redundant devices to the basic design.

Several techniques have been employed in order to solve such a problem, such as Dynamic Programming [93], Linear Programming [94] or Non-Linear Programming [82]. However, from the present research point of view, addressing the problem by using Evolutionary Algorithms is of interest.

3.2.1. Solving the problem by using Evolutionary Algorithms.

Few studies were conducted regarding Evolutionary Algorithms in the Reliability field [95] before the one conducted by Coit and Smith [96]. They developed a specific Genetic Algorithm to the design optimisation of series-parallel systems and multiple component choice. Such a study sets out two alternatives: Reliability maximisation or cost minimisation, both subjected to constraints. This brings into the light the single-objective character of such an approach. Other studies that follow a similar approach were conducted after:

- Levitin et al. [84] presented an optimisation model to multi-state systems, where the redundancy allocation problem was solved. Both the system and the devices present a range of performance levels. They employed a Genetic Algorithm where the cost is minimised by considering the performance level.
- Cantoni et al. [97] coupled Genetic Algorithms and Monte Carlo Simulation to the optimal industrial plant design. Their approach follows maximising an objective where several costs (acquisitions, repairs, penalties regarding downtime and environment damages) are subtracted to the profit.
- Ouzineb et al. [98] managed the redundancy allocation problem for series-parallel systems with multi-state reparable components (attending to several reliability levels). They developed an original tabu search meta-heuristic

optimisation method. It determines the minimal cost system configuration when constraints are based on availability.

- Bendjeghaba y Ouahdi [99] studied the redundancy allocation problem for a multi-state power system by using an Ant Colony System as an optimisation method. They minimise the cost when availability restrictions are considered.
- Ying-Shen et al. [100] proposed an optimisation model based on a Genetic Algorithm in order to improve the design efficiency. Their approach follows maximising the relationship between availability and cost when the problem is subjected to restrictions. Such restrictions consider the Mean Time to Failure and the Mean Time to Repair.
- Zou et al. [101] proposed a Harmony Search Algorithm to solve reliability problems. Their approach looks for maximising the reliability based on several constraints such as cost, volume of the component and its weight.
- Wang and Li [102] proposed a hybrid algorithm based on Particle Swarm Optimisation and Local Search. The authors employed the cost under availability constraints as an objective function.
- Valian et al. [103] used an improved version of Cuckoo Search Algorithm for reliability optimisation problems where reliability is maximised by using a single objective function. Such an objective function includes constraints.
- Pourkarim et al. [104] presented a different approach in relation to the classic formulation for the redundancy allocation problem. In this case, they considered a three-states model (working with full performance, working with half performance and a failed state). They employed a Genetic Algorithm to maximise the system’s reliability when a constraint is considered. Such a constraint is form by the total cost of the system, which includes the cost of redundant components, the internal connection cost, and the technical and organizational activity costs.
- Heish [105] proposed an Evolutionary Algorithm inspired by the bacterial behaviour, which is applied to the redundancy allocation problem in multi-level systems. As an objective function, the system reliability is maximised when cost constraints are considered.

- Xu et al. [106] focused the redundancy allocation problem for multi-state systems that are subjected to common failure causes. They proposed a novel Discrete Bat Algorithm in which they minimise the cost based on availability constraints.

Other authors proposed the simultaneous optimisation of several objectives by aggregation. In this case, an objective function is built in which objectives are accumulated and (sometimes) weighted. Although multiple objectives are optimised at the same time, a single objective function is attended. Some studies under this philosophy are shown as follows:

- Elegbede y Adjallah [107] described a methodology based on Genetic Algorithms in which the authors optimised availability and cost for series-parallel reparable systems. They converted the multi-objective optimisation problem into a single-objective optimisation problem by using a weighting technique.
- Sahoo et al. [108] employed a Genetic Algorithm to solve the optimum design of systems based on evaluating reliability intervals. As objective functions, they considered maximum reliability and minimum cost, weight and volume. Their approach consisted of building a single objective function where the objectives are related.
- Li et al. [109] proposed a two-stages approach for system reliability optimisation. At first, a set of Pareto solutions is identified by applying Multi-objective Evolutionary Algorithms (NSGA and NSGA-II). Next, the front is reduced by using a Multi-objective Selection Algorithm. As an objective, the authors maximise a function to measure the relative performance in which profit and cost are related.
- Azadeh et al. [110] studied an optimisation problem to planning redundancies for a system with warm stand-by redundancy (failure probability bigger than zero and lower than the failure probability for an operating device) where the devices are considered as non-reparable and multi-state. Genetic Algorithms were used to solve the problem and as objective functions they employed

acquisition cost, time to develop the activities and reliability. They used a single function, which involves such parameters.

However, other authors considered the redundancy allocation problem by employing Evolutionary Algorithms from a multi-objective point of view. This is the interesting focus regarding the present research. Next, a significant set of contributions in such a field are cited:

- Bussaca et al. [111] employed a Multi-objective Genetic Algorithm to optimise the design of a safety system by considering redundancy allocation. As objective functions, they used, on the one hand, a profit function in which several costs (acquisition, installation, repairs, and penalties due to missed delivery of the agreed service) are subtracted regarding the benefits, and on the other hand, the system reliability. The authors considered that the devices are subjected to constant failure rates.
- Marsaguerra et al. [112] coupled Genetic Algorithms and Monte Carlo Simulation to solve the network optimum design problem. They looked for maximising the estimated network reliability and minimising its associated variance. Not only component types with uncertain regarding reliability but also redundancy levels were considered as decision variables.
- Greiner et al. [5] introduced original methodologies for the optimum design of safety systems from a multi-objective point of view (based on fault trees which are evaluated by using the weight method) by employed several Evolutionary Multi-objective Algorithms (SPEA2, NSGA-II and NSGA-II with controlled elitism). As objective functions, the authors considered unavailability and cost.
- Tian and Zuo [113] proposed a multi-objective optimisation model to redundancy allocation for multi-state series-parallel systems. The authors employed a Genetic Algorithm to solve the optimisation model based on physical programming. As objective functions the authors maximise the performance of the systems and minimise cost and weight.
- Salazar et al. [7] developed a multi-objective formulation to solve several problems regarding the optimum design of systems. The authors used

NSGA-II as an optimisation methods and reliability and cost as objective functions.

- Zhao et al. [114] employed the first Ant Colony Algorithm to reliability optimisation based on series-parallel systems. As objective functions they considered reliability, cost and weight.
- In order to interpret the high number of obtained solutions when the Pareto optimum set is achieved for a problem, Taboada et al. [115] presented two methods to reduce the solutions on a Pareto optimum set. The first one is based on a pseudo-ranking scheme and the second one uses crowding techniques proper from data mining. They solved the redundancy allocation problem to demonstrate the efficacy of the methods. They used the genetic algorithm NSGA in order to achieve an initial set of optimum solutions. As objective functions, they maximise reliability and minimise cost and weight.
- Chiang and Cheng [116] proposed a Multi-objective Genetic Algorithm based on simulated annealing to solve optimisation and redundancy allocation problems for series-parallel reparable systems. They considered two cases based on references above cited; Elegbede et al. [107], where maximum availability and minimum cost are pretended, and Busacca et al. [111], where both availability and the profit function, which subtracts costs to benefits, are maximised.
- Limbourg and Kochs [117] applied feature models (an original method from software engineering for complex design) and a Multi-objective Evolutionary Algorithm (derived from NSGA-II with external repository) for probabilistic purposes. As objective functions they maximise the life distribution of the system and minimise its cost.
- Tobaoda et al. [118] introduced a Multi-objective Evolutionary Algorithm based on Genetic Algorithms to solve multiple objective multi-state reliability optimization design problems. They evaluate the reliability system indexes by using an approach based on Universal Moment Generating Functions. As objective functions they maximise the system's reliability and minimise both cost and weight.

- Kumar et al. [119] proposed a multi-objective formulation and a methodology to solve the multi-level redundancy allocation problem by introducing a hierarchical environment based on Genetic Algorithms (SPEA2 and NSGA-II). As objective functions, reliability and cost are considered.
- Huang et al. [120] proposed a niched Pareto Genetic Algorithm based to solve reliability design problems. By using such an approach, they look for a high number of feasible solutions in order to supply a high number of alternatives to choose by the decision maker. As objective functions they employed reliability and cost.
- Lins et al. [121] coupled Genetic Algorithms and Monte Carlo Simulation to solve the redundancy allocation problem for series-parallel reparable systems subjected to corrective maintenance activities. As objective functions they considered reliability and cost.
- Lins et al. [122] coupled Genetic Algorithms and Monte Carlo Simulation to solve the redundancy allocation problem for series-parallel reparable systems subjected to imperfect repairs. As objective functions they employed availability and cost, where acquisition, repair, maintenance teams and unavailability costs were considered.
- Chambari et al. [123] formulated the redundancy allocation problem for non-reparable systems. The decision is taken by choosing between active and cold-standby components and deciding the redundancy level regarding each subsystem. The authors considered both non-constant failure rates and imperfect transitions to redundant cold-standby components. They employed two metaheuristics, the Genetic Algorithm NSGA-II and optimisation by Particle Swarm Optimisation. As objective functions they used maximum reliability and minimum cost.
- Safari [124] proposed a variant for the Genetic Algorithm NSGA-II to solve his new mathematic formulation in order to supply solutions to the redundancy allocation problem. They considered both active and cold-standby devices. As objectives they employed reliability and cost.
- Khalili-Damghani et al. [125] proposed a novel multi-objective Particle Swarm Optimization method for solving reliability redundancy allocation problems.

They considered two states for the devices. As objective functions the authors considered reliability, which is maximised, and cost and weight, which are minimised.

- Zoulfaghari et al. [13] presented a Non-Linear Programming mixed model to analyse the availability optimisation for a system with a specific structure, which uses both repairable and non-repairable components. In order to localise the solution to the proposed model, the authors developed an efficient Genetic Algorithm to maximise reliability and minimise cost.
- Jiansheng et al. [126] introduced uncertainty theory regarding failure rates, repair rates and other involved coefficients to solve the redundancy allocation problem for repairable series-parallel systems. In order to solve such a problem, they proposed an algorithm based on Artificial Bee Colony. As objective functions the authors employed reliability and cost.
- Ardakan et al. [127] solved the redundancy allocation problem by proposing to use mixed redundancies, which are formed by the combination of active and cold-standby redundancies or without failure up to start operating. They employed NSGA-II as an optimisation method and reliability and cost as objective functions.
- Ghorabaee et al. [128] considered the redundancy allocation problem with *k-out-of-n* sub-systems. As objective functions the authors considered reliability and cost when weight is considered as a constraint. The authors were based on NSGA-II as an optimisation method, although they introduced a modification to preserve the diversity and to manage the constraints.
- Amiri and Khajeh [129] considered the redundancy allocation problem for repairable systems. They used NSGA-II as an optimisation method and reliability and cost as objective functions.
- Jahromi and Feizabadi [130] developed a formulation to solve the redundancy allocation problem when components were not considered homogeneous. Reliability and cost were taken as objective functions whereas NSGA-II was used as an optimisation method.
- Kayedpour et al. [131] developed an integrated algorithm to solve reliability design problems considering instantaneous availability, repairable

components and a selection of configuration strategies (parallel, cold or warm) based on Markov processes and the NSGA-II method. As an optimisation method, the authors took maximum availability and minimum cost.

- Samanta and Basu [132] proposed an Attraction-based Particle Swarm Optimisation (APSO) model to solve availability allocation problems for systems with repairable components. The authors considered the non-linear behaviour for the system by introducing fuzzy theory to manage uncertainties. They used availability and cost as objective functions. The traditional Particle Swarm Optimisation model was improved in this case.
- Sharifi et al. [133] presented an original multi-objective model to solve the redundancy allocation problem for systems with sub-systems considered as weighted-k-out-of-n parallel. They used NSGA-II as an optimisation method and reliability and cost as objective functions.
- De Paula et al. [134] proposed a solution to the redundancy allocation problem in which dependency between failures for redundant devices is considered. They employed a stochastic approach based on Markov chains and next, they solved the multi-objective problem by using the Genetic Algorithm NSGA-II. As objective functions they maximise availability and minimise cost. As a result, they show the number of redundant devices and the percentage of resources to use when maintenance activities must be done.
- Chambari et al. [135] proposed a bi-objective simulation-based optimisation model to redundancy allocation in series-parallel systems with homogeneous components. The authors maximise the system reliability and minimise the cost, while NSGA-II is used as optimisation method. Optimal component types, the redundancy level, and the redundancy strategy (active, cold standby, mixed or K-mixed) with imperfect switching were considered.

3.2.2. Considerations for the optimum design in the present research.

In the Reliability field, the most widely studied multi-objective design optimisation problems look for maximising reliability and minimising cost. Similar objectives are followed in the present research, although in this case, the objective to maximise consists of availability since repairable systems are exclusively considered.

The present research addresses the design optimisation problem regarding redundancy allocation by stages. In a first stage, a small dimension problem is solved in which a few design alternatives are considered. In a second stage, a bigger dimension problem is considered, in which the number of redundant devices must be chosen for each device included in the basic design of the system. Such stages allow testing the methodology at first in a lower complex case study, due to the fact that the present research not only looks for the optimum design of the system (as in the bibliography above cited) but also looks for determining simultaneously the optimum maintenance strategy (considering both corrective and preventive maintenance), as it was explained before.

3.3. Preventive maintenance optimisation.

As it was exposed before, de Jonge and Scarf [92] distinguish maintenance tasks in two blocks, corrective and preventive maintenance activities. Although some references cited regarding the systems design optimisation attend to corrective maintenance [13,118,122,125,129,131,132], the present research considers both corrective (repairs after failure) and preventive maintenance. Such preventive maintenances activities are attended next.

The necessity of considering preventive maintenance activities was identified by industries a few decades ago. Before companies would be aware of considering preventive maintenance in order to improve the efficiency and reliability of processes, maintenance activities were carried out after failure. When a repairable system is not available, such a system enters an unproductive phase [136] where not only resources are not generated, but also, they are consumed until recovering the system's available state. A repairable system is not available because either a failure (after such a failure a repair or corrective maintenance activity is required) or a programmed shutdown in order to conduct a preventive maintenance activity. When a preventive maintenance activity is carried out, the unproductive phase is better controlled than when repairs must be performed because of a failure. Such a situation is due to circumstances such as the spares are located and available or maintenance human teams are trained and prepared.

This research considers the preventive maintenance activities planning. Several techniques have been employed to solve such a problem by different authors. For instance, Integer Programming by Kralj et al. [137] or Mixed Integer Linear Programming by Charest et al. [138]. However, the present research attends to solve the problem by using Evolutionary Algorithms.

3.3.1. Preventive Maintenance Planning by using Evolutionary Algorithms.

Evolutionary Algorithms have been widely employed to solve the preventive maintenance planning problem, both from a single-objective point of view and a multi-objective point of view. Some references can be cited regarding the single-objective optimisation by employing Evolutionary Algorithms.

- Marseguerra and Zio [139] coupled Genetic Algorithms and Monte Carlo Simulation within the context of plant logistic management. This involves taking decisions regarding maintenance and repairs strategies. Monte Carlo simulation allows considering practical aspects such as stand-by operation modes, deteriorating repairs, sequences of periodic maintenances, number of repair teams available for different kinds of repair interventions. The objective function consists of maximising a profit function in which several cots are considered.
- Tsai et al. [140] developed a periodic preventive maintenance plan for a system subjected to deteriorated component. Simple preventive maintenance and preventive replacement are considered as preventive

maintenance activities. The degradation of components is modelled by a dynamic reliability equation, whereas the effect from preventive maintenance activities is based on an age reduction model. The combination of activities for a component regarding the maintenance cost and the improvement of the life of the system are considered. In order to decide the best combination of activities related to each preventive maintenance period, a Genetic Algorithm is employed. Such a Genetic Algorithm maximises an index, which measures the best combination of activities.

- Bris et al. [141] presented a methodology to minimise the cost derived from preventive maintenance activities for series-parallel systems based on the Birnbaum importance factor. The authors employed Monte Carlo simulation to evaluate the system availability, and Genetic Algorithms to minimise the cost while some constraints regarding the computed availability are considered. Afterwards, Samrout et al. [142] started from this approach and applied the Ant Colony Algorithm. In this case, the results were improved.
- Lapa et al. [143] defined inspections as a specific class of preventive maintenance activity and maintenance as a non-periodic task. From such ideas the authors developed a methodology to plan and optimise the survival test policy base on Genetic Algorithms, minimising the average system unavailability during the mission time.
- Lapa et al. [144] presented a methodology to evaluate the preventive maintenance policy based on a cost-reliability model, which allows using flexible intervals between maintenance activities. In order to solve the problem, the authors employed a Genetic Algorithm, in which the objective function is a linear combination of the impact from a specific maintenance policy and the cost regarding such a policy.
- Hadivi [145] presented a novel approach based on Genetic Algorithms to optimise the maintenance calendar for a nuclear plant during its renovation. To do that, the cessation of electricity supply is considered. As an objective function, three parameters are weighted: risk, cost and the losses incurred when a maintenance schedule is created.

- Wang and Lin [146] presented a Genetic Algorithm to minimise the preventive maintenance cost for series-parallel systems. Such an algorithm considers the intrinsic properties of reparable systems, when the structure reliability of block diagrams and the maintenance priorities of components are attended. In this way, the component importance is measured, the more relevant components are identified, and the maintenance priority is stablished. The optimum maintenance periods are determined by minimising the total cost subjected to reliability constraints.
- Wang and Lin [147] determined the optimum maintenance periods for the component of series-parallel systems by using an improved Particle Swarm Optimisation method. The cost derived from the periodic preventive maintenance is minimised as an objective function.
- Lin and Wang [148] presented a Hybrid Genetic Algorithm to optimise the periodic preventive maintenance for series-parallel systems. The Genetic Algorithm considers the intrinsic properties of repairable systems, when the structure reliability of block diagrams, the maintenance priorities of components and their maintenance periods are attended. In this way, the component importance is measured, the more relevant components are identified, and the maintenance priority is stablished. The optimum maintenance periods are determined by minimising the total cost subjected to availability constraints.
- Wong et al. [149] proposed a method to plan both production management and the maintenance tasks, when several types of maintenance tasks and needed resources are considered. A Genetic Algorithm is considered in which the time interval between the first job and the final job is minimised (makespan).
- Zade and Fakhrzad [150] solved the periodic maintenance planning problem for a machine with non-resumable jobs. The authors minimised the time between the first job of the first machine and the final job of the last machine (makespan). Moreover, two strategies are attended: on the one hand, to develop the maintenance after a determined time period, and on the other and, to consider the maximum number of jobs that a special tool can

develop. A dynamic Genetic Algorithm is proposed to minimise an objective function that depends on the needed time to carry out the maintenance tasks, the jobs without workers (because other jobs were assigned to such workers) and work times.

- Zheng et al. [151] suggested using condition-based maintenance as a preventive maintenance strategy. They solve the flexible job shop scheduling problem when preventive maintenance is included. An integrated Genetic Algorithm is applied to plan flexible jobs and afterwards the preventive maintenance is inserted to the solution by an insertion algorithm. As an objective function, the time between the first and the last job (included preventive maintenance) is minimised.
- Canh Vu et al. [152] presented a dynamic maintenance grouping strategy for multi-component systems based on the economic cost. Such a cost may be improved due to grouping maintenance activities or get worse because of shutdowns of the system. The authors developed a Genetic Algorithm to maximise the economic profit derived from grouping maintenance activities.
- Yin et al. [153] developed an integrated model of statistical process control and maintenance decisions. Depending on a monitoring policy based on calendar, it is carried out corrective maintenance after failure or preventive maintenance in case of loss of production.
- Xiao et al. [154] proposed a model to the simultaneous optimisation of production scheduling and machine group preventive maintenance for systems by employing a Genetic Algorithm. Such a Genetic Algorithm minimises the global cost due to loss of production, preventive maintenance, minimal repair for unexpected failures and tardiness.
- Maatouk et al. [155] solved the preventive maintenance optimisation problem for multi-state series-parallel systems. They employed a Genetic Algorithm controlled by fuzzy logic. The cost function is minimised with the restriction of the required availability at determined time.
- Zhang and Zeng [156] studied the joint optimisation of the strategy for periodic condition-based opportunistic preventive maintenance and a policy for the provision of spare parts. Depending on the state of deterioration of

the system and spare parts inventory, maintenance activities are ordered. The authors employed a Genetic Algorithm to minimise a cost function in which inspections, maintenance and spare parts are considered.

- Yahyatabar y Najafi [157] solved the problem of minimising the periodic preventive maintenance for series-parallel systems. The authors determine the number of preventive maintenance activities for each system device regarding reliability constraints. An Invasive Weed Optimization Algorithm were used in which the cost due to preventive maintenance actions subjected to reliability constraints is minimised.
- Rahmati et al. [158] developed an integrated condition-based maintenance and stochastic flexible job shop scheduling problem. Such a problem attends to both corrective and preventive maintenance. To do that, the degradation level of the component and a threshold are compared in order to start the preventive maintenance tasks. A Harmony Search Optimisation Algorithm is used to solve the problem. The time between the first and the last task is minimised in this case.
- Dahia et al. [159] proposed a maintenance model that considers three actions: minimal repairs, a periodic overhaul and a complete renewal. Employing a Genetic Algorithm as an optimisation method, the minimum cost for preventive maintenance is determined for multi-component systems when reliability constraints are attended.

Other authors dealt with the various objective simultaneous optimisation, although they do that by aggregation. In this case, the authors build a single objective function in which the objectives are accumulated and (in occasions) weighted. Some studies under such a consideration are:

- Quan et al. [160] considered to find out a cost-effective schedule to achieve a balanced solution between the task to plan and the size work force. To do that, the authors employed a non-conventional Multi-objective Evolutionary Algorithm by introducing a form of utility theory to find Pareto optimal solutions. As objective functions, they minimise simultaneously the number of workers and time to develop a set of preventive maintenance tasks.

- Carlos et al. [161] employed Particle Swarm Optimisation and a tolerance interval-based approach to manage uncertainty, in order to achieve a flexible variation rank to develop the preventive maintenance activities instead of supplying a constant time interval. The authors used a single objective function in which the effectiveness of the availability and the cost are weighted.
- Balaji et al. [162] formulated the maintenance scheduling problem by exact time intervals for power generation units. They used a mixed integer optimization mathematic model which is attended by Differential Evolution in order to minimise the operational cost. To do that, the authors build a cost function in which production, operation and maintenance cost are considered.
- Zhang et al. [163] proposed a new economical optimisation model to make up decisions about non-periodic maintenance for deteriorating system. The authors designed a Particle Swarm Optimisation Algorithm that combines heuristic rules to solve the multi-objective problem. Such a problem is formulated in order to minimise the total cost, which considers the maximum maintenance periods, the maximum number of inspections per maintenance period and the inspection interval per maintenance period.

Nevertheless, the present research studies how to solve the preventive maintenance scheduling problem by employing Multi-objective Evolutionary Algorithms. Under such an umbrella, it is possible to cite the studies conducted by several authors:

- Muñoz et al. [164] presented an approach based on Genetic Algorithms and focused on the global and constrained optimisation of surveillance and maintenance of components based on risk and cost criteria.
- Marseguerra et al. [165] coupled Genetic Algorithms and Monte Carlo simulation in order to optimise profit and availability when maintenance and repair strategies must be considered. The Monte Carlo simulation was used to model the system's degradation while the Genetic Algorithm was used to

determine the optimal degradation level beyond which preventive maintenance must be performed.

- Martorell et al. [166] proposed a methodology for the Integrated Multi-Criteria Decision-Making to determine the parameters of technical specifications and maintenance equipment based on safety. The authors employed a Genetic Algorithm when reliability, availability and maintenance were the considered criteria.
- Gao et al. [167] studied the flexible job shop scheduling problem attending to availability constraints which affect maintenance tasks. They used a Genetic Algorithm in order to minimise the makespan (time that elapses from the start of work to the end), time expended on a machine and the total time expended on all machines.
- Oyarbide-Zubillaga et al. [168] coupled Discrete Event Simulation and Genetic Algorithms (NSGA-II) to identify the optimal preventive maintenance frequency for multi-equipment systems under cost and profit criteria.
- Berrichi et al. [169] proposed a method to solve the simultaneous production and maintenance scheduling problem. The authors used the Weighted Sum Genetic Algorithm (WSGA) and NSGA-II as optimisation methods to compare their performances. As objective functions, they worked with makespan and unavailability due to maintenance tasks.
- Sánchez et al. [170] employed Genetic Algorithms for the optimisation of testing and maintenance tasks with unavailability and cost as objective functions. The authors considered the epistemic uncertainty in relation to imperfect repairs.
- Berrichi et al. [171] solved the joint production and preventive maintenance-scheduling problem by using a Multi-objective Ant Colony Algorithm and taking availability and cost as objective functions.
- Moradi et al. [172] studied simultaneously the production and preventive maintenance scheduling problem in order to minimise the global time invested in production tasks and the unavailability due to preventive maintenance activities. They used four Genetic Algorithms: NSGA-II, NRGA (Non-ranking Genetic Algorithm), CDRNSGA-II (NSGA-II with Composite

Dispatching Rule and active scheduling) and CDRNRGA (NRGA with Composite Dispatching Rule and active scheduling).

- Wang and Pham [173] employed a Genetic Algorithm to estimate the preventive maintenance interval by considering imperfect repairs and the number of preventive maintenance activities before a component needs to be replaced. They used availability and cost as objective functions.
- Ben Ali et al. [174] developed an elitist Genetic Algorithm to deal with the production and maintenance-scheduling problem. To do that, the authors minimise both makespan and cost.
- Hnaien and Yalaoui [175] considered the production and maintenance-scheduling problem by minimising the makespan and the delay between the real and the theoretical maintenance frequency for two machines. They used NSGA-II and SPEA2, including two novel versions based on the Johnson Algorithm.
- Suresh and Kumarappan [176] presented a model for the maintenance scheduling of generators employing hybrid Improved Binary Particle Swarm Optimisation (IBPSO). As objective functions, the authors considered a reduction in the loss of load probability and minimisation of the annual supply reserve ratio deviation for a power system.
- Li et al. [177] presented a novel Discrete Artificial Bee Colony Algorithm for the flexible job-shop scheduling problem in which maintenance activities are considered. The authors employed as objective functions the makespan, the total workload of machines and the workload of the critical machine.
- Gao et al. [178] studied the preventive maintenance considering the dynamic interval for multi-component systems. The authors solved the problem by using Genetic Algorithms whereas availability and cost were considered as objective functions.
- Wang and Liu [179] considered the optimisation of parallel-machine-scheduling integrated with two kinds of resources (machines and moulds) and preventive maintenance planning. The authors employed makespan and availability as objective functions and NSGA-II as an optimisation method.

- Piasson et al. [180] proposed a model to solve the problem of optimising the reliability-centred maintenance planning of an electric power distribution system. The authors employed NSGA-II in order to achieve the Pareto optimal front and, as objective functions, they minimised the cost due to maintenance activities and maximised the index of reliability of the whole system.
- Sheikhalishahi et al. [181] presented an open shop scheduling model that considers human errors and preventive maintenance. They considered three objective functions: human error, maintenance and production factors. The authors used NSGA-II and SPEA2 as optimisation methods. As well as that, they used another Evolutionary Algorithm, the Multi-Objective Particle Swarm Optimisation (MOPSO) method.
- An et al. [182] built an integrated optimisation model including the flexible job-shop scheduling problem to reduce the energy consumption in the manufacturing sector. The authors considered the degradation effects and imperfect maintenance. They proposed a Hybrid Multi-objective Evolutionary Algorithm considering the makespan, total tardiness, total production cost and total energy consumption as objective functions.
- Boufellouh and Belkaid [183] proposed a bi-objective model, which determines the production scheduling, the maintenance planning and the resource supply rate decisions in order to minimise the makespan and the total production costs (including maintenance, resource consumption and resource inventory costs). The authors used NSGA-II and BOPSO (Bi-Objective Particle Swarm Optimization) as Evolutionary Optimisation Algorithms.
- Bressi et al. [184] proposed a methodology to minimise the present value of the life cycle maintenance costs and maximise the life cycle quality level of the railway track-bed by considering different levels of reliability. They used a Genetic Algorithm to achieve optimal solutions.
- Zhang and Yang [185] proposed a multi-objective model of maintenance planning and resource allocation for wind farms by using NSGA-II. The

authors considered the implementation of maintenance tasks by considering the minimal total resources and at the minimal penalty cost.

3.3.2. Considering the optimum preventive maintenance strategy in the present research.

The preventive maintenance scheduling optimisation problem more studied from the multi-objective point of view consists of maximising availability (or minimise unavailability) and minimising costs. Such objectives are explored in the present research.

Particularising, the periodic preventive maintenance scheduling optimisation problem is considered so the main target consists of determining the periodic optimum interval between preventive maintenance tasks regarding each one of the system's devices. Furthermore, several time units are explored so preventive maintenance tasks can be defined by using accuracy levels as it was claimed between the objectives of the present research.

3.4. Simultaneous optimisation of the design and the preventive maintenance scheduling.

Up to now, several studies developed from different authors in the Reliability field have been cited. Such studies face on the optimisation of either the system's structural design or its preventive maintenance scheduling. However, the present research considers the simultaneous optimisation of both the system's structural design (by redundancy allocation) and its preventive maintenance scheduling (by determining the periodic time to conduct preventive maintenance tasks). The objectives considered are maximum availability and minimum cost. Some studies regarding such a research line were carried out before, as an example, Zhu et al. [186] considered the redundancy allocation problem and the sequential preventive maintenance scheduling when imperfect repairs are considered. The authors presented a stochastic programming model which is subjected to uncertainty regarding how the system is going to be used in future. However, the present research focuses on Evolutionary Algorithms (due to their demonstrated power) and Discrete Event Simulations (due to its capacity of representing the behaviour of systems).

3.4.1. Simultaneous optimisation of systems design and their preventive maintenance scheduling by using Evolutionary Algorithms.

Several authors focused on solving the problem from a single-objective point of view.

- Levitin and Lisnianski [187] presented the first formulation of the joint redundancy and maintenance optimisation problem for multi-state systems by using a Genetic Algorithm as an optimisation method. The objective function consisted of adding the acquisition costs, the maintenance costs and the penalties due to unsatisfied demand.
- Monga and Zuo [188] developed a model to design series-parallel systems based on reliability. Such a model considers the deterioration of components in order to minimise the life cycle cost. Moreover, the authors proposed several equations to model the effects of preventive maintenance in the failure rates of systems and quantify the protection rate. They optimise the annual average cost of the system by using Genetic Algorithms.
- Nourelfath et al. [189] formulated a joint redundancy and imperfect preventive maintenance planning optimisation model based on Markov processes and universal moment generating function, in order to evaluate availability and cost for multistate systems using Genetic Algorithms and Tabu search.

A few studies were conducted up to now under the umbrella of Multi-objective Evolutionary Algorithms.

- In Galván et al. [8], a methodology for Integrated Safety System Design and Maintenance Optimisation based on a bi-level evolutionary process was presented. While the inner loop is devoted to find the optimum maintenance strategy for a given design, the outer loop looks for the optimum system's design. The authors used Multi-objective Evolutionary Algorithms based on domination criterion (such as NSGA-II) as optimisation methods and several codifications were compared. Cost and unavailability were used as objective functions.

- Okasha and Frangopol [190] considered the simultaneous optimisation of design and maintenance during the life cycle by using Genetic Algorithms (NSGA-II). They studied the system reliability, redundancy and life-cycle cost as objective functions.
- Adjoul et al. [191] described an original approach to the simultaneous optimisation of design and maintenance of multi-component industrial systems improving their performances with reliability and cost as objective functions. They used a two-level Genetic Algorithm based on NSGA-II: the first optimises the design based on reliability and cost, and the second one optimises the dynamic maintenance plan.

3.4.2. Proposed solution.

This research studies the simultaneous optimisation of design and preventive maintenance strategy by coupling Multi-objective Evolutionary Algorithms and Discrete Event Simulation; a technique that has achieved good results in the Reliability field. Coupling Multi-objective Evolutionary Algorithms and Discrete Simulation has been studied, on the one hand, in order to optimise the structural design both from a single objective point of view [97] and from a multi-objective point of view [112]. On the other hand, it has been studied in order to optimise the preventive maintenance scheduling both from a single objective point of view [139][141] and from a multi-objective point of view [165][168]. Moreover, only a few works have been developed looking at the design and corrective maintenance strategy simultaneously [121][122]. Nevertheless, to the knowledge of the author of the present research, coupling Multi-objective Evolutionary Algorithms and Discrete Event Simulation has not been explored for the joint optimisation of the design and preventive maintenance strategy as in the current study. Galván et al. [8] coupled the Non-Sorting Genetic Algorithm (NSGA-II) and Monte Carlo Simulation in order to achieve the optimum design and surveillance test intervals. In this study, the main objective regarding the preventive maintenance strategy consists of supplying the optimum period of time to conduct such activities.

As it was indicated in the Chapter II of the present research, several state-of-the-art Multi-objective Evolutionary Algorithms are studied (NSGA-II, GDE3, SMS-EMOA, MOEA/D and MOEA/D-DE). The NSGA-II method has been widely employed in the Reliability field. However, do not occur the same for the rest of the cited state-of-the-art Multi-objective Evolutionary Algorithms, which have not been so widely explored.

3.5. Multi-objectivisation.

In order to improve the performance for single objective optimisation problems, several authors have considered using multi-objective algorithms with exclusive dependency on the genotypic values [192]. Such a technique has been termed multi-objectivisation. Although Loius and Rawlins previously discussed its principles [193], the term multi-objectivisation was firstly used by Knowles et al. [194]. Two types of multi-objectivisation were distinguished by the authors. On the one hand, decomposition, which is based on decomposing the main objective into several components. On the other hand, aggregation, which is based on considering some additional objectives (helper-objectives) that are used in combination with the main objective. Multi-objectivisation has been used to manage complex optimisation problems in several fields [195-198] where some advantages were reported in terms of performance.

The use of multi-objectivisation for multi-objective problems has not received so much attention. Ishibuchi et al. [199] solved a problem with two objectives after transforming such a problem into a four-objective problem under a decomposition approach. Using an aggregation approach, Zheng et al. [200] introduced a helper-task to promote positive inter-task knowledge transfer in a multi-task optimisation problem. Here, the multi-objectivisation technique is explored under the decomposition approach, in order to compare the performance achieved when two and three objectives are used. Availability and Cost are the objectives considered for the two-objective problem. In this case, the Cost considers both Acquisition and Operating Costs. In order to decompose the Cost, the multi-objectivisation of this objective consists of separating Acquisition and Operating Costs. Therefore, the objectives for the three-objective problem are Availability, Acquisition Cost and Operating Cost.

4. CHAPTER 4: RELIABILITY PROBLEM HANDLED: METHODOLOGY.

4.1. Determining the Availability from Functionability Profiles.

The concept Availability was defined in the Chapter I of the present research as “the fraction of the total time in which systems are available to perform their required function” [12]. Technical systems are developed and built to fulfil a determined function. Therefore, the “functionality” is an important feature, which is related to the capacity of systems to perform its mission. Moreover, systems must satisfy some requirements that are named as “satisfactory features” (e.g., flow or density). Furthermore, specifying the “operation conditions” under which the system must operate is needed (e.g., humidity or temperature). Such three aspects come together on the “Functionability Profile” (FP) concept, which was introduced by Knezevic [201]. It is defined as “the capacity of systems to fulfil the required function under determined features when such systems are used as it is specified”.

Speaking in general, systems fulfil their function at the beginning of their useful life. Nevertheless, irreversible changes appear over time so variations regarding the system behaviour take place. The deviation of such variations regarding the satisfactory features brings into the light the occurrence of the failure. Such a failure causes the transition from the operating to the failure state. The capacity to fulfil the required function for reparable systems can be recovered by carrying out a corrective maintenance task. The operating process for the system up to failure and its subsequent recovery can be graphically shown by its Functionability Profile. Such a Functionability Profile describes how the state of a reparable system fluctuates between the operating (time to failure) and the recovery (time to repair) state along its mission time. An example of a Functionability Profile can be seen in the Figure 4.1.

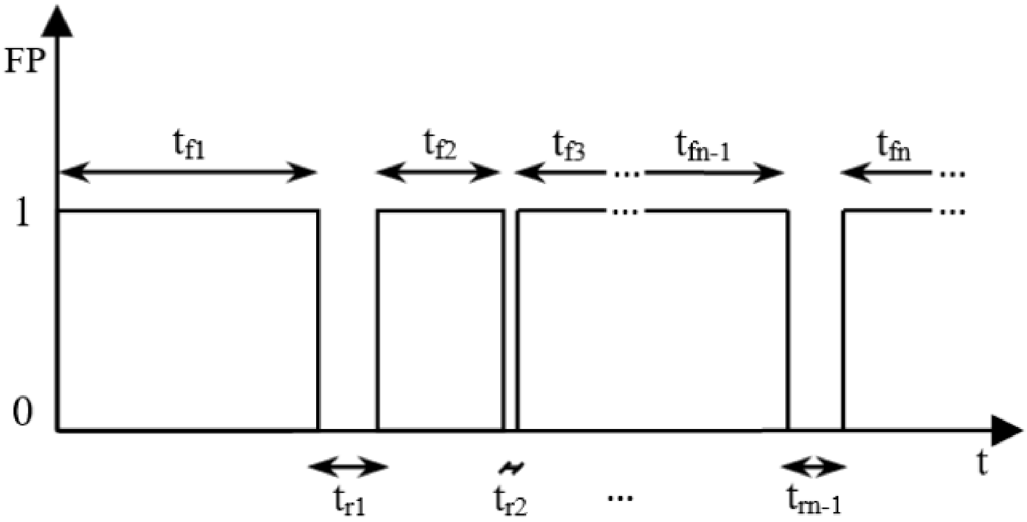


Figure 4.1: Functionability Profile.

The Functionability Profile follows operating times $(t_{f1}, t_{f2}, \dots, t_{fn})$ and recovery times $(t_{r1}, t_{r2}, \dots, t_{rn})$. Up to now, operating times are considered as times to failures (TF) and recovery times as times to repair (TR). After a failure, a corrective maintenance activity is conducted. Within the high level of the Functionability Profile, the system is fulfilling its function. Conversely, within the low level, the system is being recovered by conducting corrective maintenance. From the Figure 4.1, it can be seen that the failure occurs after a period of satisfactory operation t_f and next, a period of recovery t_r is needed in order to recover the operating state.

Once finalised the mission time, both operating and recovery times are known. As it was shown in the Chapter I of the present research, when the devices of a system follow exponential failure and repair intensities (constant failure and repair rates), the Availability of the system can be computed by employing the Equation 4.1, which appears at reference [12].

$$A = \frac{MTTF}{MTTF + MTTR} \tag{4.1}$$

Where $MTTF$ denotes the Mean Time to Failure and $MTTR$ denotes the Mean Time to Repair. When the devices of a system do not follow exponential failure and repair intensities, a simulation approach can be suitable. The behaviour of the system can be simulated along its mission time. Therefore, at the end of the process, operating and recovery times can be characterised. Once known such information, the Availability will be able to be computed by using the Equation 4.2.

$$A = \frac{\sum_{i=1}^{n} t_{fi}}{\sum_{i=1}^{n} t_{fi} + \sum_{i=1}^{m} t_{rj}} \tag{4.2}$$

Where n denotes the total number of operating times (times to failure), t_{fi} denotes the i-th operation time, m denotes the total number of recovery times (times to repair) and t_{rj} denotes the j-th recovery time.

4.2. Building the Functionability Profile from Discrete Event Simulation.

Using Discrete Event Simulation has shown its power because it allows analysing complex systems much more accurately due to a more realistic representation of their behaviour in practice. Employing such a technique, the devices' Functionability Profile of a system can be built by generating random numbers, which follow a specific distribution function. In this way, times to failure (TF) and times to repair (TR) can be generated so a good approximation regarding the true Functionability Profile for the devices can be built and finally, the Functionability Profile of the whole system. Once the system's Functionability Profile is known, it is possible to compute the Availability by employing the Equation 4.2.

The process consists of building the life cycle of each device, which moves between the operating and the repair states. To do that, a random number that follows the distribution function regarding times to failure (TF) is generated (as commented in the Chapter I), so the device is operating until the failure takes place at such a moment. Next, a new random number that follows the distribution function regarding times to repair (TR) is generated (as commented in the Chapter I), so the device does not recover the operating state up to finish such a repair. The process is repeated until concluding the mission time. Such a process is schematised as follows:

1. The system mission time is set, so the process continues for each device included in the design.
2. The Functionability Profile (FP) of the device, which is included in the design, is initialised.
3. A random time to failure t_f and a time to repair after failure t_r are generated, both following their respective distribution function. Both are included in the device Functionability Profile.
4. The third point is repeated up to generate the complete device's Functionability Profile.
5. From the second to the fourth points are repeated up to generate the Functionability Profiles regarding the devices included in the system design.
6. Once such Functionability Profiles are built, the system's Functionability Profile can be built, by following the structure of the system regarding each step of time.

4.3. Modifying the Functionability Profile.

In the previous section, it was shown how to build the devices' Functionability Profile of a system and finally, the system's Functionability Profile. It is possible to improve the system Availability by including preventive maintenance activities in the devices' Functionability Profiles.

Up to now, it has been explained that the Functionability Profile represents a repeated cycle, which is formed by operating and recovery times. The operating times were considered as times to failure (TF) and the recovery times were considered as times to repair (TR) after failure. Conducting preventive maintenance for the devices included in the system's design involves modifying their respective Functionability Profiles. This is due to the fact that an operating time could be not only a time to failure (TF) but also a time to start a preventive maintenance task (TM). On the other hand, a recovery time could be not only a time to repair (TR) but also a time to conduct a preventive maintenance task (TCM). Such a situation will be considered regarding the Equation 4.2. Therefore, for the present research, the main reasons why a system is not available are the failure (after such a failure, a time to repair is needed) or the preventive maintenance (a time to conduct such an activity is needed).

When the unavailability of the system is due to conduct a preventive maintenance task, the unproductive phase is more controlled because it consists of a scheduling shutdown, so human teams and materials are prepared and available, as examples.

Preventive maintenance activities allow modelling to users the system’s Functionability Profile. In this way, such users can be sure that the system satisfy the desired functions. They are interested in keeping the available state for the system as time as possible in order to achieve the best performance. When the system is operating, earnings are generated regarding its availability. Conversely, when such a system must be recovered, investments must be applied in order to return the operating status. Thus, the users of the system are interested in managing its Functionability Profile in order to improve the system’s Availability.

Therefore, it is needed to modify the procedure to build the Functionability Profile, which was shown in the previous section, in order to include preventive maintenance tasks. Again, the process consists of building the life cycle of the system regarding each device included in the system design, which move between operating and recovery states. For the previous process, a random number that follows the time to failure distribution functions regarding each device was needed in order to characterise an operating time. In this occasion, a random number that follows the time to start a preventive maintenance task (TM) distribution function regarding each device is needed too. In this way, if the time to failure (TF) is smaller than the time to start a preventive maintenance task (TM), such a failure takes place. Therefore, this section of the device’s Functionability Profile is formed by the time to failure

(TF), which is followed by the corresponding time to repair (TR). Conversely, if the time to failure (TF) is bigger than the time to start a preventive maintenance task (TM), such a preventive maintenance task takes place. In this case, this section of the device's Functionability Profile is formed by the time to start a preventive maintenance task (TM), which is followed by the corresponding time to conduct such a preventive maintenance task (TCM). The process is repeated up to finish the mission time. Such a process is schematised as follows:

1. The system mission time is set, so the process continues for each device included in the design.
2. The Functionability Profile (FP) of the device, which is included in the design, is initialised.
3. A random time to start a preventive maintenance task (TM) is supplied by the corresponding decision variables from the Evolutionary Algorithm. On the other hand, a time to conduct a preventive maintenance task (TCM) is randomly generated.
4. A random time to failure (TF) is generated, which follows its distribution function.
5. In case of TM > TF, the failure takes place prior to conduct the preventive maintenance task. Therefore, a time to repair (TR) must be randomly created by using the repair distribution function. This section of the device's life cycle is formed by the (TF) value as an operating time and the (TR) value as a recovery time.
6. In case of TM < TF, the preventive maintenance task is conducted prior to fail. Thus, this section of the device life cycle is formed by the (TM) value as an operating time and the (TCM) value as a recovery time.
7. The steps fourth to sixth are repeated up to generate the complete device's Functionability Profile.
8. The steps second to seventh are repeated up to generate the Functionability Profiles regarding the devices included in the system design.

9. Once such Functionability Profiles are built, the system's Functionability Profile can be built, by following the structure of the system regarding each step of time.

The procedure is shown in the Figure 4.2.

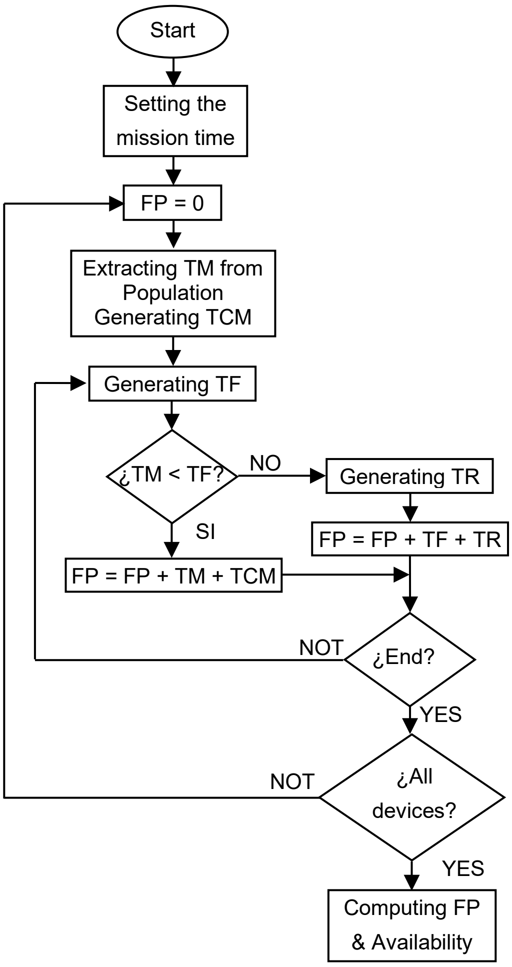


Figure 4.2: Procedure to modify the system Functionality Profile.

4.4. An example of construction and modification of the system's Functionability Profile.

In order to illustrate the procedure explained above, next, a simple example is shown. The Availability of a system is computed from its Functionability Profile point of view. The system structure is shown in the Figure 4.3.

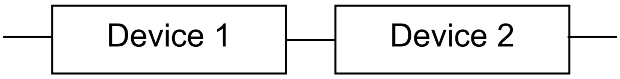


Figure 4.3: System with two devices in series.

As a started point, it is considered that the individual (solution) from the population employed to compute the system's Availability supplies a time to start a preventive maintenance task (TM) for the devices 1 and 2 of 5- and 7-time units, respectively. Moreover, it is considered that their respective times to conduct a preventive maintenance time (TCM) have been generated so values of 2- and 3-time units were achieved, respectively.

In order to generate the Functionability Profile for the device 1, a time to failure (TF) must be generated, following its corresponding distribution function. Such a time to failure is generated as it was explained in the Chapter I. As an example, considering that the time to failure follows an exponential distribution function with Mean Time to Failure of 5-time units, and the random number generated reaches a value of 0.3, the process supplied a time to failure (TF) of 6-time units.

$$TF = -5 \cdot \ln(0.3) \cong 6$$

In this case, the time to failure (TF) overcomes the time to start a preventive maintenance task (TM) of 5-time units. Therefore, such a preventive maintenance task takes place before the failure. This section of the device's Functionability Profile will be formed by 5-time units due to the time to start the preventive maintenance task (TM) and 2-time units due to the time to conduct such a preventive maintenance task (TCM) for the device 1. Such a situation is shown in the Figure 4.4.

Device 1	1	1	1	1	1	0	0											

Figure 4.4: Including TM + TCM in the Functionability Profile.

Next, a new time to failure (TF) is generated by following the corresponding distribution function. In this occasion, the random number generated was 0.45, so the time to failure (TF) achieved is 4-time units.

$$TF = -5 \cdot \ln(0.45) \cong 4$$

Since the achieved time to failure (TF) is lower than the time to start a preventive maintenance task (TCM), the failure takes place before starting the preventive maintenance task. In this occasion, a time to repair (TR) is needed so it is generated as a random number that follows the corresponding distribution function. A normal distribution with mean 5 and standard deviation of 2 is considered. As it was explained in the Chapter I, a set of 12 random numbers is generated, as an example: 0.136 – 0.245 – 0.123 – 0.489 – 0.389 – 0.214 – 0.312 – 0.785 – 0.345 – 0.614 – 0.822 – 0.503. The time to repair (TR) is computed as follows:

$$X = \sum_{i=1}^{12} X_i = 4{,}977$$

$$TR = (X - 6)\sigma + \mu = (4{,}977 - 6)2 + 5 \cong 3$$

Therefore, this section of the device Functionability Profile will be formed by 4-time units due to the time to failure (TF) and the 3-time units due to the time to repair (TR) for the device 1. Such a situation is shown in the Figure 4.5.

Device 1	1	1	1	1	1	0	0	1	1	1	1	0	0	0				

Figure 4.5: Including TF + TR in the Functionability Profile.

The process is repeated till completing the mission time, as it is shown the Figure 4.6.

Device 1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	1	0	0

Figure 4.6: Functionability profile for the device 1.

The process is repeated for the device 2. As an example, it is considered that the result achieved is as it is shown in the Figure 4.7.

Device 2	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	0	0	0

Figure 4.7: Functionability profile for the device 2.

From the devices’ Functionability Profiles and following the serial-parallel logical for the system’s structure (serial in this case), the system’s Functionability Profile can be generated, as it is shown in the Figure 4.8.

Device 1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	1	0	0
Device 2	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	0	0	0
System	1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0

Figure 4.8: Building the system’s Functionability Profile.

The system was in operating state for 8-time units while it was in failure state for 10-time units. The system’s Availability is computed by using the Equation 4.2 as follows:

$$A = \frac{8}{8+10} = 0.44$$

4.5. Multi-objective Optimisation of the design and maintenance strategy.

In the Chapter II of the present research, the basic principles of multi-objective optimisation by using Evolutionary Algorithms were presented. The details to consider when the system's design and its maintenance strategy pretends to be jointly optimised are shown next.

4.5.1. Objective functions.

The joint multi-objective optimisation of the system design and its maintenance strategy will supply a set of feasible and balanced solutions among objectives, which will be distributed along the non-dominated solutions front. In the present research, maximum Availability and minimum Cost are the objectives to consider. The Availability is computed by employing the objective function supplied by the Equation 4.2. Regarding such an Equation, as operating times t_{fi}, both times to failure and times to start a preventive maintenance task can be considered, whereas, as recovery times t_{ri}, both times to repair and times to conduct a preventive maintenance task can be respectively considered. The Cost can be computed by considering several aspects such as acquisition costs or costs due corrective and preventive maintenance tasks. The basic objective function employed to compute the Operating Cost is shown in the Equation 4.3, where C denotes the system's Operation Cost, which is quantified in economic units, q denotes the global number of corrective maintenance tasks, cc_i denotes the cost due to the i-th corrective maintenance task, p denotes the global number of preventive maintenance tasks and cp_j denotes the cost due to the j-th preventive maintenance task. Both cc_i and cp_j are computed by multiplying the cost per hour regarding each type of maintenance task and the number of hours dedicated to such maintenance tasks.

$$C = \sum_{i=1}^{q} cc_i + \sum_{i=1}^{p} cp_j \tag{4.3}$$

In addition, it is possible to consider the Acquisition and Replace (if this one is needed) Costs due to each device included in the system design as it is shown in the Equation 4.4.

$$C = \sum_{k=1}^{d} \left[c_{ak} + \sum_{i=1}^{q} (D_i + cc_i) + \sum_{j=1}^{p} cp_j \right] \tag{4.4}$$

Where C denotes the total Cost, which is quantified in economic units, d denotes the maximum number of devices included in the system's design, c_{ak} denotes the acquisition cost regarding the k-th device, q denotes the global number of corrective maintenance tasks regarding the k-th device, D_i denotes the cost regarding the i-th replace (if it is needed and it is related to corrective maintenance tasks) for the k-th device, cc_i denotes the cost in relation to the i-th corrective maintenance task for the k-th device, p denotes the count of preventive maintenance tasks regarding the k-th device and cp_j denotes the cost regarding the j-th preventive maintenance task for the k-th device. Again, both cc_i and cp_j are computed by multiplying the cost per hour regarding each type of maintenance task and the number of hours dedicated to such maintenance tasks.

Note that when the multi-objectivization technique is applied, the total Cost is decomposed between the Acquisition Cost, which is computed by using the Equation 4.4 when $\sum_{k=1}^{d} c_{ak}$ is considered exclusively, and the Operating Cost, which is computed by using $\sum_{k=1}^{d}\left[\sum_{i=1}^{q}(D_i + cc_i) + \sum_{j=1}^{p} cp_j\right]$ exclusively.

4.5.2. Decision variables, constraints and encoding.

The decision variables are a fundamental element on optimisation due to such variables form the chromosome, which represent possible solutions to the problem. In the present research, two types of decision variables are used, on the one hand, decision variables regarding the design and, on the other hand, decision variables regarding the maintenance strategy. The decision variables regarding the design consider design alternatives by redundancy allocation for the sub-systems that form the whole system. The decision variables regarding the maintenance strategy express a schedule to conduct preventive maintenance tasks regarding the devices included in the design alternative, when maximum Availability and minimum Cost are desired.

Constraints are limits to the values that the decision variables can reach. In the case of decision variables regarding the design, the limit consists of the maximum number of devices that can be allocated for each sub-system. Such a limit can be subjected to budget constraints or available space, as examples. In the case of decision variables that refers to preventive maintenance, a premature preventive maintenance task could be conducted if a minimum time were not considered. Conversely, a reckless preventive maintenance task could be conducted if a maximum time were not defined. Therefore, the decision variables must present feasible values between limits.

In order to explore the solutions to the problem in depth, in the present research, both real and binary encoding are explored. Moreover, as an extension of the binary encoding, the Gray code is explored too. A mix between real and binary encoding is avoided so solutions are achieved from a real or binary approach. However, the decision variables regarding the design could have binary or integer nature, whereas the decision variables regarding maintenance have an integer nature. Therefore, some transformation must be done in order to achieve their true values as follows.

4.5.2.1. Real encoding.

For real encoding, the decision variables included in the chromosome take real values between 0 and 1. Therefore, at the moment of evaluating the objective functions, the decision variables must be transformed according to their range of feasible values. Next, in order to show how this technique works, two optimisation design problems are considered. Firstly, decision variables with binary nature are considered because they refer to the inclusion or not of a redundant device. Secondly, decision variables with integer nature are considered because they refer to the number of redundant devices that can be included as redundancies in a sub-system. Regarding the maintenance strategy, both problems consider decision variables with integer nature.

Decision variables with binary nature for the design.

In this problem, both decision variables with binary nature (regarding design) and decision variables with integer nature (regarding periodic preventive maintenance) coexist. However, both types of decision variables are codified as real numbers, which reach values between 0 and 1 along the evolutionary process. Therefore, when the objective functions are going to be evaluated, the decision variables must be transformed between their respective limits as follows:

- The decision variables with binary nature (regarding design) must be round to the nearest integer. When the value supplied is 0, the respective device is not included in the design whereas 1 implies the opposite. An illustrative example is shown in the Figure 4.9. For a system formed by a single device (Device 1), a design alternative is considered in order to include a parallel second device (Device 2) as a redundant device.

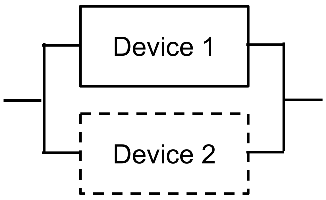


Figure 4.9: A device (Device 1) with a parallel redundancy (Device 2).

In this case, a single decision variable regarding design is needed because it would be used in order to decide the inclusion or not inclusion of the Device 2 in the system. Such a decision variable has integer nature so 0 indicates that the device is not included and 1 the opposite. The alternatives to consider at the moment of evaluating the objective functions are:

  - → A decision variable value smaller than 0.5 implies not to consider the inclusion of the device in the system design.
  - → A decision variable value equal or bigger than 0.5 implies to consider the inclusion of the device in the system design.

- The decision variables with integer nature (regarding maintenance) must be transformed because they must achieve values within the interval $[TM_{min}, TM_{max}]$, where TM_{min} denotes the minimum operating time to start a scheduled preventive maintenance task for the referred device and TM_{max} denotes the maximum operating time to start a scheduled preventive maintenance task for the referred device. Both values must be previously stablished. The Equation 4.5 shows how the transformation must be done.

$$TM_i = round\ (TM_{min_i} + M_i \cdot (TM_{max_i} - TM_{min_i})) \tag{4.5}$$

Where TM_i is the true value for the time to start a preventive maintenance task for the device i , TM_{min_i} and TM_{max_i} denote the minimum and maximum values that the time to start a preventive maintenance task can take for the device i and M_i is the value of the decision variable referred to the device i. As an example, considering a device with a TM_{min} of 4,380 hours and a TM_{max} of 8,760 hours whose decision variable regarding preventive maintenance presents a value of 0.423, the true value for the time to start a preventive maintenance task reaches a value of 4,380 + 0.423 · (8,760 − 4,380) = 6,233 hours.

Note that such transformations take place on purpose of evaluating the objective functions. Therefore, such a procedure does not affect to the values of the decision variables from the chromosome, which are exclusively modified by the evolutionary process.

Finally, the chromosome would be formed by a number of decision variables equal to the number of considered redundancies and the number of times to start a preventive maintenance task, one per device to be considered for the design alternatives. Following the previous example, which is referred by the Figure 4.9, three decision variables would form the chromosome; one of them to consider the inclusion of Device 2 as a redundant device (D_1) and two of them to consider the optimum time to start a preventive maintenance task (M_1, M_2) (one regarding the Device 1 and other one regarding the Device 2, which is not considered if it is not included in the design).

$$[D_1][M_1, M_2] \rightarrow Mínimum\ (Unavailability, Cost)$$

Decision variables with integer nature for the design.

In this case, decision variables with integer nature are considered for both the design and the maintenance strategy. As in the previous case, both types of decision variables are codified as real numbers that reach values between 0 and 1 along the evolutionary process. Therefore, when the objective functions are going to be evaluated, the decision variables must be transformed between their respective limits as follows:

- The decision variables regarding the structure of the system can take values between 1 and d_i, where d_i denotes the maximum number of devices that can be included in the sub-system i. As an example, the Figure 4.10 shows, a system which is formed by two sub-systems ($i = 2$), with a maximum number of 2 devices for the sub-system 1 ($d_1 = 2$) and 3 for the sub-system 2 ($d_2 = 3$).

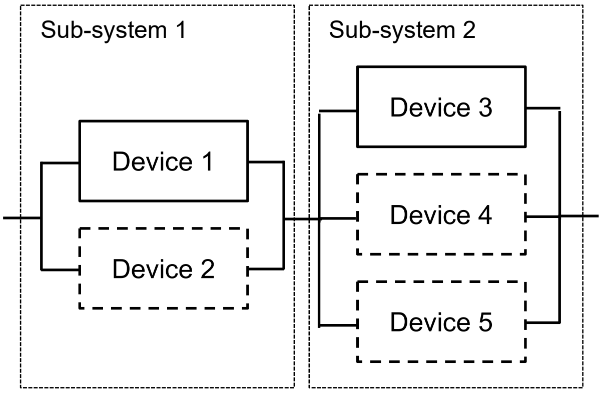


Figure 4.10: A System with 2 sub-systems with 2 and 3 devices, respectively.

Such variables must be transformed in order to evaluate the objective functions because they must achieve values within the interval $[1, d_i]$, where d_i presents values previously fixed. The Equation 4.6 shows the base to transform such variables.

$$E_i = round\_floor(N_{min_i} + D_i \cdot d_i) \qquad (4.6)$$

Where E_i is the number of devices included in the sub-system i, N_{min_i} is the minimum number of devices included in the sub-system i and D_i is the value that the decision variable takes for the sub-system i. It can be seen that the real number achieved is rounded down because the number of devices is an integer. As an example, if it is considered a sub-system with at least 1 device, a maximum of 3 devices ($d_i = 3$) and a decision variable X with a value of 0.5342, the number of devices E would be 2. A restriction must be considered in case of a value of 1 for D (it almost impossible but as a real number, it can achieve values from 0 to 1). In this case, $E_i = d_i$.

- The decision variables with integer nature (regarding maintenance) must be transformed by following the procedure previously explained regarding how to manage the decision variables with binary nature for the design (see the Equation 4.5).

Finally, the chromosome would be formed by a number of decision variables equal to the number of sub-systems considered for the system's structure (one per sub-system, where the number of its devices is included) and the periodic times to start a preventive maintenance task (one decision variable per device that can be included in the design). Considering the previous example, which is referred in the Figure 4.10, 7 decision variables are needed in order to code such an example; 2 of them regarding the devices to include for each sub-system (D_1, D_2) and 5 regarding the maintenance strategy (M_1, M_2, M_3, M_4, M_5), one per device that can be included in the design.

$$[D_1, D_2][M_1, M_2, M_3, M_4, M_5] \rightarrow Mínimum\ (Unavailability, Cost)$$

4.5.2.2. Natural binary encoding.

In this case, all the decision variables included in the chromosome are binary and they take values between 0 and 1. As in the previous section, the decision variables must be transformed according to their range of feasible values. Next, in order to show how this technique works, the way to codify the chromosome is shown when the design optimisation is considered. Decision variables with binary nature are considered because they refer exclusively to include or not a redundant device. Regarding the maintenance strategy, decision variables with integer nature are considered. However, they use binary encoding so they must be transformed as it is shown.

<u>Decision variables with binary nature for the design.</u>

In this problem, both the decision variables with binary nature (regarding design) and the decision variables with integer nature (regarding periodic preventive maintenance) coexist. However, both types of decision variables are codified as binary numbers, which take values of 0 or 1 along the evolutionary process. Therefore, when the objective functions are going to be evaluated, the decision variables must be transformed between their respective limits as follows:

- The decision variables with binary nature do not need to be processed so when the value supplied is 0, the respective device is not included in the design whereas 1 implies the opposite. Reconsidering the example from the Figure 4.9, where a basic system is formed by a single device (Device 1) and such a system could contain a parallel second device (Device 2) as a redundancy. A single decision variable regarding the design is needed because it would be used in order to decide the inclusion or not inclusion of the Device 2 in the system. Due to such a decision variable takes a binary value, its transformation is not needed. Reconsidering the example from the Figure 4.10, which consists of a system formed by two sub-systems with a maximum number of 2 redundant devices for the sub-system 1 and 3 for the sub-system 2, 3 decision variables would be needed (one of them for the sub-system 1 – Device 2, and two of them for the sub-system 2 – Devices 4 and 5). The devices 1 and 3 are the main devices so they are always considered. Due to such decision variables take a binary value, their transformation is not needed, as in the previous example.

- The decision variables with integer nature (regarding maintenance) must use binary encoding in order to be included in the chromosome. Next, they must be transformed in integer numbers in order to evaluate the objective functions. Such decision variables must achieve values within the interval $[TM_{min}, TM_{max}]$, where TM_{min} denotes the minimum time to start a scheduled preventive maintenance task for the referred device and TM_{max} denotes the maximum time to start a scheduled preventive maintenance task for the referred device. Both values must be previously stablished.
  - → In order to include the decision variables in the chromosome, the binary scale must consider as a minimum value the value $TM_{max} - TM_{min}$. A decision variable value smaller than 0.5 implies not to consider the inclusion of the device in the system design. Such a situation involves a high probability of overcoming the feasible values for the considered scale. As an example, it is possible to consider a device with TM_{min} and TM_{max} values of 2,920 and 8,760 hours, respectively. The difference $TM_{max} - TM_{min}$ reaches a value of 5,840 hours, which is the minimum steps to consider in the scale. In order to have such a number of steps, a number n of bits is needed, which must satisfy $2^n \geq 5840$. Therefore, $n = 13$ bits. However, a 13 bits scale can take $2^{13} = 8192$ steps. Therefore, a relationship between the step values of the binary scale and the equivalent real scale is needed in order to evaluate the objective functions. Such a relationship can be achieved by using the Equation 4.7.

$$R = \frac{TM_{max} - TM_{min}}{2^n} \tag{4.7}$$

Reconsidering the example, R takes a value of:

$$R = \frac{8{,}760 - 2{,}920}{2^{13}} = 0.712890625$$

In order to achieve the true value of the time to start a preventive maintenance task, the Equation 4.8 must be employed.

$$TM_i = round\left(TM_{min_i} + B_i \cdot R_i\right) \tag{4.8}$$

Where TM denotes the integer value of the time to start a preventive maintenance task for the device i, TM_{min_i} denotes the minimum time to start a preventive maintenance task for the device i, B denotes the decimal value which is equivalent to the binary number supplied by the decision variables for the device i and R_i denotes the scale factor previously computed by using the Equation 4.7. Following the example, in case of decision variable values of 1100110001110, which denotes an integer number of 6,542, the transformed value would be $6{,}542 \times 0.712890625 = 4{,}663.7 \approx 4{,}664$. Such a value is included in the scale of feasible values with a maximum of 5,840 steps. In order to achieve the true value of the time to start a preventive maintenance task within the interval $[TM_{min}, TM_{max}]$, adding the TM_{min} value is needed, so a TM value of 6,542 achieved from the chromosome is equivalent to a true value of the time to start preventive maintenance task of $TP_{min} + 4{,}664 = 2{,}920 + 4{,}664 = 7{,}584$ hours.

As it was explained before, such transformations take place on purpose of evaluating the objective functions. Therefore, such a procedure does not affect to the decision variables in the chromosome, which are exclusively modified by the evolutionary process.

Finally, the chromosome would be formed by a number of decision variables equal to the number of considered redundancies (1 bit per each sub-system) and the number of bits needed to represent the range of values $TM_{max} - TM_{min}$, for the devices that can be included in the system design.

4.5.2.3. Gray code.

As in the case of binary encoding, the decision variables included in the chromosome takes values of 0 or 1 due to the Gray code is employed. The difference between both codifications (binary and Gray) consists of using the Gray code, which is named reflected binary too. Such a code is a way to represent numbers in which the codification of two neighbour numbers differs in only one bit [202]. The decimal numbers between 0 and 7 are represented in the Table 4.1 when both the binary and the Gray code are used.

Decimal number	Binary number	Gray number
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Table 4.1: Binary and Gray numbers.

When binary encoding is used, consecutive numbers can differ in more than one bit. This is the case of the numbers 1 (codified as 001) and 2 (codified as 010). Both numbers differ in two bits. The more extreme case takes place between the numbers 3 (codified as 011) and 4 (codified as 100), where all bits differ. Conversely, when the Gray code is employed, two consecutive numbers differ in only one bit as it is shown in the Table 4.1. Employing the Gray code makes easier to optimise regular functions (functions that do not suddenly vary), because small changes in the code have small changes on the function values as a result. Apart from this, the binary encoding presents a better behaviour in problems with a high number of local minima (in the order of the middle of points in the research space). A successful application in Evolutionary Multicriteria Optimization when Gray code is used was presented by Greiner et al. [203]. The used encoding may have a big impact in the performance of an Evolutionary Algorithm so it must not be ignored [15]. This is the reason why studying both the binary and the Gray code is considered for the present research.

As in the previous case, the decision variables must be transformed at the moment of evaluating the objective functions. Such transformations are like the ones exposed in the previous section, so they are not explained again.

4.5.2.4. Other considerations regarding the decision variables.

Attending to the way to study the domain (discrete or continuous) for the considered problem, some encoding alternatives were shown before. Another interesting aspect to analyse consists of the accuracy regarding the solutions to the problem.

Preventive maintenance tasks can be scheduled based on several time units. Such tasks could be scheduled by using, for instance, the hour, the days, the week or the month as a time unit. Groups of them could be used, such as each three days or two weeks, in case of being useful to users. It could be non-significant to schedule a preventive maintenance task attending on a specific hour. It could enough to consider it within a range of 24 hours of a day.

Although such a circumstance has not impact in case of employing real encoding, it could have when binary encoding is used. As explained before, when real encoding is used, a change in the scale must be done from the real decision variables to the used time unit. Therefore, such a time unit could be the hour, the day, the week or any other. Conversely, such a circumstance is not valid when binary encoding is used because the length of the chromosome must represent the complete scale between feasible values. This is the reason why the larger the chromosome, the more the time unit accuracy to be analysed. If one week is used as a time unit, 1 bit will be needed. However, if the day is used, 3 bits will be needed (it is not enough using a number of bits $n = 2$, because $2^n = 4$ scale positions and 7 positions are needed. At least a number of bits $n = 3$ must be used because $2^n = 8$ available scale positions). The increment is higher in case of using the hour. The possible impact due to the chromosome length is considered for the present research.

4.5.3. Parameterizing the optimization methods considered.

As it was explained in the Chapter II, five state-of-the-art Multi-objective Evolutionary Algorithms are considered along the present research. These are NSGA-II *(Non-dominated Sorting Genetic Algorithm II)*, GDE3 *(Third Evolution Step of Generalized Differential Evolution)*, SMS-EMOA *(Multi-objective Selection based on Dominated Hypervolume)*, MOAE/D *(Multi-objective evolutionary algorithm based on decomposition)* and MOEA/D-DE, which is like MOAE/D but using Differential Evolution.

Setting the parameters is important in order to address a successful process. Next, the main parameters regarding the method employed are defined, and their values are exposed. The parameter tested values were chosen from experiments, recommendations from literature and common practices in the evolutionary algorithms field:

- Mutation Rate (PrM): The expectation of the number of genes mutating. The central value is equivalent to the inverse of the decision variables. Two more probabilities, one above and the other below the central value (1.5/decision variables and 0.5/decision variables, respectively) have been set to the methods that use Simulated Binary Crossover (SMS-EMOA, MOEA/D and NSGA-II). For binary encoding, the central value is explored because more types of crossover are explored.
- Mutation Distribution index (disM): The distribution index of polynomial mutation. This is set to the typical value of 20 for the present case study. A thorough study was previously conducted, and the parameter showed a low impact in the performance.

- Crossover Probability (PrC): The probability of doing crossover when the Simulated Binary Crossover is used. The crossover operator has an impact on the creation of new individuals. It is set to values of 0.9 and 1 for the present research.
- Crossover Distribution index (disC): This is the crossover distribution index when the Simulated Binary Crossover is used. It is 20 for the present research. A thorough study was previously conducted, and the parameter showed a low impact in the performance.
- Crossover Rate (CR): The crossover operator has the function of mixing the genetic information among chromosomes to create new individuals. In Differential Evolution (GDE3, MOEA/D-DE), each gene is crossed (or not) depending on a probability variable referred as the Crossover Rate. The typical value for the Crossover Rate is between 0.1 and 1.0 [15]. For the present research, the Crossover Rate parameter is set to 0.9 given that a large CR often speeds convergence [70].
- Scale Factor (F): In Differential Evolution, the mutation operator alters the genes of the chromosome by adding a scaled difference vector from two chosen chromosomes to a third chromosome. The difference vector is scaled by using the Scale Factor. The typical value for the Scale Factor is between 0.4 and 0.9 [15]. For the present research, values of 0.4, 0.5 and 0.6 are tested.
- Scalarizing function (Approach): The MOEA/D method decomposes a multi-objective optimisation problem into different single-objective sub-problems by using a set of weighted vectors and a scalarizing function. Typical scalarizing functions for MOEA/D include the weighted sum, Tchebycheff and Penalty-based Boundary Intersection (PBI). Following results from the Ref. [204], in which the Tchebycheff approach performed well, such an approach is used for the present research.
- Probability of choosing parents locally (δ): A value used widely [71,205,206] is 0.9.
- Replacement mechanism (n_r): The replacement mechanism improves the quality of the population in terms of dominance, and it also maintains the

diversity. A high-quality offspring solution could replace most current solutions in favour of its neighbouring solution [205], which implies a decrease in diversity. The parameter n_r is used to establish the maximum number of solutions that can be replaced by a high-quality offspring. A proposed empirical rule [206] consists of considering n_r = 0.01·N, as being N the population size (note that n_r must be an integer value).

As population sizes, 50, 100 and 150 individuals are considered. The population size plays a crucial role on maintaining the equilibrium between exploration and exploitation. When optimal solutions are searched, populations with excessive size could lead to slow convergences, whereas populations with few individuals could lead to premature stagnation, converging to local optimums [32,207,208]. Each configuration is executed 21 times (for statistical purposes) and 10,000,000 evaluations is used as the stopping criterion.

4.6. Employed indicator.

In the Chapter II of the present research, the employed indicator was exposed. Such an indicator is the Hypervolume [72].

4.7. Results analysis.

As it was exposed above, the employed indicator to measure the performance of the experiments in this research is the Hypervolume indicator. Once defined the set of experiment to conduct, 21 executions per each Multi-objective Evolutionary Algorithm configuration are executed for statistical purpose. In this way, softening the effect due to the random generation of the initial population is pretended. Therefore, launching only one execution could supply an optimistic or pessimistic hypervolume value. In that case, the result achieved for the experiment could not be conclusive.

Once the end of the evolutionary process is reached, the resulted information is extracted, exploited and analysed. It is interesting to observe the evolution of the hypervolume indicator all along the optimisation process so graphic information is supplied regarding the average value achieved by the 21 executions till reaching the stopping criterion. An example can be seen in the Figure 4.11.

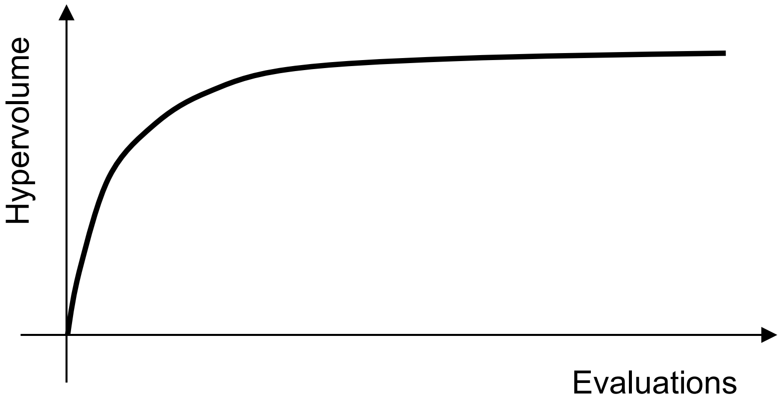


Figure 4.11: Hypervolume evolution vs. the number of evaluations.

On the other hand, box plots are created regarding the achieved Hypervolume values at the end of the process, so the results are described visually and information in relation to the group of data is brought into the light. In this way, three quartiles (the second one coincides with the median), the maximum and the minimum values are shown. An example can be seen in the Figure 4.12.

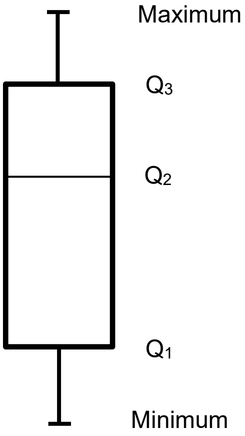


Figure 4.12: An example of box plot.

Besides the visual information previously cited, the Average, Mean, Maximum, Minimum and Standard Deviation values regarding the reached hypervolume from 21 executions is shown for each experiment.

Once developed the experiments, a procedure to compare their performances is needed. As a general rule, the followed procedure consists of conducting a rigorous statistical test, which was proposed by García and Herrera [209]. It starts by carrying out the Friedmans' test, which is a non-parametric test that allows detecting significance differences between the achieved performances and rejecting the null hypothesis (H_0) in such a case. The null hypothesis establishes that the mean of a set represents the mean of other one $(H_0\colon \theta_1 = \theta_2 = \cdots = \theta_k)$. In order to conduct the test, the ranks are computed from the achieved results. The rank 1 is set to the best, the rank 2 is set to the next best and so on till set a rank to each result. The *p*-value is computed by employing the Friedman's test, which stablishes if the null hypothesis is accepted or rejected. The *p*-value is a useful datum, which represents the smallest significant value that can result in the rejection of the null hypothesis. The *p*-value provides information about whether a statistical hypothesis test is significant (or not), and it also indicates how significant the result is: The smaller the *p*-value (< 0.05), the stronger the evidence against the null hypothesis.

In case of rejection (differences found), a post-hoc test is carried out in order to find out the concrete pairwise comparisons which produce such differences. The procedure to conduct multiple comparisons followed in this research was described by García and Herrera. However, some exceptions are applied. Sometimes, although the Friedman's test claims that differences exist, such a procedure does not detect them due to the accuracy of the post-hoc test. In this case, the signed-rank Wilcoxon test is employed as it was recommended by Benavoli et al. [210].

Finally, the non-dominated front is achieved, which is formed by the best-balanced Availability-Cost solutions. Such a front presents an hypervolume value, which is computed by employing the method proposed by Fonseca et al. [211].

4.8. Software platform.

In order to implement the code that allows optimising both the design and the maintenance strategy, employing the PlatEMO [212] platform (1.6 version) was considered. Programmed in MATLAB, the open-source PlatEMO platform includes more than 160 Multi-objective Evolutionary Algorithms, more than 300 multi-objective test problems and several widely used performance indicators. Therefore, a new problem was implemented in PlatEMO. A work flux diagram is shown in the Figure 4.13, in order to explain how the created functions are related.

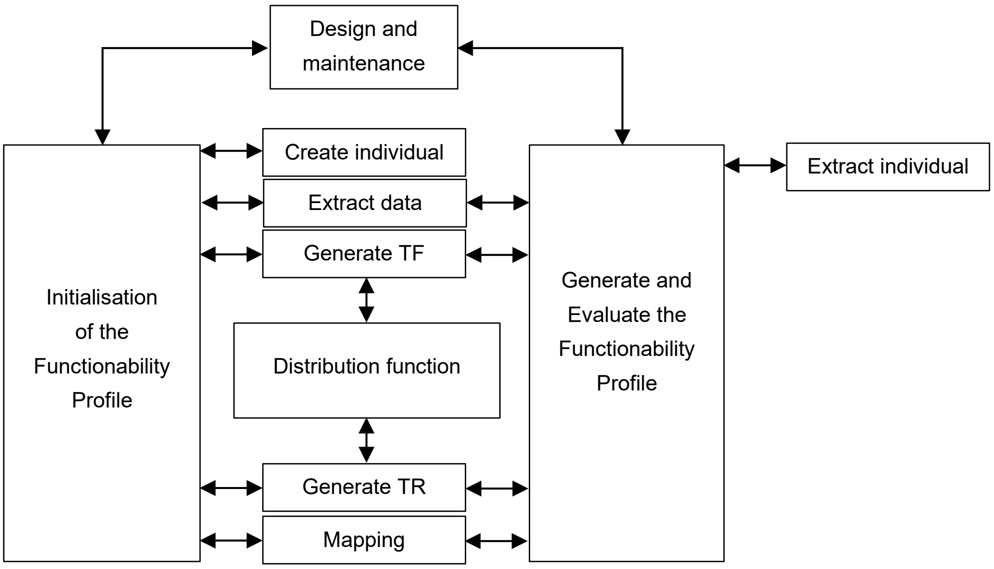


Figure 4.13: Diagram of relationships between generated functions.

Next, a short description of the programmed codes is presented.

- Design and maintenance optimisation: PlatEMO uses a file as the basis of the main problem, which allows initiating the process (*Initialisation of the Functionability Profile*). Generation after generation, it launches the section of the code that sets up and computes the objective functions (*Generate and Evaluate the Functionability Profile*). Moreover, it connects the implemented code with the rest of the platform, which launches the selected Multi-objective Evolutionary Algorithm. It receives as input parameters the type of operation ("initiate" or "evaluate"), the object that stores the parameters values (failure rates, recovery rates, mission time, …) and the decision variables (population). As output parameters, the decision variables values, the objective functions values and the design alternative (setting 0 for the preventive maintenance period regarding non-selected devices) are returned.

- Initialisation of the Functionability Profile: This function receives as input parameters the decision variables (population) and the object where the values of the parameters of interest are stored. It starts randomly generating the individuals of the population (*create individual*) and next, the Functionability Profile is built, which is needed in order to evaluate the objective functions. To do that, the relevant data for the analysed device are extracted (*extract data*) and next, the time to failure and the time to repair are generated (*generate TF – generate TR*). Once generated the Functionability Profiles for all devices (as it was explained in the Section 4.3), the system's Functionability Profile is built (*mapping*). Once built the system's Functionability Profile, the objective functions are evaluated (following the Equations 4.2 and 4.3 or 4.4). The function returns the decision variables, the objective function values, and the design alternative.
- Create individual: This function receives the object which stores the values of the parameters of interest. It starts randomly generating an individual of the population and next, it is transformed to its true values (as it was exposed in the Section 4.5.2), because they are needed in order to build the Functionability Profile. The function returns the decision variable values (population) and the representation of the design.
- Extract data: This function receives the object which stores the values of the parameters of interest. Depending on the type of device, the information to consider is extracted (minimum and maximum time to failure, minimum and maximum time to repair, failure rate, …). The function returns such information.
- Generate TF: This function receives the needed information to generate a time to failure that follows the respective distribution function. The code

allows generating exponential, normal, logarithmic and Weibull distribution functions. It is easy to extend to other distribution functions. The function returns the generated time to failure.

- Generate TR: This function receives the needed information to generate a time to repair that follows the respective distribution function. The code allows generating exponential, normal, logarithmic and Weibull distribution functions. It is easy to extend to other distribution functions. The function returns the generated time to repair.
- Mapping: This function receives as input parameters the generated Functionability Profiles for the included devices in the system's design. From them, the system's Functionability Profile is generated by considering its structure. The function returns such a Functionability Profile so the objective functions can be computed.
- Generate and evaluate the Functionability Profile: Once the process has been initiated, when the second generation is reached, the evolutionary process continues with this function. From the Figure 4.13, it can be seen that this function connects to almost all the functions which were accessed when the Functionability Profile was initiated. However, in this case, the function that allows extracting an individual from the population is called whereas generating a new population is avoided. This function receives as input parameters the decision variables (population) and the object where the values of the parameters of interest are stored. It starts extracting the individuals of the population (*extract individual*) and next, the Functionability Profile is built, which is needed in order to evaluate the objective functions. The function returns the decision variables, the objective function values and the design alternative.
- Extract individual: This function receives the population or decision variable values and the object which stores the values of the parameters of interest. The information is used in order to transform the decision variables to their true values (as it was exposed in the Section 4.5.2), due to such values are needed to build the devices' Functionability Profiles. The function returns the

values of the decision variables (population) and their representation regarding the design by their transformations.

4.9. Parallel executions.

Due to the hardness of the problem, a High-Performance Computer (HPC) was used in the optimisation process. The HPC is composed by 28 calculation nodes and one access or front-end node. Each calculation node consists of 2 processors Intel Xeon E5645 Westmere-EP with 6 cores each and 48 GB of RAM memory, allowing 336 executions to be simultaneously run.

Note that PlatEMO is designed to employ Windows as an operative system. The employed HPC uses Linux as an operative system, so some changes were developed to PlatEMO in order to create a compiled file to execute from commands.

4.9.1. Executing PlatEMO by using commands.

PlatEMO can be executed by employing its friendly User Interfase or by commands. In order to use the HPC, employing commands is needed. Depending on the method and the operator used in PlatEMO, several commands can be employed. Next, the main command used for the present research are explained.

4.9.1.1. Executing when real encoding is considered.

Depending on the optimisation method, several parameters are needed.

Using the NSGA-II method:

The employed command is as follows:

```
-algorithm,@NSGAII,-problem,@TESIS-operator,@EAreal,-N,50,-evaluation,1000,-EAreal_parameter,PrM-disM-PrC-disC
```

Next, the details regarding each parameter are commented.

- *-algorithm:* It denotes that the next parameter consists of the method to employ.
- *@NSGAII:* It denotes that the method to employ is the NSGA-II.
- *-problem:* It denotes that the next parameter consists of the problem to solve.
- *@TESIS:* It denotes the name of the problem to solve (Design and Maintenance).
- *-operator:* It denotes that the next parameter consists of the operator to use.
- *@EAreal:* It denotes that real encoding is going to be used.
- *-N:* It denotes that the next parameter consists of the population size.
- *50:* It denotes that (as an example) the population size is 50 individuals.
- *-evaluation:* It denotes that the next parameter consists of the number of evaluations of the objective functions, which is considered as a stopping criterion.
- *1000:* It denotes that (as an example) 1000 evaluations of the objective functions are going to be considered.
- *-EAreal_parameter:* It denotes that next, the configuration parameters for the operator are going to be specified. These are the mutation rate (*PrM*), the mutation distribution index (*disM*), the crossover probability (*PrC*) and the crossover distribution index (*disC*), respectively.

Using the SMS-EMOA method:

The employed command is as follows:

```
-algorithm,@SMSEMOA,-problem,@TESIS,-operator,@EAreal,-N,100,-evaluation,1000,-EAreal_parameter,PrC-disC-PrM-disM
```

It can be seen that the command line is quite similar. However, the method to use in this case is written as *SMSEMOA*.

Using the MOEA/D method:

The employed command is as follows:

```
-algorithm,@MOEAD,-problem,@TESIS,-operator,@EAreal,-N,100,-evaluation,1000,-EAreal_parameter,PrC-disC-PrM-disM,-MOEAD_parameter,approach
```

In this case, besides the different inclusion of the method (*MOEAD*), a specific parameter for this method is needed, which is named *approach*, or scalarising function. The implemented in PlatEMO scalarising functions are as follows:

- *Approach* = *1*. Intersection Limit based on Penalties (PBI).
- *Approach* = *2*. Tchebycheff.
- *Approach* = *3*. Normalised Tchebycheff.
- *Approach* = *4*. Modified Tchebycheff.

Using the MOEA/D-DE method.

The employed command is as follows:

```
-algorithm,@MOEADDE,-problem,@TESIS,-operator,@DE,-N,100,-evaluation,1000,-DE_parameter,CR-F-PrM-disM,-MOEADDE_parameter,δ-n_r
```

In this case, the method is assigned as *MOEADDE* and differential evolution parameters must be included. Besides the mutation rate (*PrM*) and the mutation distribution index (*disM*), the Crossover Rate (*CR*) and the scale factor (*F*) must be supplied. Furthermore, the proper parameters of the method must be supplied; the probability of choosing parents locally (δ) and the replacement mechanism (n_r)

Using the GDE3 method.

The employed command is as follows:

```
-algorithm,@GDE3,-problem,@TESIS,-operator,@DE,-N,100,-evaluation,1000,-DE_parameter,CR-F-PrM-disM
```

Besides the reference to the method GDE3, the differential evolution parameters must be included, like in the case of using MOEA/D-DE.

4.9.1.2. Executing when binary encoding is considered.

For the present research, two methods were explored when binary encoding was used: NSGA-II and SMS-EMOA. The used command could be as:

```
-algorithm,@NSGAII,-problem,@TESIS,-operator,@EAbinary,-N,50,-evaluation,10000000,-EAbinary_parameter,PrC
```

In this case, the operator is referenced as @*EAbinary* and the crossover probability is used as a single parameter.

4.9.2. Executing PlatEMO by using the HPC.

Once described how PlatEMO is executed by using commands, a code to read such parameters and to launch PlatEMO in the High-Performance Computer is needed. Such a code builds the command to execute PlatEMO and finally, receives the results at the end of the process. Such results consist of the hypervolume evolution and the final population.

5. CHAPTER 5: APPLICATIONS. EXPERIMENTAL RESULTS.

In the present chapter, the methodology which was presented in the Chapter IV is applied when several studies are faced. To do that, a case study is presented and analysed.

1. The first study looks for exploring the efficiency and knowledge of the methodology, which consists of optimising simultaneously systems design alternatives and their maintenance strategies (including preventive maintenance). Finding the best set of non-dominated solutions is searched for by employing the system availability with taking into consideration the associated operational cost, while automatically selecting the system devices. Multi-objective Evolutionary Algorithms and Discrete Event Simulation are coupled. Each solution supplied by the Multi-Objective Evolutionary Algorithm is analysed by employing Discrete Event Simulation in a procedure that looks at the effect of including periodic preventive maintenance tasks all along the mission time. An industrial case study is solved and a comparison of the performance of five state-of-the art Multi-objective Evolutionary Algorithms is handled. A real encoding approach is explored, and the hour is used as a time unit to define the operating and recovery times. Furthermore, the effect of the discrete event sampling size is analysed with useful insights about the synergies of Multi-objective Evolutionary Algorithms and Discrete Event Simulation. Finally, the methodology is expanded to more complex systems which are successfully solved.
2. Once the methodology was thoroughly studied under a real encoding approach when the hour is used as a time unit, such a methodology is deeper explored in a second study. This consists of extending the case study when binary encoding is used, and both the day and the week are employed as a time unit. Some applications might be benefited depending on the encoding. Moreover, the flexibility regarding the time unit to schedule preventive maintenance tasks could have a positive impact when such tasks must be managed.

3. With the third study, which is applied to the same case study, the effect of multi-objectivisation is explored when the Cost objective considers not only the Operational Cost but also the Acquisition Cost. Therefore, the cost objective is considered on both a two-objective approach (Availability-Cost) and a three-objective approach, where the Cost objective is decomposes between Operational and Acquisition Cost (Availability-Operational Cost-Acquisition Cost).

Once the methodology was thoroughly analysed, it is extended to other engineering field. The three studies cited above refer to the hydraulic engineering field. In order to test the applicability of the methodology, a fourth study is developed in which such a methodology is applied to a Substation Communication Networks (SCN) architecture in the energy field.

5.1. Case study.

A containment spray injection system (CSIS) of a nuclear power plant is used along the present chapter to develop several experiments. Such a system presents a simplified model that is shown in the Figure 5.1.

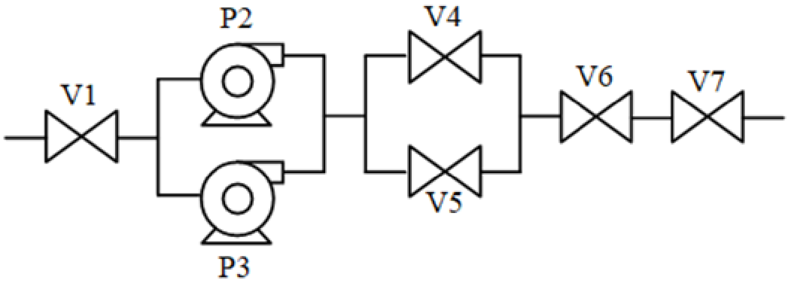


Figure 5.1: Containment spray injection system.

The model is made by employing impulsion pumps (P_i) and cut valves (V_i). The mission of the CSIS consists of the injection of borated water into the containment to wipe radioactive contamination released after a loss of coolant accident. As it is shown, the redundancies are limited to the valve V4 and the pump P2. Such devices may be or not included in the design of the system. Although the case study is based on the application presented by Greiner et al. [5], the data employed in this case were updated (and previously used by Cacereño et al. [213]). Such data, are defined as follows:

- Life Cycle: System mission time. It uses the hour as a time unit.
- Corrective Maintenance Cost: The cost involved in developing a repair task to recover the system after a failure. It is expressed in economic units per hour.
- Preventive Maintenance Cost: The cost involved in developing a Preventive Maintenance task. It is expressed in relation to the Corrective Maintenance Cost.
- Pump TF_{min}: Minimum operation time to failure for a pump without Preventive Maintenance. It uses the hour as a time unit.
- Pump TF_{max}: Maximum operation time to failure for a pump without Preventive Maintenance. It uses the hour as a time unit.
- Pump $TF\ \lambda$: Failure rate for a pump, which follows an exponential failure distribution. It is expressed as failures per hour.
- Pump TR_{min}. Minimum time to repair or duration of a corrective maintenance task for a pump. It uses the hour as a time unit.
- Pump TR_{max}. Maximum time to repair or duration of a corrective maintenance task for a pump. It uses the hour as a time unit.
- Pump $TR\ \mu$. Mean for the normal distribution followed for the time to repair assumed for a pump. It uses the hour as a time unit.
- Pump $TR\ \sigma$. Standard deviation for the normal distribution followed for the time to repair assumed for a pump. It uses the hour as a time unit.
- Pump TM_{min}. Minimum operation time to start a scheduled preventive maintenance task for a pump. It uses the hour as a time unit.
- Pump TM_{max}. Maximum operation time to start a scheduled preventive maintenance task for a pump. It uses the hour as a time unit.
- Pump TCM_{min}. Minimum time to conduct a preventive maintenance task for a pump. It uses the hour as a time unit.

- Pump TCM_{max}. Maximum time to conduct a preventive maintenance task for a pump. It uses the hour as a time unit.
- Valve TF_{min}. Minimum operation time to failure for a valve without preventive maintenance. It uses the hour as a time unit.
- Valve TF_{max}. Maximum operation time to failure for a valve without preventive maintenance. It uses the hour as a time unit.
- Valve $TF\ \lambda$. Failure rate for a valve, which follows an exponential failure distribution. It is expressed as failures per hour.
- Valve TR_{min}. Minimum time to repair or duration of a corrective maintenance task for a valve. It uses the hour as a time unit.
- Valve TR_{max}. Maximum time to repair or duration of a corrective maintenance task for a valve. It uses the hour as a time unit.
- Valve $TR\ \mu$. Mean for the normal distribution followed for the time to repair assumed for a valve. It uses the hour as a time unit.
- Valve $TR\ \sigma$. Standard deviation for the normal distribution followed for the time to repair assumed for a valve. It uses the hour as a time unit.
- Valve TM_{min}. Minimum operation time to start a scheduled preventive maintenance task for a valve. It uses the hour as a time unit.
- Valve TM_{max}. Maximum operation time to start a scheduled preventive maintenance task for a valve. It uses the hour as a time unit.
- Valve TCM_{min}. Minimum time to conduct a preventive maintenance task for a valve. It uses the hour as a time unit.
- Valve TCM_{max}. Maximum time to conduct a preventive maintenance task for a valve. It uses the hour as a time unit.

The values employed regarding each parameter are summarised in the Table 5.1.

Parameter	Value	Source
Life cycle or mission time	700.800 hours	-
Corrective Maintenance Cost	0,5 economic units/hour	MRI*
Preventive Maintenance Cost	0,125 economic units/hour	MRI*
Pump TF_{min}	1 hour	MRI*
Pump TF_{max}	70.800 hours	MRI*
Pump $TF\ \lambda$	159,57x10^{-6} failures/hour	OREDA 2009
Pump TR_{min}	1 hour	MRI*
Pump TR_{max}	24,33 hours	$\mu + 4\sigma$
Pump $TR\mu$	11 hours	OREDA 2009
Pump $TR\sigma$	3,33 hours	$(\mu - TR_{min})/3$
Pump TM_{min}	2.920 hours	MRI*
Pump TM_{max}	8.760 hours	MRI*
Pump TCM_{min}	4 hours	MRI*
Pump TCM_{max}	8 hours	MRI*
Valve TF_{min}	1 hour	MRI*
Valve TF_{max}	70.800 hours	MRI*
Valve $TF\ \lambda$	44,61x10^{-6} failures/hour	OREDA 2009
Valve TR_{min}	1 hour	MRI*
Valve TR_{max}	20,83 hours	$\mu + 4\sigma$
Valve $TR\mu$	9,5 hours	OREDA 2009
Valve $TR\sigma$	2,83 hours	$(\mu - TR_{min})/3$
Valve TM_{min}	8.760 hours	MRI*
Valve TM_{max}	30.040 hours	MRI*
Valve TCM_{min}	1 hours	MRI*
Valve TCM_{max}	3 hours	MRI*
*MRI = Machinery Reliability Institute		

Table 5.1: Reliability and cost data.

The data were obtained from specific literature [90], expert judgement (based on the professional experience from the Machinery & Reliability Institute (MRI), Alabama, USA) or mathematics relationships. In this sense, the $TR\ \sigma$ for valves and pumps have been set in relation to the μ of their respective normal distribution functions and their TCM_{min} previously established. Regarding the TCM_{max}, it is known that the 99.7% of the values of a normally distributed variable are included into the interval $\mu\ \pm\ 3\sigma$. In this case, the interval is extended to $\mu\ \pm\ 4\sigma$, taking into account anecdotal further values. The optimisation objectives consist of maximising the system Availability and minimising, the Operational Cost due to shutdowns (both because the system is being repaired and because the system is being maintained) and, in some cases the Acquisition Cost. To do that:

- Establishing the optimum period to perform the preventive maintenance tasks for the system's devices is needed, and,

- Deciding the inclusion of the redundant devices P2 and/or V4 by evaluating design alternatives is needed. Including redundant devices will improve the system Availability but it will also increase the Costs.

5.2. Testing the methodology.

5.2.1. Background.

Achieving the physical assets optimal performance is of critical importance when new industrial facilities are projected and built. Getting integrated design alternatives and maintenance strategies help to the decision makers to improve such performance. Coupling Evolutionary Algorithms and Discrete Event Simulation has been explored both in relation to systems design and maintenance strategies. However, it was not simultaneously considered when both the corrective and the preventive maintenance - consisting of supplying the optimum period of time to conduct a preventive maintenance task - are taken into account.

As it was explained above, the proposed methodology is thoroughly explored in this first study in order to find the best set of non-dominated solutions when the system’s Availability and the Operational Cost are considered as objective functions. The industrial case study previously presented is solved and a comparison of the performance of five state-of-the art Multi-objective Evolutionary Algorithms is handled. A real encoding approach is explored, and the hour is used as a time unit. A deep discussion is faced regarding the effect of the discrete event sampling size with useful insights about the synergies of Multi-objective Evolutionary Algorithms and Discrete Event Simulation. Finally, the methodology is expanded to more complex systems which are successfully solved.

5.2.2. Detailing the study.

As it was commented above, the system’s Availability and its Operational Cost are considered as the objective functions for this first study. Such objectives are computed by using the Equations 4.2 and 4.3 from the Chapter IV. Maximum Availability and minimum Operational Cost are desirable. The more investment in preventive maintenance, the greater the system Availability. Conversely, it implies the growth of unwanted Operational Cost and constitutes a conflict between objectives. The methodology which was shown in the Chapter IV is applied to the containment spray injection system (CSIS) of a nuclear power plant. The structure of the system is shown in the Figure 5.1. A comparative between the performances achieved from several multi-objective optimisation methods (SMS-EMOA, MOEA/D, MOEA/D-DE, NSGA-II y GDE3) and configurations of them is developed.

Evolutionary Algorithms employ a population of individuals (chromosomes), which represent candidate solutions to the problem to solve. As real encoding is used for the present study, each chromosome will be formed by a string of real numbers, which take 0 as the minimum value and 1 as the maximum value. Each string will be codified as $[D_1\ D_2\ M_1\ M_2\ M_3\ M_4\ M_5\ M_6\ M_7]$, where the presence of the redundant devices P2 and V4, is defined by the decision variables D_1 and D_2, respectively. The optimum time to start a preventive maintenance task regarding each device is denoted by the decision variables M_1 to M_7. As it was explained in the section 4.5.2.1 of the Chapter IV, such variables must be transformed in order to build the system's Functionability Profile and evaluate the objective functions.

- The decision variables D_1 and D_2 are rounded to the nearest integer, where 0 indicates that the device is not included and 1 the opposite.
- The decision variables M_1 to M_7 must be transformed to values within the interval $[TM_{min}, TM_{max}]$ by following the Equation 4.5 from the Chapter IV. The applied transformations are shown in the Equations 5.1 and 5.2 for pumps and valves, respectively.

$$TM_{1,4,5,6,7} = round\ (2{,}920 + M_{1,4,5,6,7} \cdot (8{,}760 - 2{,}920)) \qquad (5.1)$$

$$TM_{2,3} = round\ (8{,}760 + M_{2,3} \cdot (35{,}040 - 8{,}760)) \qquad (5.2)$$

The set of parameters to configure the experiments are shown in the Table 5.2. In the Section 4.5.3 from the Chapter IV, information regarding both the meaning of each variable and the process to set the values is supplied.

Method	Population (N)	PrM	disM	PrC	disC	CR	F	Approach	δ	n_r
SMS-EMOA	50 -100 -150	0.5-1-1.5	20	0.9	20	-	-	-	-	-
MOEA/D	50 -100 -150	0.5-1-1.5	20	0.9	20	-	-	Tchebycheff	-	-
MOEA/D-DE	50 -100 -150	1	20	-	-	0.9	0.4-0.5-0.6	-	0.9	1
NSGA-II	50 -100 -150	0.5-1-1.5	20	0.9	20	-	-	-	-	-
GDE3	50 -100 -150	1	20	-	-	0.9	0.4-0.5-0.6	-	-	-

Table 5.2: Parameters to configure the experiments.

Each method was executed by using population sizes (N) of 50, 100 and 150 individuals respectively. Population size plays a crucial role in maintaining the equilibrium between exploration and exploitation. When optimal solutions are searched, populations with excessive size could lead to slow convergences, whereas populations with few individuals could lead to premature stagnation, converging to local optimums [32,207,208]. Nine different configurations of the five methods were simulated and each configuration was executed 21 times (for statistical purposes). A total of 10,000,000 evaluations was used as the stopping criterion.

In order to evaluate the objective functions, building the system's Functionability Profile (as it was exposed in the Section 4.3 from the Chapter IV) is needed. Scale factors in relation to the value of the objective functions must be used in order to achieve a dispersed nondominated front with the unit as a maximum value (normalised objective functions values). The values were obtained by using a practical approach. The values of the scale factors are extracted from the values of the objective functions when the optimisation process starts. This approach assumes that the values of the objective functions will improve over the course of the evolutionary process. The scale factors were set as follows:

- The scale factor employed to compute the Operational Cost was 1,700 economic units.
- The scale factor used to compute the system Unavailability was 0.003.

Finally, a two-dimensional reference point is needed in order to compute the Hypervolume indicator. The reference point must cover the values limited by the scale factors, which restrict the values of the objective functions to a maximum of one. The reference point was set to (2,2). After achieving a compiled version of PlatEMO, which includes the code to solve the systems design and maintenance strategy problem, this was executed by using the High-Performance Computer. In the Section 4.8 from the Chapter IV, the codes generated and included in the PlatEMO software platform are shown.

5.2.3. Results and discussions.

In the Section 4.7 from the Chapter IV, the followed procedure to show the results is explained. This consists of showing the evolution of the hypervolume (average from 21 executions per configuration) in relation to the number of evaluations. Moreover, box plots are created regarding the achieved values at the end of the process, so the results are visually exposed and information in relation to the group of data is brought into the light. Besides the visual information previously cited, the Average, Mean, Maximum, Minimum and Standard Deviation regarding the reached hypervolume values (from 21 executions) is shown for each experiment. Next, a rigorous statistical test is conducted in order to find differences between performances. Finally, the non-dominated front is achieved, which is formed by the best-balanced Availability-Cost solutions.

5.2.3.1. Results from each Multi-objective Evolutionary Algorithms.

The consumed computational time is shown in the Table 5.3. The average time denotes the computational time regarding each one of twenty-one executions and nine different configurations (real time consumed). The sequential time denotes the computational time that would have been needed in case of not employing the High-Performance Computer. The computational time demonstrates the importance of using the High-Performance Computer, which allows parallel processes. The relationship between methods, configurations and identifiers can be seen in the Table 5.4 (columns 1 to 3).

Method	Average time	Sequential Time
SMS-EMOA	5 days, 17 hours and 32 minutes	2 years, 11 months, 18 days and 17 hours
MOEA/D	5 days, 14 hours and 46 minutes	2 years, 10 months, 27 days and 5 hours
MOEA/D-DE	2 days, 16 hours and 40 minutes	1 year, 4 months, 22 days and 15 hours
NSGA-II	2 days, 18 hours and 18 minutes	1 year, 5 months, 5 days and 3 hours
GDE3	2 days, 18 hours and 38 minutes	1 year, 5 months, 7 days and 16 hours
TOTAL		**10 years, 1 month, 20 days and 9 hours**

Table 5.3: Computational cost (consumed time).

Method	Identifier	Configuration	Average	Median	Max.	Min.	St. Deviation	Av. Rank
SMS-EMOA	ID1	N = 50 - PrM = 0.5	2.2878	2.2876	2.3158	2.2604	0.0135	5.619
	ID2	N = 100 - PrM = 0.5	2.2910	2.2940	2.3228	2.2573	0.0156	4.476
	ID3	N = 150 - PrM = 0.5	**2.2982**	**2.2978**	2.3223	2.2687	0.0142	**3.428**
	ID4	N = 50 - PrM = 1.0	2.2831	2.2803	2.3079	2.2663	0.0114	6.381
	ID5	N = 100 - PrM = 1.0	2.2887	2.2829	2.3252	2.2659	0.0160	5.523
	ID6	N = 150 - PrM = 1.0	2.2919	2.2914	**2.3330**	2.2665	0.0168	4.857
	ID7	N = 50 - PrM = 1.5	2.2871	2.2861	2.3137	2.2595	0.0174	5.333
	ID8	N = 100 - PrM = 1.5	2.2894	2.2875	2.3211	2.2690	0.0160	5.381
	ID9	N = 150 - PrM = 1.5	2.2950	2.2952	2.3126	**2.2710**	**0.0101**	**4.000**
p-Value								0.018
MOEA/D	ID1	N = 50 - PrM = 0.5	2.2563	2.2502	2.2912	2.1834	0.0256	4.380
	ID2	N = 100 - PrM = 0.5	2.2420	2.2510	2.2723	2.1689	0.0251	5.523
	ID3	N = 150 - PrM = 0.5	2.2392	2.2448	2.2898	2.1283	0.0322	5.857
	ID4	N = 50 - PrM = 1.0	2.2374	2.2443	2.2750	2.1540	0.0329	5.904
	ID5	N = 100 - PrM = 1.0	2.2495	2.2580	2.2993	2.2058	**0.0248**	4.714
	ID6	N = 150 - PrM = 1.0	**2.2592**	**2.2664**	2.3116	**2.2071**	0.0278	**4.190**
	ID7	N = 50 - PrM = 1.5	2.2482	2.2534	2.2894	2.1737	0.0266	5.190
	ID8	N = 100 - PrM = 1.5	2.2448	2.2547	2.2868	2.1493	0.0346	4.904
	ID9	N = 150 - PrM = 1.5	2.2575	2.2524	**2.3150**	2.2029	0.0258	**4.333**
p-Value								0.292
MOEA/D-DE	ID1	N = 50 - F = 0.4	2.2689	2.2694	2.2999	2.2462	0.0147	5.857
	ID2	N = 100 - F = 0.4	2.2767	2.2728	**2.3261**	2.2508	0.0162	4.428
	ID3	N = 150 - F = 0.4	2.2743	2.2770	2.3030	2.2546	**0.0138**	4.809
	ID4	N = 50 - F = 0.5	2.2676	2.2655	2.3098	2.2474	0.0160	6.050
	ID5	N = 100 - F = 0.5	**2.2828**	**2.2851**	2.2364	**2.2554**	0.0157	**3.571**
	ID6	N = 150 - F = 0.5	2.2764	2.2770	2.3152	2.2469	0.0157	4.523
	ID7	N = 50 - F = 0.6	2.2647	2.2636	2.3041	2.2307	0.0143	6.619
	ID8	N = 100 - F = 0.6	2.2764	2.2786	2.3110	2.2449	0.0202	4.761
	ID9	N = 150 - F = 0.6	2.2757	2.2752	2.3062	2.2395	0.0159	**4.333**
p-Value								0.005
NSGA-II	ID1	N = 50 - PrM = 0.5	2.2831	2.2859	2.3180	2.2611	0.0162	5.809
	ID2	N = 100 - PrM = 0.5	2.2864	2.2872	2.3007	2.2625	**0.0118**	5.095
	ID3	N = 150 - PrM = 0.5	2.2955	**2.3011**	2.3227	2.2635	0.0186	**4.142**
	ID4	N = 50 - PrM = 1.0	2.2801	2.2820	2.3070	2.2606	0.0141	6.571
	ID5	N = 100 - PrM = 1.0	2.2944	2.2941	**2.3390**	**2.2714**	0.0163	4.190
	ID6	N = 150 - PrM = 1.0	2.2874	2.2871	2.3217	2.2623	0.0174	5.333
	ID7	N = 50 - PrM = 1.5	2.2898	2.2879	2.3277	2.2534	0.0240	5.285
	ID8	N = 100 - PrM = 1.5	2.2931	2.2895	2.3266	2.2658	0.0166	4.557
	ID9	N = 150 - PrM = 1.5	**2.2957**	2.2941	2.3281	2.2592	0.0158	**4.000**
p-Value								0.035
GDE3	ID1	N = 50 - F = 0.4	2.2851	2.2853	2.3166	2.2613	0.0125	5.619
	ID2	N = 100 - F = 0.4	2.2864	2.2886	2.3018	2.2663	**0.0093**	5.381
	ID3	N = 150 - F = 0.4	2.2852	2.2905	2.3000	2.2666	0.0116	5.809
	ID4	N = 50 - F = 0.5	**2.2957**	2.2884	2.3376	2.2679	0.0209	**4.571**
	ID5	N = 100 - F = 0.5	2.2875	2.2827	2.3194	2.2627	0.0171	5.571
	ID6	N = 150 - F = 0.5	2.2927	2.2932	2.3210	**2.2684**	0.0137	**4.142**
	ID7	N = 50 - F = 0.6	2.2946	2.2898	**2.3575**	2.2540	0.0224	4.571
	ID8	N = 100 - F = 0.6	2.2897	**2.2929**	2.3236	2.2555	0.0149	4.666
	ID9	N = 150 - F = 0.6	2.2901	2.2897	2.3191	2.2610	0.0124	4.666
p-Value								0.439

Table 5.4: Id's, config., Hyperv. statistics and statistical test.

Next, some figures regarding the hypervolume evolution in relation to the number of evaluations and each Multi-objective Evolutionary Algorithm are shown. Firstly, the global evolution is shown. Secondly, the evolution from 9 to 10 million evaluations (the end of the process) is shown.

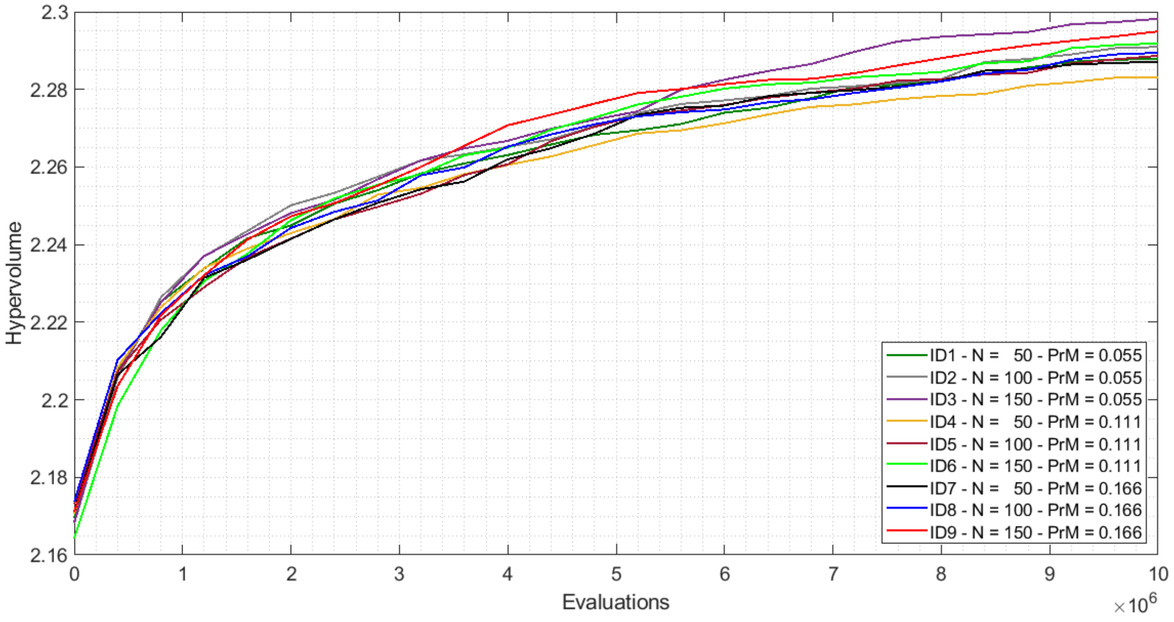


Figure 5.2: Hypervolume average vs. evaluations (SMS-EMOA).

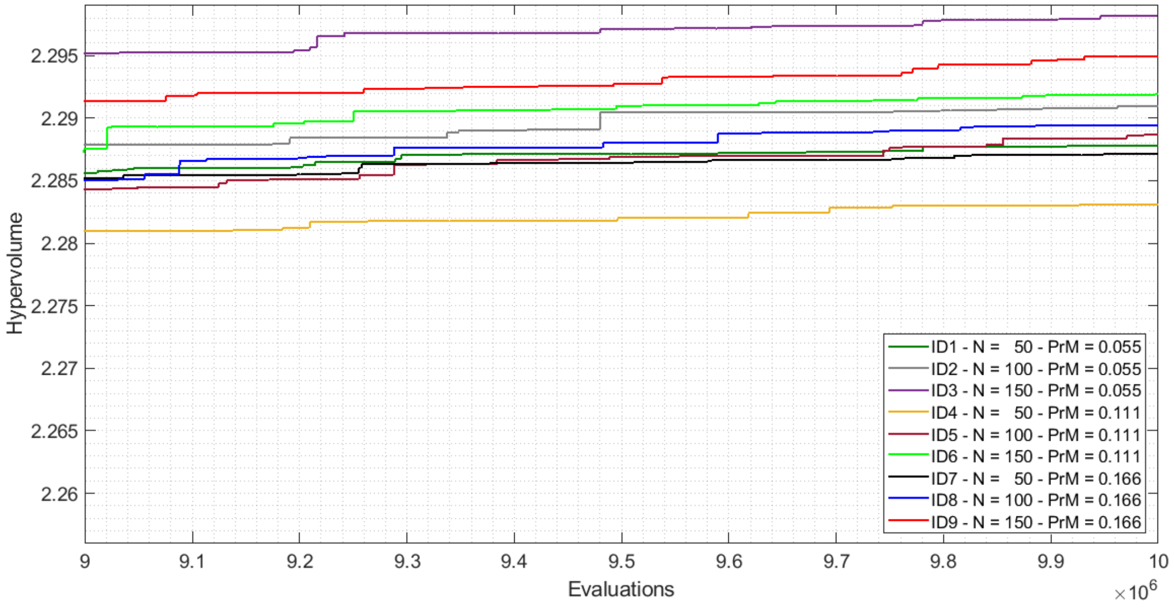


Figure 5.3: Hypervolume average vs. evaluations, detail (SMS-EMOA).

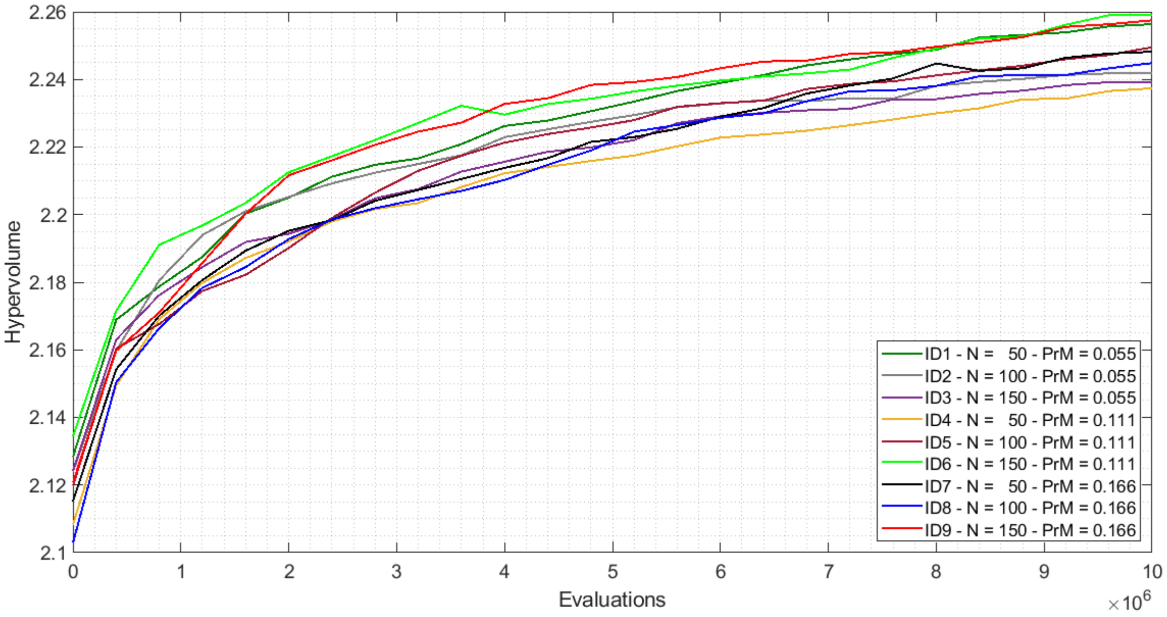


Figure 5.4: Hypervolume average vs. evaluations (MOEA/D).

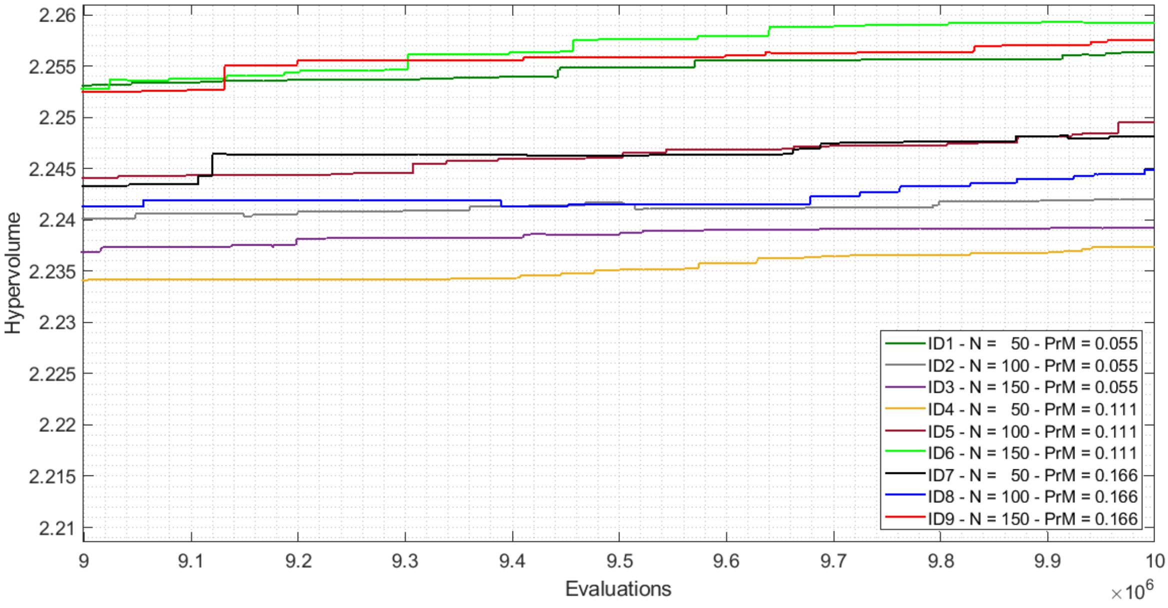


Figure 5.5: Hypervolume average vs. evaluations, detail (MOEA/D).

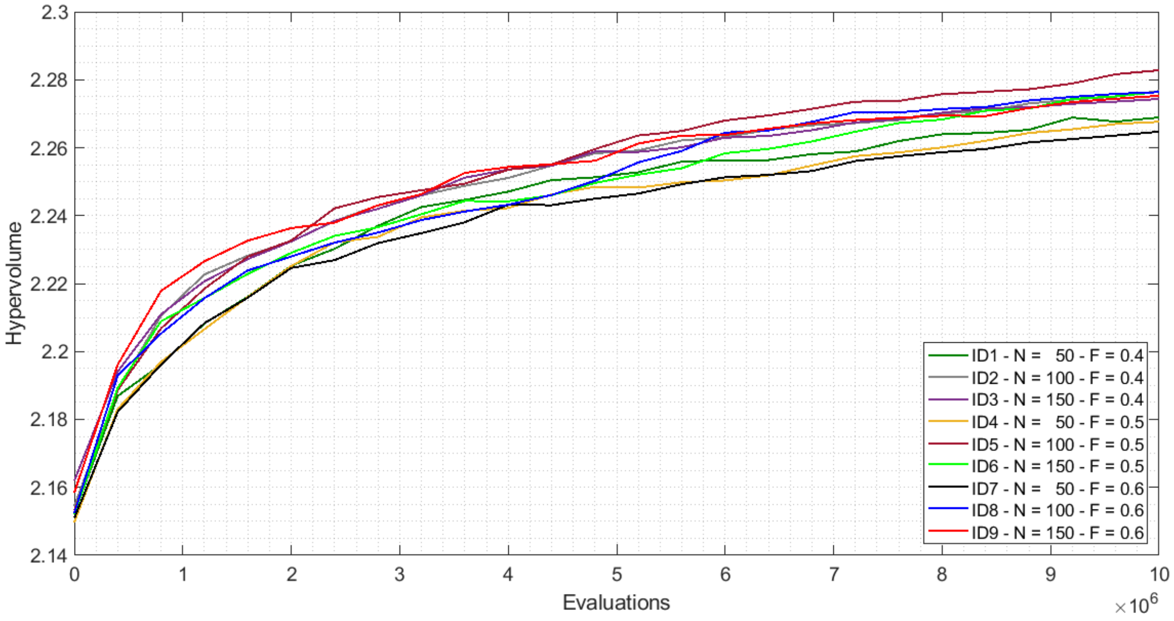


Figure 5.6: Hypervolume average vs. evaluations (MOEA/D-DE).

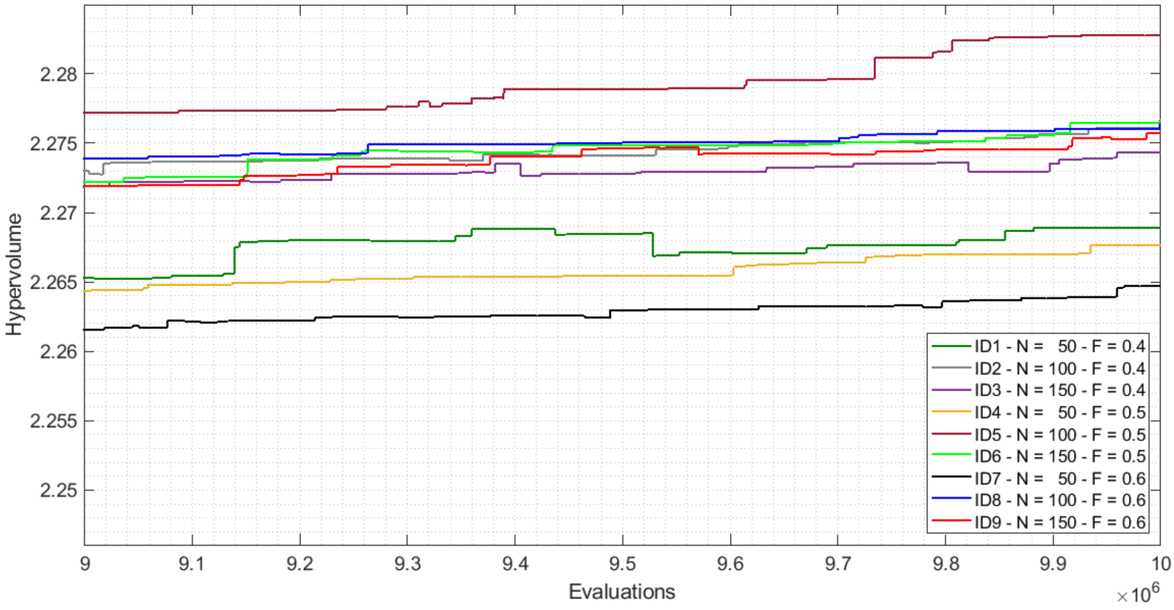


Figure 5.7: Hypervolume average vs. evaluations, detail (MOEA/D-DE).

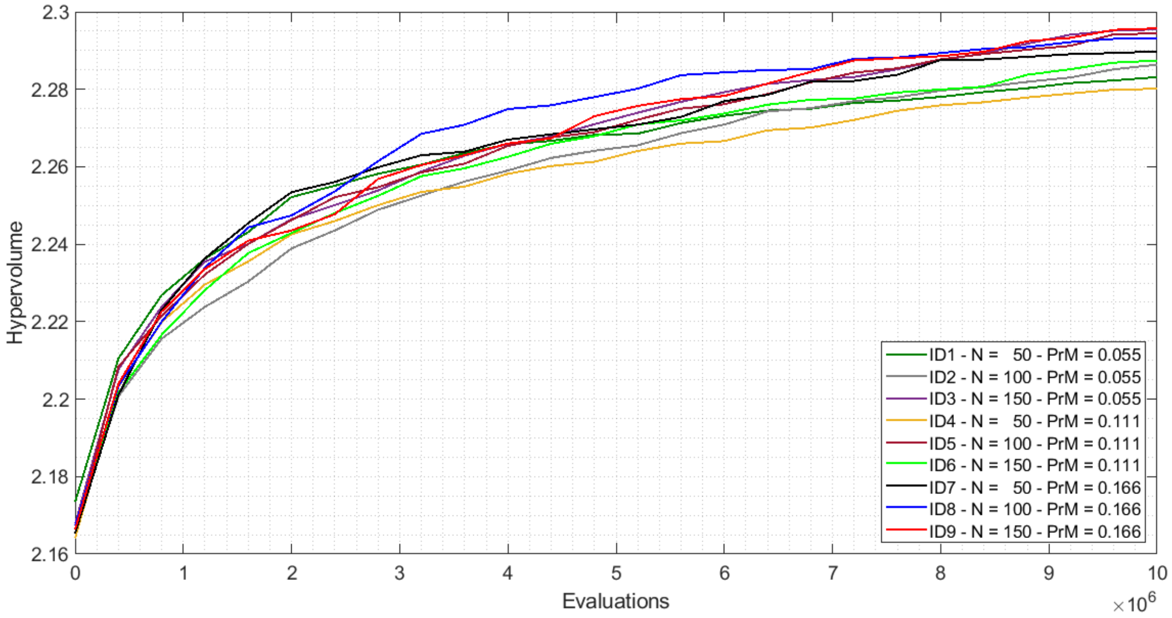


Figure 5.8: Hypervolume average vs. evaluations (NSGA-II).

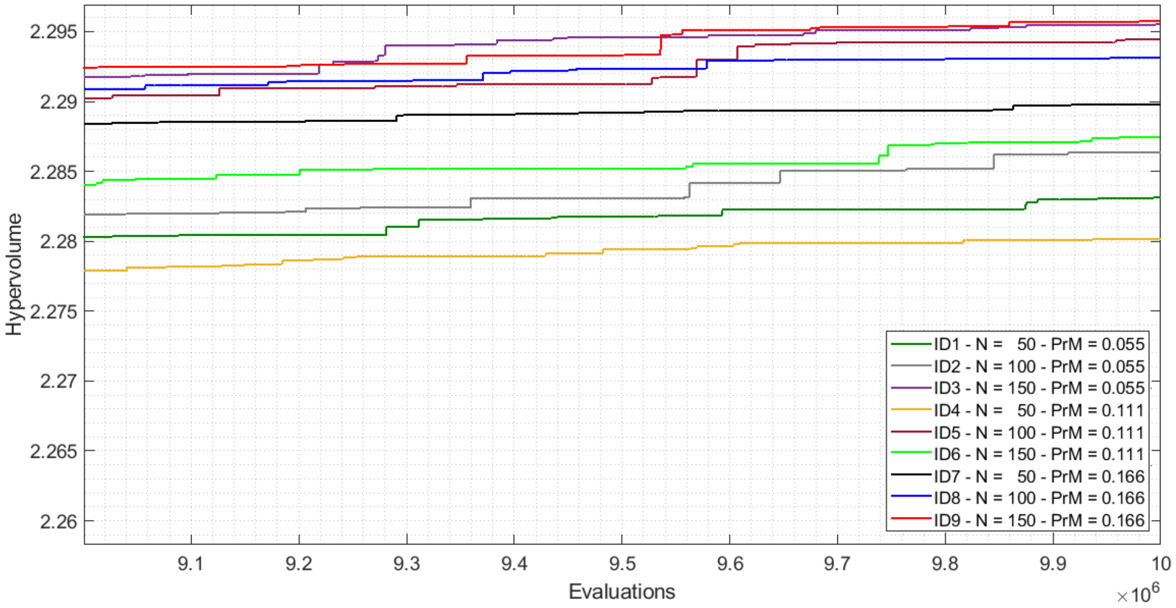


Figure 5.9: Hypervolume average vs. evaluations, detail (NSGA-II).

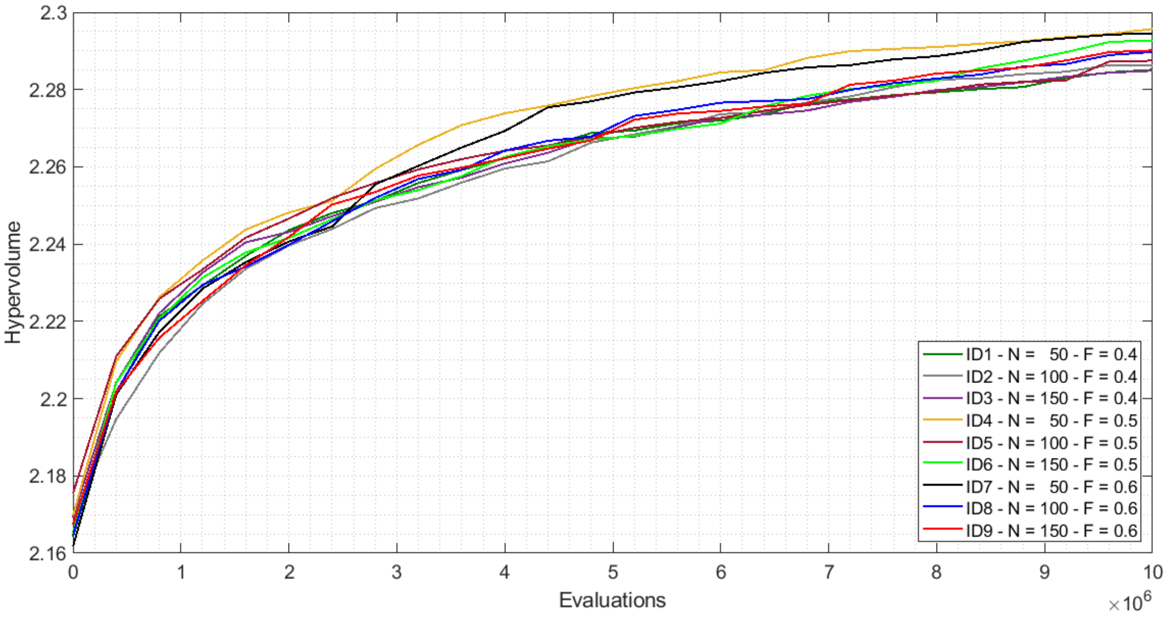


Figure 5.10: Hypervolume average vs. evaluations (GDE3).

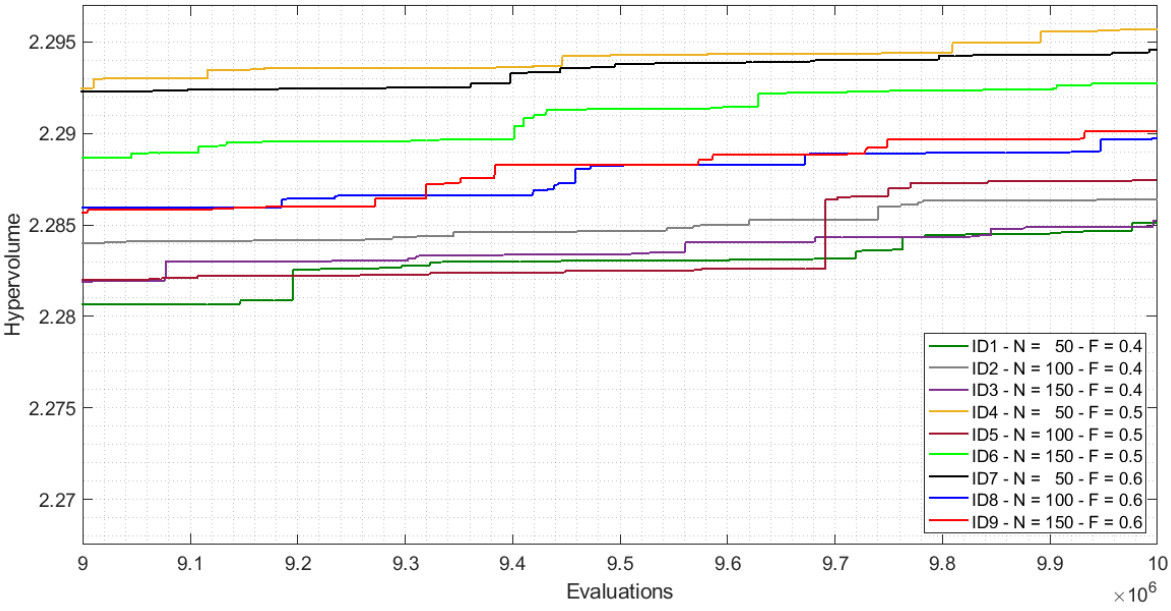


Figure 5.11: Hypervolume average vs. evaluations, detail (GDE3).

At the end of the process, it can be seen that:

- When the SMS-EMOA method is used, the configuration with identifier ID3 (population of 150 individuals and 0.5 gene per chromosome as a mutation rate) reaches the highest Hypervolume average value.
- When the MOEA/D method is employed, the configuration with identifier ID6 (population of 150 individuals and 1.0 gene per chromosome as a mutation rate) reaches the highest Hypervolume average value.
- When the MOEA/D-DE method is employed, the configuration with identifier ID5 (population of 100 individuals and 0.5 as an F parameter) reaches the highest Hypervolume average value.
- When the NSGAII method is used, the configuration with identifier ID9 (population of 150 individuals and 1.5 gene per chromosome as a mutation rate) reaches the highest Hypervolume average value.
- When the GDE3 method is employed, the configuration with identifier ID4 (population of 50 individuals and 0.5 as an F parameter) reaches the highest Hypervolume average value.

Box plots of the Hypervolume values distribution at the end of the process are shown between the Figures 5.12 to 5.16.

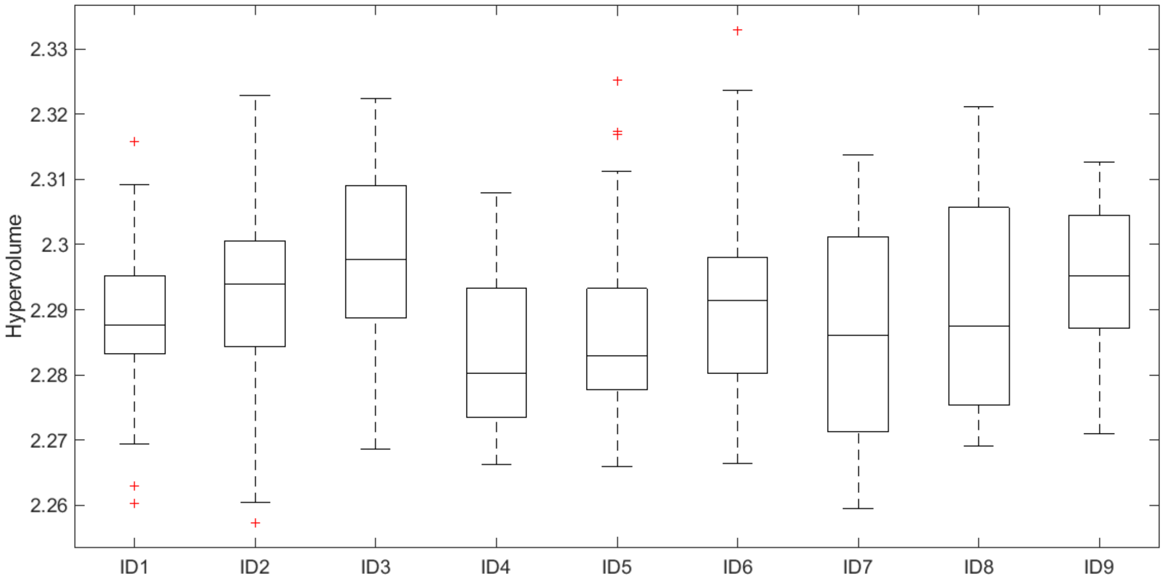


Figure 5.12: Hypervolume Box plots (SMS-EMOA, ID’s as in the Table 5.4).

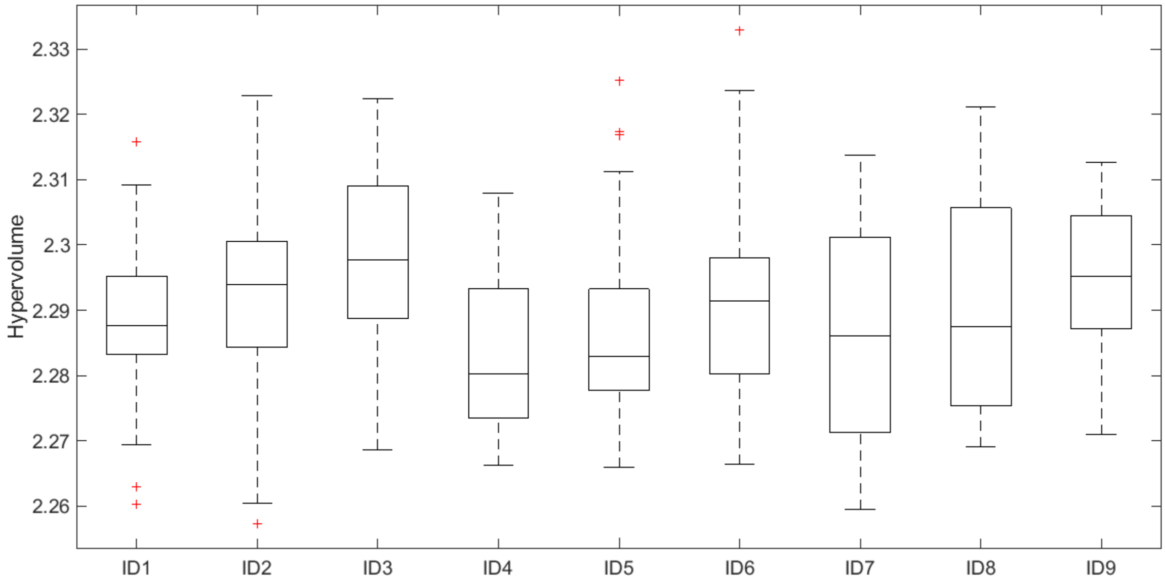


Figure 5.13: Hypervolume Box plots (MOEA/D, ID’s as in the Table 5.4).

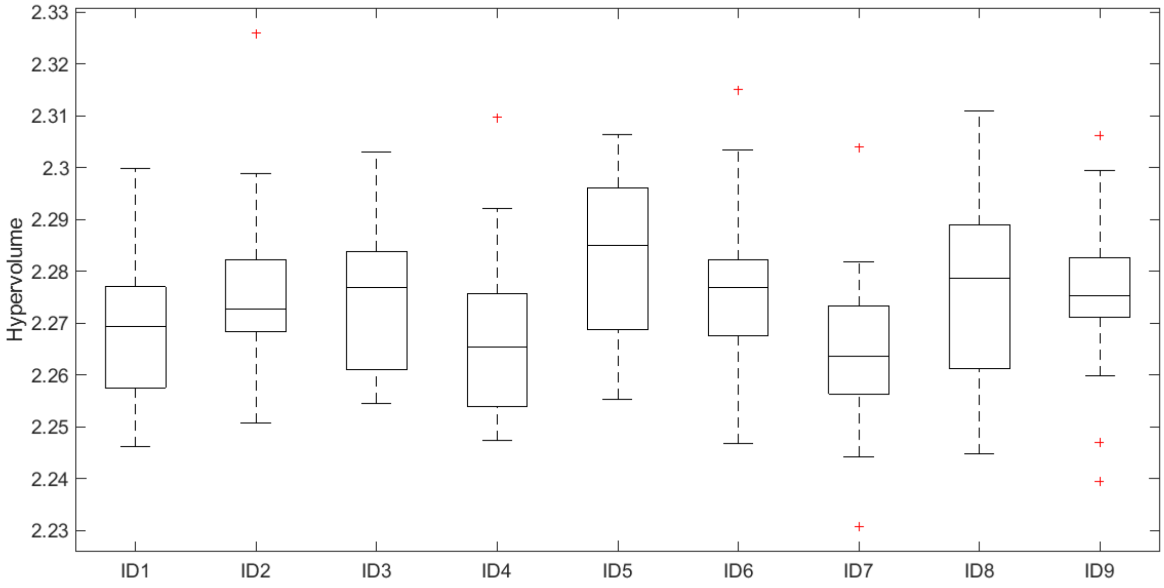


Figure 5.14: Hypervolume Box plots (MOEA/D-DE, ID’s as in the Table 5.4).

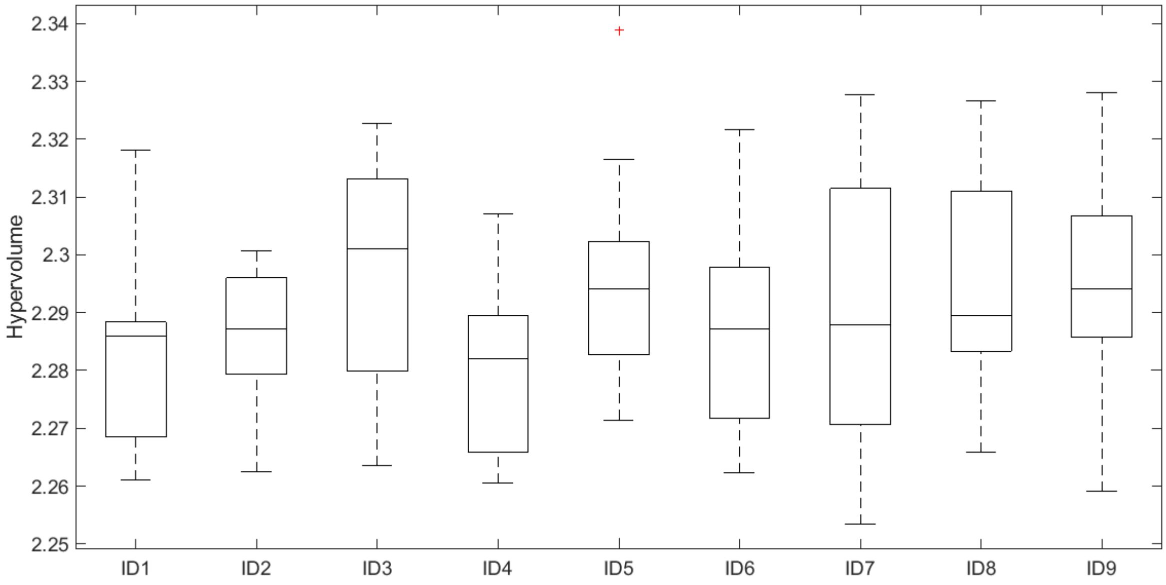


Figure 5.15: Hypervolume Box plots (NSGA-II, ID’s as in the Table 5.4).

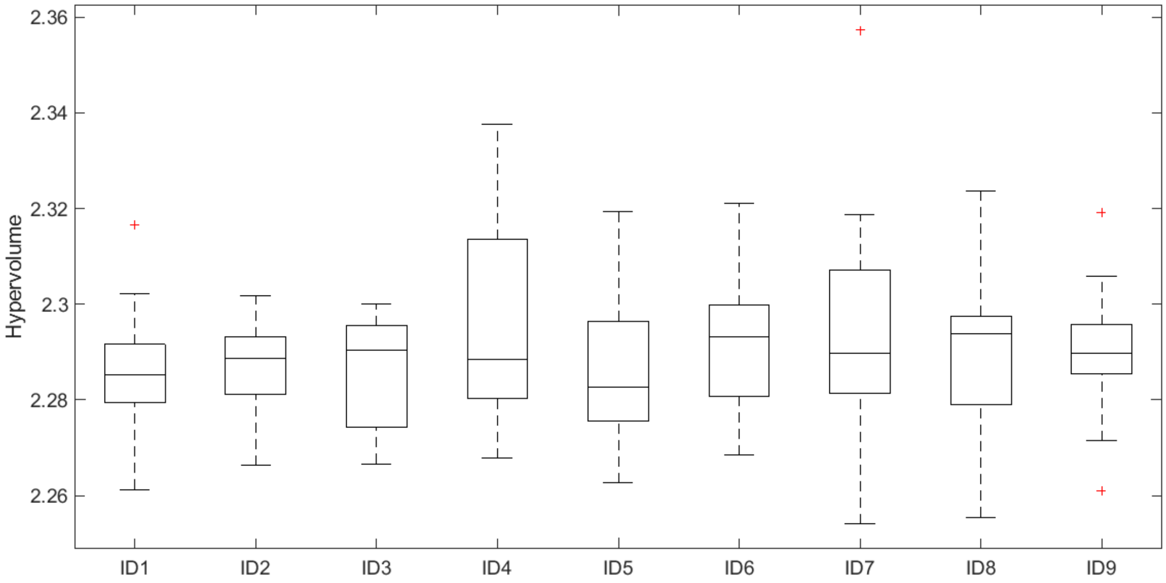


Figure 5.16: Hypervolume Box plots (GDE3, ID’s as in the Table 5.4).

Such box plots summarise the statistical information included in the Table 5.4 (columns 4 to 8), which makes reference to the Hypervolume Average, Median, Maximum, Minimum and Standard Deviation values reached at the end of the process. It is possible to conclude:

- Attending to the SMS-EMOA method, the configuration with identifier ID3 (population of 150 individuals and 0.5 gene per chromosome as a mutation rate) reaches the highest Hypervolume Average and Median values, the configuration with identifier ID6 (population of 150 individuals and 1 gene per chromosome as a mutation rate) presents the highest Hypervolume Maximum value, while the configuration with identifier ID9 (population of 150 individuals and 1.5 gene per chromosome as a mutation rate) reaches the highest Hypervolume Minimum and presents the lowest Standard Deviation values.
- Regarding the MOEA/D method, the configuration with identifier ID6 (population of 150 individuals and 1 gene per chromosome as a mutation rate) reaches the highest Hypervolume Average, Median and Minimum values, the configuration with identifier ID9 (population of 150 individuals and 1.5 gene per chromosome as a mutation rate) reaches the highest Hypervolume Maximum value and the configuration with identifier ID5 (population of 100 individuals and 1 gene per chromosome as a mutation rate) presents the lowest Standard Deviation value.
- In relation to the MOEA/D-DE method, the configuration with identifier ID5 (population of 100 individuals and 0.5 as an F parameter) reaches the highest Hypervolume Average, Median and Minimum values, the configuration with identifier ID2 (population of 100 individuals and 0.4 as an F parameter) reaches the highest Hypervolume Maximum value and the configuration with identifier ID3 (population of 150 individuals and 0.4 as an F parameter) presents the lowest Standard Deviation value.
- Attending to the NSGA-II method, the configuration with identifier ID9 (population of 150 individuals and 1.5 gene per chromosome as a mutation) reaches the highest Hypervolume Average value, the configuration with identifier ID3 (population of 150 individuals and 0.5 gene per chromosome as

a mutation rate) reaches the highest Hypervolume Median value, the configuration with identifier ID5 (population of 100 individuals and 1 gene per chromosome as a mutation rate) reaches the highest Hypervolume Maximum and Minimum values, and the configuration with identifier ID2 (population of 100 individuals and 0.5 gene per chromosome as a mutation rate) presents the lowest Hypervolume Standard Deviation value.

- Finally, regarding the GDE3 method, the configuration with identifier ID4 (population of 50 individuals and 0.5 as an F parameter) reaches the highest Hypervolume Average value, the configuration with identifier ID8 (population of 100 individuals and 0.6 as an F parameter) reaches the highest Hypervolume Median value, the configuration with identifier ID7 (population of 50 individuals and 0.6 as an F parameter) reaches the highest Hypervolume Maximum value, the configuration with identifier ID6 (population of 150 individuals and 0.5 as an F parameter) reaches the highest Hypervolume Minimum value and the configuration with identifier ID2 (population of 100 individuals and 0.4 as an F parameter) presents the lowest Hypervolume Standard Deviation value.

In order to determine if one of the nine configurations per method performs better than any other, a statistical significance hypothesis test was conducted. The Average Ranks computed by using the Friedman's test are shown in the Table 5.4 (column 9). It can be seen that:

- Regarding the SMS-EMOA method, the configuration with identifier ID3 (population of 150 individuals and 0.5 gene per chromosome as a mutation rate) presents the best Average Rank (in order to maximise the Hypervolume, the Average Rank must be as low as possible). However, the *p*-value computed (0.018) implies that the null hypothesis (H_0) can be rejected (*p*-value<0.05). Therefore, it is possible to conclude that, in the studied conditions, there are configurations that perform better than others. In order to find the concrete pairwise comparisons that produce differences, a post-hoc test was conducted. The Shaffer's test was used to compare the configuration with identifier ID3, which produced the best Average Rank in relation to the Friedman's test, with the rest of configurations. The adjusted *p*-values achieved inform the rejection or acceptance of the null hypothesis. The null hypothesis states that there are no significant differences among the behaviour of the configurations. The result regarding the comparisons is shown in the Table 5.5. It is possible to conclude that the configuration with identifier ID3 performs better than the configuration with identifier ID4 (population of 50 individuals and 1 gene per chromosome as a mutation rate) but is not possible to establish that the configuration with identifier ID3 performs better than any other. The configurations with identifiers ID3 (with population of 150 individuals and 0.5 gene per chromosome as a mutation rate) and ID9 (with population of 150 individuals and 1.5 gene per chromosome as a mutation rate) presented the lowest Average Ranks (as it is shown in the Table 5.4), so they are selected for the final comparison study among methods.

Method	Test	Comparison	*p*-value	Conclusion
SMS-EMOA	Shaffer	**ID3 - ID4**	**0.0171 < 0.05**	**The null hypothesis is rejected**
		ID1 - ID3	0.2673 > 0.05	The null hypothesis is not rejected
		ID3 - ID5	0.3687 > 0.05	The null hypothesis is not rejected
		ID3 - ID8	0.5847 > 0.05	The null hypothesis is not rejected
		ID3 - ID7	0.6779 > 0.05	The null hypothesis is not rejected
		ID3 - ID6	2.0013 > 0.05	The null hypothesis is not rejected
		ID2 - ID3	4.7330 > 0.05	The null hypothesis is not rejected
		ID3 - ID9	5.4885 > 0.05	The null hypothesis is not rejected
MOEA/D-DE	Shaffer	**ID5 - ID7**	**0.0111 < 0.05**	**The null hypothesis is rejected**
		ID4 - ID5	0.0790 > 0.05	The null hypothesis is not rejected
		ID1 - ID5	0.1915 > 0.05	The null hypothesis is not rejected
		ID3 - ID5	2.6919 > 0.05	The null hypothesis is not rejected
		ID5 - ID8	2.8612 > 0.05	The null hypothesis is not rejected
		ID5 - ID6	4.1567 > 0.05	The null hypothesis is not rejected
		ID2 - ID5	4.6574 > 0.05	The null hypothesis is not rejected
		ID5 - ID9	4.7752 > 0.05	The null hypothesis is not rejected
NSGA-II	Wilcoxon	**ID4 - ID9**	**0.0117 < 0.05**	**The null hypothesis is rejected**
		ID1 - ID9	**0.0250 < 0.05**	**The null hypothesis is rejected**
		ID2 - ID9	0.0680 > 0.05	The null hypothesis is not rejected
		ID6 - ID9	0.1592 > 0.05	The null hypothesis is not rejected
		ID7 - ID9	0.3754 > 0.05	The null hypothesis is not rejected
		ID8 - ID9	0.5202 > 0.05	The null hypothesis is not rejected
		ID5 - ID9	0.59012 > 0.05	The null hypothesis is not rejected
		ID3 - ID9	0.9584 > 0.05	The null hypothesis is not rejected

Table 5.5: *P*-values from the hypothesis tests.

- Attending to the MOEA/D method, the configuration with identifier ID6 (population of 150 individuals and 1.0 gene per chromosome mutation rate) presents the best Average Rank. However, the *p*-value computed (0.292) implies that the null hypothesis (H_0) can not be rejected (*p*-value>0.05), so it is possible to conclude that, in the studied conditions, no one single configuration performs better than any other. However, the configurations with identifiers ID6 (population of 150 individuals and 1.0 gene per chromosome as a mutation rate) and ID9 (population of 150 individuals and 1.5 gene per chromosome as a mutation rate) presented the lowest Average Ranks (as it shown in the Table 5.4), so they are selected for the final comparison study between methods.
- In relation to the MOEA/D-DE method, the configuration with identifier ID5 (population of 100 individuals and 0.5 as an F parameter) presents the best Average Rank. However, the *p*-value computed (0.005) implies that the null hypothesis (H_0) can be rejected (*p*-value<0.05), so, in the studied conditions, there are configurations that perform better than others. In order to find the concrete pairwise comparisons that produce differences, a post-hoc test was carried out. The Shaffer's test was used to compare the configuration with identifier ID5, which produced the lowest Average Rank in relation to the Friedman's test, with the rest of the configurations. The result related to the comparisons is shown in the Table 5.5. The configuration with identifier ID5 performs better than the configuration with identifier ID7 (population of 50 individuals and 0.6 as an F parameter) but it is not possible to establish that the configuration with identifier ID5 performs better than any other. The configurations with identifiers ID5 (population of 100 individuals and 0.5 as an F parameter) and ID9 (with population of 150 individuals and 0.6 as an F parameter) presented the lowest Average Ranks when the Friedman's test was used (as it is shown in the Table 5.4), so they were selected for the final comparison study among methods.
- Regarding the NSGA-II method, the configuration with identifier ID9 (population of 150 individuals and 1.5 gene per chromosome as a mutation rate) presents the best Average Rank. However, the *p*-value computed

(0.035) implies that the null hypothesis (H_0) can be rejected (*p*-value<0.05), so, in the studied conditions, there are some configurations that perform better than others. In order to find the concrete pairwise comparisons that produce such differences, a post-hoc test was conducted. The Shaffer's test was used to compare the configuration with identifier ID9, which produced the best Average Rank in relation to the Friedman's test, with the rest of configurations. With the obtained results (all post-hoc comparisons produced adjusted *p*-values bigger than 0.05), it is not possible to conclude that the configuration with identifier ID9 performs better than any other. While the Friedman's test establishes that significant differences exist among configurations, the accuracy level of the Shaffer's test does not allow for such a determination, as it was explained by Benavoli et al. [210]. They recommend carrying out the pairwise comparisons of the post-hoc analysis by using the Wilcoxon signed-rank test. The results of the Wilcoxon signed-rank test, in which the configuration with identifier ID9 is compared with the rest of the configurations, are shown in the Table 5.5. The configuration with identifier ID9 performs better than both the configuration with identifier ID1 (population of 50 individuals and 0.5 gene per chromosome as a mutation rate) and the configuration with identifier ID4 (population of 50 individuals and 1 gene per chromosome as a mutation rate). The configurations with identifiers ID9 (with population of 150 individuals and 1.5 gene per chromosome as a mutation rate) and ID3 (with population of 150 individuals and 0.5 gene per chromosome as a mutation rate) presented the lowest Average Ranks (as it is shown in the Table 5.4), so they were selected for the final comparison study among methods.

- Finally, for the GDE3 method, the configuration with identifier ID6 (population of 150 individuals and 0.5 as an F parameter) presents the best Average Rank. However, the *p*-value obtained (0.439) implies that the null hypothesis (H_0) cannot be rejected (*p*-value>0.05), so, in the studied conditions, no one configuration performs better than any other. However, the configurations with identifiers ID6 (population of 150 individuals and 0.5 as an F parameter) and ID4 (population of 50 individuals and 0.5 as an F parameter), presented the lowest Average Ranks (as it is shown in the Table 5.4), so they were selected for the final comparison study among methods.

The best accumulated non-dominated solutions obtained from the last generation of the evolutionary process for all executions and all configurations of each method are used to compute their respective accumulated Hypervolume values (as it was described by Fonseca et al [211]). Such values are shown in the Table 5.6 where, as it is expected, each method reaches a higher value than the shown Hypervolume Maximum values in the Table 5.4 (column 6), respectively.

Methods	Hypervolume Accumulated Value
SMS-EMOA	2.4087
MOEA/D	2.3844
MOEA/D-DE	2.3991
NSGA-II	2.4068
GDE3	2.4057
All methods	**2.4179**

Table 5.6: Hypervolume Accumulated Values.

5.2.3.2. Comparing all methods.

In the Table 5.3, both the sequential and the actual computational time, which were taken by the global optimisation process, are shown. The computational cost shows the vital importance of employing the High-Performance Computer. Previously, the configurations with the best Average Ranks according to the Friedman's test from all the analysed methods were selected to be globally compared. These configurations are shown in the Table 5.7 (columns 2 and 3).

Identifier	Method	Configuration	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	SMS-EMOA	N = 150 – PrM = 0.5	**2.2982**	2.2978	2.3223	2.2687	0.0142	**3.666**
ID2	SMS-EMOA	N = 150 – PrM = 1.5	2.2950	2.2952	2.3126	**2.2710**	**0.0101**	3.809
ID3	MOEA/D	N = 150 – PrM = 1.0	2.2592	2.2664	2.3116	2.2071	0.0278	7.999
ID4	MOEA/D	N = 150 – PrM = 1.5	2.2575	2.2524	2.3150	2.2029	0.0258	8.666
ID5	MOEA/D-DE	N = 100 – F = 0.5	2.2828	2.2851	2.3064	2.2554	0.0153	5.952
ID6	MOEA/D-DE	N = 150 – F = 0.6	2.2757	2.2752	2.3062	2.2395	0.0155	7.047
ID7	NSGA-II	N = 150 – PrM = 1.5	2.2957	2.2941	2.3281	2.2592	0.0158	4.190
ID8	NSGA-II	N = 150 – PrM = 0.5	2.2955	**2.3011**	2.3227	2.2635	0.0186	4.523
ID9	GDE3	N = 50 – F = 0.5	2.2957	2.2884	**2.3376**	2.2679	0.0209	4.476
ID10	GDE3	N = 150 – F = 0.5	2.2927	2.2932	2.3210	2.2684	0.0137	4.666
p-Value								$8.2 \cdot 10^{-11}$

Table 5.7: Id's, config., Hyperv. statistics and statistical test (all methods).

The Hypervolume average values evolution in relation to the evaluations number is shown in the Figure 5.17. The evolution from 9 to 10 million evaluations (the end of the process) is shown in the Figure 5.18. The configuration with identifier ID1 (with SMS-EMOA as an optimisation method, population of 150 individuals and 0.5 gene per chromosome as a mutation rate) reaches the highest Hypervolume Average value at the end of the process.

Moreover, box plots of the Hypervolume values distribution at the end of the process are shown in the Figure 5.19. They represent the statistical information supplied by the Table 5.7 (columns 4 to 8), in which it can be seen that the configuration with identifier ID1 (with SMS-EMOA as an optimisation method, population of 150 individuals and 0.5 gene per chromosome as a mutation rate) reaches the highest Hypervolume Average value, the configuration with identifier ID8 (with NSGA-II as an optimisation method, population of 150 individuals and 0.5 gene per chromosome as a mutation rate) reaches the highest Hypervolume Median value, the configuration with identifier ID9 (with GDE3 as an optimisation method, population of 50 individuals and 0.5 as an F parameter) presents the highest Hypervolume Maximum value, while the configuration with identifier ID2 (with SMS-EMOA as an optimisation method, population of 150 individuals and 1.5 gene per chromosome as a mutation rate) presents both the highest Hypervolume Minimum value and the lowest Hypervolume Standard Deviation value.

In order to determine of any configuration performs better than any other, a statistical significance hypothesis test was carried out. The Average Ranks computed by using the Friedman's test are shown in the Table 5.7 (column 9). The configuration with identifier ID1 (with SMS-EMOA as an optimisation method, population of 150 individuals and 0.5 gene per chromosome as a mutation rate) produces the lowest Average Rank. However, the *p*-value computed ($8.2594 \cdot 10^{-11}$) implies that the null hypothesis (H_0) can be rejected (*p*-value < 0.05), so, in the studied conditions, there are configurations that perform better than others.

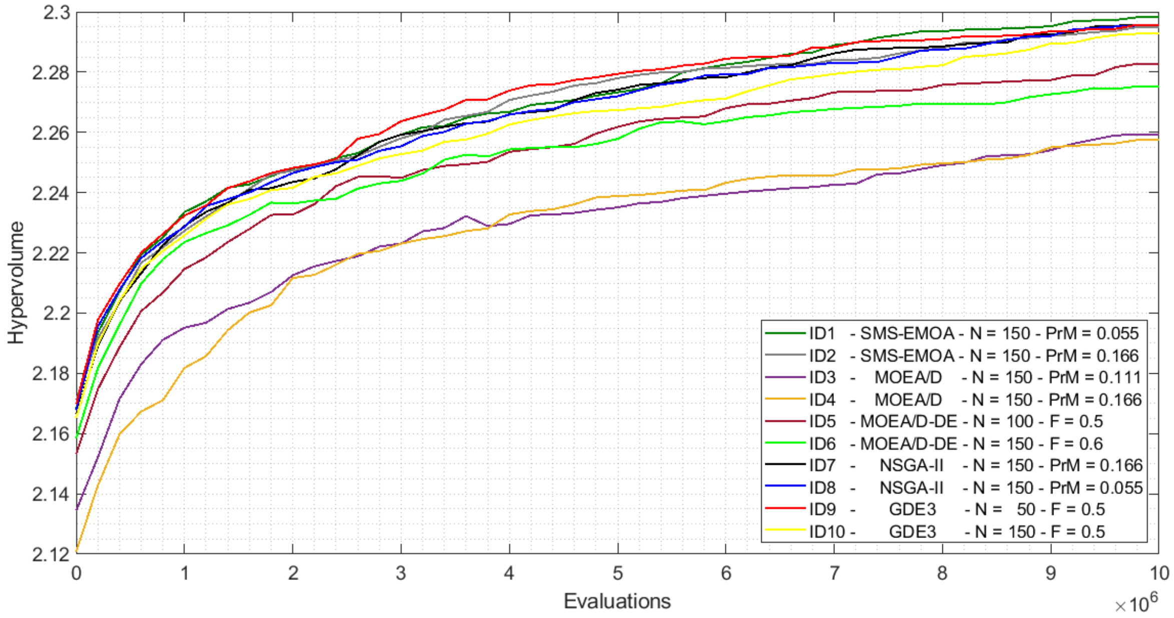


Figure 5.17: Hypervolume average vs. evaluations (all methods).

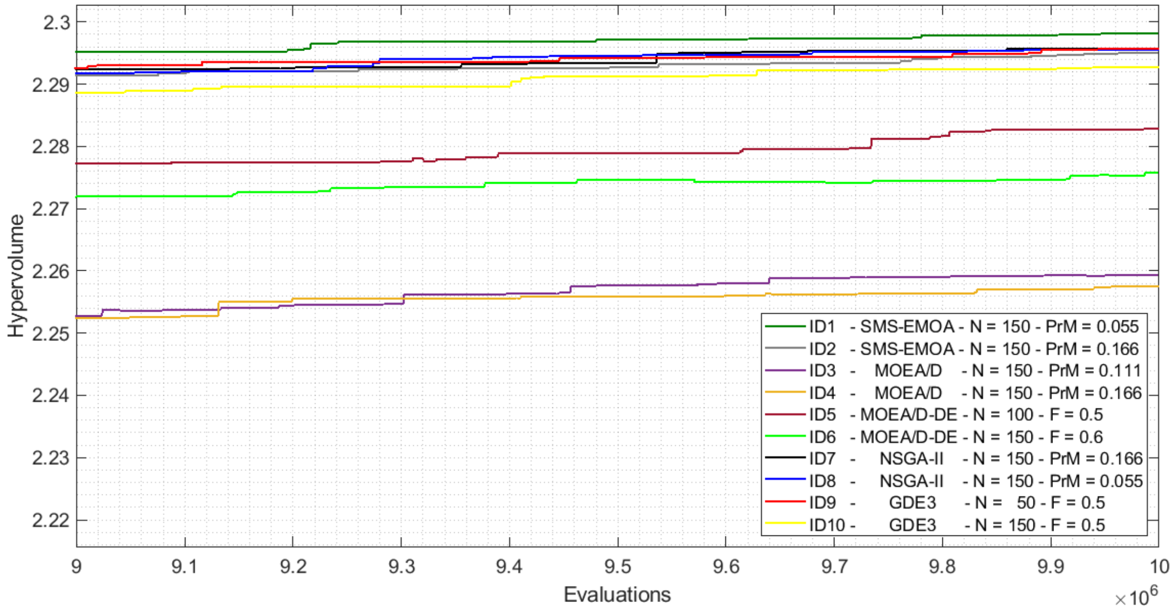


Figure 5.18: Hypervolume average vs. evaluations, detail (all methods).

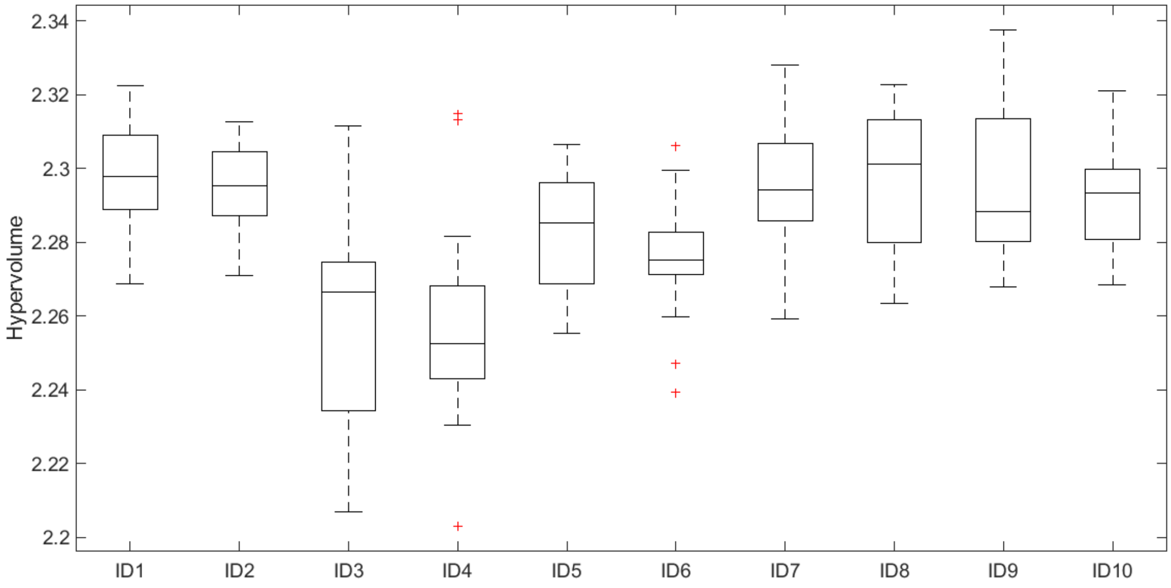


Figure 5.19: Hypervolume Box plots (ID's. as in the Table 5.7).

In order to find the concrete pairwise comparisons that produce such differences, a post-hoc test was conducted. The Shaffer's test was used to compare the configuration with identifier ID1 (with SMS-EMOA as an optimisation method, population of 150 individuals and 0.5 gene per chromosome as a mutation rate), which produces the lowest Average Rank regarding the Friedman's test, with the rest of configurations. The results regarding the comparisons are shown in the Table 5.8. In the conditions of the experiment, the configuration with identifier ID1 performs better than the configurations with identifiers ID3 - ID4 (with MOEA/D as an optimisation method) and ID6 (with MOEA/D-DE as an optimisation method, population of 150 individuals and 0.6 as an F parameter) but is not possible to conclude that the configuration with identifier ID1 performs better than any other. The best accumulated non-dominated solutions obtained were used to compute the accumulated Hypervolume, whose value was 2.4179 and it is shown in the Table 5.6. As it is expected, the value is higher than 2.4087, the maximum accumulated value achieved after the evolutionary process for the SMS-EMOA method.

Comparison	*p*-value	Conclusion
ID1 - ID4	**$3.929\cdot10^{-6} < 0.05$**	**The null hypothesis is rejected**
ID1 - ID3	**$1.267\cdot10^{-4} < 0.05$**	**The null hypothesis is rejected**
ID1 - ID6	**0.0085 < 0.05**	**The null hypothesis is rejected**
ID1 - ID5	0.3463 > 0.05	The null hypothesis is not rejected
ID1 - ID10	4.8365 > 0.05	The null hypothesis is not rejected
ID1 - ID8	5.3842 > 0.05	The null hypothesis is not rejected
ID1 - ID9	5.3842 > 0.05	The null hypothesis is not rejected
ID1 - ID7	5.3842 > 0.05	The null hypothesis is not rejected
ID1 - ID2	5.3842 > 0.05	The null hypothesis is not rejected

Table 5.8: *P*-values from the hypothesis tests.

The results of the analysis bring into light the better performance of the methods based on Indicators (SMS-EMOA) and Non-dominance (NSGA-II or GDE3) in comparison to the methods based on Decomposition (MOEA/D or MOEA/D-DE). However, the operator which creates new individuals does not appear to have a significant effect, since methods that use Simulated Binary Crossover (SMS-EMOA or NSGA-II) supplied a similar performance to a method that uses Differential Evolution (GDE3).

Id	Q	Cost [e.u.]	V_1 [h]	P_2 [h]	P_3 [h]	V_4 [h]	V_5 [h]	V_6 [h]	V_7 [h]
1	0.002898	861.38	27381	0	8760	0	34918	35040	35040
2	0.002812	877.12	17745	0	8760	0	34540	23997	34502
3	0.002500	993.50	31882	0	8735	32414	31364	27007	31499
4	0.002439	999.13	29494	0	8139	18708	24634	30543	26541
5	0.002396	1041.38	24766	0	8579	29319	24758	24663	32581
6	0.002370	1101.38	34380	0	8760	19098	11756	32930	35040
7	0.001640	1371.38	27808	8677	8760	0	26387	17704	28967
8	0.001504	1407.12	16434	8727	7582	0	18220	30929	35040
9	0.001495	1419.50	33877	8593	8671	0	34445	26119	33225
10	0.001410	1421.50	20519	8524	8754	0	25417	24732	34207
11	0.001316	1422.38	35040	8735	7927	0	23549	28082	21479
12	0.001281	1450.62	35040	7249	8257	0	25526	30376	35040
13	0.001246	1461.50	32755	8636	8220	0	22498	33016	30125
14	0.001214	1465.12	32667	8358	7756	0	30657	34271	29494
15	0.000973	1507.50	34928	7929	7908	11683	24182	34593	26275
16	0.000940	1523.38	30443	8130	8462	29999	34282	34286	34317
17	0.000876	1577.88	34470	8760	7246	27465	29267	35040	34730
18	0.000789	1595.88	33247	7286	8050	32526	10911	29935	34312
19	0.000749	1698.62	28281	8338	7367	19399	17608	32848	31220

Table 5.9: Non-dominated solutions.

The non-dominated solutions to the problem provided at the end of the evolutionary process from all executions, all configurations and all methods are shown in the Figure 5.20. In the Table 5.9, all optimum solutions belonging to the achieved non-dominated front are shown. The Unavailability (Q) is shown as a fraction, the Cost is shown in economic units and the rest of the variables represent, for the respective devices, the optimum times to allow scheduling the preventive maintenance tasks when the hour is used as a time unit.

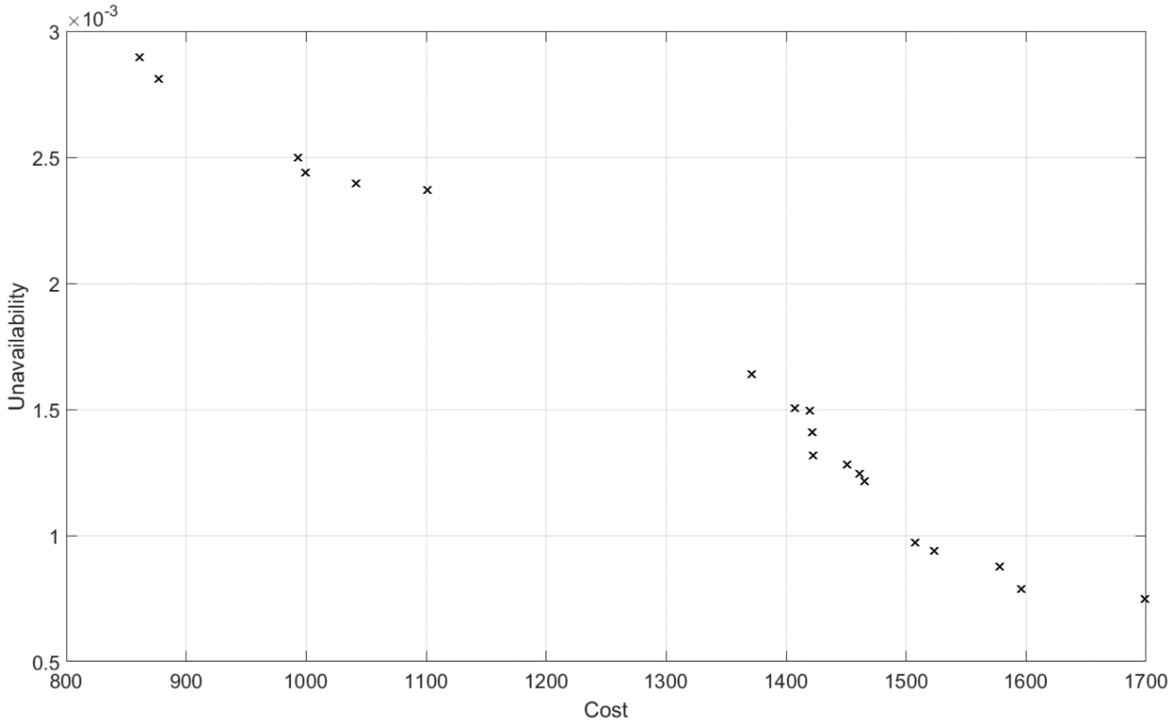


Figure 5.20: Accumulated non-dominated front.

The solution with the lowest Cost (ID1) (861.38 economic units) presents the biggest Unavailability (0.002898). These values are followed by periodic optimum times to start a maintenance task (hours) measured from the moment in which the system mission time starts (time to perform the preventive maintenance task is not included). For the solution ID1, the periodic optimum times to start a preventive maintenance task for the devices P2 and V4 are not supplied. This is because the design alternative does not include such devices. The opposite case shows the biggest Cost (ID19) (1,698.62 economic units) and the lowest Unavailability (0.000749). For the solution ID19, periodic optimum times to start a preventive maintenance task are supplied for all devices. This is because the design alternative includes the devices P2 and V4 as redundant devices. Other optimum solutions were found in these two solutions and can be seen in the Table 5.9. The decision makers should decide which is the preferable design when their individual requirements are considered. Depending on the application case, the company might have a cost threshold (e.g., due to financial budget constraints) and observing the attained non-dominated solutions, the solution/design with best non-availability corresponding to that cost could be chosen. Alternatively, the company might have a non-availability threshold (e.g., due to a legal norm) and observing the attained nondominated solutions, the solution/design with best cost corresponding to that non-availability could be chosen.

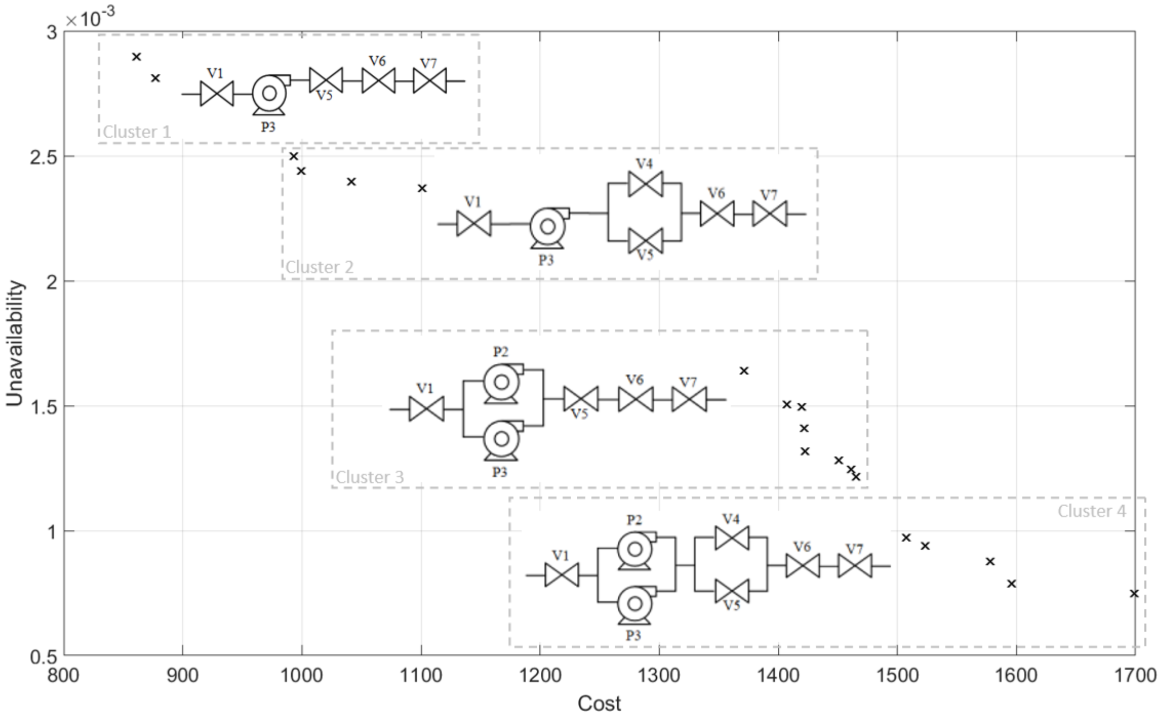


Figure 5.21: Clustered accumulated non-dominated front and design options.

Moreover, the solutions have been clustered in the Figure 5.21 according to their final design. Such solutions are shown in ascending order regarding the Cost from ID1 to ID19 and from the left to the right respectively. The solutions contained in the Cluster 1 (the solutions 1 to 2, see also the Table 5.9) are the solutions in which non-redundant devices have been included in the design. In this case, devices placed in series are exclusively contained in the system. These solutions present the lowest Cost and the biggest Unavailability. The solutions contained in the Cluster 2 (the solutions 3 to 6, see also the Table 5.9) are the solutions in which a redundant valve has been included in the design as a parallel device. These solutions present a bigger Cost and a lower Unavailability than the solutions contained in the Cluster 1. The solutions contained in the Cluster 3 (the solutions 7 to 14, see also the Table 5.9) are the solutions in which a redundant pump has been included in the design as a parallel device. These solutions present a higher Cost and a lower Unavailability than the solutions contained in the Clusters 1 and 2. Finally, the solutions contained in the Cluster 4 (the solutions 15 to 19, see also the Table 5.9) are the solutions in which both a redundant valve and a redundant pump have been included in the design as parallel devices. These solutions present the biggest Cost and the lowest Unavailability.

5.2.4. Discussion.

From the previous study, the configuration with the best behaviour based on the hypervolume indicator from the Friedman's test point of view was found (SMS-EMOA as an optimisation method, population of 150 individuals and 0.5 gene per chromosome as a mutation rate). Such a configuration is taken as a reference of the proposed methodology analysis in this subsection. Next, a discussion is opened regarding two interesting aspects: first, the effect of the sampling size when Discrete Event Simulation and Multi-objective Evolutionary Algorithms are coupled, and second, the quantification of the economic cost savings when the methodology is applied.

5.2.4.1. Discrete Event Simulation coupled to Multi-objective Evolutionary Algorithms: the effect of sampling size.

The proposed methodology employs a single discrete simulation per individual of the population in order to characterise the system behaviour and then, the objective functions are evaluated. Next, the effect of varying the sampling size is analysed with an equivalent number of fitness evaluations. Summarizing the procedure: the Functionability Profile of the system was built sample size times for each individual of the population and the objective functions (Availability and Operational Cost) were computed after as many times. The configuration of the case study with the best average rank from the Friedman's test point of view (SMS-EMOA, population size of 150 individuals, and a mutation rate of 0.5 gene per chromosome, see Table 5.7, column 9) was taken as a reference (and mentioned as 'direct SMS-EMOA'). In addition to this case (sample size equal to 1), sample sizes of 10, 100 and 1000 for each solution evaluated by the multi-objective evolutionary algorithm were tested. To foster equivalent purpose (attain the best non-dominated solutions) this procedure is equivalent to execute multiple simulations (as many as the chosen sample size) taking the minimal extreme value as a representative of the distribution achieved. However, in multi-objective optimization the non-dominated direction (non-dominated lower extreme value) of each solution is not known a priori and it depends on its relative position versus other nondominated solutions. Therefore, cases of taking as minimal extreme value either: 1) minimal unavailability, 2) minimal cost, or 3) a minimal equal weighted unavailability-cost (which is equivalent also to the Manhattan distance of both objectives) have been tested. The proposed methodology (single sample size, direct SMS-EMOA) is compared to those nine combinations (three minimal extreme values with three sample sizes each) and with a standard random search as an optimization baseline; all cases sharing an equivalent total stopping criterion of 10.000.000 evaluations of the fitness functions and being executed in 21 independent runs each. The set of configurations is shown in the Table 5.10.

Identifier	Criterion	Sampling size	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	Min. Unavailab.	10	2.2905	2.2827	2.3242	**2.2692**	0.0173	4.523
ID2	Min. Cost.	10	2.2869	2.2889	2.3095	2.2649	0.0116	4.619
ID3	Min. Unav.+ Min. Cost.	10	2.2823	2.2799	2.3119	2.2602	0.0131	5.857
ID4	Min. Unavailab.	100	2.2898	2.2878	2.3208	2.2662	0.0156	4.095
ID5	Min. Cost.	100	2.2841	2.2855	2.3122	2.2565	0.0141	5.285
ID6	Min. Unav.+ Min. Cost.	100	2.2804	2.2798	2.3069	2.2595	**0.0105**	6.095
ID7	Min. Unavailab.	1000	2.2769	2.2737	**2.3601**	2.2473	0.0232	7.190
ID8	Min. Cost.	1000	2.2658	2.2614	2.3240	2.2479	0.0182	8.714
ID9	Min. Unav.+ Min. Cost.	1000	2.2833	2.2848	2.3295	2.2467	0.0203	5.857
ID10	Random Search	1	2.2408	2.2386	2.2865	2.2149	0.0175	10.571
ID11	Direct SMS-EMOA	1	**2.2982**	**2.2978**	2.3223	2.2687	0.0142	**3.190**
p-Value								6.6712·10−11

Table 5.10: Id's, config., Hyperv. statistics and statistical test (multiple simulations).

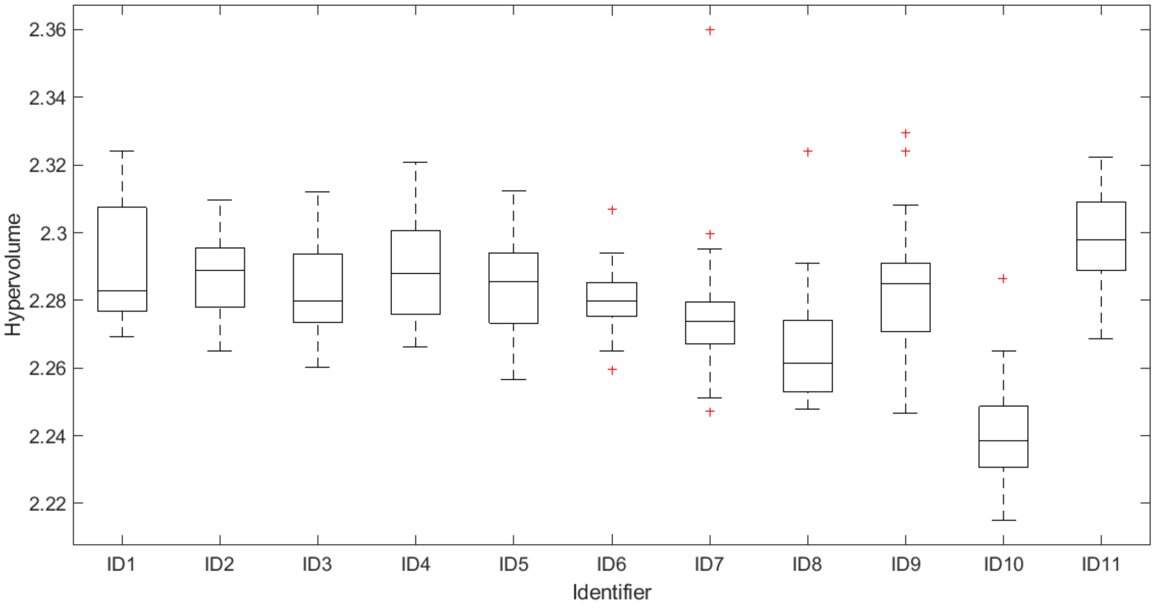


Figure 5.22: Hypervolume box plots (multiple simulations).

Box plots of the Hypervolume values distribution at the end of the process are shown in the Figure 5.22. As it is expected, the method based on random search (the configuration ID10) presents the worst performance. Moreover, the configuration with identifier ID11 (which uses the direct SMSEMOA) shows the highest Hypervolume median and average values. The configuration with identifier ID7 (which looks for minimum unavailability when 1000 evaluations of the objective functions per individual are computed) presents the highest Hypervolume maximum value and the configuration with identifier ID1 (which looks for minimum unavailability when 10 evaluations of the objective functions per individual are computed) supplies the highest Hypervolume minimum value. These, and other measures obtained, are shown in the Table 5.10 (columns 4 to 8).

In order to quantify if any one of the configurations performed better than any other, a statistical significance hypothesis test was conducted. The average ranks computed by following the Friedman's test are shown in the Table 5.10 (column 9). The configuration with identifier ID11, which uses the direct SMS-EMOA, produced the lowest average rank. After a similar number of evaluations, the direct SMS-EMOA achieved the first order regarding the hypothesis test. Moreover, the *p*-value computed ($6.6712 \cdot 10^{-11}$) implies that the null hypothesis (H_0) can be rejected (*p*-value < 0.05), so, in the studied conditions, there are configurations that perform better than others. In order to find the concrete pairwise comparisons that produce such differences, a post-hoc test was conducted. The Shaffer's test was used to compare the configuration with identifier ID11 with the rest of configurations. The results regarding the comparisons are shown in the Table 5.11.

Comparison	*p*-value	Conclusion
ID10 - ID11	**$3.049 \cdot 10^{-11} < 0.05$**	**The null hypothesis is rejected**
ID8 - ID11	**$3.052 \cdot 10^{-6} < 0.05$**	**The null hypothesis is rejected**
ID7 - ID11	**0.003 < 0.05**	**The null hypothesis is rejected**
ID6 - ID11	0.1679 > 0.05	The null hypothesis is not rejected
ID3 - ID11	0.3304 > 0.05	The null hypothesis is not rejected
ID9 - ID11	0.3304 > 0.05	The null hypothesis is not rejected
ID5 - ID11	1.2602 > 0.05	The null hypothesis is not rejected
ID2 - ID11	3.5815 > 0.05	The null hypothesis is not rejected
ID1 - ID11	4.0463 > 0.05	The null hypothesis is not rejected
ID4 - ID11	4.5206 > 0.05	The null hypothesis is not rejected

Table 5.11: *P*-values from the hypothesis tests (multiple simulations).

The hypothesis test shows that the direct SMS-EMOA achieved the best average rank from the Friedman's test point of view. Moreover, it shows significant differences regarding some of the configurations (ID7, ID8 and ID10) so a better behaviour is expected when the direct SMS-EMOA is used. Next, the configurations with the best average rank from each extreme studied (minimum unavailability, minimum cost and minimum weighted unavailability-cost) were selected to compare the results in front of using the direct SMS-EMOA. Their non-dominated solutions are shown in the Figure 5.23. It can be seen that the maximum Hypervolume is covered when the direct SMS-EMOA is used (with a value of 2.3832). Finally, the accumulated non-dominated front from the Figure 5.23 is shown in the Figure 5.24. It can be seen that there are not solutions that follow minimising the unavailability (marked as a □) on the left side of the Figure 5.24 as is expected. It is because the solutions on the left side present a best cost, which is contrary to obtaining more solutions with a smaller unavailability. Conversely, there are not solutions that follow minimising the cost (marked as a △) on the right side of the Figure 5.24 as is expected. It is because the solutions on the right side present best unavailability, which is contrary to obtaining more economical solutions. It can be seen how the solutions supplied by the direct SMSEMOA (marked as an ×) are spread along the non-dominated front.

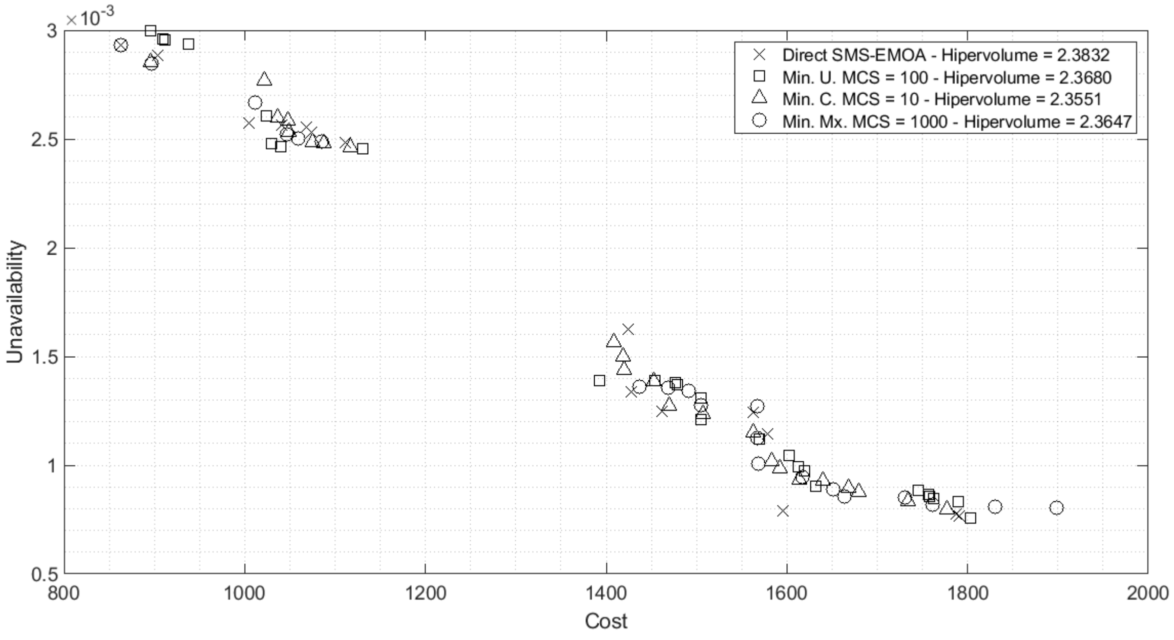


Figure 5.23: Accumulated non-dominated solutions (multiple simulations).

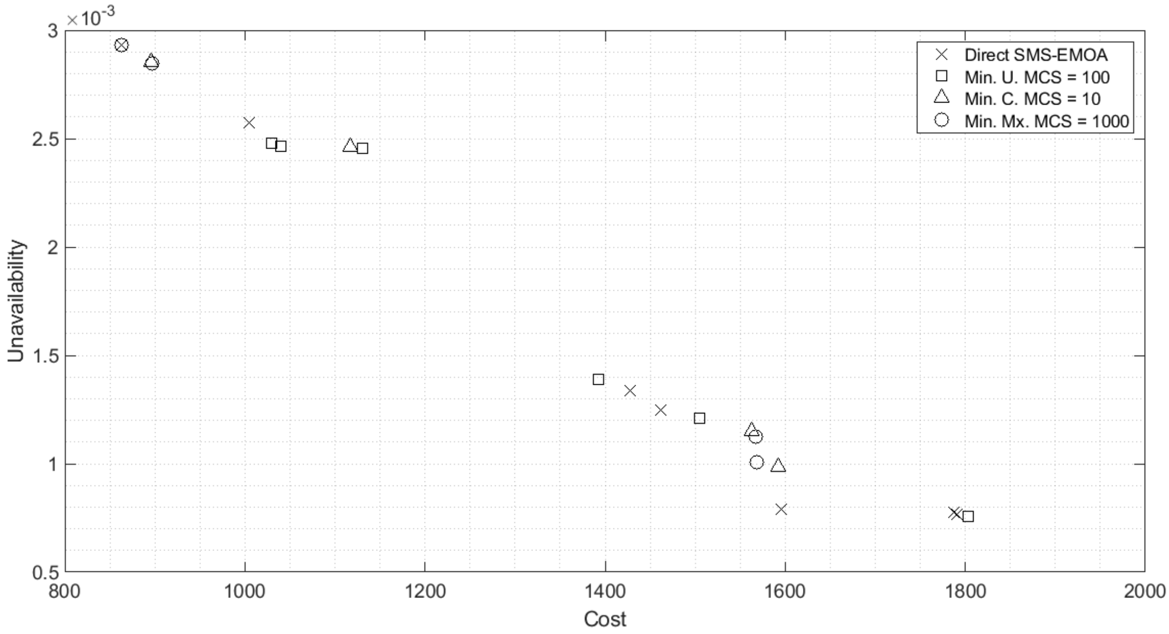


Figure 5.24: Accumulated non-dominated front (multiple simulations).

Summarising, the results of sampling size analysis enhance the benefits of the proposed methodology showing the positive synergy among Discrete Event Simulation and Multi-objective Evolutionary Algorithms, where only a single simulation per individual is enough in the fitness function evaluation to attain very competitive results.

A further analysis of the non-dominated solutions of the previous results based in their representative averages instead of their best values attained was conducted: For each non-dominated solution at the final generation (after 10.000.000 evaluations) of each of the 21 independent executions of the previous experiments, 10.000 discrete simulations were executed, and their objective functions were computed. Then, the unavailability and cost averages were used as representative values of the distribution of each solution. In this way, for each extreme non-dominated solution achieved previously, the centre of its distribution was located. These central solutions are the solutions that would be achieved by executing a

Monte Carlo simulation when considering the average as the representative value of the distribution. Box plots of the Hypervolume values distribution regarding each configuration are shown in the Figure 5.25. Statistical information regarding this experiment is shown in the Table 5.12. It can be seen that the configuration with identifier ID3 (which looks for minimum weighted unavailability-cost when 100 evaluations of the objective functions per individual are considered) presents both the best median and maximum Hypervolume values.

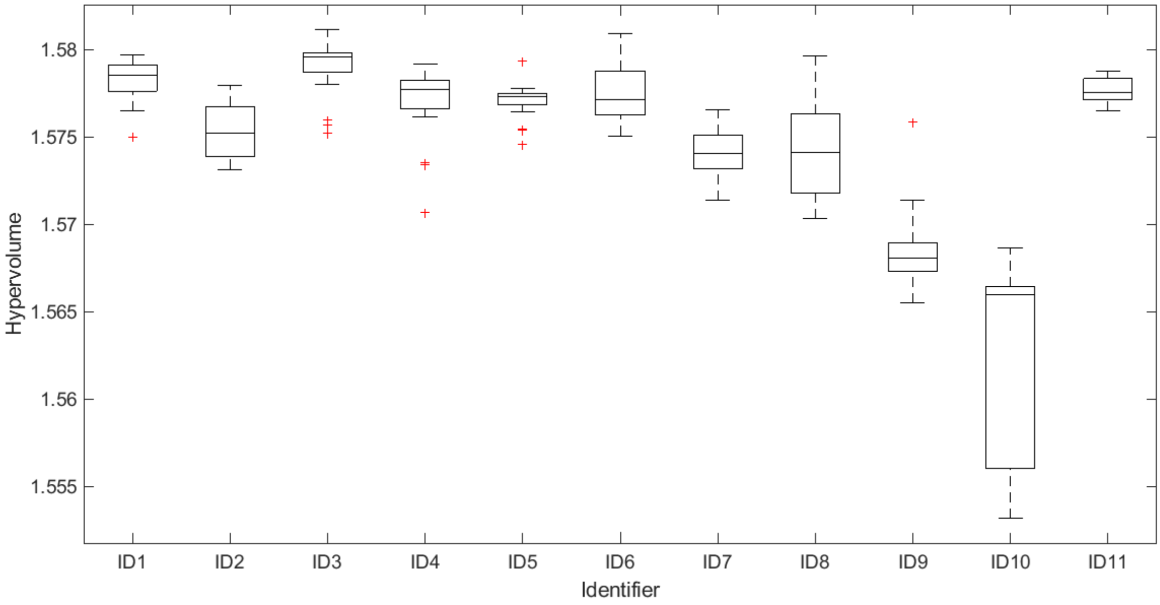


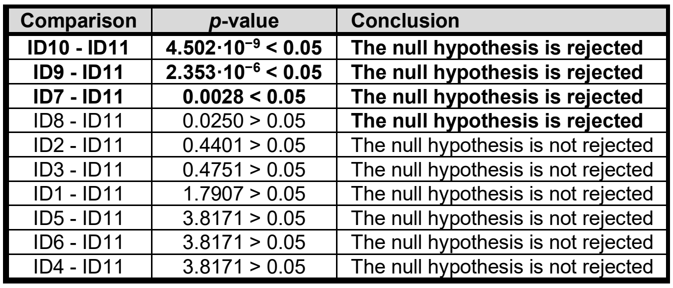
Figure 5.25: Hypervolume box plots (simulated centres).

Identifier	Criterion	Sampling size	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	Min. Unavailab.	10	1.5783	1.5785	1.5797	1.5750	0.0012	2.809
ID2	Min. Cost.	10	1.5754	1.5773	1.5780	1.5732	0.0015	6.619
ID3	Min. Unav.+ Min. Cost.	10	**1.5790**	**1.5796**	**1.5812**	1.5752	0.0015	**1.904**
ID4	Min. Unavailab.	100	1.5770	1.5777	1.5792	1.5707	0.0021	4.476
ID5	Min. Cost.	100	1.5771	1.5773	1.5793	1.5746	0.0009	4.809
ID6	Min. Unav.+ Min. Cost.	100	1.5774	1.5772	1.5809	1.5751	0.0016	4.523
ID7	Min. Unavailab.	1000	1.5741	1.5741	1.5766	1.5714	0.0015	8.285
ID8	Min. Cost.	1000	1.5743	1.4741	1.5796	1.5703	0.0027	7.666
ID9	Min. Unav.+ Min. Cost.	1000	1.5684	1.5681	1.5759	1.5655	0.0023	9.809
ID10	Random Search	1	1.5626	1.5660	1.5686	1.5532	0.0060	10.857
ID11	Direct SMS-EMOA	1	1.5777	1.5776	1.5788	**1.5765**	**0.0007**	4.238
p-Value								9.26·10−11

Table 5.12: Id's, config., Hyperv. statistics and statistical test (centres).

In order to establish if any configuration performed better than any other, a statistical significance hypothesis test was conducted. The average ranks computed by using the Friedman's test are shown in the Table 5.12. The configuration with identifier ID3 (which looks for minimum weighted unavailability-cost when 10 evaluations of the objective functions per individual are used) produced the lowest average rank, while the proposed methodology, the configuration with identifier ID11 -direct SMS-EMOA- was ranked third out of eleven configurations. Moreover, the *p*-value computed ($9.2629 \cdot 10^{-11}$) implies that the null hypothesis (H_0) can be rejected (*p*-value < 0.05), so, in the studied conditions, there are configurations that perform better than others. In order to find the concrete pairwise comparisons that produce such differences, a post-hoc test was carried out. The Shaffer's test was used to compare the proposed methodology ID11 -direct SMS-EMOA- with the rest of configurations. The results regarding the comparisons are shown in the Table 5.13. The configuration with identifier ID11 performs better than ID7, ID8, ID9 and ID10, while no other configuration outperforms the proposed methodology. Moreover, the Hypothesis test shows that non-significant differences were found among the direct SMS-EMOA and configuration ID3 with the best results from the Friedman's test point of view. However, the way to know a priori which would be the best values and their influence in the optimization outcome, either of the sampling size parameter or either of the minimum extreme value parameter is not determined. For example, if we focus on each value of sampling size, we could observe that in the case of sampling size 10 (ID1 to ID3), the best ordered case was the minimum equal-weighted unavailability-cost extreme (ID3), while in the case of sampling size 100 (ID4 to ID6), the best ordered case was the cost extreme (ID4), and finally in the case of sampling size 1000 (ID7 to ID9), the best ordered case was the unavailability extreme (ID8). Therefore, depending on the sampling sizes all the minimum extreme directions could have been the best options according to the attained results of the experiment. Hence, there are many parameters to explore in very expensive computational processes. On the contrary, the proposed methodology is parameter-less due to a single sampling size and therefore due to its implicit management of the non-dominated solutions by the selection operator of the evolutionary multi-objective algorithm.

In summary, the proposed methodology as it is shown in the insight results and discussion of the case study, is a computationally efficient and robust approach (non-parameter dependent regarding the number of samples or the minimal search direction) versus the use of Monte Carlo simulation-based approaches when facing the multi-objective optimization reliability problem handled.



Comparison	*p*-value	Conclusion
ID10 - ID11	**$4.502 \cdot 10^{-9} < 0.05$**	**The null hypothesis is rejected**
ID9 - ID11	**$2.353 \cdot 10^{-6} < 0.05$**	**The null hypothesis is rejected**
ID7 - ID11	**0.0028 < 0.05**	**The null hypothesis is rejected**
ID8 - ID11	0.0250 > 0.05	**The null hypothesis is rejected**
ID2 - ID11	0.4401 > 0.05	The null hypothesis is not rejected
ID3 - ID11	0.4751 > 0.05	The null hypothesis is not rejected
ID1 - ID11	1.7907 > 0.05	The null hypothesis is not rejected
ID5 - ID11	3.8171 > 0.05	The null hypothesis is not rejected
ID6 - ID11	3.8171 > 0.05	The null hypothesis is not rejected
ID4 - ID11	3.8171 > 0.05	The null hypothesis is not rejected

Table 5.13: *P*-values from Shaffer's test from simulated centres (centres).

5.2.4.2. Quantification of the operation cost saved.

In order to evaluate the cost savings attained by the proposed methodology in the case study by using the direct SMS-EMOA, a comparison with a standard random search has been tested. Therefore, each individual of the population (the design – devices involved in- and maintenance strategy) was randomly generated and the objective functions were evaluated after. The total number of solutions generated was equivalent to the stopping criterion of the direct SMS-EMOA (10,000,000) at each of the 21 independent executions. The configuration of the case study with best average rank from the Friedman's test point of view (SMS-EMOA, population size of 150 individuals, and mutation rate of 0.5 gene per chromosome, as it is seen in the Table 5.7, column 9) was taken as reference. In both compared cases (direct SMS-EMOA and random search), the 21 independent executions were ordered by their Hypervolume values, and the median case (11th ordered) was taken as reference, as is shown in the Figure 5.26.

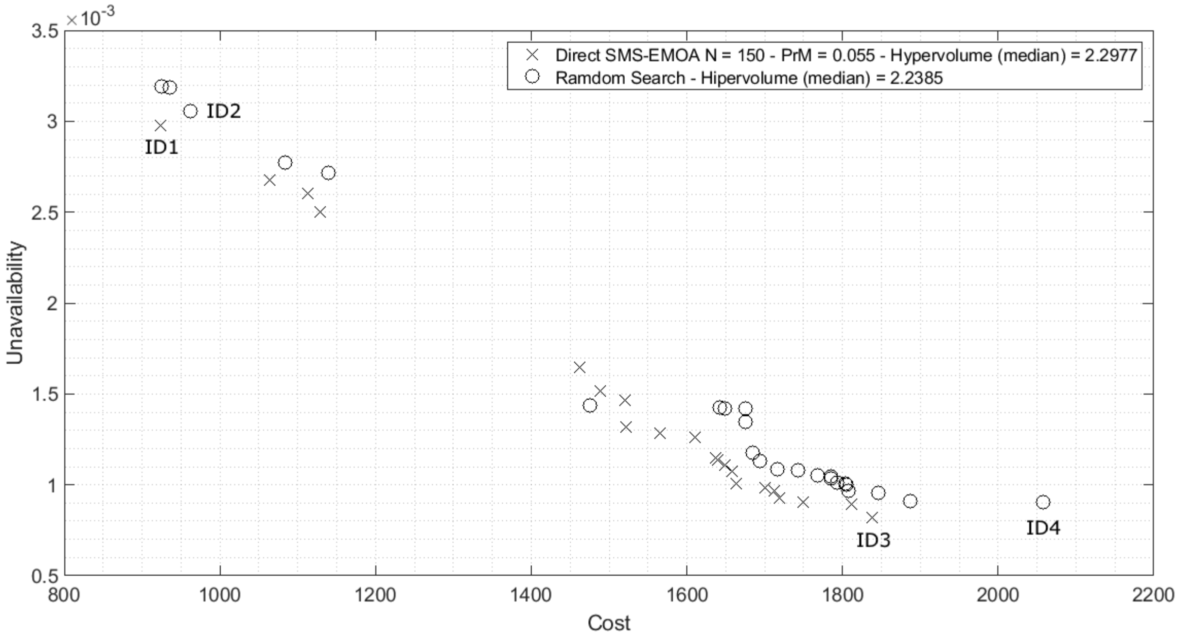


Figure 5.26: Non-dominated fronts -direct SMS-EMOA and random search- (median case out of 21 independent executions).

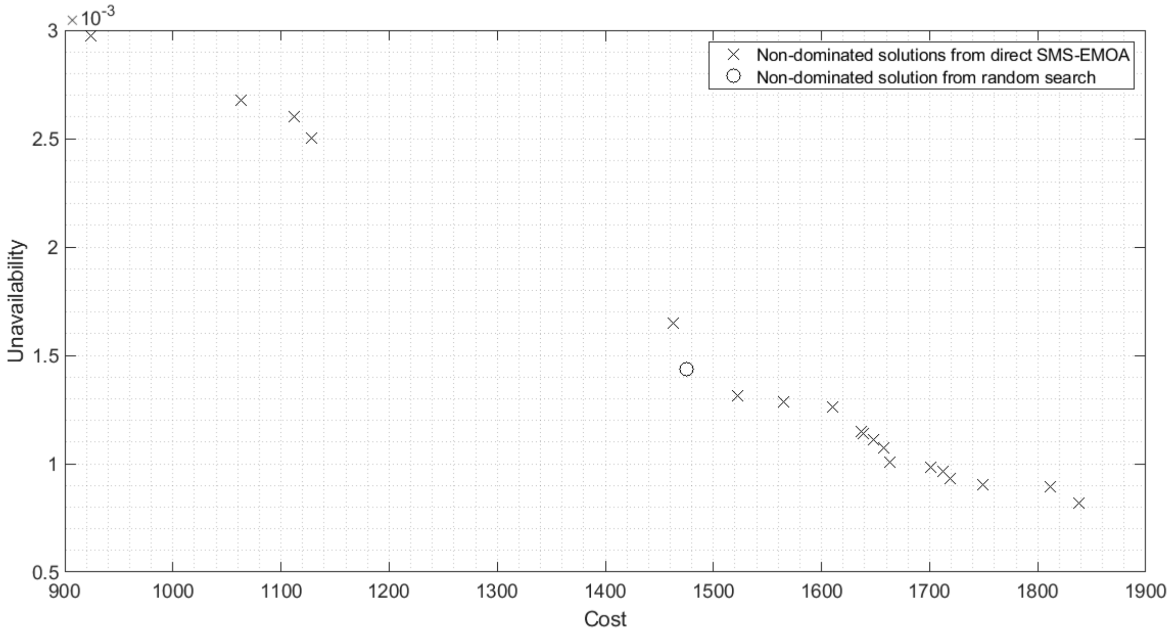


Figure 5.27: Global non-dominated front (median direct SMS-EMOA and median random search).

The Hypervolume covered by the methodology (2.2977) is better than the Hypervolume covered by the random search (2.2385); hence, the non-dominated front obtained by the proposed methodology is better than the non-dominated front obtained by the random search. From those sets, their joint non-dominated front is shown in the Figure 5.27, where all solutions except a single one (by chance) were attained by the proposed methodology.

In order to quantify what is the benefit of using the direct SMS-EMOA, characteristic solutions identified as ID1 and ID3 (taken from direct SMS-EMOA) and ID2 and ID4 (taken from random search) have been chosen to compare. These solutions are shown both in the Table 5.14 and in the Figure 5.26. Comparing the solutions with the best cost from the Table 5.14 (ID1 and ID2), it can be seen that the solution ID1 (achieved from the direct SMS-EMOA) is not only more economic but also more reliable than the solution ID2 (achieved from the random search). Comparing the economic cost of the solutions ID1 and ID2, the solution ID1 presents an improvement of a 4%. The better the unavailability, the bigger the impact of the methodology in terms of cost benefits. Comparing the solutions with the best unavailability from the Table 5.14 (ID3 and ID4), it can be seen that the solution ID3 (achieved from the direct SMS-EMOA) is not only more economic but also more reliable than the solution ID4 (achieved from the random search). The difference between the solutions ID3 and ID4 reaches an economic cost of a 10% lower. Then in the conditions of the experiment, using the direct SMS-EMOA produces a positive impact not only from the economic point of view but also from the availability point of view.

Id	Method	Q	Cost [ue]	V_1 [h]	P_2 [h]	P_3 [h]	V_4 [h]	V_5 [h]	V_6 [h]	V_7 [h]
1	SMS-EMOA	0.002974	923.75	25434	0	8691	0	21781	33457	31259
2	Random search	0.003055	962.12	15810	0	8621	0	16584	23113	24667
3	SMS-EMOA	0.000819	1838.37	33858	7700	8678	16795	24190	33248	28461
4	Random Search	0.000903	2057.75	34514	4711	4946	29724	28028	25410	32408

Table 5.14: Extreme solutions taken from the Figure 5.26.

5.2.5. Testing the methodology in more complex applications.

Two applications are faced in order to demonstrate the viability of applying the methodology. The Application Case A consists of an extension of the main case study by adding a second branch. The Application Case B consists of a system whose structure is more complex, and the number and kind of devices is bigger.

5.2.5.1. Application Case A: The case study with double branch.

This application consists of an extension of the case study, which is a basic model of a containment spray injection system. In this case, a second branch is included as it is shown in the Figure 5.28. A similar structure was analysed by Greiner et al [5]. As in the case study, it is necessary to establish the optimum period to perform a preventive maintenance activity for the system devices and it is necessary to decide whether to include redundant devices such as the pump P2, the pump P9, the Valve V4 and the Valve V11 by evaluating design alternatives. Hence, the number of components of the Application Case A can vary from 10 to 14 automatically as an outcome from the evolutionary algorithm search. The chromosomes are codified as follows:

$$[D_1\ D_2\ D_3\ D_4\ M_1\ M_2\ M_3\ M_4\ M_5\ M_6\ M_7\ M_8\ M_9\ M_{10}\ M_{11}\ M_{12}\ M_{13}\ M_{14}]$$

The presence of redundant devices P2, V4, P9 and V11 is defined by the decision variables D_1, D_2, D_3 and D_4 respectively, and the optimum time to start a preventive maintenance activity in relation to each device is represented by the decision variables M_1 to M_{14}. They must be transformed to evaluate the objective functions as claimed in the Section 4.5.2.1 of the Chapter IV. This application was executed attending to the configuration that presented the best behaviour from the case study, as it is shown in the Table 5.15.

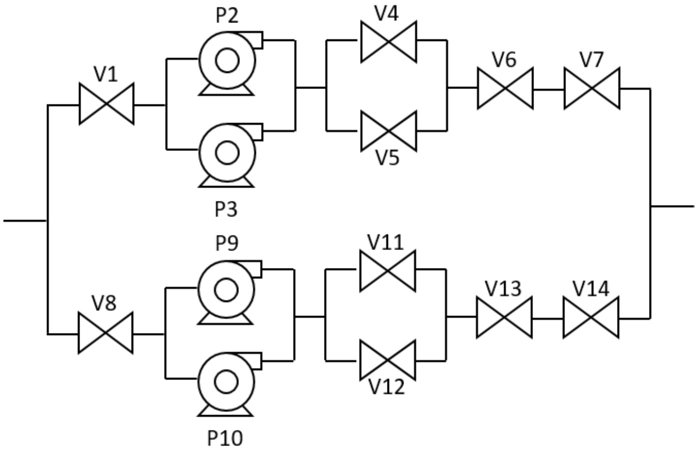


Figure 5.28: Application Case A: Double branch CSIS.

Description	Setting
Method	SMS-EMOA
Population	150
Mutation rate	0.5
Mutation distribution index	20
Crossover probability	0.9
Crossover distribution index	20
Number of evaluations	10.000.000
Number of executions	21

Table 5.15: Parameters configuration for the Application Cases A and B.

The scale factors in relation to the value of the objective functions used with the purpose of achieving an equally dispersed non-dominated front with the unit as a maximum value of each objective were as follows:

- The scale factor employed to compute the Cost was 4,500 economic units.
- The scale factor employed to compute the system Unavailability was 0.00004.

Firstly, the experiment was conducted based on the proposed methodology. Additionally, to tune the effect of automatic devices selection, a second problem was run where the structural design was based on the mandatory selection of all devices. Box plots of the Hypervolume values distribution at the end of the process are shown in the Figure 5.29. The configuration with identifier ID1, which uses the proposed methodology, presents the best median Hypervolume value. Statistical information regarding the Hypervolume reached when 21 independent executions were carried out is shown in the Table 5.16. The configuration with identifier ID1 presents the best statistics. Moreover, the configuration with identifier ID1 presented the best average rank from the Friedman’s test point of view, and the *p*-value achieved of $4.592{\cdot}10^{-6}$ establishes that the configuration ID1 performs better than the configuration ID2. The nondominated solutions achieved by the proposed methodology (marked as a ×) are shown in the Figure 5.30. The accumulated Hypervolume computed in this case out of 21 independent executions reached a value of 3.1146. Moreover, the non-dominated solutions are detailed in the Table 5.17. It can be seen that the devices P2, V4, P9 and V11 are not included in the design. The non-dominated solutions achieved based on the mandatory selection (marked as an O) are shown in the Figure 5.30. The accumulated Hypervolume computed in this case out of 21 independent executions reached a value of 3.1004. It can be seen that the hypervolume covered by the non-dominated solutions achieved by the proposed methodology is bigger than the hypervolume covered by the non-dominated solutions achieved by the mandatory selection of all devices; also, the former non-dominated solutions, which are identified as ID1, ID2 and ID3, dominate the latter non-dominated solutions. The proposed methodology was able to find optimum non-dominated solutions for the system.

Finally, the Figure 5.31 shows the non-dominated solutions achieved both for the case study (marked as a ×) and for the Application Case A (marked as a △). It can be seen that both fronts are complementary. The set of solutions of the non-dominated front achieved by the case study present a lower cost and a bigger unavailability than the set of solutions of the non-dominated front achieved by the Application Case A. The Application Case A is more reliable (lower unavailability) and more expensive (higher cost) since its solutions are composed by two parallel branches and with more components than the case study. Nevertheless, this is the reason why a bigger economic investment is needed to maintain the system. The decision makers should determine whether the benefit of applying better unavailability designs supports the increase in economic investment.

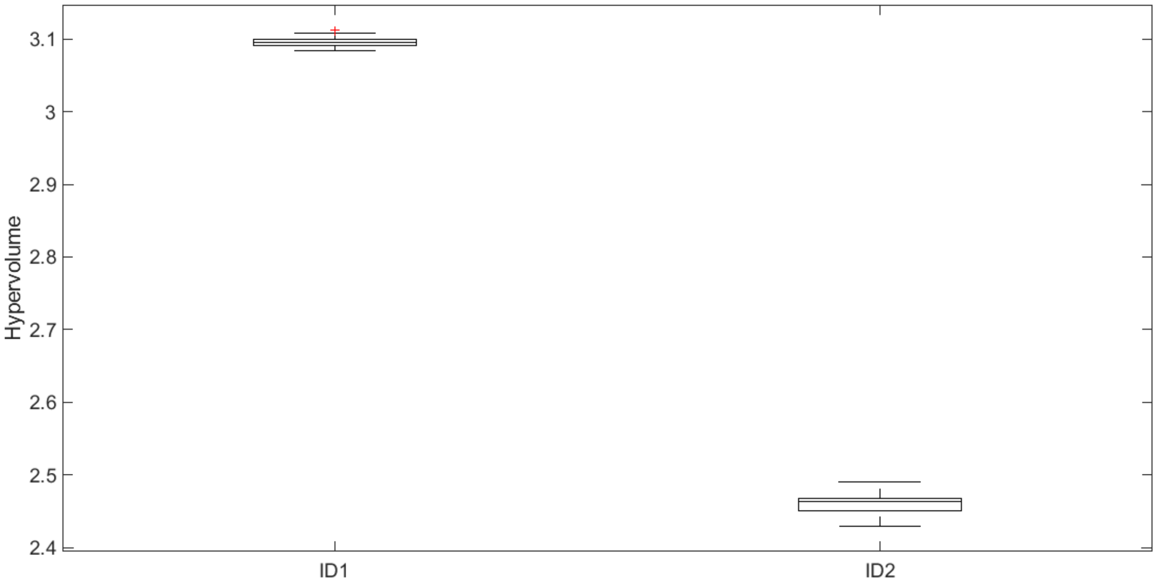


Figure 5.29: Box plots of Hypervolume (App. Case A).

Identifier	Criterion	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	Proposed methodology	**3.0963**	**3.0958**	**3.1130**	**3.0834**	**0.0082**	**1.000**
ID2	Mandatory selection	2.4600	2.4628	2.4904	2.4286	0.0146	2.000
p-Value							$4.59 \cdot 10^{-6}$

Table 5.16: Id’s, config., Hyperv. statistics and statistical test (App. Case A).

Id	Q	Cost [ue]	V_1 [h]	P_2 [h]	P_3 [h]	V_4 [h]	V_5 [h]	V_6 [h]	V_7 [h]	V_8 [h]	P_9 [h]	P_{10} [h]	V_{11} [h]	V_{12} [h]	V_{13} [h]	V_{14} [h]
1	0.000012842465	1990.75	11231	0	8105	0	19999	27523	21384	17516	0	8430	0	21274	21836	30564
2	0.000001426940	1994.62	13651	0	8745	0	11242	21832	26738	23110	0	8754	0	24284	23167	31445
3	0.000000000000	2032.00	17883	0	8352	0	32021	17390	20798	29361	0	8467	0	17758	19825	22320

Table 5.17: Non-dominated solutions (App. Case A).

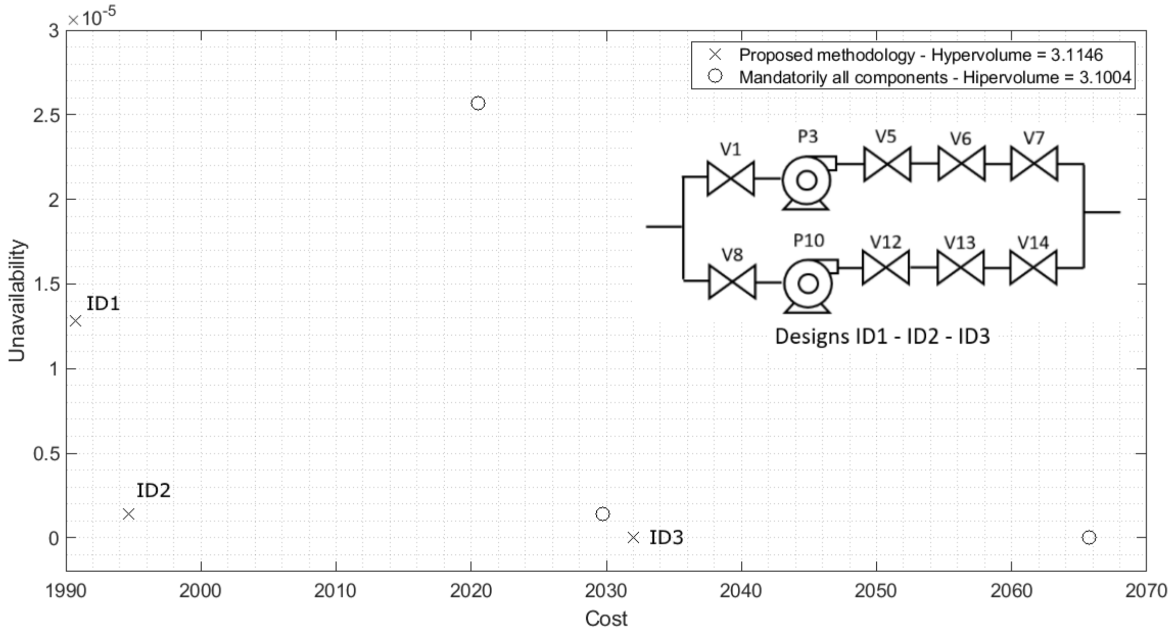


Figure 5.30: Accumulated non-dominated solutions and designs (App. Case A).

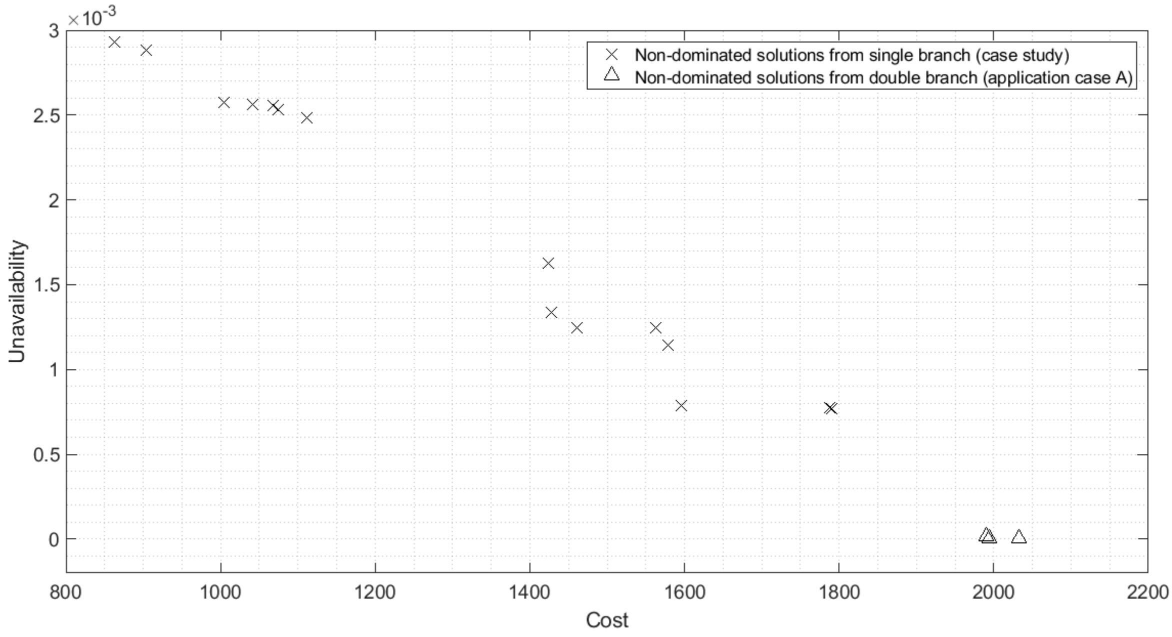


Figure 5.31: Accumulated non-dominated solutions from both the case study and the App. Case A.

5.2.5.2. Application Case B: An extended model for the Containment Spray System of a Nuclear Power Plant.

The Application Case B is based on an industrial case presented by Galván et al. [8] It consists of a Pressured Water Reactor (PWR) Containment Spray System, which is designed to provide particular and different functions inside the containment of a PWR, such as the borated water injection function. In the case study, a simplified model of this system was studied. In this case, an extended model is studied as it is shown in the Figure 5.32.

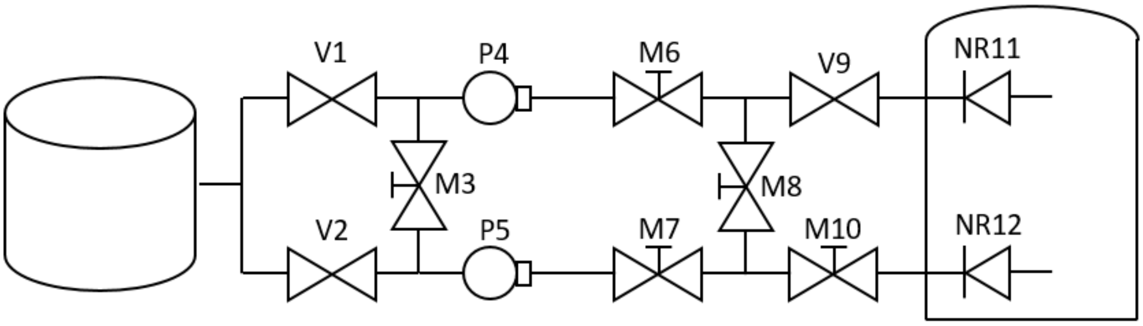


Figure 5.32: Application Case B: Base line.

The system consists of two separated trains, each one formed by centrifugal pumps and valves to control the flow of borated water from the Refuelling Water Storage Tank. The main devices are the single Valves (V1, V2 and V9), the Motor Driven Pumps (P4 and P5), the Motor Operated Valves (M3, M6, M7, M8 and M10) and One Way Valves (NR11 and NR12). The aim is the simultaneous optimisation of the system structural design (with automatic selection of devices) and its maintenance strategy with some considerations:

- Each position may locate a maximum of three redundant devices in parallel, so the maximum number of devices is thirty-six as is shown in the Figure 5.33,
- The devices V1, P4, V9, M10, NR11 and NR12 are mandatory as it is shown in the Figure 5.34,
- When the device M8 is not included in the design, the line is considered a tube,

- When the device P5 is not included in the design, the device M7 cannot be included,
- The device M3 might be included in the design when the device V2 or/and the device P5 is/are included,
- As in the Application Case A, to tune the effect of automatic devices selection, a second problem was run where the structural design was based on the mandatory selection of a minimum of a device per position.

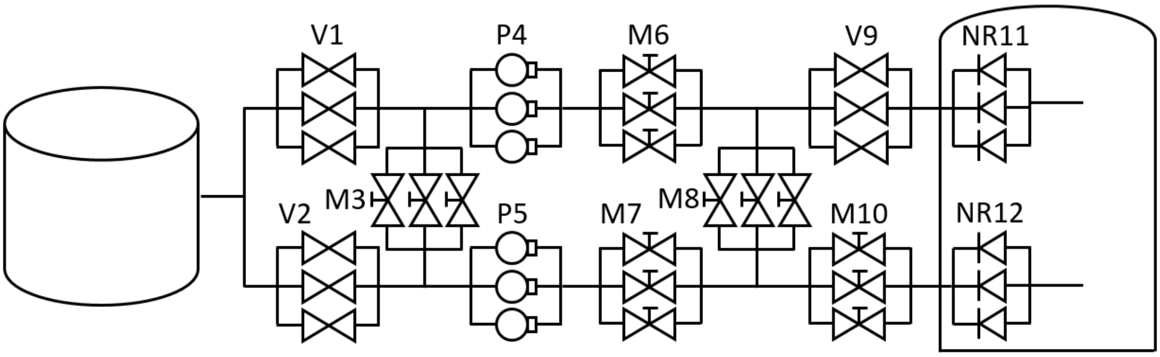


Figure 5.33: Application Case B: The most complex possible design.

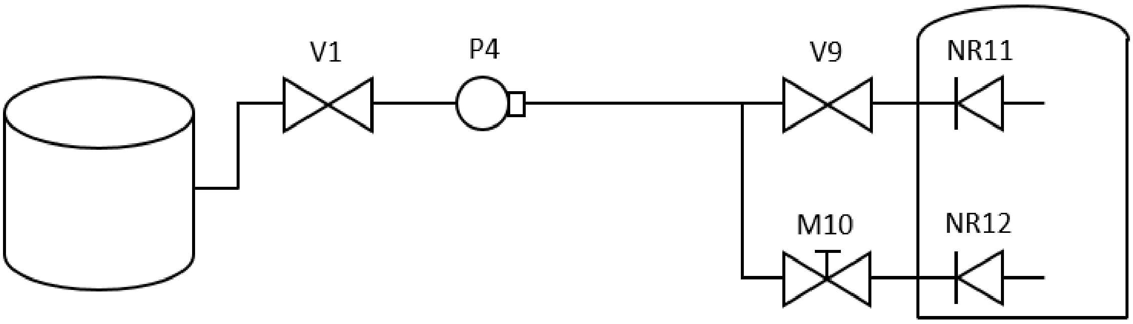


Figure 5.34: Application Case B: The simplest possible design.

As in the case study, the chromosome codification includes the period of time to conduct a preventive maintenance task for the system’s devices. Moreover, it is necessary to decide the inclusion of redundant devices by evaluating design alternatives; the number of devices of the Application Case B may vary from 6 to 36 automatically as an outcome from the evolutionary algorithm search.

In this application, two types of chromosome codifications are considered and compared:

- Long Chromosome codification: Formed by 66 decision variables, which are 30 for the design and 36 for the maintenance strategy. Six components are mandatory so 30 design decision variables are necessary to decide whether the rest of the devices are or not included in the design. The system can contain a maximum of 36 devices so this is the number of the decision variables for the preventive maintenance strategy,
- Short Chromosome codification: Formed by 48 decision variables, which are 12 for the design and 36 for the maintenance strategy. In this case, the 12 design decision variables are scaled to entire values so they may take a maximum value of three (the required transformation was explained in the Section 4.5.2.1 from the Chapter IV, Equation 4.6). As in the previous case, the system may contain a maximum of 36 devices so this is the number of the decision variables for the preventive maintenance strategy.

In summary, in this section, four problems solving Application Case B were executed: long chromosome and the proposed methodology, short chromosome and the proposed methodology, long chromosome and mandatory selection of a minimum of a device per position, and short chromosome and mandatory selection of a minimum of a device per position.

The data used are shown in the Table 5.18. As in the Application Case A, this Application Case B was executed attending to the multi-objective evolutionary optimisation configuration which presented the best performance in the case study, which is shown in the Table 5.15. The scale factors in relation to the value of the objective functions used with the purpose of achieving an equally dispersed non-dominated front with the unit as a maximum value of each objective were as follows:

- The scale factor used to compute the Cost was 6,000 economic units,
- The scale factor used to compute the system Unavailability was 0.00083.

Parameter	Value	Source
Life cycle or mission time	.438,000 hours	-
(V) Corrective Maintenance Cost	15 units/hour	Galván et al. [8]
(P) Corrective Maintenance Cost	15 units/hour	Galván et al. [8]
(M) Corrective Maintenance Cost	20 units/hour	Galván et al. [8]
(NR) Corrective Maintenance Cost	15 units/hour	Galván et al. [8]
(V) Preventive Maintenance Cost	15 units/hour	Galván et al. [8]
(P) Preventive Maintenance Cost	15 units/hour	Galván et al. [8]
(M) Preventive Maintenance Cost	20 units/hour	Galván et al. [8]
(NR) Preventive Maintenance Cost	15 units/hour	Galván et al. [8]
(V) TF_{min}	1 hour	MRI*
(V) TF_{max}	70.800 hours	MRI*
(V) $TF\ \lambda$	5,83x10^{-6} failures/hour	Galván et al. [8]
(V) TR_{min}	1 hour	MRI*
(V) TR_{max}	5 hours	$\mu + 3\sigma$
(V) $TR\mu$	3 hours	Galván et al. [8]
(V) $TR\sigma$	0.67 hours	$(\mu - TR_{min})/3$
(V) TM_{min}	8,760 hours	MRI*
(V) TM_{max}	35,040 hours	MRI*
(V) TCM_{min}	1 hours	MRI*
(V) TCM_{max}	1 hours	$\mu + 3\sigma$
(V) $TCM\mu$	1 hours	Galván et al. [8]
(V) $TCM\sigma$	0 hours	$(\mu - TR_{min})/3$
(P) TF_{min}	1 hour	MRI*
(P) TF_{max}	70.800 hours	MRI*
(P) $TF\ \lambda$	3,89x10^{-6} failures/hour	Galván et al. [8]
(P) TR_{min}	1 hour	MRI*
(P) TR_{max}	47 hours	$\mu + 3\sigma$
(P) $TR\mu$	24 hours	Galván et al. [8]
(P) $TR\sigma$	7.67 hours	$(\mu - TR_{min})/3$
(P) TM_{min}	2,920 hours	MRI*
(P) TM_{max}	8,760 hours	MRI*
(P) TCM_{min}	1 hours	MRI*
(P) TCM_{max}	1 hours	$\mu + 3\sigma$
(P) $TCM\mu$	4 hours	Galván et al. [8]
(P) $TCM\sigma$	1 hours	$(\mu - TR_{min})/3$
(M) TF_{min}	1 hour	MRI*
(M) TF_{max}	70.800 hours	MRI*
(M) $TF\ \lambda$	5,9x10^{-6} failures/hour	Galván et al. [8]
(M) TR_{min}	1 hour	MRI*
(M) TR_{max}	5 hours	$\mu + 3\sigma$
(M) $TR\mu$	3 hours	Galván et al. [8]
(M) $TR\sigma$	0.67 hours	$(\mu - TR_{min})/3$
(M) TM_{min}	8,760 hours	MRI*
(M) TM_{max}	35,040 hours	MRI*
(M) TCM_{min}	1 hours	MRI*
(M) TCM_{max}	1 hours	$\mu + 3\sigma$
(M) $TCM\mu$	1 hours	Galván et al. [8]
(M) $TCM\sigma$	0 hours	$(\mu - TR_{min})/3$
(NR) TF_{min}	1 hour	MRI*
(NR) TF_{max}	70.800 hours	MRI*
(NR) $TF\ \lambda$	5,9x10^{-6} failures/hour	Galván et al. [8]
(NR) TR_{min}	1 hour	MRI*
(NR) TR_{max}	5 hours	$\mu + 3\sigma$
(NR) $TR\mu$	3 hours	Galván et al. [8]
(NR) $TR\sigma$	0.67 hours	$(\mu - TR_{min})/3$
(NR) TM_{min}	8,760 hours	MRI*
(NR) TM_{max}	35,040 hours	MRI*
(NR) TCM_{min}	1 hours	MRI*
(NR) TCM_{max}	1 hours	$\mu + 3\sigma$
(NR) $TCM\mu$	1 hours	Galván et al. [8]
(NR) $TCM\sigma$	0 hours	$(\mu - TR_{min})/3$
*MRI = Machinery Reliability Institute		

Table 5.18: Data set (App. Case B).

Box plots of the Hypervolume values distribution at the end of the process are shown in the Figure 5.35. The identifiers (ID) are referred in the Table 5.19. It can be seen that the configuration with identifier ID3, which employs the short chromosome and the proposed methodology, presents the best median Hypervolume value. Statistical information regarding the Hypervolume reached when 21 independent executions were carried out is shown in the Table 5.19. The configuration with identifier ID3 presents the best Hypervolume average, median, minimum and standard deviation values, whereas the configuration with identifier ID1, which uses the long chromosome and the proposed methodology, presents the best Hypervolume maximum value. Moreover, the configuration with identifier ID3 presents the best average rank from the Friedman's test point of view, and the *p*-value achieved of $7.666 \cdot 10^{-11}$ establishes that the configuration ID3 performs better than some other.

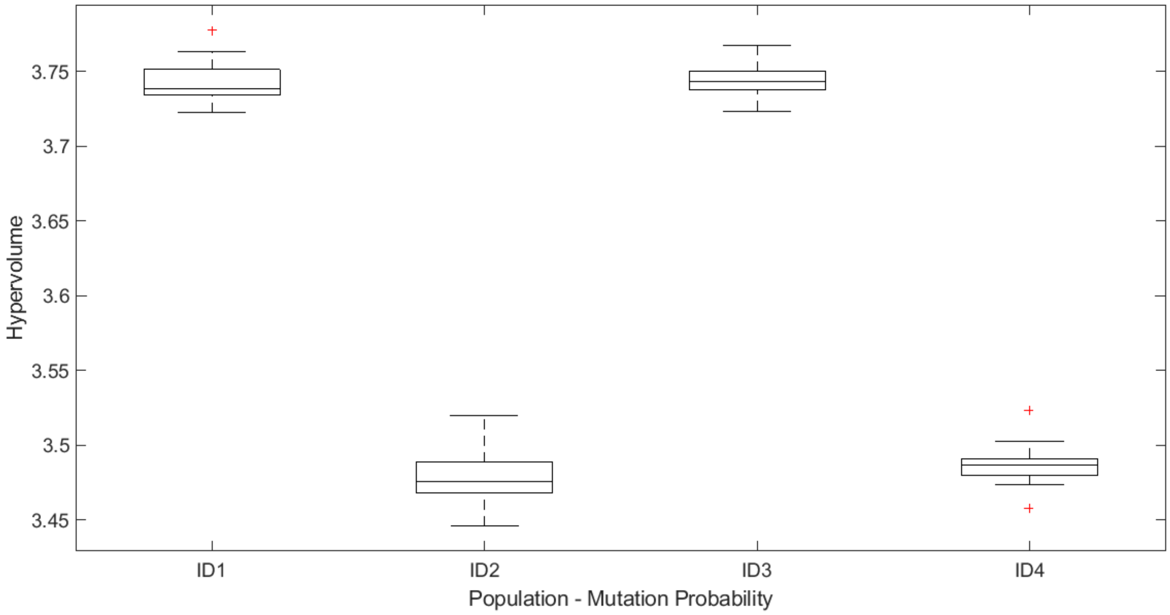


Figure 5.35: Box plots of Hypervolume (App. Case B).

Identifier	Criterion	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	Long chromosome - Proposed methodology	3.7434	3.7380	**3.7779**	3.7224	0.0135	1.5714
ID2	Long chromosome - Mandatory selection	3.4778	3.4757	3.5197	3.4457	0.0181	3.6190
ID3	Short chromosome - Proposed methodology	**3.7438**	**3.7434**	3.7676	**3.7233**	**0.0108**	**1.4285**
ID4	Short chromosome - Mandatory selection	3.4859	3.4867	3.5230	3.4573	0.0145	3.3809
p-Value							$7.66 \cdot 10^{-11}$

Table 5.19: Id's, config., Hyperv. statistics and statistical test (App. Case B).

Finally, the Shaffer's test was used to compare the configuration with identifier ID3 and the rest of configurations. Statistical significant difference was found regarding the configurations with identifiers ID2 and ID4 (both with mandatory selection of devices) as it is shown in the Table 5.20. Therefore, in the studied conditions, the best performance is achieved when the proposed methodology is employed. Furthermore, the best order from the Friedman's test is achieved when only one decision variable per position for the system design is used (the short chromosome).

Comparison	*p*-value	Conclusion
ID2 - ID3	**$2.304 \cdot 10^{-7} < 0.05$**	**The null hypothesis is rejected**
ID3 - ID4	**$2.868 \cdot 10^{-6} < 0.05$**	**The null hypothesis is rejected**
ID1 - ID3	$1.1001 > 0.05$	The null hypothesis is not rejected

Table 5.20: *P*-values from Shaffer's test (App. Case B).

The non-dominated solutions achieved when the long chromosome and the proposed methodology are used (marked as ×) are shown in the Figure 5.36. The accumulated Hypervolume computed out of 21 independent executions is 3.7809. The non-dominated solutions achieved when the long chromosome and the mandatory selection of a minimum of a device per position, are used (marked as O) cover an accumulated Hypervolume computed out of 21 independent executions of 3.5196. The non-dominated solutions achieved when the short chromosome and the proposed methodology are used (marked as □) cover an accumulated Hypervolume computed out of 21 independent executions of 3.7702. Finally, the non-dominated solutions achieved when the short chromosome and the mandatory selection of a minimum of a device per position are used (marked as △) cover an accumulated Hypervolume computed out of 21 independent executions of 3.5230. In the Figure 5.37, the non-dominated solutions, which belong to the non-dominated front from all configurations, were extracted. The detail of the solutions is shown in the Table 5.21. Solutions L1, L2 and L3 are solutions supplied when the long chromosome and the proposed methodology are used. Solutions S1, S2 and S3 are solutions supplied when the short chromosome and the proposed methodology are used. Finally, the SM1 solution is a solution achieved when using the short chromosome and the mandatory selection of devices. Each solution presents its cost, unavailability and periodic times to start a preventive maintenance activity regarding each device included in the design, as it is shown in the Table 5.21. The design alternatives are shown in the Figures 5.38 to 5.43. It can be seen that the solution L1 (the less expensive design), whose design is shown in the Figure 5.38, belongs to the simplest design, as it is shown in the Figure 5.34. On the contrary, it can be seen that the solution SM1, whose design is shown in the Figure 5.43, is the more reliable solution with an unavailability equal to 0.0. This is because the system was kept in the operating state all along the mission time for the discrete simulation that describes its behaviour. All the four tested methods were able to achieve best solutions with same unavailability value, as it is shown in the bottom right part of the Figure 5.36, although with slight differences in the cost attained.

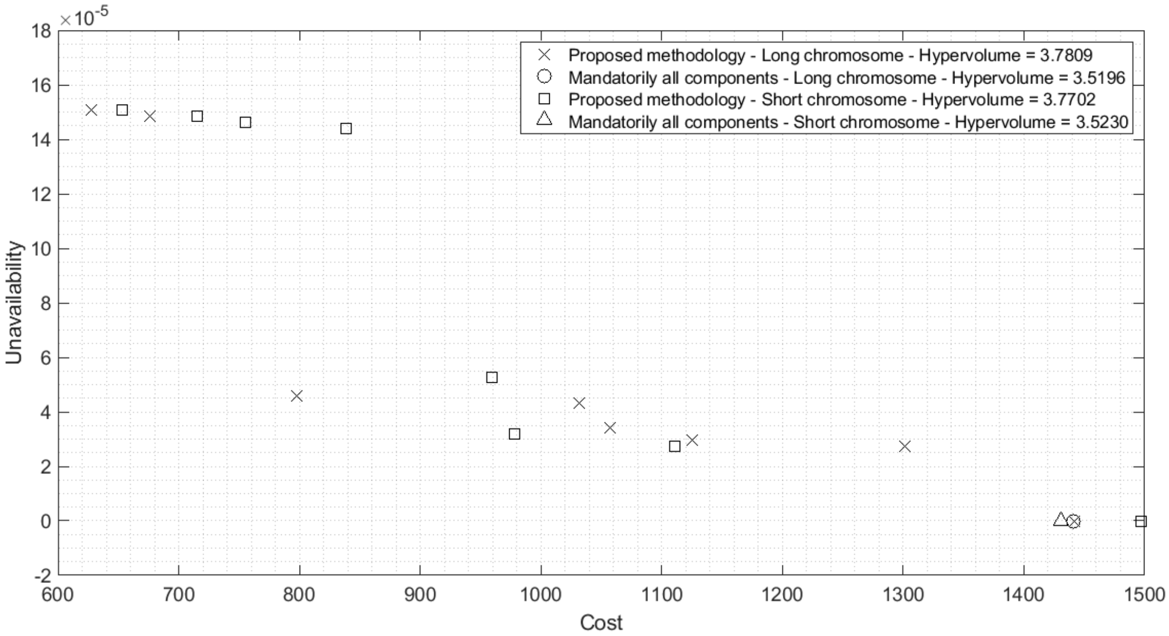


Figure 5.36: Accumulated non-dominated solutions (App. Case B).

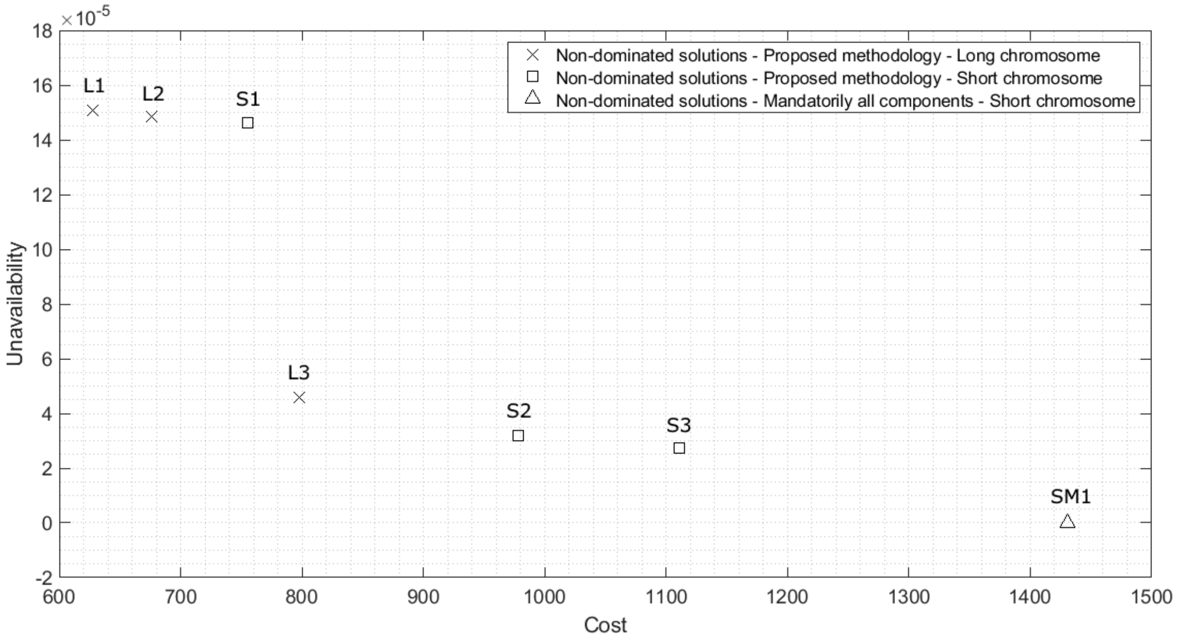


Figure 5.37: Global non-dominated front (App. Case B).

Id	Q	Cost [ue]	V_1 [h]	V_2 [h]	M_3 [h]	P_4 [h]	P_5 [h]	M_6 [h]	M_7 [h]	M_8 [h]	V_9 [h]	M_{10} [h]	NR_{11} [h]	NR_{12} [h]
L1	0.000150684	628.00	34812	-	-	8603	-	-	-	-	35040	10643	27565	17670
L2	0.000148401	676.00	35040	-	-	8751	-	-	-	-	32096	30768	21823-28100	24906
S1	0.000146118	755.00	33167	-	-	8437	-	-	-	10149	33617	18317	35040	9822
L3	0.000045662	798.00	33668	-	-	8590	8356	-	-	-	32598	24984	11763	24185
S2	0.000031963	978.00	34390	-	-	8452-7787	-	-	-	-	32201	29526-10617	17637	30224-18493
S3	0.000027397	1,111.00	34820	-	-	8127	7921	-	-	-	31937	33602	23771	23035
SM1	0.000000000	1,431.00	35038	33698	24164	8214	8750	20442	20313-20820	29945-21985	31666	23579	15473	28511

Table 5.21: Non-dominated solutions (App. Case B, id’s as in the Figure 5.37).

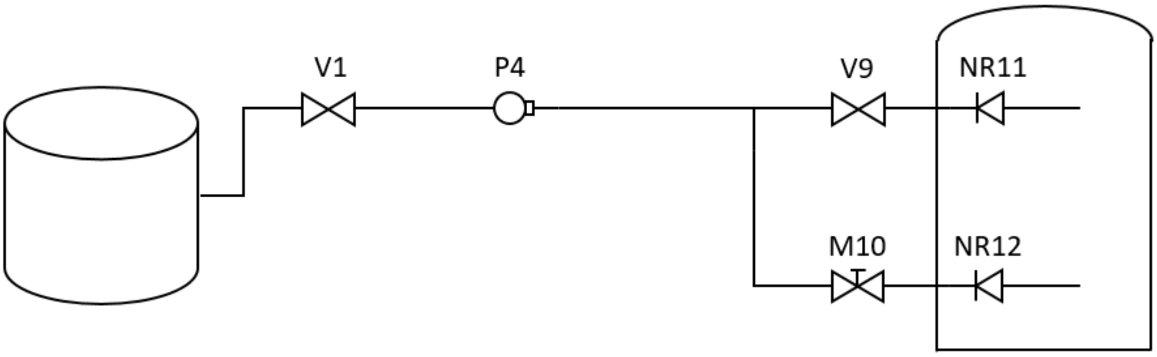


Figure 5.38: L1 solution (Cost = 628.00 - Q = 1.5068·10^{-4}).

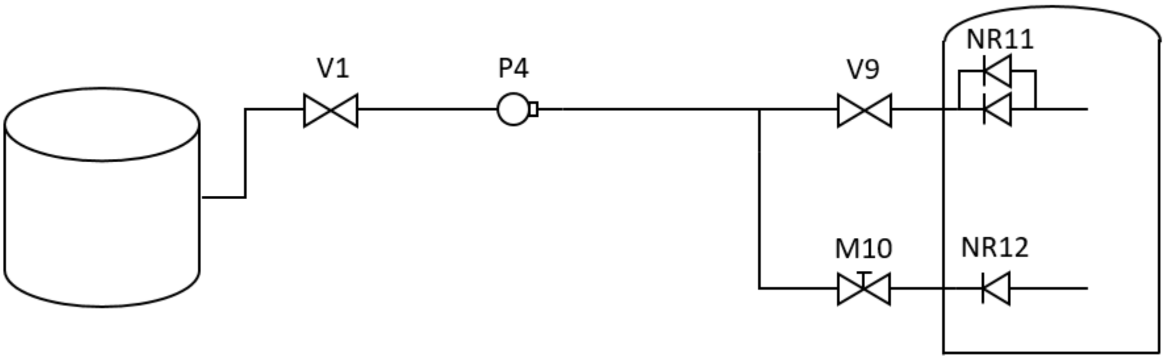


Figure 5.39: L2 solution (Cost = 676.00 - Q = 1.4844·10^{-4}).

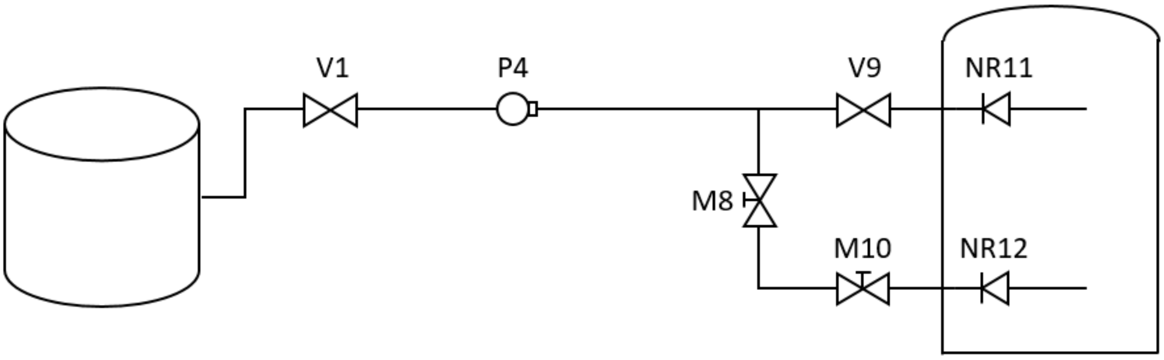


Figure 5.40: S1 solution (Cost = 755.00 - Q = 1.4611·10^{-4}).

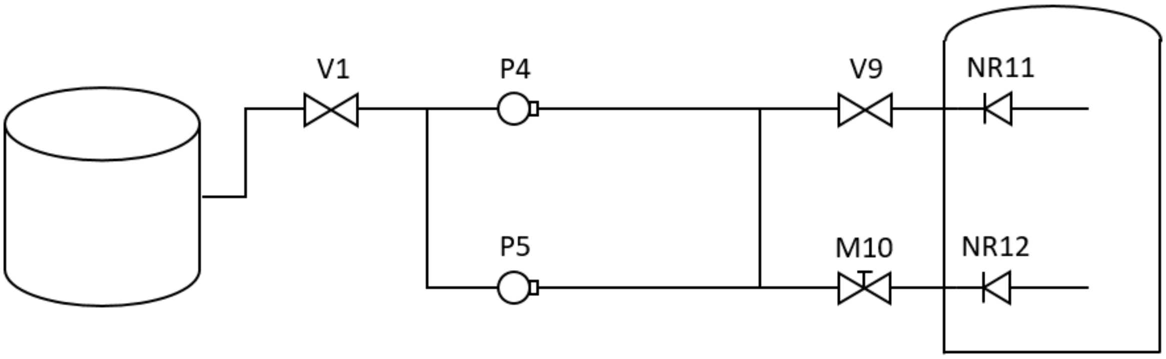


Figure 5.41: L3 and S3 solutions (Cost = 798.00 - Q = 4.5662·10^{-5}, Cost = 1111.00 - Q = 2.7397·10^{-5}, respectively).

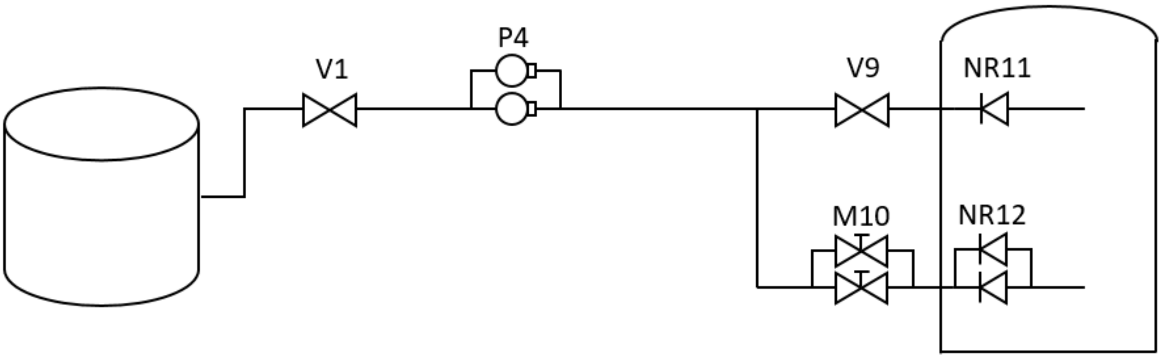


Figure 5.42: S2 solution (Cost = 978.00 - Q = 3.1963·10^{-5}).

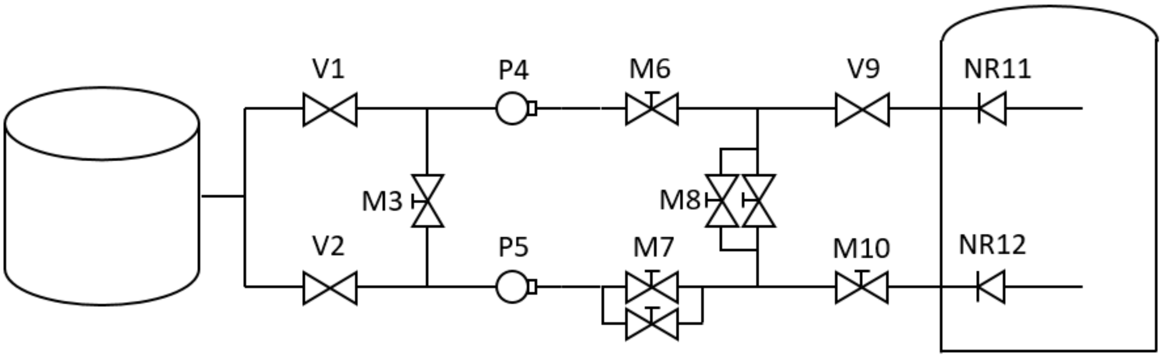


Figure 5.43: SM1 solution (Cost = 1431.00 - Q = 0.0000).

5.2.5.3. Discussion.

The application of the proposed methodology to both application cases demonstrates that its generalisation and scalability is viable. It has been possible to extend the methodology to more complex systems and balanced unavailability-cost solutions could be found with automatic selection of system devices. It was interesting to compare the proposed methodology with the cases with mandatory selection of devices.

In the Application Case A, the proposed methodology avoids choosing single devices located in parallel. Once the non-dominated solutions from the mandatory selection of devices are achieved, it could be seen that the achieved non-dominated solutions from the proposed methodology dominated the previously cited solutions. Hence, the solutions with single devices located in parallel are not optimal solutions. They were rejected by the Multi-objective Evolutionary Algorithm along the evolutionary process.

Regarding the Application Case B, two chromosome codifications (short and long) were explored. Statistically non-significant differences were found regarding the length of the chromosome, however, the short chromosome resulted firstly ordered by the Friedman’s test. It can be seen that solutions from both codifications belong to the accumulated non-dominated front. Nevertheless, generally speaking, the solutions with better cost and worse unavailability were achieved from the long chromosome; conversely, the solutions with worse cost and better unavailability were achieved from the short chromosome. Therefore, depending on the features of the solutions to achieve it could be better to use one or another codification to solve complex problems in the proposed methodology. In any case, the proposed methodology with either of the two tested types of chromosome codifications were able to attain a distributed set of non-dominated solutions along the whole front as it is shown in the Figures 5.35 (Hypervolume distributions) and 5.36.

The proposed methodology is a powerful tool for decision makers (e.g.: chief of engineering, manager of company) in order to plan the systems with simultaneous optimum maintenance cost and unavailability and automatic selection of systems components. It is possible to attain an optimum set of non-dominated solutions with minimum cost and minimum unavailability: it can be observed as the set of solutions where for each value of cost, the best unavailability is shown, or alternatively, the set of solutions where for each value of unavailability the best value of cost is shown.

5.3. Testing encoding and time units.

5.3.1. Background.

This study is currently published [213]. The methodology proposed to promote the improvement of the system Availability by optimising the design and the maintenance strategy by using Multi-objective Evolutionary Algorithms and Discrete Event Simulation was previously explored. Several Multi-objective Evolutionary Algorithms were employed under a real encoding approach and the hour was used as a time unit. In that case, the best ordered configuration from the Friedman's test point of view was obtained when the SMS-EMOA was employed as an optimisation method, with a population size of 150 individuals and a mutation rate of 0.5 gene per chromosome. However, using the NSGA-II method resulted very competitive due to the fact that non-statistically significant differences were found regarding the performance from such methods. The NSGA-II method is one of the most widely employed in order to solve Reliability problems, how it can be checked by reviewing the Chapter III of the present research.

In this second study, the previous one is extended and a deeper exploration using the NSGA-II method is developed. The case study previously presented is analysed again. However, in this case, the possible impact on the solutions as a result of different encodings, parameter configurations and chromosome lengths, which affect the accuracy levels when scheduling preventive maintenance, are attended.

Then, some binary encoding alternatives are explored, looking for possible advantages and disadvantages to encode the real-world problem considered as the case study. Moreover, an accuracy-level experiment for the preventive maintenance strategy is considered. The first part of the study determines the optimum periodical time to start a preventive maintenance activity when the hour is used as a time unit. Two more levels are explored in the second part of the study: the day and the week as a time unit. There are preventive maintenance tasks whose accuracy level in time can be of little importance. It may not be important to determine the exact instant for their development, being enough to define the day or the week. Therefore, the effect of several chromosome lengths is explored looking for improving the evolutionary process. Summarising the scope of the present study:

- In this job, seven encoding alternatives are thoroughly explored: Real encoding (with Simulated Binary Crossover), Binary encoding (with 1-point Crossover), Binary encoding (with 2-point Crossover), Binary encoding (with Uniform Crossover), Gray encoding (with 1-point Crossover), Gray encoding (with 2-point Crossover) and Gray encoding (with Uniform Crossover). Their performances are compared by using the Hypervolume indicator and statistical significance tests.
- Additionally, three accuracy levels on time are explored for the binary encoding; hours, days and weeks, in order to analyse the effect on both the chromosome length in the evolutionary search and the final non-dominated set of solutions. Their performances are compared by using the Hypervolume indicator and statistical significance tests.

5.3.2. Description of the conducted experiments.

Two sets of experiment comparisons were developed: firstly, comparing several encodings (real, binary and Gray), and secondly, comparing several accuracies in binary encoding.

5.3.2.1. Comparing encodings.

In the present study, testing whether there is a significant difference between the performances of the different encodings is intended. Depending on the encoding type, each individual is codified as follows:

Real encoding: The chromosome is formed by strings of real numbers (with 0 as a minimum value and 1 as a maximum value) following the shape $[D_1\ D_2\ M_1\ M_2\ M_3\ M_4\ M_5\ M_6\ M_7]$. The presence of the redundant devices P2 and V4, is defined by the decision variables D_1 and D_2, respectively. The optimum time to start a preventive maintenance task for each system device is denoted by the decision variables M_1 to M_7. The values of the decision variables must be scaled and rounded, i.e., D_1 and D_2 are rounded to the nearest integer (0 implies that the respective device is not selected whereas 1 implies the opposite). M_1 to M_7 are scaled between the respective TM_{min} and TM_{max} (depending on the device) and rounded to the nearest integer by using the Equation 4.5 from the Chapter IV (e.g., the decision variable M_1 denotes the time to start a preventive maintenance task for the valve V1, whose TM_{min} and TM_{max} reach values of 8,760 hours and 35,040 hours, respectively. If the value of the decision variable M_1 is 0.532, the value of the time to start a preventive maintenance task is 8,760 + 0.532 x (35,040 - 8,760) ≈ 23 hours).

Binary encoding: The chromosome is formed by binary number strings that can vary between 0 and 1, where the total number of bits is 103 and they are:

- D_1: This denotes the presence of the pump P2 in the system design (0 implies that the respective device is not considered whereas 1 implies the opposite).
- D_2: This denotes the presence of the valve V4 in the system design (0 implies that the respective device is not considered whereas 1 implies the opposite).
- M_3 to M_{17}: These denote the time to start a preventive maintenance task for the valve V1. A binary scale that allows representing the numbers from TM_{min} and TM_{max} is needed. TM_{min} has a value of 8,760 hours and TM_{max} has a value of 35,040 hours. Therefore, 35,040 - 8760 = 26,280 steps are

needed, where the step 0 denotes a time of 8,760 hours and the step 26,279 denotes a time of 35,040 hours. A binary scale with at least 26,280 steps involves using 15 bits (as 26,280 steps lies between 2^{14} = 16,384 and 2^{15} = 32,768). Since 26,280 steps are needed and 32,768 are available on the scale, an equivalent relationship must be used. Each step in the scale of 32,768 steps denotes 26,768 / 32,768 = 0.8020019531 steps in the scale of 26,768 steps. Therefore, it is possible to calculate the time to start a preventive maintenance task (in hours) by using the scale change that is shown by the Equation 5.3 (based on the Equation 4.6 from the Chapter IV), where B represents the decimal value of the binary string M_3 to M_{17} (e.g., if the values of the decision variables in binary encoding are 101101100011101, the decimal value in the scale of 32,768 steps is 23,325. If 26,768 steps are scaled, the number achieved is 23,325 x 0.8020019531 = 18,707 steps. Therefore, the time to start a preventive maintenance task reaches 8,760 + 18,707 ≈ 25 hours).

$$TM = round\ (TM_{min} + B \cdot 0.8020019531) \tag{5.3}$$

- M_{18} to M_{30}: These denote the time to start a preventive maintenance task for the pump P2. A binary scale that allows representing numbers from TM_{min} and TM_{max} is needed. TM_{min} has a value of 2,920 hours and TM_{max} has a value of 8,760 hours. Therefore, 8,760 - 2,920 = 5,840 steps are needed, where the step 0 denotes the time of 2,920 hours and the step 5,839 represents the time of 8,760 hours. A binary scale with at least 5,840 steps involves using 13 bits (as 5,840 steps lies between 2^{12} = 4,096 and 2^{13} = 8,192). Since 5,840 steps are needed and 8,142 are available on the scale, an equivalent relationship must be used. Each step in the scale of 8,142 represents 5,840 / 8,192 = 0.712890625 steps on a scale of 5840 steps. Therefore, it is possible to calculate the time to start a preventive maintenance task (hours) by using the scale change that is shown by the Equation 5.4, where B represents the decimal value of the binary string M_{18} to M_{30} (e.g., if the values of the decision variables in binary encoding are

1011011000111, such a value on a scale of 8,192 steps is 5,831. If 5,840 steps are scaled, the number achieved is 5,831 x 0.712890625 = 4,157 steps. Therefore, the true time to start a preventive maintenance task reaches 2,920 + 4,157 ≈ 7 hours).

$$TM = round\ (TM_{min} + B \cdot 0.712890625) \tag{5.4}$$

- M_{31} to M_{43}: These denote the time to start a preventive maintenance task for the pump P3. The behaviour of its encoding is like the behaviour explained for the pump P2.
- M_{44} to M_{58}: These denote the time to start a preventive maintenance task for the valve V4. The behaviour of its encoding is like the behaviour explained for the valve V1.
- M_{59} to M_{73}: These denote the time to start a preventive maintenance task for the valve V5. The behaviour of its encoding is like the behaviour explained for the valve V1.
- M_{74} to M_{88}: These denote the time to start a preventive maintenance task for the valve V6. The behaviour of its encoding is like the behaviour explained for the valve V1.
- M_{89} to M_{103}: These denote the time to start a preventive maintenance task for the valve V7. The behaviour of its encoding is like the behaviour explained for the valve V1.

Gray encoding: Some details regarding the Gray code were previously presented in the Section 4.5.2.3 of the Chapter IV. In addition to the standard binary encoding, the Gray code is used in this research.

Therefore, the performance of the real, the standard binary and the Gray encodings are compared.

5.3.2.2. Comparing Accuracies.

A second experiment is developed, which consists of studying the possible impact depending on the size of the chromosome. In the encoding experiment, the hour is used by the chromosomes as a measure of time. In this case, based on the idea that the exact hour to develop a preventive maintenance task is not needed, the day and the week are used as measures of time. Therefore, in these cases, the solution regarding the preventive maintenance strategy consists of supplying the time to start a preventive maintenance task with the day or the week as a time unit, respectively. The consequence is a reduction in the size of the chromosome, which is applied to the binary encoding as follows:

Binary encoding - Days: The chromosome is formed by binary number strings that can vary between 0 and 1, where the total number of bits is 73 and they are:

- D_1: This denotes the presence of the pump P2 in the system design (0 implies that the device is not selected whereas 1 implies the opposite).
- D_2: This denotes the presence of the valve V4 in the system design (0 implies that the device is not selected whereas 1 implies the opposite).
- M_3 to M_{13}: These denote the time to start a preventive maintenance task for the valve V1. A binary scale that allows representing the numbers from TM_{min} to TM_{max} expressed in days as a time unit is needed. TM_{min} has a value of 8,760 hours (equivalent to 365 days) and TM_{max} has a value of 35,040 hours (equivalent to 1,460 days). Therefore, 1,460 - 365 = 1,095 steps are needed, where the step 0 denotes the time of 365 days, and the step 1,094 denotes the time of 1,460 days. A binary scale with at least 1,095 steps involves using 11 bits (due to the fact that 1,095 lies between 2^{10} = 1,024 and 2^{11} = 2,048). Since 1095 steps are needed and 2048 are available on the scale, an equivalent relationship must be used. Each step on the scale of 2,048 steps represents 1,095 / 2,048 = 0.5346679688 steps on a scale of 1,095 steps. Therefore, it is possible to achieve the time to start a preventive maintenance activity (days) by using the scale change that is shown by the Equation 5.5, where B represents the decimal value of the binary string M_3

to M_{13} (e.g., if the values of the decision variables on the binary encoding are 10110110001, the decimal value on the scale of 2,048 steps will be 1,457. Scaling to a scale of 1,095 steps, the number achieved is 1,457 x 0.5346679688 = 779 steps. Therefore, the time to start a preventive maintenance task reaches 779 + 365 = 1,144 days).

$$TM = round\ (TM_{min} + B \cdot 0.5346679688) \tag{5.5}$$

- M_{14} to M_{21}: These denote the time to start a preventive maintenance task for the pump P2. A binary scale that allows representing the numbers from TM_{min} to TM_{max} expressed in days as a time unit is needed. TM_{min} has a value of 2,920 hours (equivalent to 122 days) and TM_{max} has a value of 8,760 (equivalent to 365 days). Therefore, 365 - 122 = 243 steps are needed, where the step 0 denotes the time of 122 days, and the step 242 represents the time of 365 days. A binary scale with at least 243 steps involves using 8 bits (as 243 lies between 2^7 = 128 and 2^8 = 256). Since 243 steps are needed and 256 are available on the scale, an equivalent relationship must be used. Each step in the scale of 256 represents 243 / 256 = 0.94921875 steps in the scale of 243 steps. Therefore, it is possible to achieve the true time to start a preventive maintenance task (days) by using the scale change that is shown by the Equation 5.6, where B represents the decimal value of the binary string M_{14} to M_{21} (e.g., if the values of the decision variables in binary encoding are 10110110, the value on the scale of 256 steps is 182. Scaling to a scale of 243 steps, the number achieved is 182 x 0.94921875 = 173 steps. Therefore, the true time to start a preventive maintenance task reaches 173 + 122 = 295 days).

$$TM = round\ (TM_{min} + B \cdot 0.94921875) \tag{5.6}$$

- M_{22} to M_{29}: These denote the time to start a preventive maintenance task for the pump P3. The behaviour of its encoding is like the behaviour explained for the pump P2.

- M_{30} to M_{40}: These denote the time to start a preventive maintenance task for the valve V4. The behaviour of its encoding is like the behaviour explained for the valve V1.
- M_{41} to M_{51}: These denote the time to start a preventive maintenance task for the valve V5. The behaviour of its encoding is like the behaviour explained for the valve V1.
- M_{52} to M_{62}: These denote the time to start a preventive maintenance task for the valve V6. The behaviour of its encoding is like the behaviour explained for the valve V1.
- M_{63} to M_{73}: These denote the time to start a preventive maintenance task for the valve V7. The behaviour of its encoding is like the behaviour explained for the valve V1.

Binary encoding - Weeks: The chromosome is formed by binary number strings that can vary between 0 and 1, where the total number of bits is 54 and they are:

- D_1: This denotes the presence of the pump P2 in the system design (0 implies that the device is not selected whereas 1 implies the opposite).
- D_2: This denotes the presence of the valve V4 in the system design (0 implies that the device is not selected whereas 1 implies the opposite).
- M_3 to M_{10}: These denote the time to start a preventive maintenance task for the valve V1. A binary scale that allows representing the numbers from TM_{min} to TM_{max} expressed in weeks as a time unit is needed. TM_{min} has a value of 8,760 hours (equivalent to 52 weeks) and TM_{max} has a value of 35,040 hours (equivalent to 209 weeks) so 209 - 52 = 157 steps are needed, where the step 0 represents a time of 52 weeks, and the step 156 denotes a time of 209 weeks. A binary scale with at least 157 steps involves using 8 bits (as 157 lies between 2^7 = 128 and 2^8 = 256). Since 157 steps are needed and 256 are available on the scale, an equivalent relationship must be used. Each step on a 256-steps scale represents 157 / 256 = 0.61328125 steps on the 157-steps scale. Therefore, it is possible to achieve the time to start a preventive maintenance task (weeks) by using the scale change that is

shown by the Equation 5.7, where B represents the decimal value of the binary string M_3 to M_{10} (e.g., if the values of the decision variables in binary encoding are 10110110, the decimal value in the scale of 256 steps will be 182. Working with a scale of 157 steps, the number achieved is 182 x 0.61328125 = 112 steps. Therefore, the time to start a preventive maintenance task reaches 52 + 112 = 164 weeks).

$$TM = round\ (TM_{min} + B \cdot 0.61328125) \qquad (5.7)$$

- M_{11} to M_{16}: These denote the time to start a preventive maintenance task for the pump P2. A binary scale that allows representing the numbers from TM_{min} to TM_{max} expressed in weeks as a time unit is needed. TM_{min} has a value of 2,920 hours (equivalent to 17 weeks) and TM_{max} has a value of 8,760 hours (equivalent to 52 weeks), Therefore, 52 - 17 = 35 steps are needed, where the step 0 represents the time of 17 weeks and the step 34 represents the time of 52 weeks. A binary scale with at least 35 steps involves using 6 bits (as 35 lies between 2^5 = 32 and 2^6 = 64). Since 35 steps are needed and 64 are available on the scale, an equivalent relationship must be used. Each step on the scale of 64-steps scale represents 35 / 64 = 0.546875 steps in the 35-steps scale. Therefore, it is possible to achieve the time to start a preventive maintenance task (weeks) by using the scale change that is shown by the Equation 5.8, where B represents the decimal value of the binary string M_{11} to M_{16} (e.g., if the values of the decision variables in binary encoding are 101101, the value in the scale of 64 steps will be 45. Scaling on the 35-steps scale, the number achieved is 45 x 0.546875 = 25 steps. Therefore, the true time to start a preventive maintenance task reaches 17 + 45 = 62 weeks).

$$TM = round\ (TM_{min} + B \cdot 0.546875) \qquad (5.8)$$

- M_{17} to M_{22}: These denote the time to start a preventive maintenance task for the pump P3. The behaviour of its encoding is like the behaviour explained for the pump P2.
- M_{23} to M_{30}: These denote the time to start a preventive maintenance task for the valve V4. The behaviour of its encoding is like the behaviour explained for the valve V1.
- M_{31} to M_{38}: These denote the time to start a preventive maintenance task for the valve V5. The behaviour of its encoding is like the behaviour explained for the valve V1.
- M_{39} to M_{46}: These denote the time to start a preventive maintenance task for the valve V6. The behaviour of its encoding is like the behaviour explained for the valve V1.
- M_{47} to M_{54}: These denote the time to start a preventive maintenance task for the valve V7. The behaviour of its encoding is like the behaviour explained for the valve V1.

Summarising, the longitude of the chromosome is 103 bits when the hour is used as a time unit, 73 when the day is used as a time unit and 54 when the week is used as a time unit.

5.3.2.3. NSGA-II Configuration.

The parameters used to configure the NSGA-II method are shown in Table 5.22.

Method	Experiment	Encoding	Crossover	Time unit	Population	PrM	disM	PrC	disC
NSGA-II	Encoding	Real	SBX	Hour	50 -100 -150	0.5-1-1.5	20	1	20
		Standard Binary	1-point (1PX)	Hour			-		-
		Standard Binary	2-point (2PX)	Hour					
		Standard Binary	Uniform (UX)	Hour					
		Gray	1-point (1PX)	Hour					
		Gray	2-point (2PX)	Hour					
		Gray	Uniform (UX)	Hour					
	Accuracy	Standard Binary	1-point (1PX)	Day					
		Standard Binary	2-point (2PX)	Day					
		Standard Binary	Uniform (UX)	Day					
		Standard Binary	1-point (1PX)	Week					
		Standard Binary	2-point (2PX)	Week					
		Standard Binary	Uniform (UX)	Week					

Table 5.22: Parameters to configure the experiments.

Depending on the encoding applied, specific parameters must be set, which are described below. The type of crossover used during the optimisation process is different because depends on the encoding. The Simulated Binary Crossover (SBX) is used for real encoding while one-point (1PX), two-point (2PX) and uniform crossover (UX) are used for binary and Gray encodings. The population sizes used are 50, 100 and 150 individuals. Like in the first study, the mutation rates (PrM) are 0.5, 1 and 1.5 genes per chromosome. The crossover probability (PrC) is set to 1 in all cases.

Each configuration was executed 21 times (for statistical purposes) with 10,000,000 evaluations used as the stopping criterion. Scale factors in relation to the value of the objective functions were used with the purpose of achieving a dispersed non-dominated front with the unit as maximum value. Again, the values are obtained by following a practical approach in which the values of the scale factors are extracted from the values of the objective functions when the optimisation process starts. The scale factors are as follows:

- The scale factor used to compute the Cost is 1,700 economic units.
- The scale factor used to compute the system Unavailability is 0.003.

Finally, a two-dimensional reference point is needed to compute the Hypervolume indicator. The cited point must cover the values limited by the scale factors, which restricts the values of the objective functions to a maximum of one. The reference point is set to (2,2). Again, the PlatEMO platform is used to develop the study.

5.3.3. Results.

Due to the complexity of the problem, a general-purpose calculation cluster (High-Performance Computer) was used during the computational process. The cluster is composed by 28 calculation nodes and one access node. Each calculation node consists of 2 Intel Xeon E5645 Westmere-EP processors with 6 cores each and 48 GB of RAM memory, allowing 336 executions to be run simultaneously.

Once the results were obtained, valuable information emerged. For each analysed case, the following information is provided: Firstly, information regarding the computational process is given with the purpose of showing the complexity of the problem and its computational cost. It consists of the time taken for 21 executions of the nine configurations (three population sizes and three mutation rates) related to each analysed case. Secondly, the values of the main measures achieved for the final evaluation in 21 executions are shown. These measures are the Average, Median, Minimum, Maximum and Standard Deviation values of the Hypervolume indicator (HV). Thirdly, in order to establish the existence of significant differences between the performance of the studied cases, a rigorous statistical analysis is conducted. The Friedman's test allows significant differences among results obtained to be detected, and the null hypothesis (H_0) to be rejected in that case. Finally, the Hypervolume is computed for the accumulated non-dominated solutions obtained (the non-dominated front). These represent the best equilibrium solutions among the objectives.

Once the configurations were ordered according to the Friedman's test values, one configuration from each analysed case is used for the final comparison taking the two experiments into consideration: one looking at encodings and the other looking at to accuracy. In each case, additional information is given. The Hypervolume indicator average value evolution (in 21 executions) is shown for each configuration. Moreover, box plots are given for the Hypervolume values distribution after the stopping criterion is reached. In addition, the Friedman's test is used to detect significant differences among the performance of the configurations for each experiment. Finally, the accumulated best non-dominated solutions obtained (non-dominated front) are shown.

5.3.3.1. Encoding Experiment.

Real encoding.

The results of using real encoding with simulated binary crossover are shown below. This experiment was previously developed for the first study, so the achieved results are similar. The computational time consumed is shown in the Table 5.23.

Encoding	Time Unit	Average time	Sequential Time
Real SBX	Hour	2 days, 18 hours and 18 minutes	1 year, 5 months, 5 days, 3 hours and 25 minutes
Binary 1PX	Hour	2 days, 18 hours and 38 minutes	1 year, 5 months, 7 days, 16 hours and 49 minutes
Binary 2PX	Hour	2 days, 22 hours and 41 minutes	1 year, 5 months, 13 days, 1 hours and 40 minutes
Binary U	Hour	3 days, 1 hour and 5 minutes	1 year, 6 months, 28 days, 2 hours and 34 minutes
Gray 1PX	Hour	2 days, 19 hours and 48 minutes	1 year, 5 months, 16 days, 21 hours and 27 minutes
Gray 2PX	Hour	2 days, 21 hours and 40 minutes	1 year, 6 months, 1 days, 5 hours and 42 minutes
Gray UX	Hour	2 days, 20 hours and 15 minutes	1 year, 5 months, 20 days, 11 hours and 12 minutes
TOTAL		**20 days, 3 hours and 1 minutes**	**10 years, 4 months, 6 days, 2 hours and 18 minutes**

Table 5.23: Computational cost (encoding experiment).

The average time is the computational time regarding each one of 21 executions and nine different configurations (real time consumed). The sequential time is the computational time that would have been needed in case of not using the High-Performance Computer.

The relationship between the method configurations (where N represents the population size and PrM the mutation rate) and identifiers is shown in the Table 5.24. Moreover, statistical information regarding the Hypervolume value at the end of the evolutionary process is shown (average, median, maximum, minimum and standard deviation, out of 21 independent executions). It is possible to conclude that the configuration with the identifier ID9 (with a population of 150 individuals and mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume average value, the configuration with identifier ID3 (population of 150 individuals and 0.5 gene per chromosome as a mutation rate) reaches the highest Hypervolume median value, the configuration with identifier ID5 (population of 100 individuals and 1 gene per chromosome as a mutation rate) reaches the highest Hypervolume maximum and minimum values, and the configuration with identifier ID2 (population of 100 individuals and 0.5 gene per chromosome as a mutation rate) presents the lowest Hypervolume standard deviation value.

Identifier	Configuration	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	N = 50 – PrM = 0.5	2.2831	2.2859	2.3180	2.2611	0.0162	5.809
ID2	N = 100 – PrM = 0.5	2.2864	2.2872	2.3007	2.2625	**0.0118**	5.095
ID3	N = 150 – PrM = 0.5	2.2955	**2.3011**	2.3227	2.2635	0.0186	**4.142**
ID4	N = 50 – PrM = 1.0	2.2801	2.2820	2.3070	2.2606	0.0141	6.571
ID5	N = 100 – PrM = 1.0	2.2944	2.2941	**2.3390**	**2.2714**	0.0163	4.190
ID6	N = 150 – PrM = 1.0	2.2874	2.2871	2.3217	2.2623	0.0174	5.333
ID7	N = 50 – PrM = 1.5	2.2898	2.2879	2.3277	2.2534	0.0240	5.285
ID8	N = 100 – PrM = 1.5	2.2931	2.2895	2.3266	2.2658	0.0166	4.557
ID9	N = 150 – PrM = 1.5	**2.2957**	2.2941	2.3281	2.2592	0.0158	4.000
p-Value							0.035

Table 5.24: Id’s, config., Hyperv. statistics and statistical test (real encoding).

In order to establish the best behaviour amongst the configurations, a statistical significance hypothesis test was conducted. The average ranks computed by using the Friedman’s test and the *p*-value obtained (a value bigger than 0.05 implies that the null hypothesis cannot be rejected, suggesting that all configurations perform in a similar way) are shown in the Table 5.24. It can be seen that the configuration with identifier ID3 (population of 150 individuals and mutation rate of 0.5 gene per chromosome) presents the best average rank (in order to maximise the Hypervolume, the average rank must be as low as possible), so it was selected for the final comparison study among encoding configurations.

Finally, the best accumulated non-dominated solutions obtained from the final generation for all executions and all configurations were used to compute the accumulated Hypervolume, whose value was 2.4068. As it is expected, the value is higher than 2.3390, the maximum value that is shown in the Table 5.24.

Standard Binary Encoding.

The results of using standard binary encoding with one-point, two-point and uniform crossover are shown below. The computational time consumed by each one is shown in the Table 5.23. The relationship between the method configurations and the identifiers is shown in the Table 5.25. Moreover, statistical information regarding the Hypervolume value at the end of the evolutionary process is shown.

For the binary encoding with one-point crossover (B1PX), it is possible to conclude that the configuration with identifier ID8 (population of 100 individuals and mutation rate of 1.5 gene per chromosome) presents both the highest Hypervolume average value and the highest Hypervolume median value, the configuration with identifier ID7 (population of 50 individuals and a mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume maximum value, the configuration with identifier ID2 (population of 100 individuals and a mutation rate of 0.5 gene per chromosome) presents the highest Hypervolume minimum value, and the configuration with identifier ID6 (population of 150 individuals and a mutation rate of 1 gene per chromosome) presents the lowest Hypervolume standard deviation.

Encoding	Identifier	Configuration	Average	Median	Max.	Min.	St. Deviation	Av. Rank
B1PX	ID1	N = 50 – PrM = 0.5	2.2844	2.2843	2.3297	2.2496	0.0175	5.523
	ID2	N = 100 – PrM = 0.5	2.2933	2.2927	2.3336	**2.2703**	0.0163	4.428
	ID3	N = 150 – PrM = 0.5	2.2892	2.2921	2.3313	2.2554	0.0171	4.809
	ID4	N = 50 – PrM = 1.0	2.2921	2.2913	2.3253	2.2566	0.0175	4.523
	ID5	N = 100 – PrM = 1.0	2.2846	2.2826	2.3200	2.2548	0.0156	5.666
	ID6	N = 150 – PrM = 1.0	2.2894	2.2824	2.3216	2.2697	**0.0144**	4.999
	ID7	N = 50 – PrM = 1.5	2.2821	2.2777	**2.3396**	2.2599	0.0189	6.285
	ID8	N = 100 – PrM = 1.5	**2.2943**	**2.3003**	2.3126	2.2640	0.0153	**3.714**
	ID9	N = 150 – PrM = 1.5	2.2885	2.2891	2.3394	2.2616	0.0166	5.047
p-Value								0.114
B2PX	ID1	N = 50 – PrM = 0.5	2.2877	2.2861	2.3253	2.2401	0.0216	5.428
	ID2	N = 100 – PrM = 0.5	**2.2975**	**2.2961**	2.3436	**2.2714**	0.0220	**4.428**
	ID3	N = 150 – PrM = 0.5	2.2889	2.2888	**2.3627**	2.2630	0.0218	5.238
	ID4	N = 50 – PrM = 1.0	2.2898	2.2866	2.3186	2.2612	0.0155	5.047
	ID5	N = 100 – PrM = 1.0	2.2894	2.2920	2.3056	2.2638	**0.0089**	4.523
	ID6	N = 150 – PrM = 1.0	2.2903	2.2910	2.3215	2.2644	0.0157	5.047
	ID7	N = 50 – PrM = 1.5	2.2902	2.2891	2.3483	2.2571	0.0180	4.666
	ID8	N = 100 – PrM = 1.5	2.2867	2.2865	2.3326	2.2606	0.0177	5.428
	ID9	N = 150 – PrM = 1.5	2.2861	2.2844	2.3107	2.2463	0.0178	5.190
p-Value								0.923
BUX	ID1	N = 50 – PrM = 0.5	2.2889	2.2902	2.3048	2.2611	**0.0129**	5.142
	ID2	N = 100 – PrM = 0.5	2.2920	2.2925	2.3163	2.2647	0.0155	4.142
	ID3	N = 150 – PrM = 0.5	2.2904	2.2905	2.3265	2.2576	0.0164	4.809
	ID4	N = 50 – PrM = 1.0	2.2904	2.2869	**2.3497**	**2.2702**	0.0170	4.619
	ID5	N = 100 – PrM = 1.0	**2.2972**	**2.2938**	2.3332	2.2660	0.0218	**4.190**
	ID6	N = 150 – PrM = 1.0	2.2921	2.2882	2.3488	2.2687	0.0211	5.047
	ID7	N = 50 – PrM = 1.5	2.2865	2.2847	2.3207	2.2622	0.0164	5.666
	ID8	N = 100 – PrM = 1.5	2.2858	2.2835	2.3148	2.2579	0.0142	5.666
	ID9	N = 150 – PrM = 1.5	2.2823	2.2844	2.3036	2.2562	**0.0129**	5.714
p-Value								0.397

Table 5.25: Id’s, config., Hyperv. statistics and statistical test (binary encoding).

For the binary encoding with two-point crossover (B2PX), it is possible to conclude that the configuration with identifier ID2 (population of 100 individuals and a mutation rate of 0.5 gene per chromosome) presents the highest Hypervolume average value, the highest Hypervolume median value and the highest Hypervolume minimum value. The configuration with identifier ID3 (population of 150 individuals and a mutation rate of 0.5 gene per chromosome) presents the highest Hypervolume maximum value and the configuration with identifier ID5 (population of 100 individuals and a mutation rate of 1 gene per chromosome) presents the lowest Hypervolume standard deviation value.

For the binary encoding with uniform crossover (BUX), it is possible to conclude that the configuration with identifier ID5 (population of 100 individuals and a mutation rate of 1 gene per chromosome) presents the highest Hypervolume average value and the highest Hypervolume median value. The configuration with identifier ID4 (population of 50 individuals and a mutation rate of 1 gene per chromosome) presents the highest Hypervolume maximum value and the highest Hypervolume minimum value. Finally, both the configuration with identifier ID1 (population of 50 individuals and mutation rate of 0.5 gene per chromosome) and the configuration with identifier ID9 (population of 150 individuals and mutation rate of 1.5 gene per chromosome) present the lowest Hypervolume standard deviation value.

In order to establish the best behaviour amongst the configurations, a statistical significance hypothesis test was conducted. The average ranks computed by using the Friedman's test and the *p*-value obtained (a value bigger than 0.05 implies that the null hypothesis cannot be rejected, suggesting that all configurations perform in a similar way) are shown in the Table 5.25. The configuration with identifier ID8 (population of 100 individuals and a mutation rate of 1.5 gene per chromosome) presents the best average rank for the binary encoding with one-point crossover. The configuration with identifier ID2 (population of 100 individuals and mutation rate of 0.5 gene per chromosome) presents the best average rank for the binary encoding with two-point crossover. Finally, the configuration with identifier ID5 (population of 100 individuals and a mutation rate of 1 gene per chromosome) presents the best average rank for the binary encoding with uniform crossover. These configurations are selected for the final comparison study among encoding configurations which is shown later. The best accumulated non-dominated solutions obtained from the final generation for all executions and all configurations were used to compute the accumulated Hypervolume whose values were 2.4227, 2.4378 and 2.4134 for the binary encoding with one-point, two-point and uniform crossover, respectively. As it is expected, the values are higher than 2.3396, 2.3627 and 2.3497, the maximum values that are shown in the Table 5.25, respectively.

Gray encoding.

The results of employing the Gray encoding with one-point, two-point and uniform crossover are shown below. The computational time consumed by each one is shown in the Table 5.23. The relationship between the method configurations and the identifiers is shown in the Table 5.26. Moreover, statistical information in relation to the Hypervolume value at the end of the evolutionary process is shown. For the Gray encoding with one-point crossover (G1PX), it is possible to conclude that the configuration with identifier ID2 (population of 100 individuals and mutation rate of 0.5 gene per chromosome) presents the highest Hypervolume average value, the highest Hypervolume median value and the highest Hypervolume maximum value. The configuration with identifier ID4 (population of 50 individuals and mutation rate of 1 gene per chromosome) presents both the highest Hypervolume minimum value and the lowest Hypervolume standard deviation value.

For the Gray encoding with two-point crossover (G2PX), it is possible to conclude that the configuration with identifier ID9 (population of 150 individuals and mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume average value, the highest Hypervolume median value, the highest Hypervolume maximum value and the highest Hypervolume minimum value. The configuration with identifier ID5 (population of 100 individuals and mutation rate of 1 gene per chromosome) presents the lowest Hypervolume standard deviation value.

Encoding	Identifier	Configuration	Average	Median	Max.	Min.	St. Deviation	Av. Rank
G1PX	ID1	N = 50 – PrM = 0.5	2.2880	2.2846	2.3280	2.2652	0.0148	4.857
	ID2	N = 100 – PrM = 0.5	**2.2938**	**2.2954**	**2.3556**	2.2640	0.0198	4.333
	ID3	N = 150 – PrM = 0.5	2.2833	2.2855	2.3040	2.2626	0.0109	6.047
	ID4	N = 50 – PrM = 1.0	2.2922	2.2905	2.3175	**2.2762**	**0.0108**	4.285
	ID5	N = 100 – PrM = 1.0	2.2918	2.2893	2.3347	2.2647	0.0177	**4.142**
	ID6	N = 150 – PrM = 1.0	2.2830	2.2791	2.3043	2.2611	0.0123	6.095
	ID7	N = 50 – PrM = 1.5	2.2894	2.2926	2.3171	2.2526	0.0171	4.333
	ID8	N = 100 – PrM = 1.5	2.2827	2.2847	2.3149	2.2608	0.0139	5.714
	ID9	N = 150 – PrM = 1.5	2.2894	2.2836	2.3224	2.2692	0.0169	5.190
p-Value								0.081
G2PX	ID1	N = 50 – PrM = 0.5	2.2860	2.2814	2.3192	2.2577	0.0149	5.333
	ID2	N = 100 – PrM = 0.5	2.2869	2.2906	2.3206	2.2556	0.0197	5.285
	ID3	N = 150 – PrM = 0.5	2.2857	2.2833	2.3112	2.2519	0.0124	4.571
	ID4	N = 50 – PrM = 1.0	2.2880	2.2818	2.3186	2.2659	0.0166	5.000
	ID5	N = 100 – PrM = 1.0	2.2843	2.2836	2.3004	2.2658	**0.0098**	5.142
	ID6	N = 150 – PrM = 1.0	2.2900	2.2870	2.3256	2.2640	0.0162	**4.285**
	ID7	N = 50 – PrM = 1.5	2.2832	2.2783	2.3205	2.2617	0.0177	5.476
	ID8	N = 100 – PrM = 1.5	2.2848	2.2842	2.3140	2.2585	0.0142	5.619
	ID9	N = 150 – PrM = 1.5	**2.2936**	**2.2911**	**2.3364**	**2.2684**	0.0178	**4.285**
p-Value								0.683
GUX	ID1	N = 50 – PrM = 0.5	2.2846	2.2791	2.3334	2.2540	0.0199	4.857
	ID2	N = 100 – PrM = 0.5	2.2833	2.2801	2.3060	2.2634	**0.0109**	5.047
	ID3	N = 150 – PrM = 0.5	2.2843	2.2852	2.3077	2.2665	0.0129	4.999
	ID4	N = 50 – PrM = 1.0	2.2871	2.2807	**2.3358**	**2.2683**	0.0174	4.904
	ID5	N = 100 – PrM = 1.0	**2.2898**	2.2858	2.3167	2.2611	0.0165	4.333
	ID6	N = 150 – PrM = 1.0	2.2832	2.2848	2.3048	2.2615	0.0116	5.095
	ID7	N = 50 – PrM = 1.5	2.2856	2.2836	2.3140	2.2608	0.0154	5.523
	ID8	N = 100 – PrM = 1.5	2.2875	**2.2866**	2.3099	2.2524	0.0149	**4.190**
	ID9	N = 150 – PrM = 1.5	2.2803	2.2745	2.3150	2.2638	0.0163	6.047
p-Value								0.532

Table 5.26: Id’s, config., Hyperv. statistics and statistical test (Gray encoding).

For the Gray encoding with uniform crossover (GUX), it is possible to conclude that the configuration with identifier ID5 (population of 100 individuals and a mutation rate of 1 gene per chromosome) presents the highest Hypervolume average value, the configuration with identifier ID8 (population of 100 individuals and a mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume median value, the configuration with identifier ID4 (population of 50 individuals and a mutation rate of 1 gene per chromosome) presents both the highest Hypervolume maximum value and the highest Hypervolume minimum value and the configuration with identifier ID2 (population of 100 individuals and a mutation rate of 0.5 gene per chromosome) presents the lowest Hypervolume standard deviation value.

In order to establish the best behaviour amongst the configurations, a statistical significance hypothesis test was conducted. The average ranks computed by using the Friedman’s test and the *p*-value obtained are shown in the Table 5.26. It can be seen that the configuration with identifier ID5 (population of 100 individuals and mutation rate of 1 gene per chromosome) presents the best average rank for the Gray encoding with one-point crossover. It can be seen that the configurations with identifiers ID6 (population of 150 individuals and mutation rate of 1 gene per chromosome) and ID9 (population of 150 individuals and mutation rate of 1.5 gene per chromosome) present the best average rank for the Gray encoding with two-point crossover. Finally, it can be seen that the configuration with identifier ID8 (population of 100 individuals and mutation rate of 1.5 gene per chromosome) presents the best average rank for the Gray encoding with uniform crossover. These configurations were selected for the final comparison study among encoding configurations, which is shown later. The best accumulated non-dominated solutions obtained from the final generation of the evolutionary process for all executions and all configurations were used to compute the accumulated Hypervolume, whose values were 2.4130, 2.4050 and 2.4038 for the Gray encoding with one-point, two-point and uniform crossover, respectively. As expected, the values are higher than 2.3556, 2.3364 and 2.3358, the maximum values shown in the Table 5.26, respectively.

Comparing Encoding Configurations.

The total computational time consumed is shown in the Table 5.23. The computational cost shows the importance of employing the High-Performance Computer. Previously, the configurations with the best average rank according to the Friedman's test were selected to be globally compared. These configurations are shown in the Table 5.27.

Identifier	Description	Configuration	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	Real	N = 150 – PrM = 0.5	2.2955	**2.3011**	2.3227	2.2635	0.0186	4.047
ID2	Binary 1P	N = 100 – PrM = 1.5	2.2943	2.3003	2.3126	2.2640	0.0153	4.190
ID3	Binary 2P	N = 100 – PrM = 0.5	**2.2975**	2.2961	**2.3436**	**2.2714**	0.0220	**3.809**
ID4	Binary U	N = 100 – PrM = 1.0	2.2972	2.2938	2.3332	2.2660	0.0218	4.285
ID5	Gray 1P	N = 100 – PrM = 1.0	2.2918	2.2893	2.3347	2.2647	0.0177	4.904
ID6	Gray 2P	N = 150 – PrM = 1.0	2.2900	2.2870	2.3256	2.2640	0.0162	4.952
ID7	Gray 2P	N = 150 – PrM = 1.5	2.2936	2.2911	2.3364	2.2684	0.0178	4.571
ID8	Gray U	N = 100 – PrM = 1.5	2.2875	2.2866	2.3099	2.2524	**0.0149**	5.238
p-Value								0.528

Table 5.27: Id's, config., Hyperv. statistics and statistical test (enc. experiment).

The Hypervolume average values evolution versus the evaluations number is shown in the Figure 5.44. The detail for the final evaluations (last million fitness function evaluations, from 9 to 10 million) is shown in the Figure 5.45.

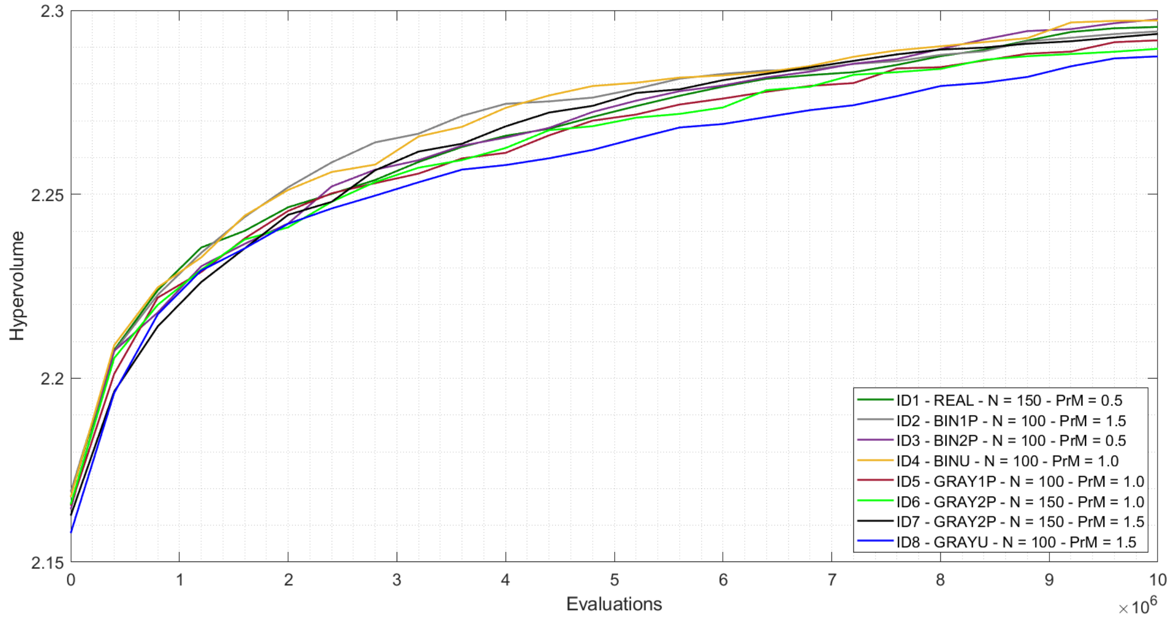


Figure 5.44: Hypervolume Average vs. evaluations (enc. experiment).

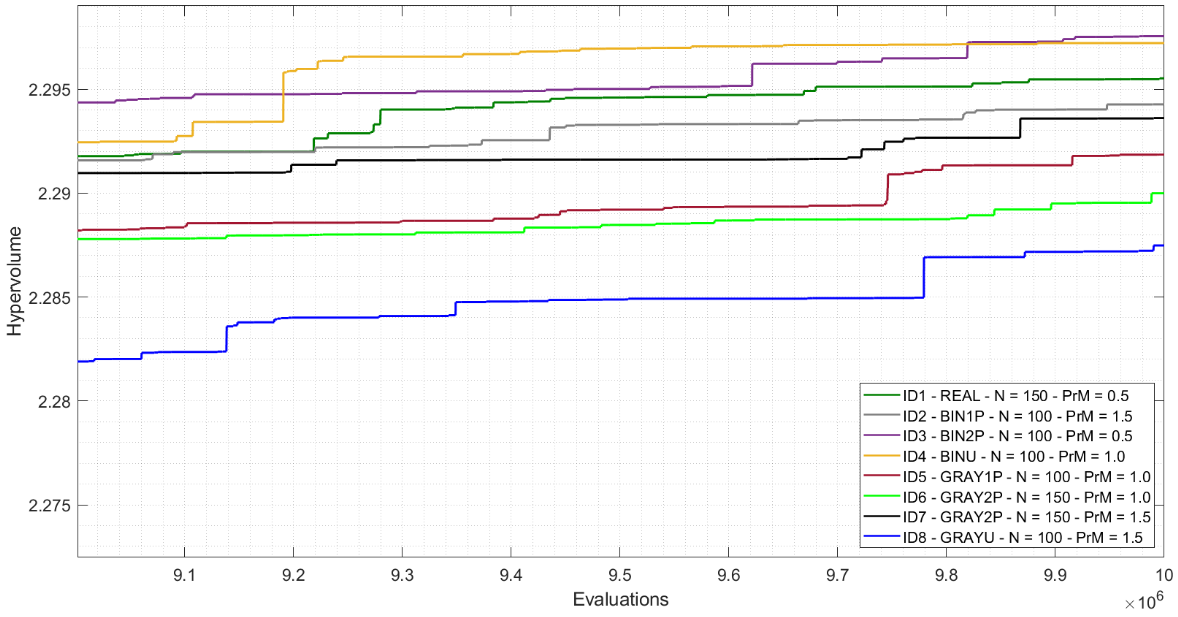


Figure 5.45: Hypervolume Average vs. evaluations, detail (enc. experiment).

It can be seen that the configuration with identifier ID3 (with binary encoding and two-point crossover, population of 100 individuals and mutation rate of 0.5 gene per chromosome) reaches the highest Hypervolume average value.

Box plots of the Hypervolume values distribution at the end of the process are shown in the Figure 5.46. Statistical information in relation to the Hypervolume value at the end of the evolutionary process is shown in the Table 5.27. It can be seen that the configuration with identifier ID3 (with Binary encoding and two-point crossover, population of 100 individuals and a mutation rate of 0.5 gene per chromosome) presents the highest Hypervolume average value, the highest Hypervolume maximum value and the highest Hypervolume minimum value. The configuration with identifier ID1 (with real encoding, population of 150 individuals and a mutation rate of 0.5 gene per chromosome) presents the highest Hypervolume median value. Finally, the configuration with identifier ID9 (with Gray encoding and uniform crossover, population of 100 individuals and a mutation rate of 1.5 gene per chromosome) presents the lowest Hypervolume standard deviation value.

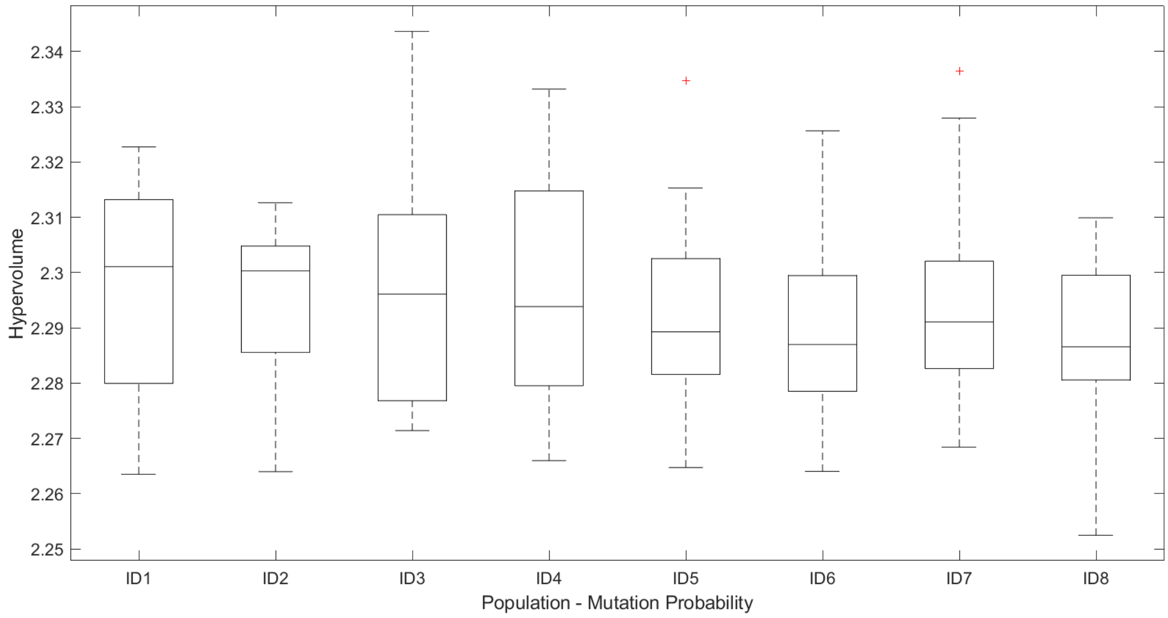


Figure 5.46: Hypervolume box plots (enc. experiment).

In order to determine if one of the configurations performs better than any other, a statistical significance hypothesis test was conducted. The average ranks computed by using the Friedman's test are shown in the Table 5.27. The configuration with identifier ID3 (with binary encoding and two-point crossover, population of 100 individuals and a mutation rate of 0.5 gene per chromosome) presents the best average rank. However, the *p*-value computed (0.528) implies that the null hypothesis (H_0) cannot be rejected (*p*-value > 0.05), so it is possible to conclude that, in the studied conditions, all configurations perform similar.

The best accumulated non-dominated solutions obtained from all the selected encodings and configurations were used to compute the accumulated Hypervolume, whose value is 2.4588. As it is expected, the value is higher than 2.4378, the maximum accumulated value obtained after the evolutionary process for the standard binary encoding with two-point crossover. This is shown in the Table 5.28.

Encoding	Hypervolume Accumulated Value
Real	2.4068
Binary 1 Point Crossover	2.4227
Binary 2 Point Crossover	2.4378
Binary Uniform Crossover	2.4134
Gray 1 Point Crossover	2.4130
Gray 2 Point Crossover	2.4050
Gray Uniform Crossover	2.4038
Global	**2.4588**

Table 5.28: Hypervolume Accumulated Value (enc. experiment).

5.3.3.2. Accuracy Experiment.

In the previous experiment, a thorough comparison of the performances of several encodings was conducted when the hour was employed as a time unit. Although non-significant differences among such performances were found, the best average rank from the Friedman's test was presented by the standard binary encoding. For this reason, the results achieved for the standard binary encoding are used in this second experiment in order to compare the performances when the day and the week are used as time units.

Standard Binary Encoding (Days).

The results of using the standard binary encoding with one-point, two-point and uniform crossover when the day is considered as a time unit are shown below. The computational time consumed by each one is shown in the Table 5.29.

Encoding	Time Unit	Average time	Sequential Time
Binary 1PX	Hour	2 days, 18 hours and 38 minutes	1 year, 5 months, 7 days, 16 hours and 49 minutes
Binary 2PX	Hour	2 days, 22 hours and 41 minutes	1 year, 5 months, 13 days, 1 hours and 40 minutes
Binary U	Hour	3 days, 1 hour and 5 minutes	1 year, 6 months, 28 days, 2 hours and 34 minutes
Binary 1PX	Day	2 days, 20 hours and 37 minutes	1 year, 5 months, 23 days, 6 hours and 49 minutes
Binary 2PX	Day	2 days, 19 hours and 12 minutes	1 year, 5 months, 12 days, 4 hours and 48 minutes
Binary U	Day	2 days, 19 hours and 12 minutes	1 year, 5 months, 12 days, 3 hours and 2 minutes
Binary 1PX	Week	2 days, 19 hours and 55 minutes	1 year, 5 months, 17 days, 18 hours and 19 minutes
Binary 2PX	Week	2 days, 19 hours and 44 minutes	1 year, 5 months, 16 days, 8 hours and 20 minutes
Binary U	Week	2 days, 18 hours and 27 minutes	1 year, 5 months, 6 days, 7 hours and 14 minutes
TOTAL		**25 days, 16 hours and 7 minutes**	**13 year, 2 months, 19 days, 23 hours and 4 minutes**

Table 5.29: Computational cost (consumed time, enc. experiment).

The relationship between the method configurations and the identifiers is shown in the Table 5.30. Moreover, statistical information regarding the Hypervolume value at the end of the evolutionary process is shown. For the binary encoding with one-point crossover and the day as a time unit (B1PX-D), it is possible to conclude that the configuration with identifier ID6 (population of 150 individuals and a mutation rate of 1 gene per chromosome) presents the highest Hypervolume average value, the highest Hypervolume median value and the highest Hypervolume minimum value. The configuration with identifier ID1 (population of 50 individuals and a mutation rate of 0.5 gene per chromosome) presents the highest Hypervolume maximum value. Finally, the configuration with identifier ID3 (population of 150 individuals and a mutation rate of 0.5 gene per chromosome) presents the lowest Hypervolume standard deviation value.

For the binary encoding with two-point crossover and the day as a time unit (B2PX-D), it is possible to conclude that the configuration with identifier ID7 (population of 50 individuals and a mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume average value, the highest Hypervolume median value and the highest Hypervolume minimum value. The configuration with identifier ID9 (population of 150 individuals and mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume maximum value and the configuration with identifier ID3 (population of 150 individuals and mutation rate of 0.5 gene per chromosome) presents the lowest Hypervolume standard deviation value.

Encoding	Identifier	Configuration	Average	Median	Max.	Min.	St. Deviation	Av. Rank
B1PX-D	ID1	N = 50 – PrM = 0.5	2.2871	2.2797	**2.3486**	2.2620	0.0240	5.523
	ID2	N = 100 – PrM = 0.5	2.2894	2.2866	2.3198	2.2652	0.0127	4.714
	ID3	N = 150 – PrM = 0.5	2.2830	2.2846	2.3021	2.2563	**0.0122**	5.571
	ID4	N = 50 – PrM = 1.0	2.2863	2.2843	2.3127	2.2622	0.0131	4.809
	ID5	N = 100 – PrM = 1.0	2.2902	2.2889	2.3222	2.2621	0.0181	4.571
	ID6	N = 150 – PrM = 1.0	**2.2913**	**2.2924**	2.3193	**2.2675**	0.0143	**4.380**
	ID7	N = 50 – PrM = 1.5	2.2893	2.2896	2.3299	2.2442	0.0196	4.619
	ID8	N = 100 – PrM = 1.5	2.2818	2.2801	2.3273	2.2530	0.0187	6.142
	ID9	N = 150 – PrM = 1.5	2.2889	2.2844	2.3112	2.2655	0.0140	4.666
p-Value								0.435
B2PX-D	ID1	N = 50 – PrM = 0.5	2.2879	2.2862	2.3202	2.2596	0.0156	4.714
	ID2	N = 100 – PrM = 0.5	2.2819	2.2790	2.3114	2.2609	0.0141	6.285
	ID3	N = 150 – PrM = 0.5	2.2870	2.2874	2.3092	2.2606	**0.0115**	4.428
	ID4	N = 50 – PrM = 1.0	2.2850	2.2827	2.3173	2.2554	0.0171	5.619
	ID5	N = 100 – PrM = 1.0	2.2869	2.2882	2.3188	2.2627	0.0168	5.333
	ID6	N = 150 – PrM = 1.0	2.2902	2.2923	2.3231	2.2561	0.0169	4.523
	ID7	N = 50 – PrM = 1.5	**2.2912**	**2.2932**	2.3249	**2.2693**	0.0144	**4.238**
	ID8	N = 100 – PrM = 1.5	2.2891	2.2875	2.3243	2.2658	0.0144	4.619
	ID9	N = 150 – PrM = 1.5	2.2865	2.2847	**2.3372**	2.2573	0.0180	5.238
p-Value								0.266
BUX-D	ID1	N = 50 – PrM = 0.5	2.2876	2.2891	2.3152	2.2643	0.0144	4.904
	ID2	N = 100 – PrM = 0.5	2.2913	2.2883	2.3356	2.2602	0.0162	4.714
	ID3	N = 150 – PrM = 0.5	2.2876	2.2892	2.3104	2.2522	0.0140	4.857
	ID4	N = 50 – PrM = 1.0	2.2806	2.2782	2.3078	2.2601	**0.0130**	6.428
	ID5	N = 100 – PrM = 1.0	2.2916	**2.2944**	2.3178	2.2586	0.0177	4.761
	ID6	N = 150 – PrM = 1.0	2.2914	2.2933	2.3333	**2.2666**	0.0169	4.666
	ID7	N = 50 – PrM = 1.5	2.2866	2.2866	2.3201	2.2596	0.0149	5.285
	ID8	N = 100 – PrM = 1.5	**2.2938**	2.2919	2.3279	2.2588	0.0219	4.809
	ID9	N = 150 – PrM = 1.5	2.2935	2.2877	**2.3461**	2.2565	0.0253	**4.571**
p-Value								0.500

Table 5.30: Id’s, config., Hyperv. statistics and statistical test (binary encoding - Days).

For the binary encoding with uniform crossover and the day as a time unit (BUX-D), it is possible to conclude that the configuration with identifier ID8 (population of 100 individuals and mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume average value, the configuration with identifier ID5 (population of 100 individuals and mutation rate of 1 gene per chromosome) presents the highest Hypervolume median value, the configuration with identifier ID9 (population of 150 individuals and mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume maximum value, the configuration with identifier ID6 (population of 150 individuals and mutation rate of 1 gene per chromosome) presents the highest

Hypervolume minimum value, and the configuration with identifier ID4 (population of 50 individuals and mutation rate of 1 gene per chromosome) presents the lowest Hypervolume standard deviation value.

In order to determine the best behaviour amongst the range of configurations, a statistical significance hypothesis test was conducted. The average ranks computed by using the Friedman's test and the *p*-value obtained are shown in the Table 5.30. The configuration with identifier ID6 (population of 150 individuals and a mutation rate of 1 gene per chromosome) presents the best average rank for the binary encoding with one-point crossover and the day as a time unit. Moreover, the *p*-value obtained of 0.435 explains that the null Hypothesis cannot be rejected so, in this case, no one configuration performs better than any other one.

Regarding the binary encoding with two-point crossover and the day as a time unit, the configuration ID7 (population of 50 individuals and a mutation rate of 1.5 gene per chromosome) presents the best average rank. Moreover, the *p*-value obtained of 0.266 explains that the null Hypothesis cannot be rejected so, no one configuration performs better than any other one. Finally, it can be seen that the configuration with identifier ID9 (population of 150 individuals and a mutation rate of 1.5 gene per chromosome) presents the best average rank for the binary encoding with uniform crossover and the day as a time unit. Furthermore, the *p*-value obtained of 0.500 explains that the null Hypothesis cannot be rejected so, no one configuration performs better than any other one. From each case, the best configurations were selected for the final comparison study between the accuracy level encodings.

The best accumulated non-dominated solutions obtained from the final generation for all executions and all configurations were used to compute the accumulated Hypervolume whose values were 2.4112, 2.4073 and 2.4319 for the binary encoding with one-point, two-point and uniform crossover with the day as a time unit, respectively. As expected, the values are higher than 2.3486, 2.3372 and 2.3461, the maximum values shown in the Table 5.30, respectively.

Standard Binary Encoding (Weeks).

The results of using the standard binary encoding with one-point, two-point and uniform crossover and the week as a time unit are shown below. The computational time consumed by each one is shown in the Table 5.29. The relationship between the method configurations and the identifiers is shown in the Table 5.31. Statistical information regarding the Hypervolume value at the end of the process is also shown. For the binary encoding with one-point crossover and the week as a time unit (B1PX-W), it is possible to conclude that the configuration with identifier ID1 (population of 50 individuals and a mutation rate of 0.5 gene per chromosome) presents the highest Hypervolume average value and the highest Hypervolume maximum value. The configuration with identifier ID4 (population of 50 individuals and a mutation rate of 1 gene per chromosome) presents the highest Hypervolume median value. Finally, the configuration with identifier ID5 (population of 100 individuals and mutation rate of 1 gene per chromosome) presents both the highest Hypervolume minimum value and the lowest Hypervolume standard deviation value.

For the binary encoding with two-point crossover and the week as a time unit (B2PX-W), it is possible to conclude that the configuration with identifier ID9 (population of 150 individuals and a mutation rate of 1.5 gene per chromosome) presents both the highest Hypervolume average value and the highest Hypervolume minimum value. The configuration with identifier ID3 (population of 150 individuals and a mutation rate of 0.5 gene per chromosome) presents the highest Hypervolume median value. The configuration with identifier ID5 (population of 100 individuals and mutation rate of 1 gene per chromosome) presents the highest Hypervolume maximum value. Finally, the configuration with identifier ID4 (population of 50 individuals and a mutation rate of 1 gene per chromosome) presents the lowest Hypervolume standard deviation value.

For the binary encoding with uniform crossover and the week as a time unit (BUX-W), it is possible to conclude that the configuration with identifier ID8 (population of 100 individuals and a mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume average value, the highest Hypervolume median value and the highest Hypervolume minimum value. The configuration with identifier ID9 (population of 150 individuals and a mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume minimum value and the configuration with identifier ID4 (population of 50 individuals and a mutation rate of 1 gene per chromosome) presents the lowest Hypervolume standard deviation value.

Encoding	Identifier	Configuration	Average	Median	Max.	Min.	St. Deviation	Av. Rank
B1PX-W	ID1	N = 50 – PrM = 0.5	**2.2930**	2.2911	**2.3430**	2.2561	0.0197	**4.571**
	ID2	N = 100 – PrM = 0.5	2.2896	2.2915	2.3205	2.2636	0.0154	4.857
	ID3	N = 150 – PrM = 0.5	2.2891	2.2869	2.3259	2.2623	0.0186	4.761
	ID4	N = 50 – PrM = 1.0	2.2862	**2.2932**	2.3117	2.2520	0.0149	5.095
	ID5	N = 100 – PrM = 1.0	2.2906	2.2897	2.3207	**2.2690**	**0.0116**	4.619
	ID6	N = 150 – PrM = 1.0	2.2854	2.2839	2.3143	2.2638	0.0131	5.761
	ID7	N = 50 – PrM = 1.5	2.2880	2.2872	2.3110	2.2595	0.0154	5.380
	ID8	N = 100 – PrM = 1.5	2.2873	2.2884	2.3184	2.2559	0.0171	4.952
	ID9	N = 150 – PrM = 1.5	2.2899	2.2887	2.3249	2.2634	0.0155	5.000
p-Value								0.921
B2PX-W	ID1	N = 50 – PrM = 0.5	2.2882	2.2876	2.3225	2.2654	0.0163	4.809
	ID2	N = 100 – PrM = 0.5	2.2868	2.2839	2.3145	2.2501	0.0154	5.238
	ID3	N = 150 – PrM = 0.5	2.2921	**2.2972**	2.3237	2.2567	0.0203	**4.380**
	ID4	N = 50 – PrM = 1.0	2.2819	2.2750	2.3103	2.2633	**0.0133**	5.999
	ID5	N = 100 – PrM = 1.0	2.2877	2.2852	**2.3578**	2.2632	0.0200	5.095
	ID6	N = 150 – PrM = 1.0	2.2905	2.2839	2.3500	2.2502	0.0211	4.761
	ID7	N = 50 – PrM = 1.5	2.2863	2.2826	2.3465	2.2546	0.0198	5.523
	ID8	N = 100 – PrM = 1.5	2.2877	2.2853	2.3198	2.2546	0.0160	4.666
	ID9	N = 150 – PrM = 1.5	**2.2923**	2.2861	2.3297	**2.2679**	0.0164	4.523
p-Value								0.643
BUX-W	ID1	N = 50 – PrM = 0.5	2.2938	2.2931	2.3336	2.2659	0.0176	4.428
	ID2	N = 100 – PrM = 0.5	2.2868	2.2893	2.3144	2.2659	0.0132	5.047
	ID3	N = 150 – PrM = 0.5	2.2912	2.2938	2.3216	2.2647	0.0158	4.333
	ID4	N = 50 – PrM = 1.0	2.2857	2.2863	2.3057	2.2579	**0.0119**	5.238
	ID5	N = 100 – PrM = 1.0	2.2849	2.2847	2.3144	2.2620	0.0139	6.142
	ID6	N = 150 – PrM = 1.0	2.2888	2.2898	2.3162	2.2645	0.0146	4.904
	ID7	N = 50 – PrM = 1.5	2.2871	2.2873	2.3256	2.2511	0.0179	5.476
	ID8	N = 100 – PrM = 1.5	**2.2955**	**2.2940**	2.3316	**2.2682**	0.0164	**4.142**
	ID9	N = 150 – PrM = 1.5	2.2885	2.2841	**2.3388**	2.2632	0.0183	5.285
p-Value								0.348

Table 5.31: Id’s, config., Hyperv. statistics and statistical test (binary encoding - Weeks).

In order to establish the best behaviour amongst configurations, a statistical significance hypothesis test was conducted. The average ranks computed by employing the Friedman’s test and the *p*-values obtained are shown in the Table 5.31. It can be seen that the configuration with identifier ID1 (population of 50 individuals and a mutation rate of 0.5 gene per chromosome) presents the best average rank for the binary encoding with one-point crossover and the week as a time unit. The configuration with identifier ID3 (population of 150 individuals and mutation rate of 0.5 gene per chromosome) presents the best average rank for the binary encoding with two-point crossover and the week as a time unit. Finally, the configuration with identifier ID8 (population of 100 individuals and mutation rate of 1.5 gene per chromosome) presents the best average rank for the binary encoding with uniform crossover and the week as a time unit. The respective *p*-values of 0.921, 0.643 and 0.348 claim that the null Hypothesis cannot be rejected so, no one configuration performs better than any other one. These configurations were selected for the final comparison study of the accuracy-level configurations.

The best accumulated non-dominated solutions obtained from the final generation, all executions and all configurations were used to compute the accumulated Hypervolume whose values were 2.4188, 2.4425 and 2.4163 for the binary encoding with one point, two point and uniform crossover with the week as a time unit, respectively. As expected, the values are higher than 2.3430, 2.3578 and 2.3388, the maximum values shown in the Table 5.31, respectively.

Comparing Accuracy-Level Configurations.

The global computational time consumed is shown in the Table 5.29. Previously, the configurations with the best average rank according to the Friedman's test were selected in order to be globally compared. These configurations are shown in the Table 5.32.

Identifier	Description	Configuration	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	Binary 1P (hour)	N = 100 – PrM = 1.5	2.2943	2.3003	2.3126	2.2640	0.0153	4.809
ID2	Binary 2P (hour)	N = 100 – PrM = 0.5	**2.2975**	2.2961	2.3436	**2.2714**	0.0220	**4.571**
ID3	Binary U (hour)	N = 100 – PrM = 1.0	2.2972	2.2938	2.3332	2.2660	0.0218	4.904
ID4	Binary 1P (day)	N = 150 – PrM = 1.0	2.2913	2.2924	2.3193	2.2675	**0.0143**	5.380
ID5	Binary 2P (day)	N = 50 – PrM = 1.5	2.2912	2.2932	2.3249	2.2693	0.0144	5.238
ID6	Binary U (day)	N = 150 – PrM = 1.5	2.2935	2.2877	**2.3461**	2.2565	0.0253	5.285
ID7	Binary 1P (week)	N = 50 – PrM = 0.5	2.2930	2.2911	2.3430	2.2561	0.0197	4.952
ID8	Binary 2P (week)	N = 150 – PrM = 0.5	2.2921	**2.2972**	2.3237	2.2567	0.0203	5.047
ID9	Binary U (week)	N = 100 – PrM = 1.5	2.2955	2.2940	2.3316	2.2682	0.0164	4.809
p-Value								0.991

Table 5.32: Id's, config., Hyperv. statistics and statistical test (acc. experiment).

The Hypervolume average values evolution versus the evaluations number is shown

In the Figure 5.47. The detail for the final evaluations (last million fitness function evaluations, from 9 to 10 million) is shown in the Figure 5.48. It can be seen that the configuration with identifier ID2 (with Binary encoding, two-point crossover, the hour as a time unit, population of 100 individuals and a mutation rate of 0.5 gene per chromosome) reaches the highest Hypervolume average value.

Box plots of the Hypervolume values distribution at the end of the process are shown in the Figure 5.49. They are ordered from the left to the right in relation to the time units of hours (H), days (D) and weeks (W) and crossover types of one-point (1PX), two-point (2PX) and uniform crossover (UX). It can be seen that the medians are ordered from the biggest one to smallest one when the hour is considered as a time unit. In this case, the greater the accuracy the bigger the Hypervolume median value. This effect cannot be seen for the rest of time units.

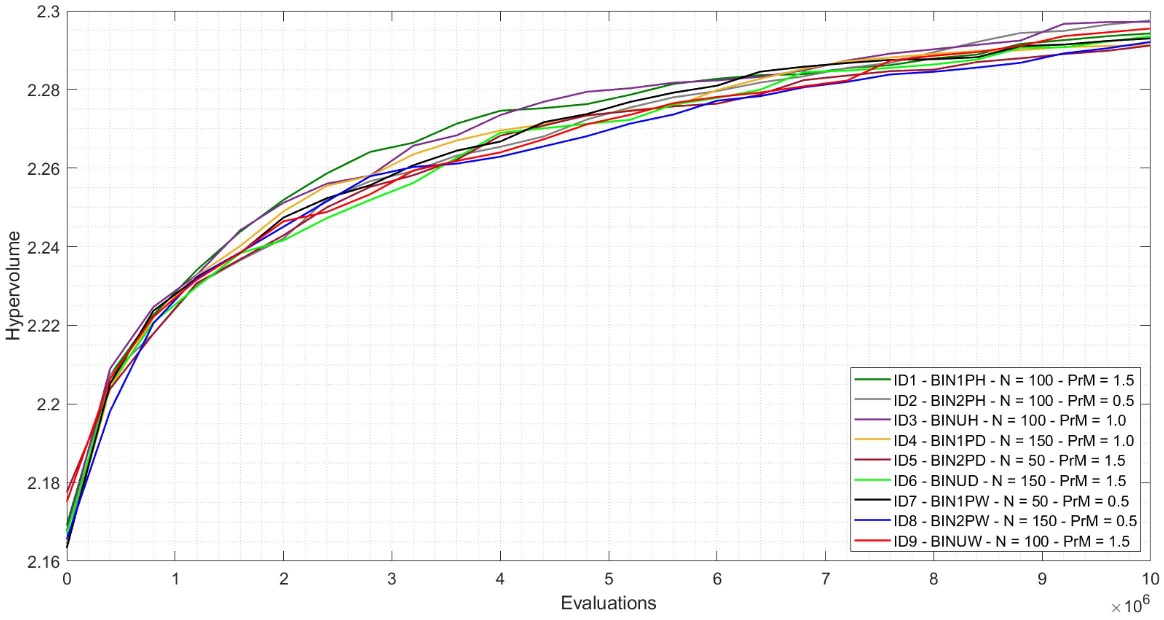


Figure 5.47: Hypervolume Average vs. evaluations (acc. experiment).

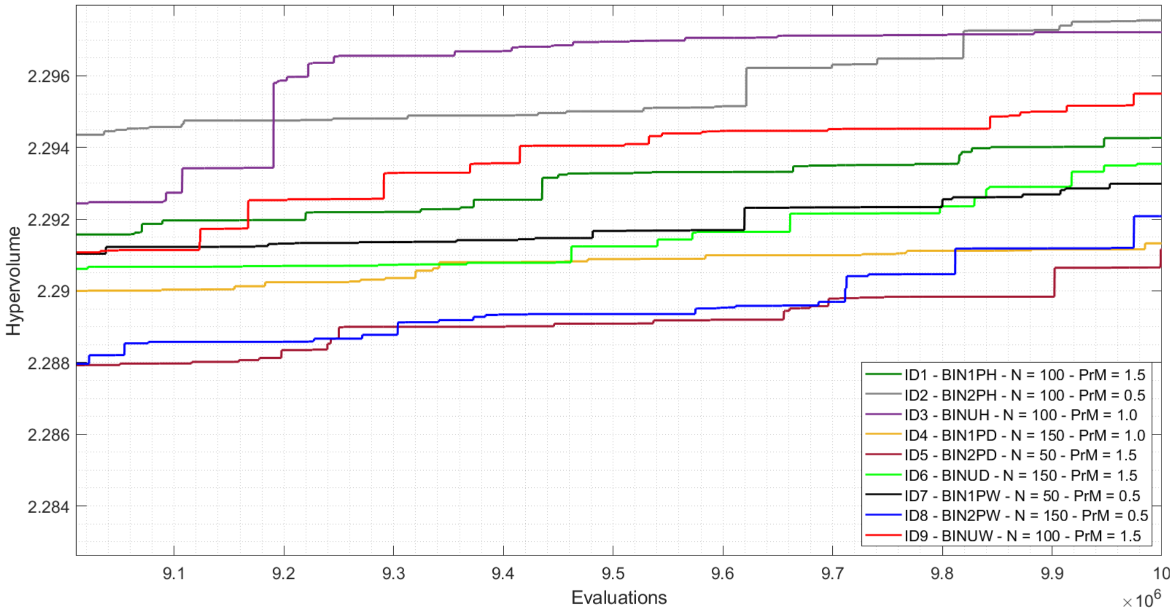


Figure 5.48: Hypervolume Average vs. evaluations, detail (acc. experiment).

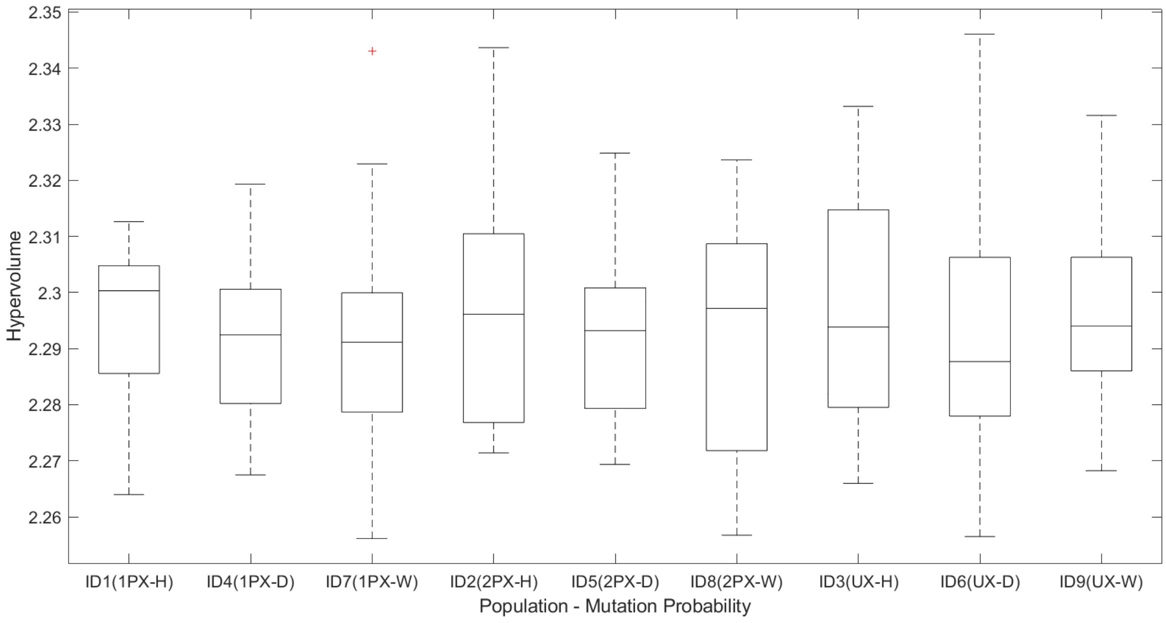


Figure 5.49: Box plots of the final Hypervolume (acc. experiment).

Statistical information in relation to the Hypervolume value at the end of the evolutionary process is shown in the Table 5.32. It can be seen that the configuration with identifier ID2 (with binary encoding, two-point crossover and the hour as a time unit, population of 100 individuals and a mutation rate of 0.5 gene per chromosome) presents both the highest Hypervolume average value and the highest Hypervolume minimum value, the configuration with identifier ID8 (with binary encoding, two-point crossover and the week as a time unit, population of 100 individuals and a mutation rate of 0.5 gene per chromosome) presents the highest Hypervolume median value. The configuration with identifier ID6 (with binary encoding, uniform crossover and the day as a time unit, population of 150 individuals and a mutation rate of 1.5 gene per chromosome) presents the highest Hypervolume minimum value. Finally, the configuration with identifier ID4 (with binary encoding, one point crossover and the day as a time unit, population of 150 individuals and a mutation rate of 1 gene per chromosome) presents the lowest Hypervolume standard deviation.

In order to determine if one of the configurations performs better than any other, a statistical significance hypothesis test was conducted. The average ranks computed by using the Friedman's test are shown in the Table 5.32. It can be seen that the configuration with identifier ID2 (with Binary encoding, two-point crossover and the hour as a time unit, population of 100 individuals and a mutation rate of 0.5 gene per chromosome) presents the best average rank. However, the *p*-value computed (0.991) implies that the null hypothesis (H_0) cannot be rejected (*p*-value > 0.05), so it is possible to conclude that, in the studied conditions, there is no configuration that performs better than any other.

The best accumulated non-dominated solutions obtained were used to compute the accumulated Hypervolume, whose value was 2.4724. As expected, the value is higher than 2.3461, the maximum accumulated value obtained after the evolutionary process for the standard binary encoding with uniform crossover and the day as a time unit. This is shown in the Table 5.33.

Encoding	Time Unit	Hypervolume Accumulated Value
Binary 1 Point Crossover	Hour	2.4227
Binary 2 Point Crossover	Hour	2.4378
Binary Uniform Crossover	Hour	2.4134
Binary 1 Point Crossover	Day	2.4112
Binary 2 Point Crossover	Day	2.4073
Binary Uniform Crossover	Day	2.4319
Binary 1 Point Crossover	Week	2.4188
Binary 2 Point Crossover	Week	2.4425
Binary Uniform Crossover	Week	2.4163
Global		**2.4724**

Table 5.33: Hypervolume Accumulated Value (acc. experiment).

5.3.3.3. Accumulated Non-Dominated Set of Designs.

The non-dominated solutions to the problem provided at the end of the evolutionary process for all executions, all configurations, all encodings and time units are shown in the Figure 5.50. All optimum solutions belonging to the achieved non-dominated front are shown in the Table 5.34. Unavailability (Q) is shown as a fraction, Cost is shown in economic units and the rest of the variables represent, for the respective devices, the optimum time to start a preventive maintenance activity with the hour, day or week as a time unit.

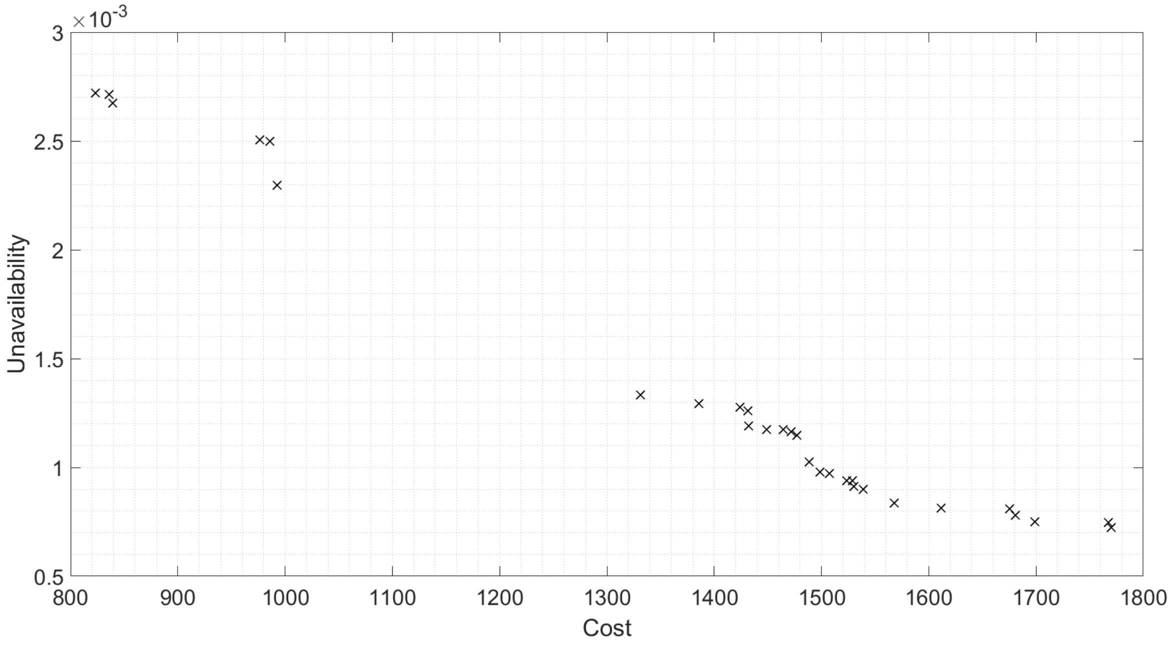


Figure 5.50: Non-dominated front.

Id	Q	Cost [e.u.]	Time Unit	V_1 [h]	P_2 [h]	P_3 [h]	V_4 [h]	V_5 [h]	V_6 [h]	V_7 [h]
1	0.002720	823.38	Hours	25408	0	8633	0	34179	34903	31386
2	0.002713	835.75	Hours	29225	0	8633	0	27070	33454	33690
3	0.002673	839.12	Weeks	206	0	50	0	200	140	151
4	0.002506	976.12	Weeks	200	0	51	160	165	182	168
5	0.002500	986.12	Hours	22388	0	6022	23776	17397	26730	22354
6	0.002295	992.75	Days	1435	0	350	830	1088	1459	1454
7	0.001334	1363.75	Days	1394	360	315	0	1125	1301	1026
8	0.001294	1385.50	Hours	30287	6093	8344	0	34503	15445	31903
9	0.001276	1424.25	Weeks	204	50	37	0	174	173	188
10	0.001260	1431.12	Hours	27536	8658	7984	0	32322	25055	32113
11	0.001189	1431.75	Hours	31040	8617	8103	0	34787	31445	29929
12	0.001174	1449.00	Weeks	178	51	50	0	182	171	195
13	0.001173	1464.38	Hours	27019	5942	5844	0	26998	24364	27915
14	0.001164	1471.62	Hours	29879	8537	8599	0	32718	33261	30100
15	0.001149	1477.12	Days	1364	353	363	0	1198	1435	1458
16	0.001026	1488.75	Hours	32511	7945	8594	9234	31316	30834	31234
17	0.000977	1498.50	Hours	32045	8752	8415	13235	20781	31501	29254
18	0.000973	1507.50	Hours	34928	7929	7908	11683	24182	34593	26275
19	0.000940	1523.38	Hours	30443	8130	8462	29999	34282	34286	34317
20	0.000939	1528.88	Hours	29939	7028	7904	21690	25711	34814	34791
21	0.000913	1530.38	Weeks	194	47	49	103	137	145	179
22	0.000898	1538.62	Weeks	131	42	49	186	156	200	152
23	0.000835	1567.75	Days	1028	363	340	1280	1430	986	1326
24	0.000813	1611.38	Hours	22516	5955	6298	17237	29568	27075	22757
25	0.000808	1675.38	Weeks	200	41	50	144	149	167	173
26	0.000781	1680.38	Hours	28624	8245	8698	27171	21444	29197	21351
27	0.000749	1698.62	Hours	28281	8338	7367	19399	17608	32848	31220
28	0.000748	1767.50	Weeks	207	44	47	107	83	201	161
29	0.000725	1770.12	Hours	30813	7371	8453	29958	16345	30776	25358

Table 5.34: Non-dominated solutions.

The solution with the lowest Cost (ID1) (823.38 economic units) represents the biggest Unavailability (0.002720). These values are followed by periodic optimum times (using the hour as a time unit in this case) measured from the moment in which the system's mission time starts (time to perform a preventive maintenance task is not included). For the solution ID1, it can be seen that periodic optimum times to start a preventive maintenance task (TM) for the devices P2 and V4 are not supplied. This is because the design alternative does not include such devices. The opposite case shows the biggest Cost (ID26) (1770.12 economic units) and the lowest Unavailability (0.000725). For the solution ID29, periodic optimum times to start a preventive maintenance task (TM) are supplied for all devices. This is because the design alternative includes the devices P2 and V4. Other optimum solutions were found in these two solutions and can be seen in the Table 5.34. The decision makers should decide which is the preferable design for them, considering their individual requirements.

Moreover, the solutions were clustered in the Figure 5.51 according to their final design. Such solutions are shown from left to right and in ascending order in relation to the Cost from ID1 to ID29. The solutions contained in the Cluster 1 (the solutions 1 to 3, see also the Table 5.34) are the solutions in which non-redundant devices were included in the design. In this case, the system contains exclusively devices placed in series. These solutions present the lowest Cost and the biggest Unavailability. The solutions contained in the Cluster 2 (the solutions 4 to 6, see also the Table 5.35) are the solutions in which a redundant valve was included in the design as a parallel device. These solutions present a bigger Cost and a lower Unavailability than the solutions contained in the Cluster 1. The solutions contained in the Cluster 3 (the solutions 7 to 15, see also the Table 5.34) are the solutions in which a redundant pump was included in the design as a parallel device. These solutions present a higher Cost and a lower Unavailability than the solutions contained in both the Clusters 1 and 2. Finally, the solutions contained in the Cluster 4 (the solutions 16 to 29, see also the Table 5.34) are the solutions in which both a redundant valve and a redundant pump were included in the design as parallel devices. These solutions present the biggest Cost and the lowest Unavailability.

The best accumulated non-dominated solutions obtained were used to compute the accumulated Hypervolume, whose value was 2.4738. As it is expected, the value is higher than the rest of the maximum accumulated values obtained after the evolutionary process for the encoding experiment and the accuracy experiment. This is shown in the Table 5.35.

Experiment	**Hypervolume Accumulated Value**
Encoding	2.4588
Accuracy	2.4724
Total	**2.4738**

Table 5.35: Hypervolume Accumulated Value.

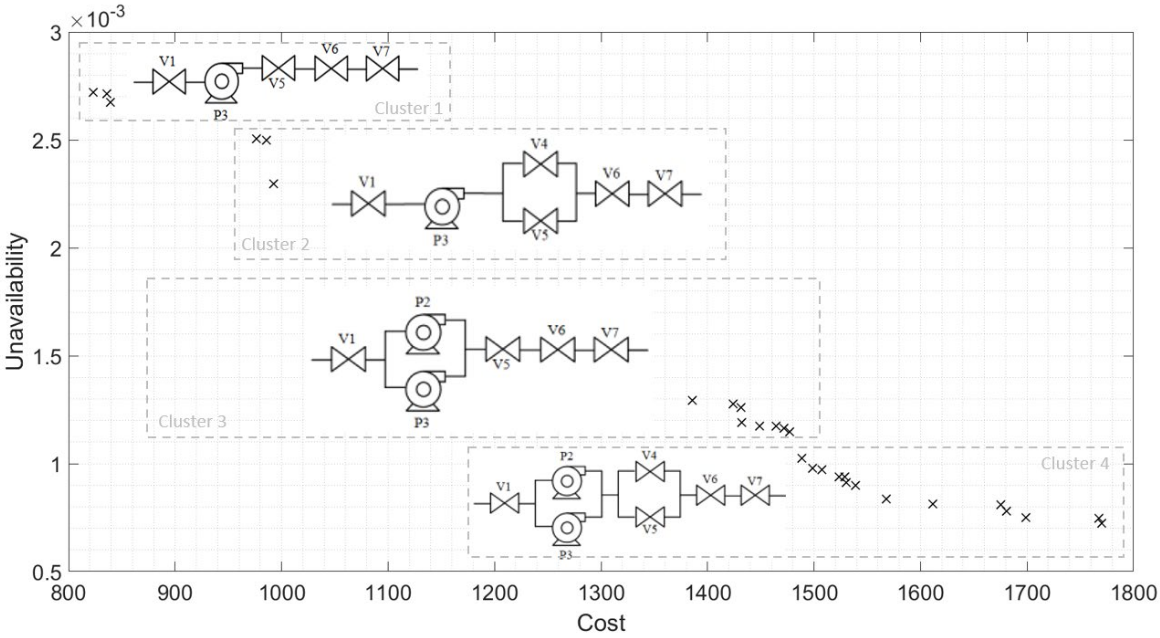


Figure 5.51: Design alternatives.

5.4. Testing Multi-objectivisation.

5.4.1. Background.

Once the methodology was explored by attending to several Multi-objective Evolutionary Algorithms, encodings and accuracy levels in relation to the employed time unit, the effect of applying “multi-objectivisation” is evaluated for the case study previously presented and studied. In this case, Multi-objective Evolutionary Algorithms and Discrete Simulation are coupled while indicator-based and dominance-based state-of-the-art multi-objective optimisers are used. Such multi-objective optimisers achieved the first orders from the Friedman’s test point of view when they were explored on the first study. Moreover, both real and binary encoding (with two-point crossover due to the results achieved from the second study) are tested. Two optimisation approaches are developed, explored and thoroughly compared by using the analysed case study along the present research; a two-objective approach, and a three-objective approach, which attends to the multi-objectivisation concept.

5.4.2. Case study.

The case study was presented in the Section 5.1. However, in this case, the conflicting objectives are Availability, Operational Cost and Acquisition Cost. Therefore, the Equations 4.2 and 4.4 from the Chapter 4 are used in order to compute such objectives. As it was exposed above, two optimisation approaches are tested. When the three-objective approach is employed, the Operational Cost and the Acquisition Cost are decomposed from the Equation 4.4 presented in the Chapter 4. The parameters used to configure the methods for the evolutionary process are shown in the Table 5.36.

Description	Setting
Methods	NSGAII & SMS-EMOA
Encoding	Real & Binary
Population	150
Mutation rate	0.5 - 1.0 - 1.5
Mutation distribution index	20
Crossover probability	1.0
Crossover distribution index	20
Number of evaluations	10.000.050
Number of executions	21

Table 5.36: Parameters configuration (multi-objectivisation experiment).

The population size of 150 individuals is employed. Six different configurations for both methods are simulated and 21 executions per configuration are conducted for statistical purposes. As a stopping criterion, a number of 10,000,000 evaluations of the objective functions is employed. To normalise the value of the objective functions, scale factors were needed. Such values were achieved by computing the objective functions at the beginning of the optimisation process. Again, this approach is based on considering that such values will enhance along the evolutionary process. When the two-objective problem is considered, the scale factors are:

- When the Cost is computed, the employed scale factor is 4,500 economic units,
- when the Unavailability of the system is computed, the considered scale factor is 0.03.

When the three-objective problem is considered:

- The scale factor of 4,500 economic units is employed when the Operational Cost is computed,
- the employed scale factor of 1,875 economic units is used when the Acquisition Cost is computed,
- the scale factor of 0.03 is considered when the Unavailability of the system is computed.

Finally, depending on the number of objectives to consider, a two- or three-dimensional reference point must be chosen to compute the Hypervolume indicator. Such points must cover the points limited by the scale factors, which normalise the values of the objective functions until a maximum value of a unit. The reference points were set to (2,2) and (2,2,2) respectively. Again, the open-source Software Platform PlatEMO was used to optimise the case study.

5.4.3. Results and discussion.

5.4.3.1. Two-objective approach.

The computational time consumed, when the High-Performance Computer was used, is shown in the Table 5.37. The relationship between the methods, the configurations and the identifiers can be seen in the Table 5.38 (columns 1 to 4).

Method	Encoding	Average time	Sequential Time
NSGAII	Real	2 days, 20 hours and 55 minutes	5 months, 28 days, 20 hours and 26 minutes
NSGA-II	Binary 2-point crossover	2 days, 21 hours and 55 minutes	6 months, 1 day, 1 hour and 4 minutes
SMS-EMOA	Real	5 days, 23 hours and 37 minutes	1 year, 12 days and 34 minutes
SMS-EMOA	Binary 2-point crossover	6 days, 3 hours and 47 minutes	1 year, 6 days, 3 hours and 47 minutes
TOTAL			**3 years, 1 month, 4 days and 1 hour**

Table 5.37: Computational cost (consumed time, 2-obj. app.).

Identifier	Method	Encoding	Mutation	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	NSGA-II	Real	0.5	**2.4628**	**2.4617**	2.5009	**2.4382**	0.0156	5.666
ID2	NSGA-II	Real	1.0	2.4611	2.4643	2.4793	2.4399	0.0113	5.952
ID3	NSGA-II	Real	1.5	2.4606	2.4601	2.4851	2.4411	0.0118	6.285
ID4	NSGA-II	Binary	0.5	2.4590	2.4582	2.4841	2.4386	0.0106	7.095
ID5	NSGA-II	Binary	1.0	2.4576	2.4564	2.4781	2.4401	**0.0096**	6.714
ID6	NSGA-II	Binary	1.5	2.4573	2.4568	2.4789	2.4428	0.0106	6.904
ID7	SMS-EMOA	Real	0.5	2.4592	2.4587	2.4859	2.4423	0.0102	6.380
ID8	SMS-EMOA	Real	1.0	2.4581	2.4578	2.4871	2.4418	0.0116	6.476
ID9	SMS-EMOA	Real	1.5	2.4582	2.4576	2.4840	2.4419	0.0112	6.714
ID10	SMS-EMOA	Binary	0.5	2.4587	2.4575	2.4806	2.4408	0.0097	6.523
ID11	SMS-EMOA	Binary	1.0	**2.4628**	2.4615	**2.5066**	2.4347	0.0165	**5.619**
ID12	SMS-EMOA	Binary	1.5	2.4555	2.4542	2.4717	2.4393	0.0112	7.666
p-Value									0.8636

Table 5.38: Id’s, config., Hyperv. statistics and statistical test (2-obj. app.).

The Figure 5.52 shows the Hypervolume average values evolution regarding the number of evaluations. The evolution from 9 to 10 million evaluations (the end of the process) is shown in the Figure 5.53.

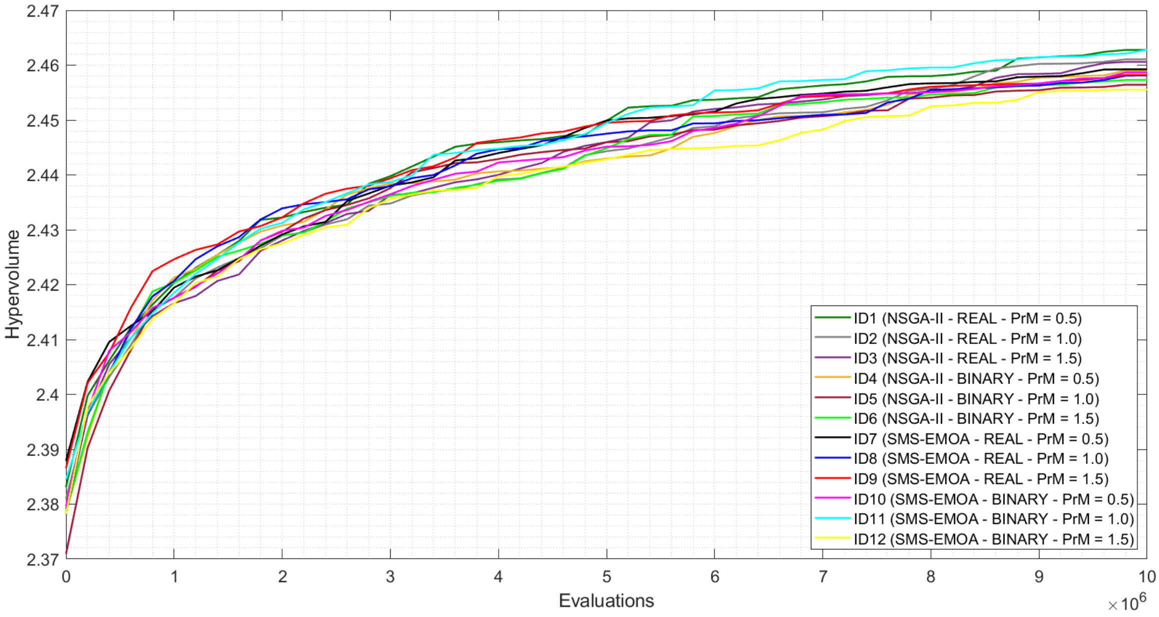


Figure 5.52: Hypervolume average vs. evaluations (2-obj. app.).

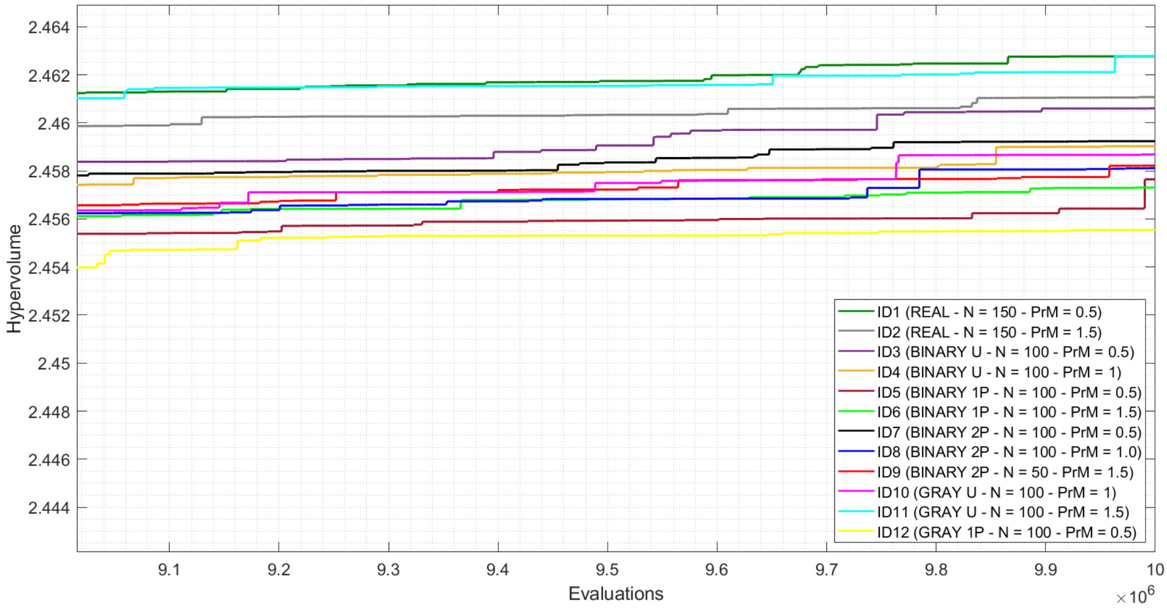


Figure 5.53: Hypervolume average vs. evaluations, detail (2-obj. app.).

Both the configuration with identifier ID1 (NSGA-II, real encoding and 0.5 gene per chromosome as a mutation rate) and the configuration with identifier ID11 (SMS-EMOA, binary encoding and 1.0 gene per chromosome as a mutation rate) present the best Hypervolume average value.

In the Figure 5.54, box plots of the Hypervolume values distribution are shown. Such box plots summarise the statistical information supplied by the Table 5.38 (columns 5 to 9). The configuration ID1 (NSGA-II, real encoding and 0.5 gene per chromosome as a mutation rate) achieves the best Hypervolume average, median and minimum values. The configuration ID11 (SMS-EMOA, binary encoding and 1 gene per chromosome as a mutation rate) achieves the same Hypervolume average value and the best Hypervolume minimum value. Finally, the configuration ID5 (NSGA-II, binary encoding and 1 gene per chromosome as a mutation rate) achieves the lowest standard deviation value.

In order to conclude if any configuration reaches a better performance, a statistical hypothesis test was conducted. The Friedman's test was employed to compute the average ranks, which are shown in the Table 5.38 (column 10). The configuration

ID11 (SMS-EMOA, binary encoding and 1 gene per chromosome as a mutation rate) reaches the best average rank. However, the *p*-value computed (0.8636) does not allow rejecting H_0 (*p*-value < 0.05). Therefore, it is not possible to establish that any configuration performs better than any other.

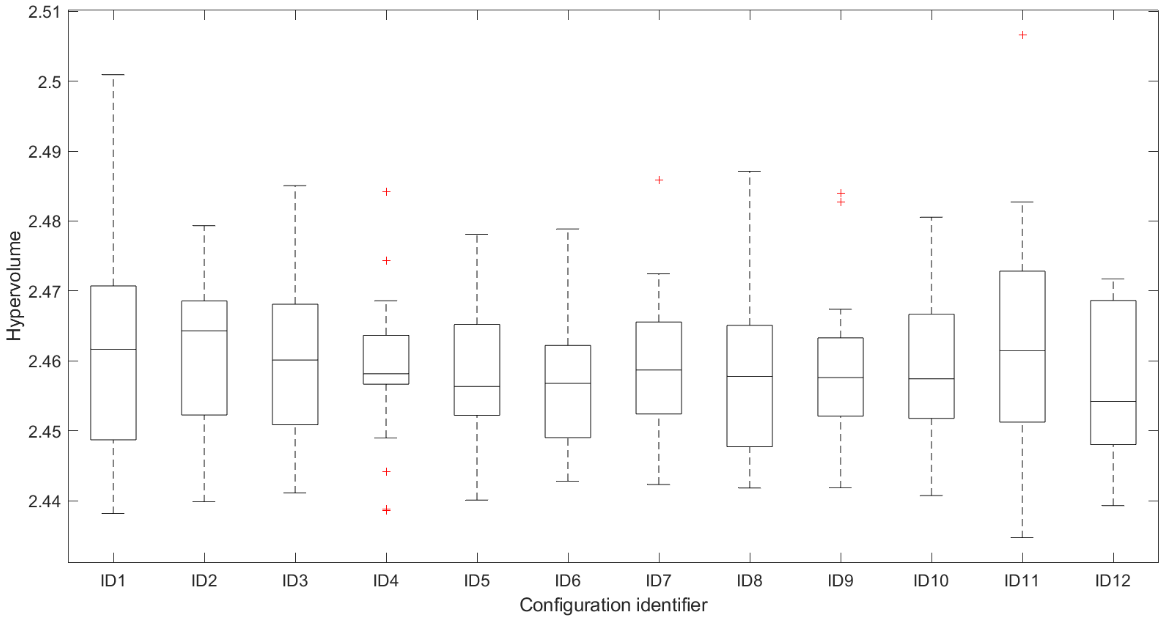


Figure 5.54: Hypervolume box plots, identifiers as in the table 5.38 (2-obj. app.).

The Figure 5.55 shows the set of non-dominated solutions achieved regarding all executions, configurations and methods. The detail of such solutions is shown in the Table 5.39, where they are ordered from the left to the right side of the figure (solutions with worse Unavailability are firstly ordered). The Unavailability (Q), the Cost and the optimum times to start a scheduled preventive maintenance task for each device are displayed in such a table. It can be seen that the solutions without redundant devices (marked as O in the Figure 5.55 and with identifiers Id1 to Id2 in the Table 5.39) are the more economic and less reliable solutions. Conversely, the solutions with both a pump and a valve as redundant devices (marked as □ in the Figure 5.55 and with identifiers Id16 to Id24 in the Table 5.39) are the more expensive and reliable solutions, as it is expected. The solutions with at least one redundant device are located between the described ones.

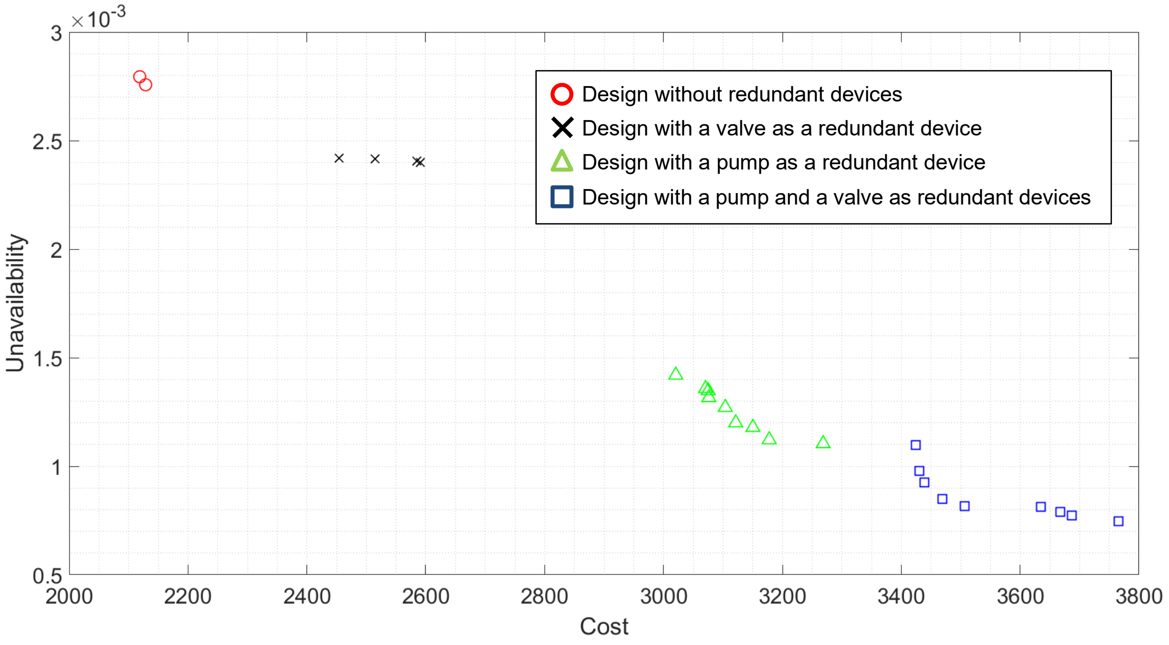


Figure 5.55: Accumulated non-dominated front (2-obj. app.).

Id	Q	Cost [e.u.]	V_1 [h]	P_2 [h]	P_3 [h]	V_4 [h]	V_5 [h]	V_6 [h]	V_7 [h]
1	0.002794	2118.75	32081	0	8268	0	32608	29911	22838
2	0.002757	2128.75	32867	0	8738	0	33767	28717	33847
3	0.002419	2454.12	34178	0	8471	22544	22477	34758	29997
4	0.002417	2514.50	31168	0	8640	24997	26369	21853	18347
5	0.002404	2585.25	27157	0	8759	25247	29928	32602	33711
6	0.002400	2591.38	33394	0	7624	27291	33090	33453	34139
7	0.001420	3020.62	30166	8752	8592	0	19308	30349	27921
8	0.001358	3070.75	27879	7766	8516	0	30022	34692	31346
9	0.001348	3075.25	32005	8218	6237	0	32457	28160	31568
10	0.001316	3076.25	30348	7793	8637	0	33060	20156	30396
11	0.001271	3103.88	29309	7856	8585	0	31712	29781	25606
12	0.001200	3121.50	27780	7589	7849	0	33659	35040	31753
13	0.001179	3150.37	33444	7785	7487	0	26641	33341	29593
14	0.001122	3177.88	33180	7576	8374	0	28795	32242	32651
15	0.001104	3268.62	28318	6858	8569	0	29571	32477	28891
16	0.001097	3424.00	34961	6689	8677	25265	21118	34641	29384
17	0.000979	3429.88	27512	8167	8198	32671	32290	34720	32839
18	0.000926	3438.38	24535	8396	8192	10411	32448	34686	33434
19	0.000850	3469.50	32166	8202	8686	31625	28822	29127	34733
20	0.000816	3507.00	30163	8443	7902	19796	34310	30607	18931
21	0.000812	3635.00	29820	7951	8679	13862	33215	29994	27626
22	0.000791	3667.62	27172	8697	8756	10660	19336	34773	32893
23	0.000773	3686.38	32641	8488	8687	10351	26703	31875	34949
24	0.000748	3765.50	27132	7088	7915	10849	14859	33567	34813

Table 5.39: Non-dominated solutions (2-obj. app.).

Finally, the accumulated Hypervolume value (computed as it was described by Fonseca et al. [211]) reaches the value 2.5425. As it is expected, such a value is higher than 2.5066, the maximum value that is displayed in the Table 5.38.

5.4.3.2. Three-objective approach.

The computational time consumed is shown in the Table 5.40.

Method	Encoding	Average time	Sequential Time
NSGAII	Real	3 days, 19 hours and 39 minutes	7 months, 27 days, 16 hours and 27 minutes
NSGA-II	Binary 2-point crossover	3 days, 20 hours and 58 minutes	8 months, 17 hour and 25 minutes
SMS-EMOA	Real	6 days, 22 hours and 50 minutes	1 year, 2 months, 12 days, 3 hours and 34 minutes
SMS-EMOA	Binary 2-point crossover	7 days, 3 hours and 3 minutes	1 year, 2 moths, 25 days, 6 hours and 21 minutes
TOTAL			**3 years, 9 months, 2 days and 21 hours**

Table 5.40: Computational cost (consumed time, 3-obj. app.).

The relationship between the methods, the configurations and the identifiers can be seen in the Table 5.41 (columns 1 to 4).

Identifier	Method	Encoding	Mutation	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	NSGA-II	Real	0.5	3.7326	3.7304	3.7649	3.7059	0.0151	6.190
ID2	NSGA-II	Real	1.0	3.7314	3.7262	3.7709	3.6988	0.0194	7.333
ID3	NSGA-II	Real	1.5	**3.7437**	**3.7426**	**3.8338**	**3.7153**	0.0251	**4.571**
ID4	NSGA-II	Binary	0.5	3.7276	3.7264	3.7617	3.6989	0.0176	7.571
ID5	NSGA-II	Binary	1.0	3.7277	3.7275	3.7539	3.6979	0.0147	7.476
ID6	NSGA-II	Binary	1.5	3.7281	3.7269	3.7601	3.6977	0.0138	7.571
ID7	SMS-EMOA	Real	0.5	3.7367	3.7322	3.7770	3.7046	0.0215	6.428
ID8	SMS-EMOA	Real	1.0	3.7382	3.7324	3.7860	3.7142	0.0171	5.619
ID9	SMS-EMOA	Real	1.5	3.7349	3.7327	3.7665	3.7049	0.0147	5.904
ID10	SMS-EMOA	Binary	0.5	3.7307	3.7331	3.7772	3.7005	0.0199	6.619
ID11	SMS-EMOA	Binary	1.0	3.7324	3.7247	3.7715	3.7128	0.0186	7.095
ID12	SMS-EMOA	Binary	1.5	3.7367	3.7372	3.7608	3.7101	**0.0126**	5.619
p-Value									0.1332

Table 5.41: Id's, config., Hyperv. statistics and statistical test (3-obj. app.).

The Figure 5.56 shows the Hypervolume average values evolution regarding the number of evaluations. The evolution from 9 to 10 million evaluations (the end of the process) is shown in the Figure 5.57. The configuration with identifier ID3 (NSGA-II, real encoding and 1.5 gene per chromosome as a mutation rate) presents the best Hypervolume average value.

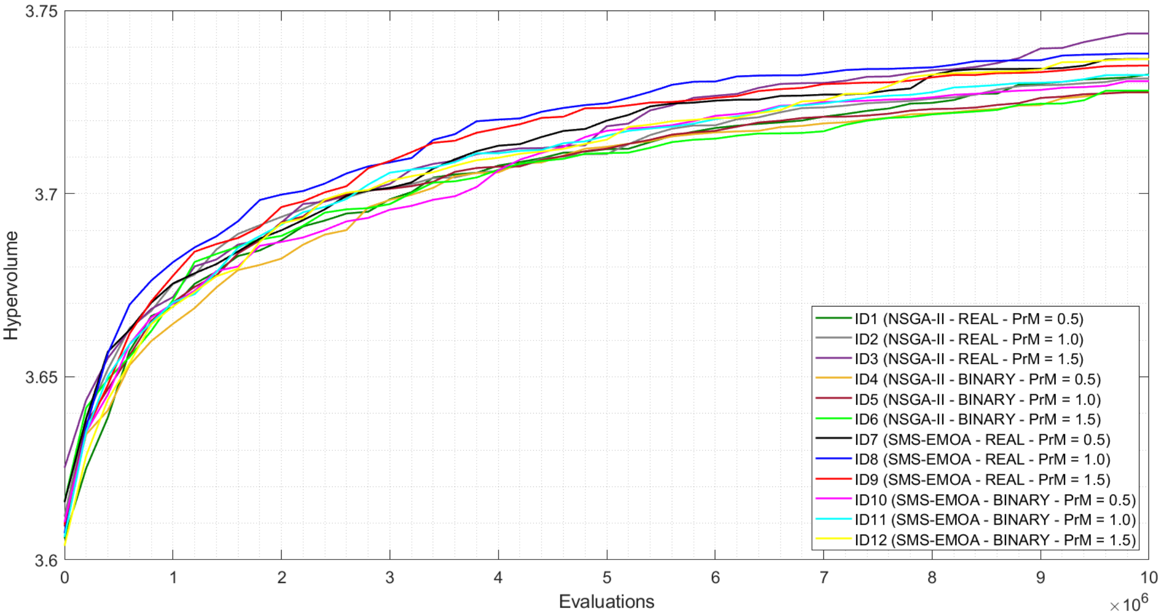


Figure 5.56: Hypervolume average vs. evaluations (3-obj. app.).

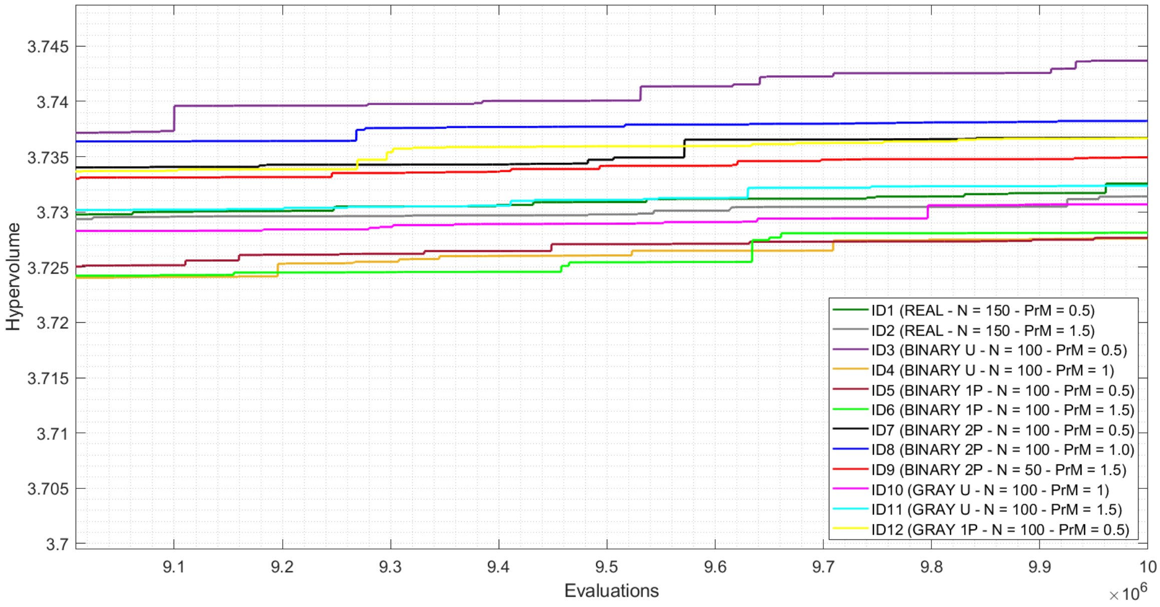


Figure 5.57: Hypervolume average vs. evaluations, detail (3-obj. app.).

In the Figure 5.58, box plots of the Hypervolume values distribution are shown. Such box plots summarise the statistical information supplied by the Table 5.41 (columns 5 to 9). It can be seen that the configuration ID3 (NSGA-II, real encoding and 1.5

gene per chromosome as a mutation rate) reaches the best Hypervolume average, median, maximum and minimum values. The configuration ID12 (SMS-EMOA, binary encoding and 1.5 gene per chromosome as a mutation rate) reaches the lowest standard deviation value.

A statistical hypothesis test was carried out in order to conclude if any configuration reaches a better performance. The Friedman's test was employed to compute the average ranks, which are shown in the Table 5.41 (column 10). The configuration ID3 (NSGA-II, real encoding and 1.5 gene per chromosome as a mutation rate) reaches the best average rank. However, the *p*-value computed (0.1332) does not allow rejecting H_0 (*p*-value<0.05). Therefore, it is not possible to establish that any configuration performs better than any other.

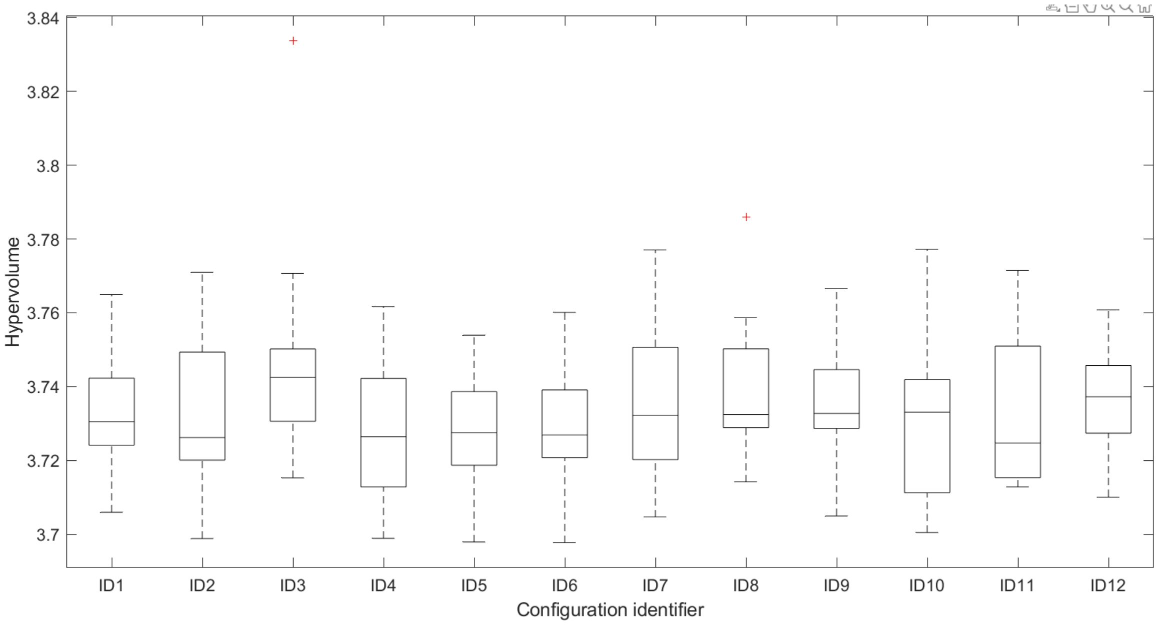


Figure 5.58: Hypervolume box plots, identifiers as in table 5.41 (3-obj. app.).

The Figure 5.59 shows the set of the non-dominated solutions achieved regarding all executions, configurations and methods. Moreover, the projections of such solutions by considering two objectives are shown in the Figures 5.60 (where the considered objectives are Unavailability and Operational Cost), 5.61 (where the considered objectives are Unavailability and Acquisition Cost) and 5.62 (where the considered objectives are Acquisition and Operational Costs). The detail of such solutions is shown in the Table 5.42, where they are ordered from the left to the right side of the Figure 5.59 (solutions with worse Unavailability are firstly ordered). The Unavailability (Q), the Operational Cost, the Acquisition Cost and the optimum times to start a scheduled preventive maintenance task for each device are displayed in such a table. It can be seen that the solution without redundant devices (marked as O in the Figure 5.59 and with identifier Id1 in the Table 5.42) are the more economic and less reliable solution. Conversely, some solutions with both a pump and a valve as redundant devices (marked as □ in the Figure 5.59 and with the identifiers Id17 to Id29 in the Table 5.42) are the more expensive and reliable solutions, as it is expected. The solutions with at least one redundant device are located between the described ones.

Id	Q	Oper. C. [e.u.]	Acq. C. [e.u.]	V_1 [h]	P_2 [h]	P_3 [h]	V_4 [h]	V_5 [h]	V_6 [h]	V_7 [h]
1	0.002814	850.00	1260.00	29239	0	8751	0	22265	31652	34055
2	0.002449	1001.75	1475.00	32279	0	8705	28939	27345	33325	32669
3	0.002439	1026.62	1475.00	33969	0	8451	15234	34630	32356	35040
4	0.002429	1049.75	1475.00	29169	0	8748	27307	27056	26913	25804
5	0.002412	1055.88	1475.00	26692	0	8587	28326	21950	32178	20455
6	0.001597	1384.88	1660.00	27433	8455	8180	0	22551	28131	25988
7	0.001453	1392.50	1660.00	28991	8322	8724	0	33158	28853	30811
8	0.001357	1398.38	1660.00	21919	8441	8709	0	31591	34723	34165
9	0.001260	1437.25	1660.00	24188	7341	8219	0	33715	32883	34964
10	0.001230	1440.88	1660.00	31821	7346	8618	0	25431	30824	28469
11	0.001186	1465.12	1660.00	29598	6453	8465	0	27403	34669	32318
12	0.001182	1494.25	1660.00	33774	8355	8679	0	33429	34617	34507
13	0.001153	1532.00	1660.00	35012	7955	8081	0	34661	30243	34901
14	0.001149	1515.38	1875.00	32693	7970	8760	27430	32350	34495	32112
15	0.001123	1531.00	1875.00	33699	8703	8650	17417	27526	33609	33323
16	0.001123	1599.25	1660.00	32817	7726	8697	0	28190	33078	32850
17	0.001092	1545.75	1875.00	33234	8221	8304	24359	17538	34918	33209
18	0.001019	1550.62	1875.00	29642	7713	8721	27292	13695	32551	34690
19	0.001012	1571.88	1875.00	33985	7684	8745	23603	27644	33321	31591
20	0.000977	1582.50	1875.00	31923	8341	8475	30895	34049	35040	33957
21	0.000933	1589.38	1875.00	31112	7754	8275	16171	32390	24051	30468
22	0.000900	1594.75	1875.00	26810	8442	8482	28415	29632	26639	32205
23	0.000859	1619.00	1875.00	22693	8752	8627	27723	34551	30350	28168
24	0.000839	1667.50	1875.00	27417	7311	8369	33214	33596	28876	21099
25	0.000792	1682.62	1875.00	27161	6209	8711	30234	23623	34641	29979
26	0.000769	1747.38	1875.00	33835	8681	8365	34534	23459	27707	26617
27	0.000763	1782.50	1875.00	25861	8286	8420	24386	24559	34795	34787
28	0.000756	1932.38	1875.00	30523	8706	8034	22339	14998	30052	29927
29	0.000721	2079.12	1875.00	32221	8738	8637	24434	27246	25775	33442

Table 5.42: Non-dominated solutions (3-obj. app.).

Moreover, it can be seen that the solutions id14 and id15 have both redundant devices in the design, however, they are less reliable than the solution id16, which has a redundant valve.

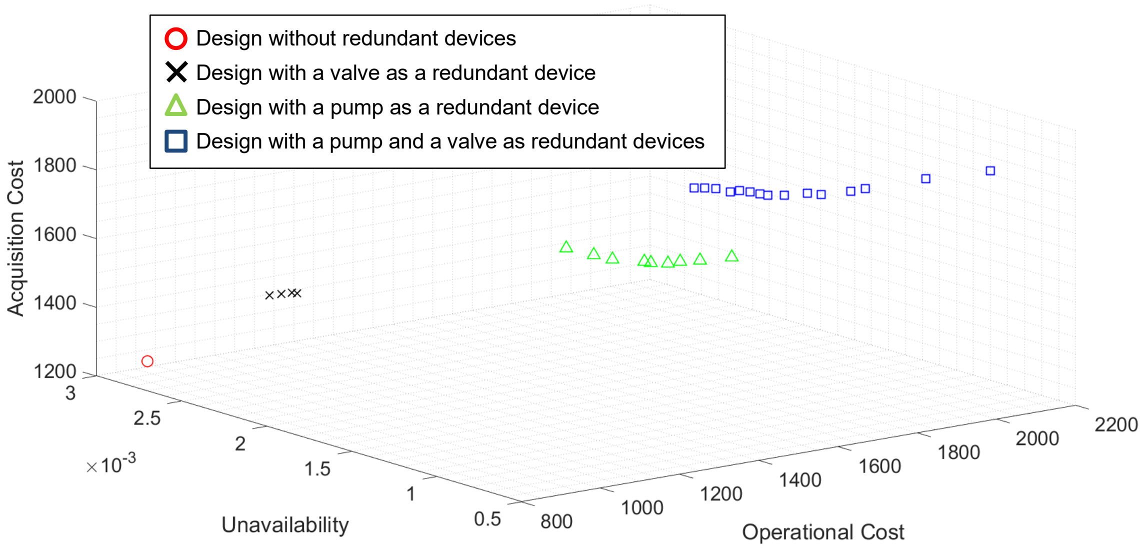


Figure 5.59: Accumulated non-dominated front (3-obj. app.).

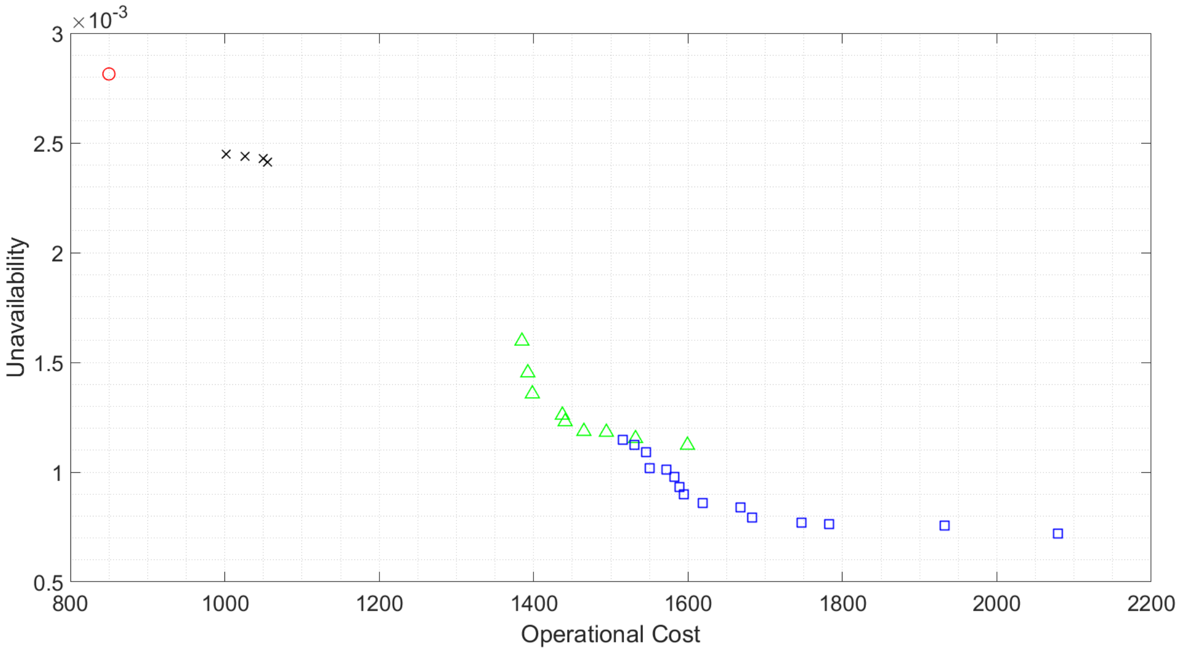


Figure 5.60: Accumulated non-dominated front (3-obj. app. Unav. - Op. Cost).

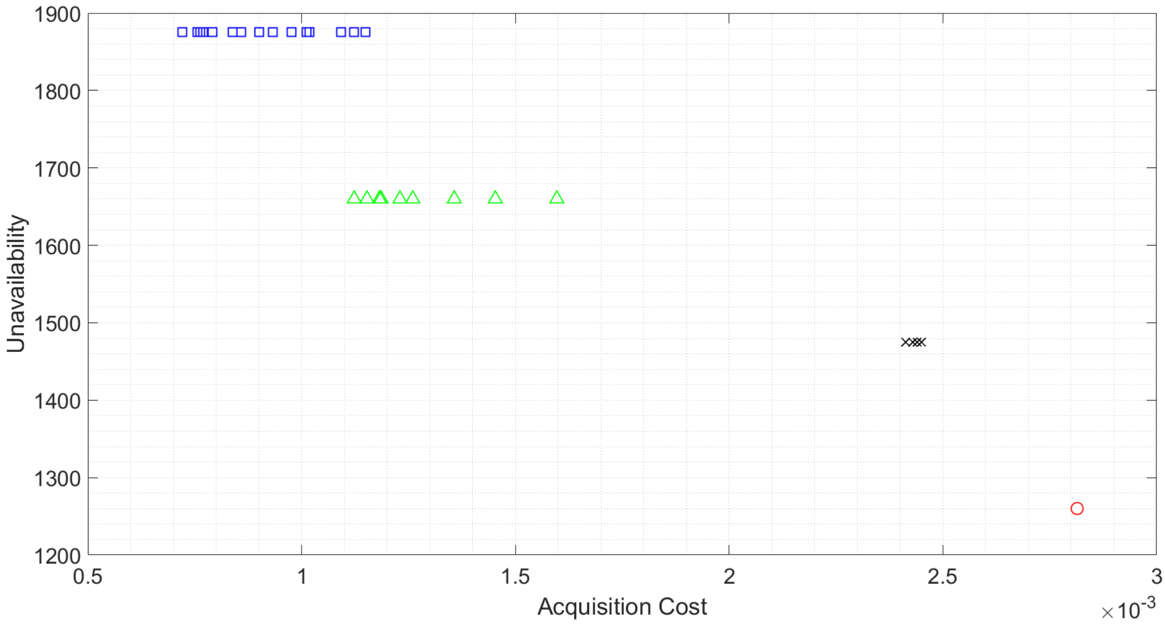


Figure 5.61: Accumulated non-dominated front (3-obj. app. Unav. - Ac. Cost).

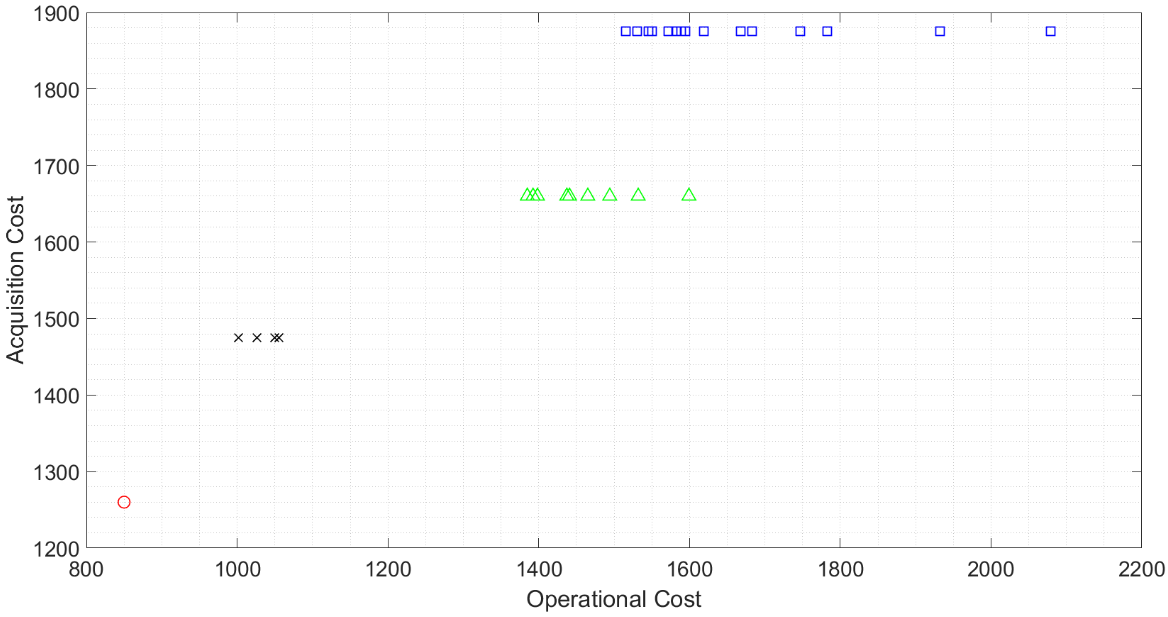


Figure 5.62: Accumulated non-dominated front (3-obj. app. Ac. - Op. Cost).

Finally, the accumulated Hypervolume value (computed as it was described by Fonseca et al. [211]) reaches a value of 3.8704. As it is expected, such a value is higher than 3.7437, the maximum value that is displayed in the Table 5.42.

5.4.3.3. Discussion.

The sections 5.4.3.1 and 5.4.3.2 show the results of solving the case study when two (Unavailability and Cost, which includes both Acquisition and Operational Costs) and three objectives (Unavailability, Acquisition Cost and Operational Cost) were considered. Under the two-objective approach, the configuration ID11 (SMS-EMOA, binary encoding and 1 gene per chromosome as a mutation rate) reaches the best average rank when the Friedman's test is applied. Under the three-objectives approach, the configuration ID3 (NSGA-II, real encoding and 1.5 gene per chromosome as a mutation rate) reaches the best average rank. It can be seen that both methods and encoding are competitive.

In order to compare the performance between both approaches, the best two configurations are taken and compared. These are the configurations ID11 (SMS-EMOA, binary encoding and 1.0 gene per chromosome as a mutation rate) and ID1 (NSGA-II, real encoding and 0.5 gene per chromosome as a mutation rate) when two objectives are considered, and the configurations ID3 (NSGA-II, real encoding and 1.5 gene per chromosome as a mutation rate) and ID12 (SMS-EMOA, binary encoding and 1.5 gene per chromosome as a mutation rate) when three objectives are considered. However, in order to compare directly the configurations regarding the two-objective approach and the configurations in relation to the three-objective approach, these ones must be transformed. Such a transformation consists of adding the Operational Cost and the Acquisition Cost before computing the Hypervolume. In the Table 5.43 (columns 1 to 5), the relationship between the methods and the configuration identifiers is shown. The Figure 5.63 shows the Hypervolume average values evolution versus the number of evaluations. The configuration with identifier ID3 (NSGA-II, real encoding and 1.5 gene per chromosome as a mutation rate) reaches the best Hypervolume average value.

Identifier	Objectives	Method	Encoding	Mutation	Average	Median	Max.	Min.	St. D.	Av. Rank
ID1	2	SMSEMOA	Binary	1.0	2.4628	2.4615	2.5066	2.4347	0.0165	3.476
ID2	2	NSGA-II	Real	0.5	2.4628	2.4617	2.5009	2.4382	0.0156	3.523
ID3	3	NSGA-II	Real	1.5	**2.7537**	**2.7505**	**2.8176**	2.7324	0.0183	**1.476**
ID4	3	SMSEMOA	Binary	1.5	2.7507	2.7494	2.7733	**2.7362**	**0.0105**	1.523
p-Value										$1.009 \cdot 10^{-11}$

Table 5.43: Id’s, config., Hyperv. statistics and statistical test (2- and 3- obj. app.).

In the Figure 5.64, box plots of the Hypervolume values distribution are shown. Such box plots summarise the statistical information supplied by the Table 5.43 (columns 6 to 10). The configuration ID3 (NSGA-II under the 3-objective approach, real encoding and 1.5 gene per chromosome as a mutation rate) shows the highest Hypervolume average, median and maximum values, whereas the configuration ID4 (SMS-EMOA under the 3-objective approach, binary encoding and 1.5 gene per chromosome as a mutation rate) reaches the highest Hypervolume minimum value and the lowest Hypervolume standard deviation value.

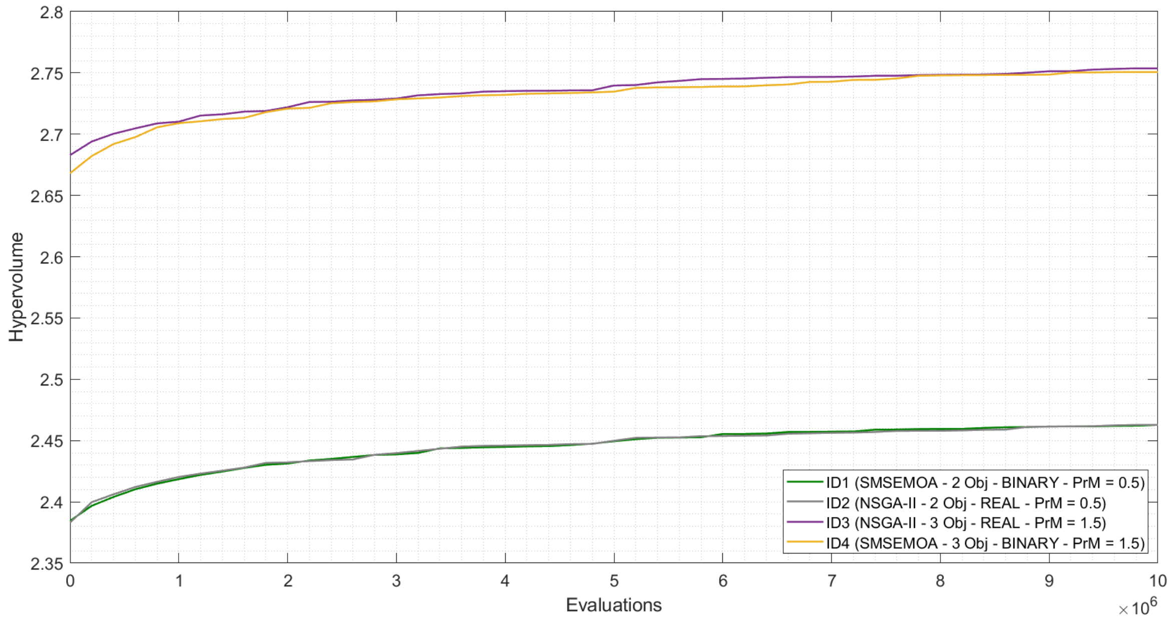


Figure 5.63: Hypervolume average vs. evaluations (2- and 3- obj. app.).

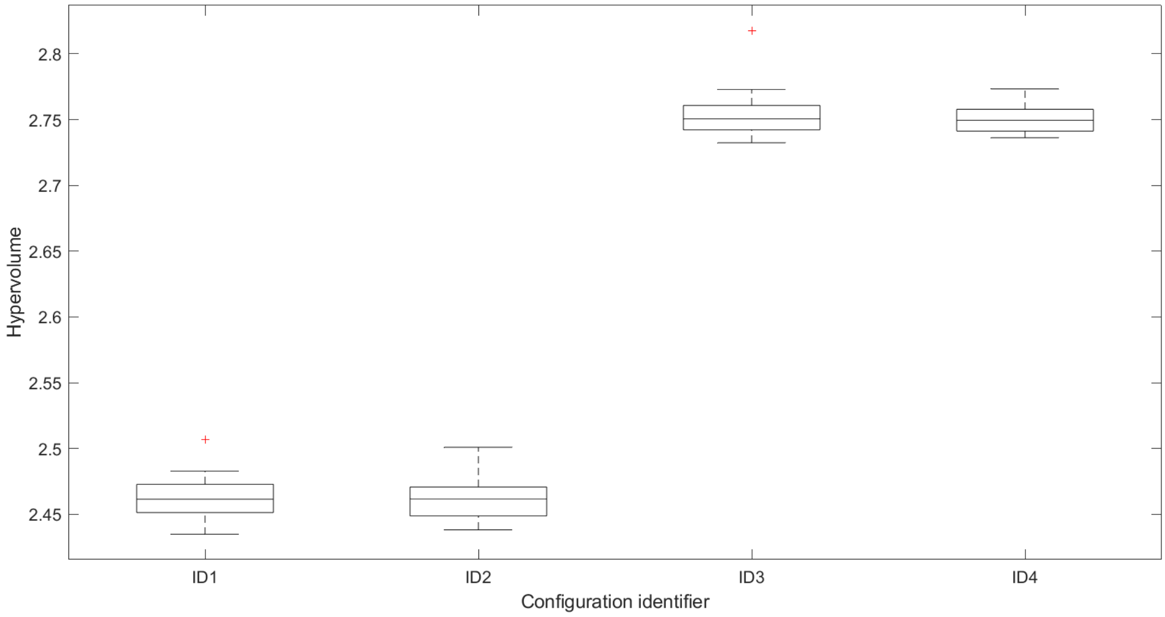


Figure 5.64: Hypervolume box plots, id's as in table 5.43 (2- and 3- obj. app.).

A statistical hypothesis test was carried out in order to conclude if any configuration reaches a better performance. The Friedman's test was employed to compute the average ranks, which are shown in the Table 5.43 (column 11). The configuration ID3 (NSGA-II under the 3-objective approach, real encoding and 1.5 gene per chromosome as a mutation rate) reaches the best average rank. However, the *p*-value computed ($1.009\cdot10^{-11}$) allows rejecting H_0 (*p*-value < 0.05). Therefore, it is possible to establish that any configuration performs better than any other. In order to find the concrete pairwise comparisons that produce such differences, a post-hoc test was conducted. The Shaffer's test was used to compare the configuration ID3, which produced the lowest Average Rank in relation to the Friedman's test, with the rest of configurations. The adjusted *p*-values obtained inform the rejection or acceptance of the null hypothesis. The null hypothesis states that there are no significant differences among the behaviour of the configurations. The result related to the comparisons is shown in the Table 5.44. It is possible to conclude that the configuration with identifier ID3 performs better than the configurations with identifiers ID1 (SMS-EMOA, binary encoding and 1.0 gene per chromosome as a mutation rate) and ID2 (NSGA-II, real encoding and 0.5 gene per chromosome as a mutation rate) when two objectives are considered but is not possible to establish that the configuration with identifier ID3 performs better than the configuration ID4. This advantage of the three-objective approach is a demonstration of how the multi-objectivisation strategy (transforming the original two-objective problem into a three-objective problem by decomposing the cost objective) has been shown successful in this reliability study

Comparison	*p*-value	Conclusion
ID2 - ID3	**$1.652\cdot10^{-6} < 0.05$**	**The null hypothesis is rejected**
ID1 - ID3	**$1.652\cdot10^{-6} < 0.05$**	**The null hypothesis is rejected**
ID3 - ID4	$1.809 > 0.05$	The null hypothesis is not rejected

Table 5.44: *P*-values from the hypothesis tests (2- and 3- obj. app.).

The non-dominated solutions achieved at the end of the evolutionary process for all configurations and methods from both the two- and the three-objective approaches are shown in the Figure 5.65. It can be seen both solutions from the two-objective problem (marked as O) and solutions from the three-objective problem (marked as ×). These solutions are shown in the Table 5.45. The decision makers should decide the preferable design by taking into account their Unavailability-Cost requirements. The front of solutions presents an accumulated Hypervolume value of 2.8318, which was computed by using the procedure supplied by Fonseca et al. [211].

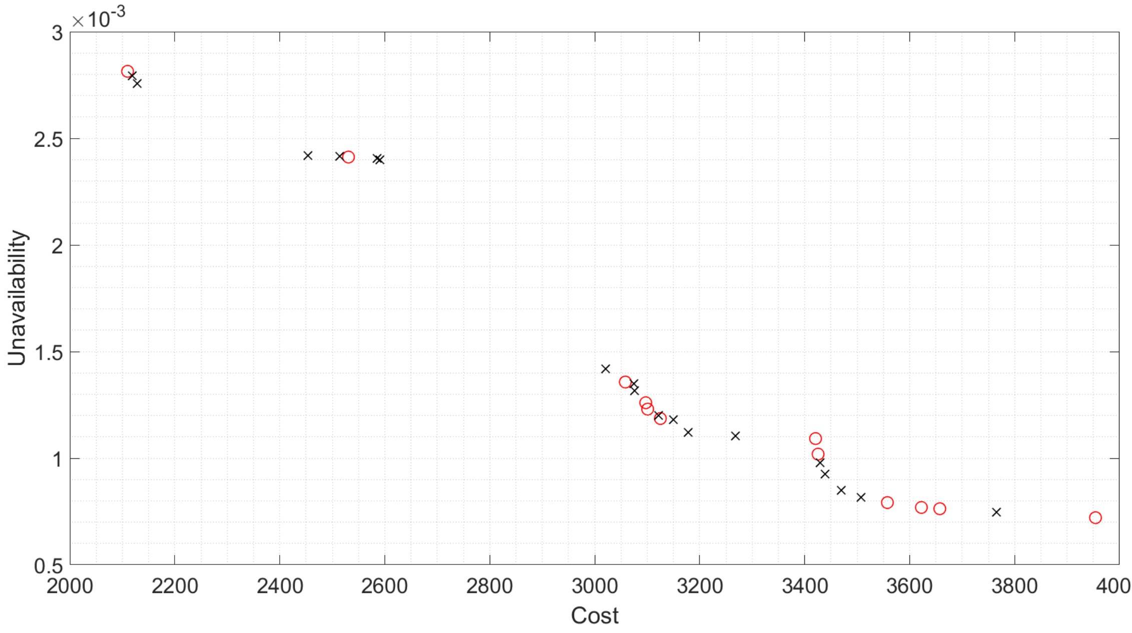


Figure 5.65: Accumulated non-dominated front (2- and 3- obj. app.).

Id	Q	Cost [e.u.]	V_1 [h]	P_2 [h]	P_3 [h]	V_4 [h]	V_5 [h]	V_6 [h]	V_7 [h]
1	0.002814	2110.00	29239	0	8751	0	22265	31652	34055
2	0.002794	2118.75	32081	0	8268	0	32608	29911	22838
3	0.002757	2128.75	32867	0	8738	0	33767	28717	33847
4	0.002419	2454.12	34178	0	8471	22544	22477	34758	29997
5	0.002417	2514.50	31168	0	8640	24997	26369	21853	18347
6	0.002412	2530.88	26692	0	8587	28326	21950	32178	20455
7	0.002404	2585.25	27157	0	8759	25247	29928	32602	33711
8	0.002400	2591.38	33394	0	7624	27291	33090	33453	34139
9	0.001420	3020.62	30166	8752	8592	0	19308	30349	27921
10	0.001357	3058.38	21919	8441	8709	0	31591	34723	34165
11	0.001348	3075.25	32005	8218	6237	0	32457	28160	31568
12	0.001316	3076.25	30348	7793	8637	0	33060	20156	30396
13	0.001260	3097.25	24188	7341	8219	0	33715	32883	34964
14	0.001230	3100.88	31821	7346	8618	0	25431	30824	28469
15	0.001200	3121.50	27780	7589	7849	0	33659	35040	31753
16	0.001186	3125.12	29598	6453	8465	0	27403	34669	32318
17	0.001179	3150.37	33444	7785	7487	0	26641	33341	29593
18	0.001122	3177.88	33180	7576	8374	0	28795	32242	32651
19	0.001104	3268.62	28318	6858	8569	0	29571	32477	28891
20	0.001092	3420.75	33234	8221	8304	24359	17538	34918	33209
21	0.001019	3425.62	29642	7713	8721	27292	13695	32551	34690
22	0.000979	3429.88	27512	8167	8198	32671	32290	34720	32839
23	0.000926	3438.38	24535	8396	8192	10411	32448	34686	33434
24	0.000850	3469.50	32166	8202	8686	31625	28822	29127	34733
25	0.000816	3507.00	30163	8443	7902	19796	34310	30607	18931
26	0.000792	3557.62	27161	6209	8711	30234	23623	34641	29979
27	0.000769	3622.38	33835	8681	8365	34534	23459	27707	26617
28	0.000763	3657.50	25861	8286	8420	24386	24559	34795	34787
29	0.000748	3765.50	27132	7088	7915	10849	14859	33567	34813
30	0.000721	3954.12	32221	8738	8637	24434	27246	25775	33442

Table 5.45: Non-dominated solutions (2- and 3- obj. app.).

5.5. Extending the methodology to other fields of reliability engineering.

The bases of this study were stablished during the development of a research stay in the Departamento de Engenharia Electrónica e de Computadores, Área Científica de Energia, Instituto Superior Técnico of Universidade de Lisboa, Portugal. In this study, the methodology and techniques previously explored are applied to a different engineering field. Reliable smart grids are required by digitised societies to guarantee communication for critical systems. The protection, control and monitoring of Substation Automation Systems (SAS) are enabled when the IEC 61850 standard for Substation Communication Networks (SCN) is employed. Such a standard allows integrating substation devices in order to enable their communication. Although the IEC 61850 standard suggests ethernet-based SCN architectures, depending on the application, the architectural design must be proposed by the system designer. Therefore, when critical systems must be considered, having tools to make smart decisions is vital.

In this study, the joint optimisation of the design and maintenance for a SCN architecture that follows the IEC 61850 standard is explored. The aim consists of supplying a set of optimum Availability-Cost solutions. Multi-objective Evolutionary Algorithms and Discrete Simulation are coupled while indicator-based and dominance-based state-of-the-art multi-objective optimisers are employed. Two optimisation approaches are considered, analysed and thoroughly compared in a case study; a two-objective approach, and a three-objective approach, which attends to the multi-objectivisation concept.

5.5.1. Background.

The improvement of technology benefits to modern societies. Such an improvement has increased the demand of energy over the past decades. Therefore, the power systems must face several challenges to cover such a demand. The IEC 61850 standard for Substation Communication Networks (SCN) enables the protection, control and monitoring of Substation Automation Systems (SAS). The architectural design is in the hands of the designers of the system so a methodology to support such decisions is useful for them. In this study, the methodology applied all along the present research is explored as follows:

- The study covers the joint optimisation of a SCN architecture design based on the IEC 61850 standard (this consists of considering the automatic selection of devices in order to be included in the design) and their preventive maintenance strategy (this consists of determining the optimum preventive maintenance times regarding the devices included in the design). This allows scheduling the preventive maintenance tasks. As objectives for the multi-objective problem, Unavailability and Cost are considered.

- Multi-objective Evolutionary Algorithms and Discrete Event Simulation are coupled. The objective function regarding the Cost objective considers both Operational and Acquisition Cost.
- Once the two-objective problem is solved, a multi-objectivisation approach is proposed where the decomposition of the Cost objective is handled: the Cost objective is decomposed between Acquisition and Operational Cost. Therefore, the problem to be solved has three objectives: Unavailability, Acquisition Cost and Operational Cost. The results when this approach is considered and a comparison with the two-objective approach are provided.

5.5.2. Case study.

The methodology is applied to a specific section of a subsystem that follows the IEC 61850 standard. The Figure 5.66 shows the T1-1 substation design under the IEC 61850 [214], which is a single bus, small transmission substation to transform energy from 220 kV. to 132 kV. Such a substation presents 5 bays (3 bay lines, 1 for bus and 1 for transformer). The methodology is employed to optimise one line bay. A star topology [215] is assumed for the line bay equipment connection; this implies that the Merging Unit (MU), the Control Intelligent Electronic Device (Cnt. IED) and the Protection Intelligent Electronic Device (Prt. IED) are attached to the Ethernet Switch (ESW) by using individual communication cables. In addition, a Time Synchronisation source (TS) is connected directly to the MU [214].

The Reliability Blocks Diagram is shown in the Figure 5.67. As it is shown, depending on the evolutionary process, a second Prt. IED may be considered as a redundant device. Typical preventive maintenance tasks regarding such devices can include, for instance, testing and calibrations of protective relays, verifying system telecommunications equipment and channels required for correct operation and functional testing [216]. To apply the methodology, detailed information regarding the considered devices is needed, which is like the information detailed for the previous studies. Such information is shown in the Table 5.46.

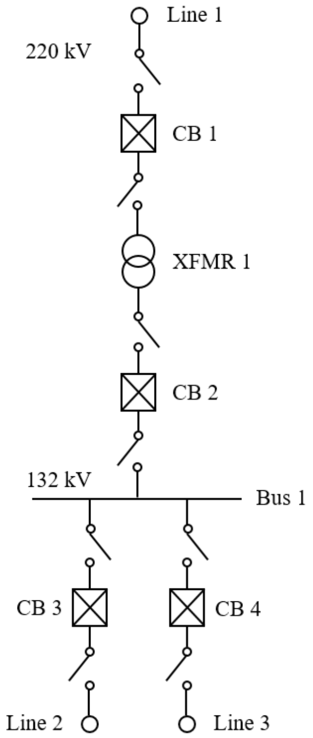


Figure 5.66: T1-1 substation layout.

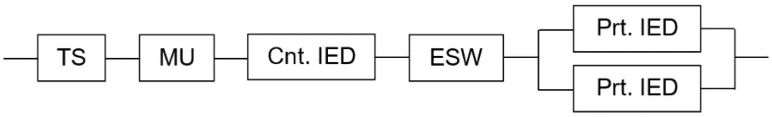


Figure 5.67: Reliability Blocks Diagram for line bay.

Parameter	Value	Source
Life cycle or mission time	525,600 hours	-
Corrective Maintenance Cost	110 €/hour	SIANI-IST
Preventive Maintenance Cost	100 €/hour	SIANI-IST
TS Acquisition cost	7,000 €	IST [217]
TS TF_{min}	1 hour	-
TS TF_{max}	525,600 hours	Mission time
TS $TF\ \lambda$	0,1x10^{-6} failures/hour	CISCO [219]
TS TR_{min}	1 hour	-
TS TR_{max}	7 hours	$TR\mu + 3 \cdot TR\sigma$
TS $TR\mu$	4 hours	CISCO [219]
TS $TR\sigma$	1 hours	$(TR\mu - TR_{min})/3$
TS TM_{min}	2,190 hours	TM_{max}/2
TS TM_{max}	4,380 hours	NERC 2007 [220]
TS TCM_{min}	1 hours	Round (TR_{min}/2)
TS TCM_{max}	4 hours	Round (TR_{min}/2)
MU Acquisition cost	3,500 €	IST [217]
MU TF_{min}	1 hour	-
MU TF_{max}	525,600 hours	Mission time
MU $TF\ \lambda$	6,0047x10^{-6} failures/hour	Scheer and Dolezilek [218]
MU TR_{min}	2 hours	-
MU TR_{max}	14 hours	$TR\mu + 3 \cdot TR\sigma$
MU $TR\mu$	8 hours	Scheer and Dolezilek [218]
MU $TR\sigma$	2 hours	$(TR\mu - TR_{min})/3$
MU TM_{min}	2,190 hours	TM_{max}/2
MU TM_{max}	4,380 hours	NERC 2007 [220]
MU TCM_{min}	1 hours	Round (TR_{min}/2)
MU TCM_{max}	7 hours	Round (TR_{min}/2)
Cnt.IED Acquisition cost	3,537.50 €	IST [217]
Cnt.IED TF_{min}	1 hour	-
Cnt.IED TF_{max}	525,600 hours	Mission time
Cnt.IED $TF\ \lambda$	6,0047x10^{-6} failures/hour	Scheer and Dolezilek [218]
Cnt.IED TR_{min}	2 hours	-
Cnt.IED TR_{max}	14 hours	$TR\mu + 3 \cdot TR\sigma$
Cnt.IED $TR\mu$	8 hours	Scheer and Dolezilek [218]
Cnt.IED $TR\sigma$	2 hours	$(TR\mu - TR_{min})/3$
Cnt.IED TM_{min}	2,190 hours	TM_{max}/2
Cnt.IED TM_{max}	4,380 hours	NERC 2007 [220]
Cnt.IED TCM_{min}	1 hours	Round (TR_{min}/2)
Cnt.IED TCM_{max}	7 hours	Round (TR_{min}/2)
ESW Acquisition cost	2,600 €	IST [217]
ESW TF_{min}	1 hour	-
ESW TF_{max}	525,600 hours	Mission time
ESW $TF\ \lambda$	9,9265x10^{-6} failures/hour	Scheer and Dolezilek [218]
ESW TR_{min}	2 hours	-
ESW TR_{max}	14 hours	$TR\mu + 3 \cdot TR\sigma$
ESW $TR\mu$	8 hours	Scheer and Dolezilek [218]
ESW $TR\sigma$	2 hours	$(TR\mu - TR_{min})/3$
ESW TM_{min}	2,190 hours	TM_{max}/2
ESW TM_{max}	4,380 hours	NERC 2007 [220]
ESW TCM_{min}	1 hours	Round (TR_{min}/2)
ESW TCM_{max}	7 hours	Round (TR_{min}/2)
Crt.IED Acquisition cost	3,537.50 €	IST [217]
Crt.IED TF_{min}	1 hour	-
Crt.IED TF_{max}	525,600 hours	Mission time
Crt.IED $TF\ \lambda$	6,0047x10^{-6} failures/hour	Scheer and Dolezilek [218]
Crt.IED TR_{min}	2 hours	-
Crt.IED TR_{max}	14 hours	$TR\mu + 3 \cdot TR\sigma$
Crt.IED $TR\mu$	8 hours	Scheer and Dolezilek [218]
Crt.IED $TR\sigma$	2 hours	$(TR\mu - TR_{min})/3$
Crt.IED TM_{min}	2,190 hours	TM_{max}/2
Crt.IED TM_{max}	4,380 hours	NERC 2007 [220]
Crt.IED TCM_{min}	1 hours	Round (TR_{min}/2)
Crt.IED TCM_{max}	7 hours	Round (TR_{min}/2)

Table 5.46: Reliability and cost data.

The life cycle or mission time considered was 525,600 hours. The Costs were supplied by Instituto Superior Técnico (IST) [217] and Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI), from Universidade de Lisboa and Universidad de Las Palmas de Gran Canaria, respectively. The failure rates were achieved from studies performed by Scheer and Dolezilek [218] and CISCO [219]. The mean for the time to repair ($TR\mu$) were defined from CISCO [219] and Kanabar and Sidhu [214]. TF_{min} was taken as 1 hour and TF_{max} was taken as the life cycle or mission time, however, due to the methodology, TM_{max} limits TF_{max} because the operational time taken into account is the smaller one. TR_{min} was set at 1 hour for the TS and 2 hours for the rest of devices, attending to data sources. It is considered that the time to repair follows a normal distribution with μ as mean, so $TR\sigma$ was set by using a mathematical relationship. As 99.7% of values from a normal distribution fall within the interval $\mu \pm 3\sigma$, so this consideration was considered to define both $TR\sigma$ and TR_{max}. TM_{max} was set to 6 months as it is claimed by NERC 2007 [220] for the minimal interval and the middle of such a quantity was set to TM_{min}. Finally, TCM_{min} and TCM_{max} were set to the middle of TR_{min} and TR_{max}, respectively. In this case:

- The optimum period to start a preventive maintenance task for each device must be established, and
- Including a Prt. IED as a redundant component must be decided. To do that, design alternatives must be evaluated. Considering the integration of a redundant device will enhance the system Availability. However, it will get worse both the Operational and the Acquisition Cost.

Both real and binary encoding are explored so two codifications for the chromosomes are considered:

- Real encoding: Each individual of the population is built by employing a real numbers string with values between 0 and 1. Such a string is codified as [D_1 M_1 M_2 M_3 M_4 M_5 M_6], where the presence of a redundant Prt. IED is denoted by the design decision variable D_1 and the optimum time to start a preventive maintenance task for each device is denoted by the maintenance decision variables M_1 to M_6. Nevertheless, in order to compute the objective functions, they must be converted:

- The decision variable D_1 is rounded at the nearest integer, so the value 1 implies that the device is included in the design and the value 0 implies the opposite.
- The decision variables M_1 to M_6 are scaled by using the Equation 5.9, which was presented as the Equation 4.5 in the Chapter IV, where TM_i is the true value of the time to start a scheduled preventive maintenance task for the i-th line bay device, M_i is the value of the corresponding decision variable regarding the i-th system device and finally, TM_{max_i} and TM_{min_i} are the limit values for the parameter TM for the i-th line bay device, when $1 \leq i \leq 6$.

$$TM_i = round\ (TM_{min_i} + M_i \cdot (TM_{max_i} - TM_{min_i})) \tag{5.9}$$

- Binary encoding: Each individual of the population is built by employing a binary numbers string with value among 0 and 1. The bits number of such a string is 73 and they are as follows:
  - The decision variable D_1 denotes the presence of a redundant Prt. IED. The value 1 implies that the device is included in the design and the value 0 implies the opposite.
  - The decision variables M_1 to M_{12} denote the optimum time to start a scheduled preventive maintenance task regarding the device TS. A binary scale to represent numbers from its TM_{min} to its TM_{max} is needed in order to achieve the value for its TM. $TM_{min_{TS}}$ and $TM_{max_{TS}}$ have values of 2,190 hours and 4,380 hours respectively. Therefore, 4,380 - 2,190 = 2,190 steps are needed where the step zero represents 2,190 hours and the step 2,189 represents 4,380 hours. To cover 2,190 steps, the binary scale must satisfy that 2^n > 2,190, where n is the number of bits. Thus, 12 bits are needed in this case. Since 2,190 steps are needed and 2^{12} = 4,096 steps are available, an

equivalent relationship must be used. Each step regarding the scale of 4,096 steps considers 2,190 / 4,096 = 0.53466796875 steps in relation to the scale of 2,190 steps. Therefore, in order to achieve the true time to start a scheduled preventive maintenance task, employing the transformation that is shown by the Equation 5.10 is needed.

$$TM_i = round\ (TM_{min_i} + M_i \cdot (0.53466796875)) \tag{5.10}$$

- The decision variables M_{13} to M_{24}, M_{25} to M_{36}, M_{37} to M_{48}, M_{49} to M_{60} and M_{61} to M_{72} denote the optimum times to start a scheduled preventive maintenance task regarding the devices MU, Cnt. IED, ESW and both Prt. IEDs, respectively. The procedure described above for the TS must be followed to achieve the true value for the time to start a scheduled preventive maintenance task regarding such devices. However, the respective TM_{min} and TM_{max} in relation to such devices must be used to achieve their TM.

The set of parameters and the used Multi-objective Algorithms are shown in the Table 5.47. They were previously defined in the present Chapter of the research.

Method	Encoding	PrM	disM	PrC	disC
SMS-EMOA	Real	0.5	20	1	20
		1.0			
		1.5			
SMS-EMOA	Binary	0.5	-	1	-
		1.0			
		1.5			
NSGA-II	Real	0.5	20	1	20
		1.0			
		1.5			
NSGA-II	Binary	0.5	-	1	-
		1.0			
		1.5			

Table 5.47: Parameters to configure the experiments.

The population size of 150 individuals was used. Six different configurations of the two methods were simulated and 21 executions per configuration were conducted for statistical purposes. A total of 10,000,000 evaluations of the objective functions was employed as a stopping criterion. In order to normalise the value of the objective functions, scale factors were employed. When the two-objective problem was considered:

- The Cost was computed by considering a scale factor of 740,000 €.
- The Unavailability of the system was computed by considering a scale factor of 0.01.

When the three-objective problem was considered:

- The Acquisition Cost was computed by using a scale factor of 24,000 €.
- The Operational Cost was computed by considering a scale factor of 740,000 €.
- The Unavailability of the system was computed by employing a scale factor of 0.01.

Finally, depending on the number of objectives to consider, a two- or three-dimensional reference point must be chosen in order to compute the Hypervolume indicator. Such points must cover the points limited by the scale factors, which normalise the values of the objectives up to a maximum value of a unit. The reference points were set to (2,2) and (2,2,2) respectively. The open-source Software Platform PlatEMO (programmed in MATLAB) was used again to optimise the case study.

5.5.3. Results and discussion.

For the optimisation process, a High-Performance Computer was employed. In this case, six calculation nodes and one front-end node were employed. Each node has two processors Intel Xeon E5645 Westmere-EP with twelve cores each and 48 GB of RAM. After achieving the results, useful information is provided:

- Data regarding the computational process is given in order to show the hardness of the problem and the computational cost. Such information consists of the time taken for 21 executions of the 3 configurations related to each method and encoding.

- Regarding each configuration, the Hypervolume indicator average value evolution (in twenty-one executions) is displayed.
- Box plots are shown for the Hypervolume values distribution at the end of the process.
- The values of several measures achieved at the end of the process are displayed. Such measures are the average, median, minimum, maximum and standard deviation values for the Hypervolume indicator.
- A meticulous hypothesis test is conducted in order to find out significant differences between the performance of the employed methods and their configurations. The Friedman's test is used to detect differences between the achieved results and rejecting the null hypothesis (H_0) in such a case. Once the differences have been detected, in order to find the concrete pairwise comparisons that produce such differences, a post-hoc test is conducted. The *p*-value denotes the lowest significant value that can conduct to reject H_0. The *p*-value supplies information regarding the significance of a statistical hypothesis test, and regarding how much significant it is: The evidence to reject H_0 appears when the *p*-value is smaller than 0.05. The procedure to conduct the pairwise comparisons followed in this paper was described by Benavoli et al. [210].
- Finally, for the accumulated best non-dominated solutions, the Hypervolume is computed [211]. Such solutions are the best-balanced solutions between the objectives.

5.5.3.1. Two-objective problem results.

The average time consumed by each execution of the method was 4,336 minutes (3 days and 16 minutes). The whole optimisation process (21 executions of 12 configurations) implies a sequential time of 1,092,743 minutes (2 years, 28 days and 20 hours, approximately). Such a computational cost brings into the light how important the use of the High-Performance Computer is, which allows parallel processes. In the Table 5.48 (columns 1 to 4), the relationship between the methods and the configuration identifiers is shown.

Identifier	Method	Encoding	Mutation	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	NSGA-II	Real	0.5	3.4404	3.4340	**3.5192**	3.4044	0.0258	7.5714
ID2	NSGA-II	Real	1.0	**3.4495**	3.4456	3.4898	3.4141	0.0192	**4.7619**
ID3	NSGA-II	Real	1.5	3.4439	3.4356	3.4786	3.4160	0.0189	6.0952
ID4	NSGA-II	Binary	0.5	3.4393	3.4428	3.4645	3.4107	**0.0126**	6.1428
ID5	NSGA-II	Binary	1.0	3.4397	3.4370	3.4746	3.4119	0.0149	6.8095
ID6	NSGA-II	Binary	1.5	3.4332	3.4340	3.4674	3.4064	0.0149	7.8571
ID7	SMS-EMOA	Real	0.5	3.4471	**3.4474**	3.4800	3.4166	0.0150	5.3333
ID8	SMS-EMOA	Real	1.0	3.4391	3.4364	3.4812	3.4125	0.0163	6.7142
ID9	SMS-EMOA	Real	1.5	3.4389	3.4421	3.4635	3.4111	0.0144	6.6666
ID10	SMS-EMOA	Binary	0.5	3.4414	3.4368	3.4916	3.4075	0.0213	7.0952
ID11	SMS-EMOA	Binary	1.0	3.4413	3.4391	3.4824	3.4095	0.0193	6.4761
ID12	SMS-EMOA	Binary	1.5	3.4401	3.4373	3.4747	**3.4215**	0.0145	6.4761
p-Value									0.2787

Table 5.48: Id's, config., Hyperv. statistics and statistical test (2-obj. app.).

The Figure 5.68 shows the Hypervolume average values evolution regarding the number of evaluations. The configuration with identifier ID2 (NSGA-II, real encoding and 1.0 gene per chromosome as a mutation rate) presents the best Hypervolume average value.

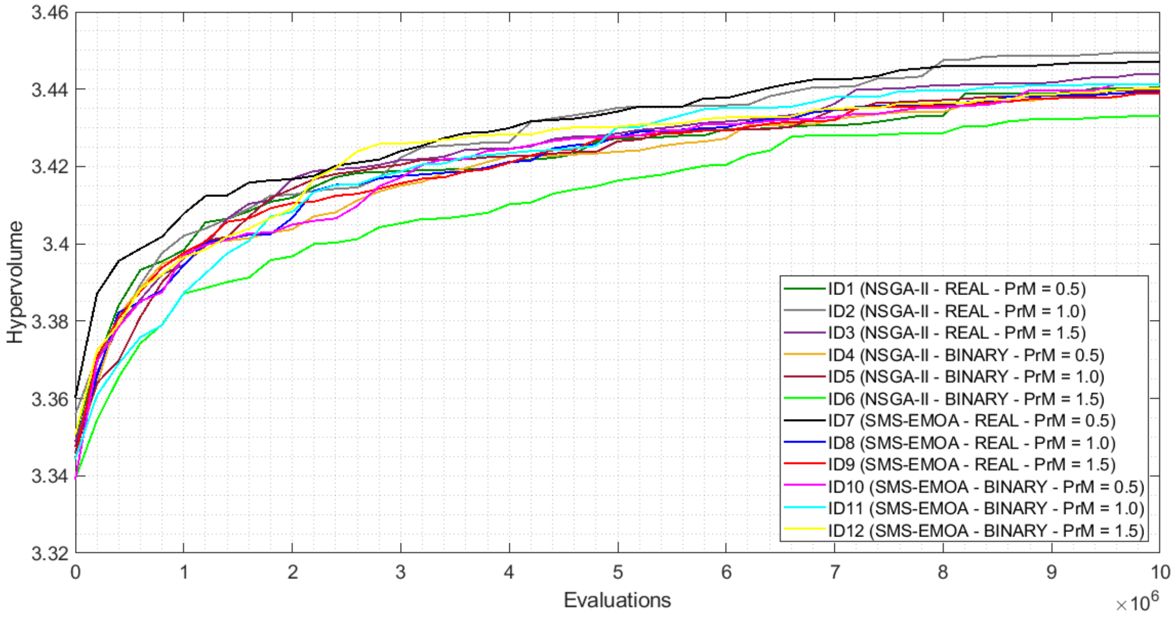


Figure 5.68: Hypervolume average vs. evaluations (2-obj. app.).

In the Figure 5.69, box plots of the Hypervolume values distribution are shown. Such box plots summarise the statistical information supplied by the Table 5.48 (columns 5 to 9).

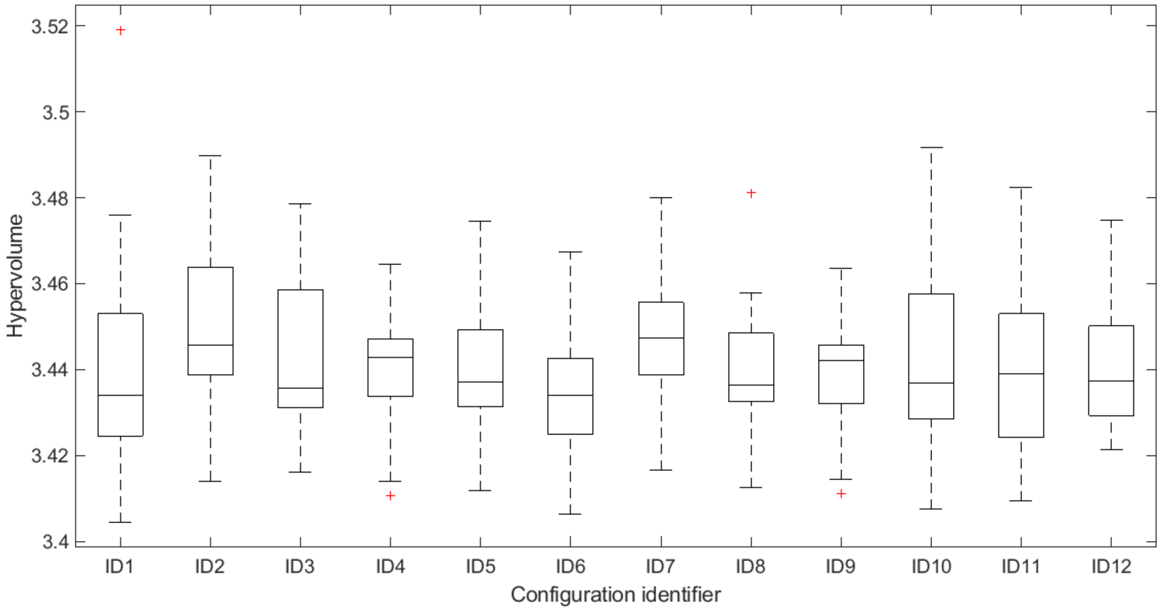


Figure 5.69: Hypervolume box plots, identifiers as in Table 5.48 (2-obj. app.).

It can be seen that the configuration ID2 (NSGA-II, real encoding and 1 gene per chromosome as a mutation rate) achieves the best Hypervolume average value, the configuration ID7 (SMS-EMOA, real encoding and 0.5 gene per chromosome as a mutation rate) achieves the best Hypervolume median value, the configuration ID1 (NSGA-II, real encoding and 0.5 gene per chromosome as a mutation rate) achieves the best Hypervolume maximum value, the configuration ID9 (SMS-EMOA, real encoding and 1.5 gene per chromosome as a mutation rate) achieves the best Hypervolume minimum value and the configuration ID4 (NSGA-II, binary encoding and 0.5 gene per chromosome as a mutation rate) achieves the lowest standard deviation value.

A statistical hypothesis test was conducted in order to conclude if any configuration reaches a better performance. The Friedman's test was employed to computed the average ranks, which are shown in the Table 5.48 (column 10). The configuration ID2 (NSGA-II, real encoding and 1.0 gene per chromosome as a mutation rate) reaches the best average rank (the average rank must be as low as possible when the problem consists of maximising the Hypervolume). However, the *p*-value computed (0.2787) does not allow rejecting H_0 (*p*-value > 0.05). Therefore, it is not possible to conclude that any configuration performs better than any other.

The Figure 5.70 shows the set of non-dominated solutions achieved regarding all executions, configurations and methods. The detail of such solutions is shown in the Table 5.49, where they are ordered from the left to the right side of the figure (solutions with worse Unavailability are firstly ordered). The Unavailability (Q), the Cost and the optimum times to start a scheduled preventive maintenance task regarding each device are displayed in such a table. The solutions without a redundant Prt. IED (marked as O in the Figure 5.70 and with identifiers Id1 and Id2 in the Table 5.70) are the more economic and less reliable solutions. Conversely, the solutions with a redundant Prt. IED (marked as × in the Figure 5.70 and with identifiers Id3 to Id8 in the Table 5.49) are the more expensive and reliable solutions, as it is expected. It can be seen that the solutions Id1 and Id2 do not present values regarding the redundant Prt. IED due to the fact that such a device is not considered for the design. As it was said above, these solutions are more economic and less reliable than the rest because they do not contain a redundant device. On the other hand, speaking in general, the times to start a preventive maintenance task present the trend of being set close to the maximum value for such a variable, which is 4,380 hours (see TM_{max} values from the Table 5.46). This is due to the fact that the optimisers work in order to maximise the time between preventive maintenance tasks as much as possible. However, some times to start a preventive maintenance task are further from such a value (for instance, in the Table 5.49, it can be seen that the solution 8 presents a value of 2,805 for the main Prt. IED), which implies more preventive maintenance tasks. In this case, the more maintenance, the more expensive and available the solution, as it is expected. Finally, the front of solutions presents an accumulated Hypervolume value (computed as it was described by

Fonseca et al [211]) of 3.5239. As it is expected, such a value is higher than 3.5192, the maximum value that is displayed in the Table 5.48.

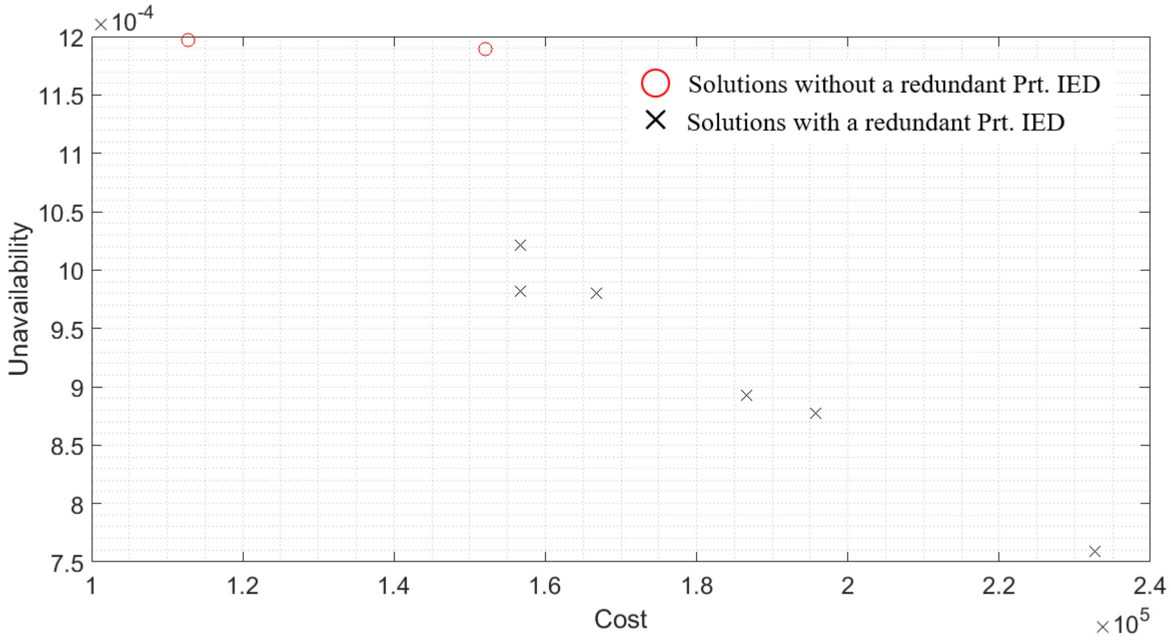


Figure 5.70: Accumulated non-dominated front (2-obj. app.).

Id	Q	Cost [€]	TS [h]	MU [h]	Cnt. IED [h]	ESW [h]	Prt. IED [h]	Prt. IED [h]
1	0.001197	112750.00	4369	4350	4298	4342	4322	0
2	0.001189	152022.50	4353	4380	4173	4377	4380	0
3	0.001022	156712.50	4372	4282	4352	4239	4235	3752
4	0.000982	156762.50	4380	4333	4369	4321	4317	3306
5	0.000980	166782.50	4349	4360	4337	4285	4022	3250
6	0.000892	186565.00	4380	4380	4082	4195	4271	3761
7	0.000877	195675.00	4376	4245	4309	4376	4183	4129
8	0.000759	232682.50	4349	4377	4380	4380	2805	3279

Table 5.49: Non-dominated solutions (2-obj. app.).

5.5.3.2. Three-objective problem results.

The average time consumed by each execution of the method was 5,945 minutes (4 days, 3 hours and 5 minutes). The whole optimisation process (21 executions of 12 configurations) implies a sequential time of 1,498,190 minutes (2 years, 10 months and 6 days, approximately). In the Table 5.50 (columns 1 to 4), the relationship between the methods and the configuration identifiers is shown.

Identifier	Method	Encoding	Mutation	Average	Median	Max.	Min.	St. Deviation	Av. Rank
ID1	NSGA-II	Real	0.5	4.0442	4.0420	4.0862	3.9998	0.0229	6.2857
ID2	NSGA-II	Real	1.0	4.0422	4.0394	4.0856	4.0144	0.0167	6.4761
ID3	NSGA-II	Real	1.5	4.0377	4.0372	4.0613	4.0066	0.0158	7.6190
ID4	NSGA-II	Binary	0.5	4.0473	4.0445	4.0716	4.0184	0.0151	5.4285
ID5	NSGA-II	Binary	1.0	4.0453	4.0413	4.0948	4.0053	0.0228	5.9523
ID6	NSGA-II	Binary	1.5	4.0436	4.0430	**4.1269**	4.0002	0.0265	6.4761
ID7	SMS-EMOA	Real	0.5	4.0527	4.0480	4.1046	4.0160	0.0259	5.3809
ID8	SMS-EMOA	Real	1.0	**4.0531**	**4.0512**	4.1035	**4.0255**	0.0191	**4.9523**
ID9	SMS-EMOA	Real	1.5	4.0487	4.0472	4.0892	4.0177	0.0192	5.8095
ID10	SMS-EMOA	Binary	0.5	4.0382	4.0383	4.1104	4.0056	0.0261	7.5714
ID11	SMS-EMOA	Binary	1.0	4.0373	4.0352	4.0704	4.0168	**0.0147**	7.3809
ID12	SMS-EMOA	Binary	1.5	4.0301	4.0308	4.0603	4.0069	0.0148	8.6666
p-Value									0.0260

Table 5.50: Id's, config., Hyperv. statistics and statistical test (3-obj. app.).

The Figure 5.71 shows the Hypervolume average values evolution regarding the number of evaluations. The configuration with identifier ID8 (SMS-EMOA, real encoding and 1.0 gene per chromosome as a mutation rate) reaches the best Hypervolume average value.

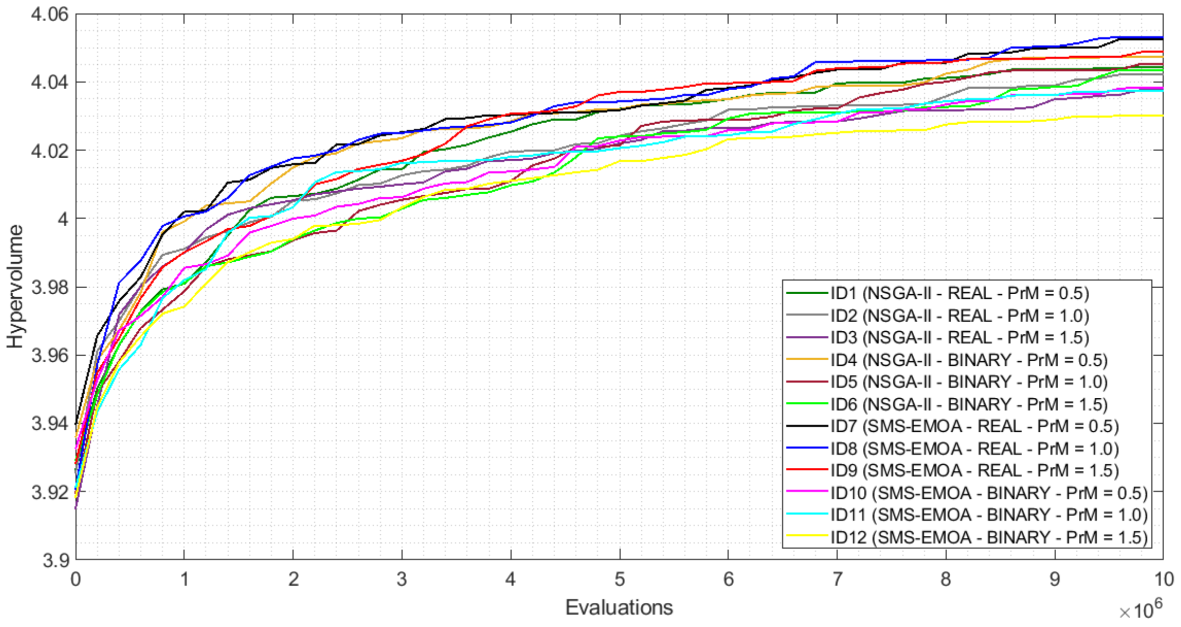


Figure 5.71: Hypervolume average vs. evaluations (3-obj. app.).

In the Figure 5.72, box plots of the Hypervolume values distribution are shown. Such box plots summarise the statistical information included by the Table 5.50 (columns 5 to 9). It can be seen that the configuration ID8 (SMS-EMOA, real encoding and 1.0 gene per chromosome as a mutation rate) achieves the best Hypervolume Average, Median and Minimum values, the configuration ID6 (NSGA-II, binary encoding and 1.5 gene per chromosome as a mutation rate) achieves the best Hypervolume Maximum value and the configuration ID11 (SMS-EMOA, binary encoding and 1.0 gene per chromosome as a mutation rate) achieves the lowest Standard Deviation value.

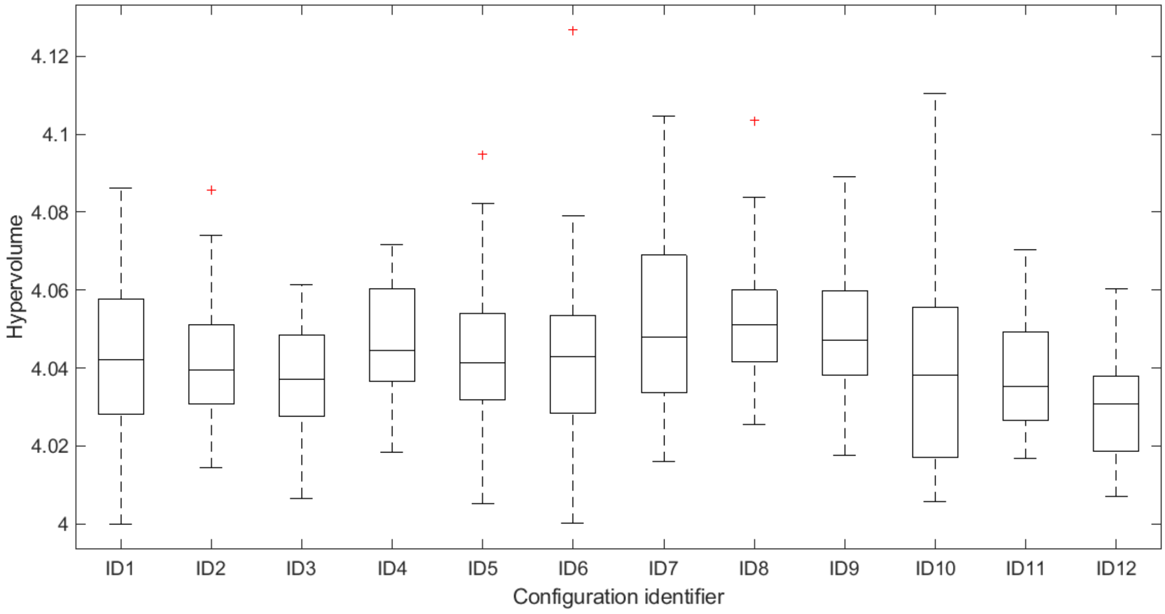


Figure 5.72: Hypervolume box plots, id’s as in the Table 5.50 (3-obj. app.).

A statistical hypothesis test was conducted to conclude if any configuration reaches a better performance. The Friedman’s test was employed to compute the average ranks, which are shown in the Table 5.50 (column 10). The configuration ID8 (SMS-EMOA, real encoding and 1.0 gene per chromosome as a mutation rate) reaches the best average rank. However, the *p*-value computed (0.0260) allows rejecting H_0 (*p*-value < 0.05). Therefore, it is possible to conclude that any configuration performs better than any other. The Wilcoxon signed-rank test is used in order to curry out the pairwise comparisons. In the Table 5.51, the results of the Wilcoxon signed-rank test are shown, where the configuration ID8 was compared with the rest of configurations. Such a configuration performs better than the configurations ID3 (NSGA-II, real encoding and 1.5 gene per chromosome as a mutation rate), ID11 (SMS-EMOA, binary encoding and 1.0 gene per chromosome as a mutation rate) and ID12 (SMS-EMOA, binary encoding and 1.5 gene per chromosome as a mutation rate).

Comparison	*p*-value	Conclusion
ID8 - ID12	**0.0017 < 0.05**	**The null hypothesis is rejected**
ID8 - ID11	**0.0057 < 0.05**	**The null hypothesis is rejected**
ID3 - ID8	**0.0117 < 0.05**	**The null hypothesis is rejected**
ID10 - ID8	0.0582 > 0.05	The null hypothesis is not rejected
ID2 - ID8	0.0680 > 0.05	The null hypothesis is not rejected
ID1 - ID8	0.2305 > 0.05	The null hypothesis is not rejected
ID8 - ID9	0.2586 > 0.05	The null hypothesis is not rejected
ID6 - ID8	0.2736 > 0.05	The null hypothesis is not rejected
ID5 - ID8	0.3392 > 0.05	The null hypothesis is not rejected
ID4 - ID8	0.4342 > 0.05	The null hypothesis is not rejected
ID7 - ID8	0.7677 > 0.05	The null hypothesis is not rejected

Table 5.51: *P*-values from Wilcoxon signed rank test (3-obj. app.).

The Figure 5.73 shows the set of achieved non-dominated solutions regarding all executions, configurations and methods. The detail of such solutions is shown in the Table 5.52, where they are ordered from the left to the right side of the figure (solutions with worse Unavailability are firstly ordered). The Unavailability (Q), the Cost and the optimum times to start a scheduled preventive maintenance task regarding each device are displayed in such a table. The solutions without a redundant Prt. IED (marked as O in the Figure 5.73 and with identifiers Id1 to Id3 in the Table 5.52) are the more economic and less reliable solutions. Conversely, the solutions with a redundant Prt. IED (marked as × in the Figure 5.73 and with identifiers Id4 to Id11 in the Table 5.52) are the more expensive and reliable solutions, as it is expected. It can be seen that the solutions Id1 to Id3 do not present values regarding the redundant Prt. IED because such a device is not included for the design. On the other hand, speaking in general, the times to start a preventive maintenance task present the trend of being set close to the maximum value for such a variable, which is 4,380 hours (see TM_{max} values from the Table 5.46). This is because the optimisers try to adjust the value as much as possible.

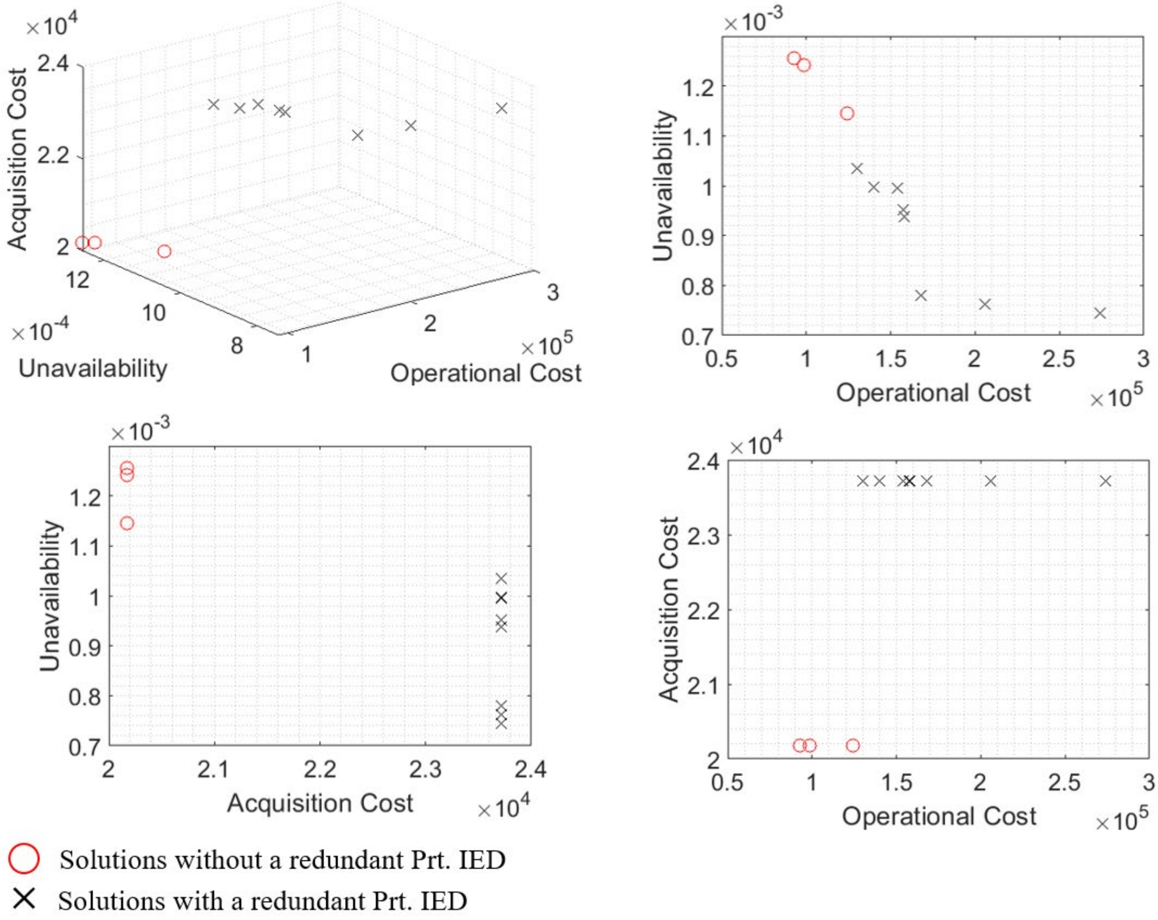


Figure 5.73: Accumulated non-dominated front (3-obj. app.).

Id	Q	O. Cost [€]	A. Cost [€]	TS [h]	MU [h]	Cnt. IED [h]	ESW [h]	Prt. IED [h]	Prt. IED [h]
1	0.001256	92645.00	20175.00	4295	3937	4270	4140	4174	0
2	0.001242	98637.50	20175.00	4302	4303	4360	4355	4050	0
3	0.001145	124582.50	20175.00	4364	4225	4088	4364	4378	0
4	0.001035	130177.50	23712.50	4282	4238	4294	3974	4263	3989
5	0.000997	139962.50	23712.50	4316	4358	4351	4328	3953	4337
6	0.000995	153815.00	23712.50	4380	4378	4279	4227	4214	4344
7	0.000951	157527.50	23712.50	4355	4369	4234	4369	4374	3726
8	0.000938	158160.00	23712.50	4275	4275	4364	4260	4364	4178
9	0.000780	167982.50	23712.50	3758	4221	4221	4365	4336	4317
10	0.000763	206047.50	23712.50	4380	4380	4380	4357	4122	3537
11	0.000744	274085.00	23712.50	4380	4380	4380	4380	3946	3889

Table 5.52: Non-dominated solutions (3-obj. app.).

Finally, the front of solutions presents an accumulated Hypervolume value (computed as it was described by Fonseca et. all [211]) of 4.1702. As it is expected, such a value is higher than 4.1269, the maximum value that is displayed in the Table 5.50.

5.5.3.3. Discussion regarding the results from both approaches.

The Sections 5.5.3.1 and 5.5.3.2 show the results of solving the case study when two (Unavailability and Cost, which includes both Acquisition and Operational Costs) and three objectives (Unavailability, Acquisition Cost and Operational Cost) were considered. The robustness of considering the two-objective approach is brought into the light since non-significant statistical differences were found among the configurations. Nevertheless, significant statistical differences were found when the three-objective approach was employed. Such a situation claims that the problem to be solved is harder. In this case, the order supplied by the Friedman's test claims that the SMS-EMOA method with real encoding reaches a better position than the NSGA-II method with real encoding. Furthermore, the SMS-EMOA method with real encoding and 0.5 and 1 gene per chromosome as mutation probabilities achieved better positions than the NSGA-II method with binary encoding. Finally, the SMS-EMOA method with real encoding achieves a better position than such a method with binary encoding. Therefore, the SMS-EMOA with real encoding and 0.5 or 1 gene per chromosome as mutation rates could be recommended to solve this problem.

The Table 5.48 shows the configurations that present the best average ranks from the Friedman's test point of view, which are ID2 (NSGA-II, real encoding and 1.0 gene per chromosome as a mutation rate) and ID7 (SMS-EMOA, real encoding and 0.5 gene per chromosome as a mutation rate) when two objectives were considered. Furthermore, the Table 5.50 shows the configurations that present the best average ranks from the Friedman's test point of view, which are ID8 (SMS-EMOA, real encoding and 1.0 gene per chromosome as a mutation rate) and ID7 (SMS-EMOA, real encoding and 0.5 gene per chromosome as a mutation rate) when three objectives were considered. All of them are considered to compare the performance between the two-objective problem and the multi-objectivised three-objective problem.

5.5.3.4. Comparing the achieved solutions.

In order to compare the solutions regarding the two-objective problem and the solutions in relation to the three-objective problem, this one must be transformed. Such a transformation consists of adding the Operational Cost and the Acquisition Cost before computing the Hypervolume. In the Table 5.53 (columns 1 to 5), the relationship between methods and configuration identifiers is shown.

Id	Objectives	Method	Encoding	Mutation	Av.	Med.	Max.	Min.	St. D.	Rank
ID1	2	NSGA-II	Real	1.0	3.4495	3.4456	3.4898	3.4141	0.0192	2.5238
ID2	2	SMSEMOA	Real	0.5	3.4471	3.4474	3.4800	3.4166	**0.0150**	2.7142
ID3	3	SMSEMOA	Real	1.0	**3.4517**	**3.4513**	**3.4965**	**3.4274**	0.0166	**2.3809**
ID4	3	SMSEMOA	Real	0.5	3.4513	3.4470	**3.4965**	3.4191	0.0225	2.3810
p-Value										0.8150

Table 5.53: Id's, conf. and Hypervolume statistics (2- and 3- obj. app.).

On the other side, in the Figure 5.74, the Hypervolume average values evolution versus the number of evaluations is displayed. The configuration with identifier ID3 (SMS-EMOA, real encoding and 1 gene per chromosome as a mutation rate) reaches the best Hypervolume average value.

In the Figure 5.75, box plots of the Hypervolume values distribution are shown. Such box plots summarise the statistical information included in the Table 5.53 (columns 6 to 10). It can be seen that the configuration ID3 (SMS-EMOA, real encoding and 1.0 gene per chromosome as a mutation rate) achieves the best Hypervolume average (Av.), median (Med.), maximum (Max.) and minimum (Min.) values. However, the configuration ID2 (SMS-EMOA, real encoding and 0.5 gene per chromosome as a mutation rate) reaches the smallest standard deviation (St. D.) value.

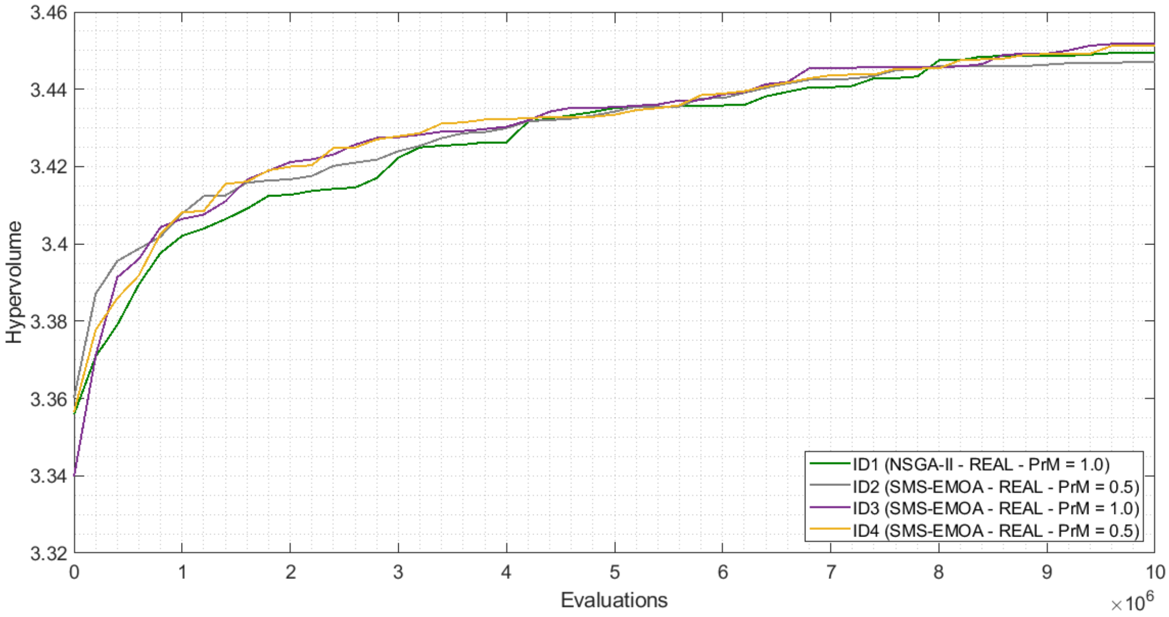


Figure 5.74: Hypervolume Average vs. evaluations (2- and 3- obj. app.).

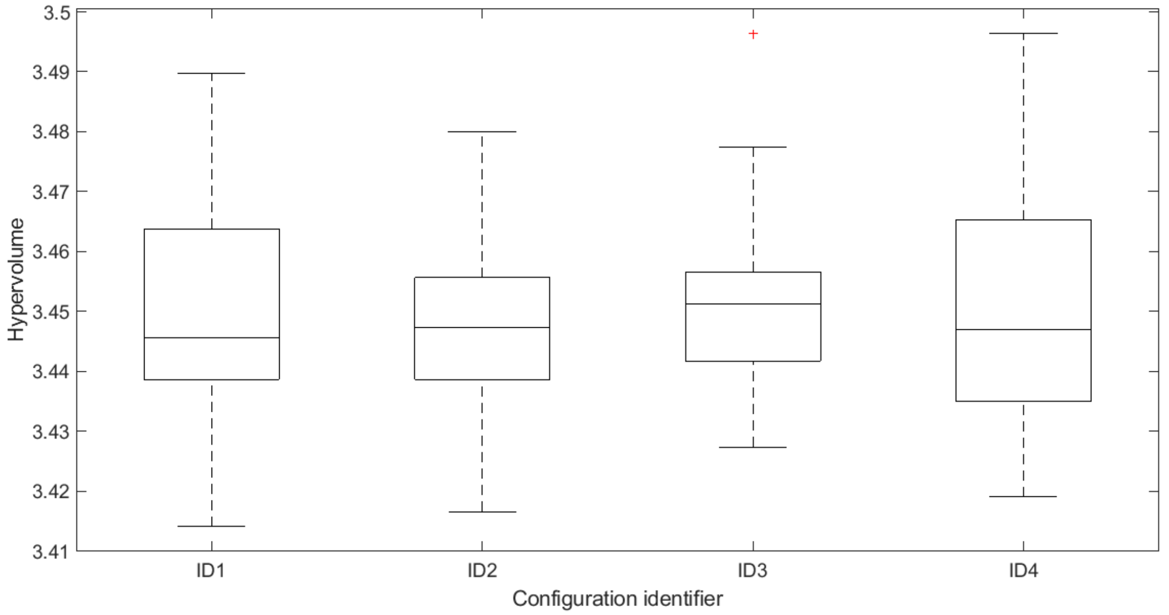


Figure 5.75: Hyperv. box plots, id's as in the Table 5.53 (2- and 3- obj. app.).

A statistical hypothesis test was carried out to conclude if any configuration reaches a better performance. The Friedman's test was employed to compute the average ranks, which are shown in the Table 5.53 (column 11). The configuration ID3 (SMS-EMOA, real encoding and 1 gene per chromosome as a mutation rate) reaches the best average rank. However, the *p*-value computed (0.8150) does not allow rejecting H_0 (*p*-value > 0.05). Therefore, it is not possible to conclude that any configuration performs better than any other. Nevertheless, the three-objective problem was better ordered regarding the Friedman's test. Thus, using the three-objective approach and the SMS-EMOA method with real encoding could be recommend in order to solve the considered problem.

The non-dominated solutions achieved at the end of the process for all configurations and methods and from both two and three-objective approaches are shown in the Figure 5.76. It can be seen both solutions from the two-objective approach (marked as O) and solutions from the three-objective approach (marked as ×). The decision makers should decide the preferable design by considering their Unavailability-Cost requirements. The front of solutions presents an accumulated Hypervolume value of 3.5539. As it is expected, the value is higher than 3.4965, the highest value that is displayed in the Table 5.53.

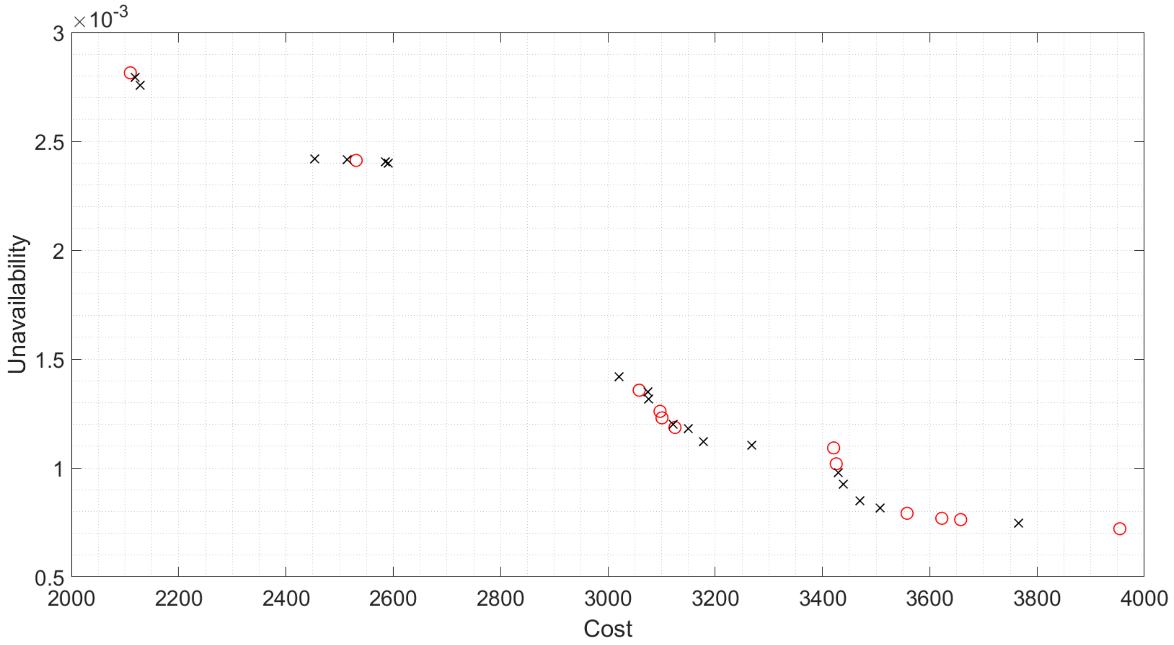


Figure 5.76: Hyperv. box plots, id's as in the Table 5.53 (2- and 3- obj. app.).

6. CONCLUSIONS AND FUTURE RESEARCH.

6.1. Conclusions.

In this research, coupling Multi-objective Evolutionary Algorithms and Discrete Event Simulation has been employed and explored in order to tackle simultaneously both the optimisation of systems' design (based on a process of including automatic structure of components and redundant devices) and also their maintenance strategy (based on the implementation of periodic preventive maintenance tasks), whilst addressing the conflict between Availability and Cost. Coupling these techniques had been previously used to explore the problems separately but not simultaneously, at least when both the corrective and the preventive maintenance - consisting in achieving the optimum period of time to conduct a preventive maintenance task - are taken into account. The Multi-objective Evolutionary Algorithm gave rise to a population of individuals, each encoding one design alternative and one preventive maintenance strategy.

Each individual represents a possible solution to the problem, where decision variables regarding the structural design and decision variables regarding the maintenance strategy coexist. Such an idea implies that decision variables with a different nature must be simultaneously considered by the evolutionary process so several transformations must be applied. On the one hand, real encoding was used when decision variables both with binary and integer nature for the design and decision variables with integer nature for the maintenance strategy coexisted. On the other hand, binary encoding was used when decision variables with binary nature for the design and decision variables with integer nature for the maintenance strategy coexisted. In all cases the above-mentioned transformations worked satisfactorily.

Once the Multi-objective Evolutionary Algorithm supplies a population of individuals, they are employed to modify and evaluate the system's Functionability Profile by employing Discrete Event Simulation. The individuals are evolved generation after generation till reaching the stopping criterion.

The explained process was firstly applied to a technical system in a case study in which the performance of five state-of-the-art Multi-objective Evolutionary Algorithms (SMS-EMOA, MOEA/D, MOEA/D-DE, NSGA-II and GDE3) was compared and a set of optimum non-dominated solutions were obtained. It can be concluded that using Multi-objective Evolutionary Algorithms and Discrete Event Simulation to address the joint optimisation of systems' design and their maintenance strategy provides Availability-Cost balanced solutions to real world problems where data based on field experience were used. Moreover, in the solved case study, regarding the Multi-objective Evolutionary Algorithms, the best performance was found when methods based on both the Hypervolume Indicator (SMS-EMOA) and Pareto dominance relation (NSGA-II and GDE3) were used, rather than when methods based on Decomposition (MOEA/D and MOEA/DDE) were used. However, the operator used to create new individuals does not appear to have a relevant effect since methods that use Simulated Binary Crossover (SMS-EMOA and NSGA-II) presented similar performance than methods that use Differential Evolution (GDE3); also, in the case of MOEA/D versus MOEA/D-DE.

Once the case study was solved, the configurations with best performance were identified and a discussion was opened. On the one hand, the effect of sampling size and its minimal extreme direction was analysed, whose results enhance the benefits of the proposed methodology by showing the positive synergy among Discrete Event Simulation and Multi-objective Evolutionary Algorithms, where only a single simulation per individual is enough in the fitness function evaluation to attain very competitive results. These results are confirmed with an analysis based on the average values of both objective functions for each non-dominated solution of each compared configuration. The proposed methodology is a computationally efficient and robust approach (non-parameter dependent regarding the number of samples or the minimal search direction) versus the use of Monte Carlo simulation-based approaches when facing the multi-objective optimization reliability problem handled.

On the other hand, the economic benefits of using the methodology to determine the optimum structural design of the system and its maintenance strategy were quantified in the case study being in the estimated interval of 4-10%.

Next, the proposed methodology scalability and generalisation was demonstrated when it was applied to two more complex applications. Both problems were satisfactorily solved, and insights about a proper chromosome codification regarding the design components were obtained from the executed experiments.

Once the methodology was defined, developed, implemented, and tested both in a case study and in more complex systems, a deeper study was conducted. Such a study consisted of: Firstly, an encoding experiment to compare the performance of seven encoding types (real, standard binary with one-point, two-point and uniform crossover, and Gray code with one-point, two-point and uniform crossover), and secondly, an accuracy level encoding which consisted of comparing the performance of using standard binary encoding with accuracy levels across a range of time units (hours, days and weeks) with impact in the form of the length of the chromosome (the smaller the time unit, the bigger the chromosome). The Multi-objective Evolutionary Algorithm used in this case was the NSGA-II method due to such a method previously resulted very competitive. Using the case study previously defined to conduct such experiments, a set of optimum non-dominated solutions were obtained for all cases.

Regarding the encoding experiment, the two-point crossover standard binary encoding resulted the best ordered method from the Friedman’s test point of view (based on the final Hypervolume indicator distributions), although no statistically significant differences were observed. Regarding the accuracy experiment, the two-point crossover standard binary encoding with the hour as a time unit resulted the best ordered method from the Friedman’s test point of view, although no statistically significant differences were observed. An important conclusion arises from this last experiment, which relates to flexibility regarding the time unit to schedule the preventive maintenance tasks. Using the hour, the day or the week as a time unit does not significantly affect the performance of the configurations so, in the studied conditions, the preventive maintenance tasks can be planned by using weeks as a time unit. This allows a better range of time for planning than if the day or the hour are used as a time unit.

Once several Multi-objective Evolutionary Algorithms, encodings and accuracy levels were studied, the previously presented case study is explored again but, in this occasion, two multi-objective approaches were considered. On the one hand, a two-objective approach where Availability and Cost are the evaluated objectives. On the other hand, a three-objective approach where Availability and Cost are the evaluated objectives again. However, in this case such a Cost is decomposed between Acquisition and Operational Cost under a multi-objectivisation approach. In order to find the more suitable approach, a thorough hypothesis test is conducted. The performance of two state-of-the-art Multi-objective Evolutionary Algorithms (SMS-EMOA and NSGA-II) is compared when several configurations of such methods are considered. Real and binary encoding are tested too. On the one hand, non-significant statistical differences were found when the two-objective approach was employed. However, the best ordered configuration was achieved when the SMS-EMOA method is used with binary encoding and 1 gene per chromosome as a mutation rate. On the other hand, no significant statistical differences were found when the three-objective approach was employed. However, the best ordered configuration was achieved when the NSGA-II method is used with real encoding and 1.5 gene per chromosome as a mutation rate.

Next, the best configurations from both experiments were compared and significant statistical differences were found. Therefore, it is possible to conclude that the applied methodology works better when multi-objectivisation is employed. Furthermore, the NSGA-II method with real encoding and 1.5 gene per chromosome as a mutation rate resulted best ordered from the Friedman’s test point of view, and with the highest hypervolume indicator average, median and maximum values. Therefore, such a configuration and approach could be recommended in order to solve the problem of achieving the optimum design alternatives and maintenance strategies, which supplied the best Availability-Cost balance.

Finally, the use of the methodology was extended to a different engineering field in order to supply reliable architectural designs of Substation Communication Networks. A specific case study is explored by applying the methodology and the conclusions previously reached. Again, two multi-objective approaches were considered. On the one hand, a two-objective approach where Availability and Cost are the evaluated objectives. On the other hand, a three-objective approach where Availability and Cost are the evaluated objectives (in this case such a Cost is decomposed in Acquisition and Operational Cost under the multi-objectivisation approach). Applying the methodology provided Availability-Cost balanced solutions. In order to find the more suitable approach, a thorough hypothesis test was conducted. This process is applied to a section of a subsystem that follows the IEC 61850 standard, which is a bay line of a single bus, small transmission substation to transform energy from 220 kV. to 132 kV. The performance of two state-of-the-art Multi-objective Evolutionary Algorithms (SMS-EMOA and NSGA-II) was compared when several configurations of such methods were considered.

In this case, on the one hand, non-significant statistical differences were found when the two-objective approach was employed. However, the best ordered configuration was achieved when the NSGA-II method was used with real encoding and 1 gene per chromosome as a mutation rate. On the other hand, significant differences were found when the three-objective approach was employed. The best performance was obtained when the SMS-EMOA method is employed with real encoding and 1 gene per chromosome as a mutation rate. Finally, the best configurations from both experiments were compared and non-significant statistical differences were found. However, the best order regarding the Friedman’s test point of view was achieved when the three-objective approach was considered. Therefore, it can be concluded that the methodology applied is robust due to the fact that non-significant statistical differences were found among using two or three objectives. However, the multi-objectivisation approach presented a best order regarding the Friedman’s test so a slight positive effect was seen.

Furthermore, the SMS-EMOA method with real encoding and 1 gene per chromosome as a mutation rate resulted best ordered from the Friedman’s test point of view, and with the highest hypervolume indicator average, median, maximum and minimum values. Therefore, such a configuration and approach could be recommended in order to achieve the optimum design alternatives and maintenance strategies, which supplied the best Availability-Cost balance. As it can be seen, the methodology has been successfully extended to solve problems from a different field on the engineering, which is promising.

6.2. Future research.

Regarding the future of this research, several lines are opened:

- The casuistry regarding the system’s Reliability that has been dealt along the present research was clearly defined and delimited. This considers basically the following aspects, which could be extended as it is commented:
    - Two states are considered for the system’s devices. A device can be either in operating or in recovery state. Multi-state systems could be considered by attending deterioration states.
    - After repairs or corrective maintenance activities, all devices recover the operating state as-good-as-new. Imperfect repairs could be considered.
    - Active redundancies are considered so the system fulfil the required function while opportune redundancies work. Other types of redundancies could be considered such as cold, warm, or hot standby redundancies.
    - Each device is considered like a single unit from the maintenance point of view. A single-unit device is a device which cannot be decomposed in lower maintainable levels. Multi-unit devices could be considered and/or several failure modes too.

  - The preventive maintenance activities are scheduled by attending to the time. Such activities might be planned by attending to the age, use or condition.
  - The preventive maintenance tasks start immediately once a device does not satisfy the required function so the continuous monitorization of the system is considered. This circumstance could be not considered so testing the device's state could be attended in order to initiate preventive maintenance tasks.
  - Non-dependencies among devices are considered so each device works isolated. Dependencies among devices could be considered.
- Regarding the Multi-objective Evolutionary Algorithms, several state-of-the-art methods were tested along the present research. They are considered as standards methods. More modern methods could be employed in order to compare their performances. Moreover, when the multi-objectivisation approach was explored, the more complex the systems, the more statistical difference regarding the performance was observed. It could be deeper explored by applying the methodology to more complex architectures.

REFERENCES.

[1]. B. Galván, Contribuciones a la evaluación cuantitativa de árboles de fallos, Tésis Doctoral, Programa de Doctorado: Física Fundamental y Aplicada, Universidad de Las Palmas de Gran Canaria, 1999, http://hdl.handle.net/10553/2059.

[2]. L. González, D. García, B. Galván, "An intrinsic order criterion to evaluate large, complex fault trees", *IEEE Transactions on Reliability*, vol. 53, no. 3, pp. 297–305, 2004, doi: 10.1109/TR.2004.833307.

[3]. B. Galván, A. Marco, J. Rolin, L. Delauney, "NeXOS contribution to the adaptation of system analysis engineering tools for mature and reliable ocean sensors", *2014 IEEE Sensor Systems for a Changing Ocean,* SSCO 2014, doi: 10.1109/SSCO.2014.7000370.

[4]. J. Rolin, L. Delauney, B. Bigourdan, F. Salvetat, B. Galván, Y. Elejalde, "Engineering approach (reliability analysis, metrology and environmental testing procedures) for newly developed marine sensors", *Oceans Aberdeen Conference*, 2017, doi: 10.1109/OCEANSE.2017.8084986.

[5]. D. Greiner, B. Galván, G. Winter. Safety Systems Optimum Design by Multicriteria Evolutionary Algorithms. In *"Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science"*, Second International Conference, EMO 2003, Faro, Portugal, 2003; Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L., Eds.; Springer: Berlin, Germany, pp. 722-736, 2003.

[6]. S. Martorell, S. Carlos, J.F. Villanueva, A.I Sanchez, B. Galván, D. Salazar, M. Cepin, "Use of multiple objective evolutionary algorithms in optimizing surveillance requirements", *Reliability Engineering & System Safety*, Vol. 91, Issue 9, pp. 1027-1038, 2006, doi:10.1016/j.ress.2005.11.038.

[7]. D. Salazar, C.M. Rocco, B.J. Galván, "Optimization of constrained multiple-objective reliability problems using evolutionary algorithms", *Reliability Engineering & System Safety*, Vol. 91, Issue 9, pp. 1057–1070, 2006, ISSN 0951-8320, doi.org/10.1016/j.ress.2005.11.040.

[8]. B. Galván, G. Winter, D. Greiner, D. Salazar, "New Evolutionary Methodologies for Integrated Safety System Design and Maintenance Optimization", In: G. Levitin (ed). *Computational Intelligence in Reliability Engineering: Evolutionary Techniques in Reliability Analysis and Optimization*, Springer: Berlin, Heidelberg, pp. 151-190, 2007, doi:10.1007/978-3-540-37368-1_5.

[9]. D. Greiner, P. Hajela. "Truss topology optimization for mass and reliability considerations—co-evolutionary multiobjective formulations", *Structural and Multidisciplinary Optimization*, Vol. 45, pp. 589-613, 2012, doi: 10.1007/s00158-011-0709-9.

[10]. D. Greiner, P. Hajela, "Influence of buckling constraints on reliability based multiobjective design of truss structures", *2nd International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, CSC 2011, Crete, Greece, 2011. http://hdl.handle.net/10553/54707.

[11]. K. Misra, Reliability Engineering: A Perspective. Handbook of Performability Engineering, Springer, Ch. 19, pp. 253–289, 2008, doi:10.1007/978-1-84800-131-2_19.

[12]. J. Andrews, T. Moss, Reliability and Risk Assessment, 2nd Edition, The American Society of Mechanical Engineers, 2002.

[13]. H. Zoulfaghari, A.Z. Hamadani, M. Ardakan, "Bi-objective redundancy allocation problem for a system with mixed repairable and non-repairable components", *ISA Transactions*, Vol. 53, Issue 1, pp. 17-24, 2014, ISSN 0019-0578, doi:10.1016/j.isatra.2013.08.002.

[14]. A. Mendes, M. Lorenzoni, "Analysis and optimization of periodic inspection intervals in cold standby systems using Monte Carlo simulation", *Journal of Manufacturing Systems*, Vol. 49, pp. 121-130, 2018, ISSN 0278-6125, doi:10.1016/j.jmsy.2018.09.006.

[15]. D. Simon, Evolutionary Optimization Algorithms, John Wiley & Sons, Inc., Hoboken, New Jersey, 2013.

[16]. C. Coello, "Multi-objective Evolutionary Algorithms in Real-World Applications: Some Recent Results and Current Challenges", In: D. Greiner,

B. Galván, J. Périaux, N. Gauger, K. Giannakoglou and G. Winter (eds). *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, Computational Methods in Applied Sciences*, Vol. 36, pp. 3-18, Springer, Cham, 2015. doi:10.1007/978-3-319-11541-2_1.

[17]. D. Greiner, D. Salazar, B. Galván, "Evolutionary Multiobjective Methods", apuntes de la asignatura Diseño Óptimo en Ingeniería, Máster en Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería. Universidad de Las Palmas de Gran Canaria Instituto de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería, SIANI.

[18]. K. Miettinen, "Some Methods for Nonlinear Multi-objective Optimization". In: E. Zitzler, L. Thiele, K. Deb, C.A. Coello, D. Corne (eds), *Evolutionary Multi-Criterion Optimization, EMO 2001, Lecture Notes in Computer Science*, Vol. 1993, 2001. Springer, Berlin, Heidelberg. doi:10.1007/3-540-44719-9_1.

[19]. D. Greiner, Optimización multiobjetivo de pórticos metálicos mediante algoritmos evolutivos, Tesis doctoral, Programa de doctorado: Sistemas inteligentes y aplicaciones numéricas en Ingeniería, Universidad de Las Palmas de Gran Canaria, 2005, http://hdl.handle.net/10553/20952.

[20]. R. Keeney, H. Raiffa, D. Rajala, "Decisions with Multiple Objectives: Preferences and Value Trade-Offs", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 9, Nº7, pp. 403 - 403, 1979, doi:10.1109/TSMC.1979.4310245.

[21]. J.S. Dyer, R.K. Sarin, "Multicriteria Decision Making, Mathematical Programming for Operations Researchers and Computer Scientics", *In: A.G. Holzman, M. Dekker (eds)*, 123-148, 1981.

[22]. P.C. Fishburn, "Lexicographic Orders, Utilities and Decision Rules: A Survey", *Management Science*, Vol. 20, Nº11, pp. 1442-1471, 1974.

[23]. A. Charnes, W.W. Cooper, "Management Models and Industrial Applications of Linear Programming", Vol. 1, Jhon Wiley & Sons, New York, 1961.

[24]. 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*, Vol. 19, pp. 357-368, 1972.

[25]. R.E. Steuer, "Multiple Criteria Optimization: Theory", *Computation and Applications*, John Wiley & Sons, 1986.

[26]. A.P. Wierzbicki, "A Mathematical Basis for Satisficing Decision Making", *Mathematical Modelling*, Vol. 3 pp. 391-405, 1982.

[27]. J.T. Buchanan, "A Naïve Approach for Solving MCDM Problems: The GUESS Method", *Journal of the Operational Research Society*, Vol. 48, Nº2, pp. 202-206, 1997.

[28]. K. Miettinen, M.M. Mäkelä, "Iterative Bundle-Based Method for Nondifferentiable Multi-objective Optimization: NIMBUS", *Optimization*, Vol. 34, Nº3, pp. 231-246, 1995.

[29]. S. Gass, T. Saaty, "The Computational Algorithm for the Parametric Objective Function", *Naval Research Logistics Quarterly*, Vol. 2, pp. 39-45, 1995.

[30]. L. Zadeh, "Optimality and Non-Scalar-Valued Performance Criteria", *IEEE Transactions on Automatic Control*, Vol. 8, pp. 59-60, 1963.

[31]. Y.Y. Haimes, L.S. Lasdon, D.A. Wismer, "On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization", *IEEE Transactions on Systems, Man. and Cybernetics*, Vol. 1, pp. 296-297, 1971.

[32]. D.E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", *Addison-Wesley Publishing Company*, Reading, Massachusetts, 1989.

[33]. C. Darwin, The Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life. Jhon Murray, Albemarle Street, 1859.

[34]. G.J. Mendel. "Versuche der Pflanzen-hybriden", *Verhandlungen des naturforschenden Vereines in Brünn*, Vol. 1, pp. 3-47, 1865.

[35]. G.J. Mendel. "Experiments in Plant Hybridisation (English article translation from the original article published in 1865)", *Journal of the Royal Horticultural Society*, Vol. 26, pp. 1–32, 1901.

[36]. N. Barricelli, "Esempi numerici di processi di evoluzione", *Methodos*, Vol. 6, pp. 45-68, 1954.

[37]. A. Fraser, “Simulation of genetic systems by automatic digital computers: I. Introduction”, *Australian Journal of Biological Sciences*, Vol. 10, Nº3, pp. 484-491, 1957.

[38]. G. Box, “Evolutionary operation: A method for increasing industrial productivity”, *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, Vol. 6, Nº2, pp. 81-101, 1957.

[39]. I. Rechenberg, “Cybernetic solution path of an experimental problem”, In Fogel, D. (ed.), *Evolutionary Computation: The Fossil Record*, pp. 301-310, 1998. Wiley-IEEE Press. First published in 1964.

[40]. L.J. Fogel, “Artificial Intelligence through Simulated Evolution”, John Wiley, New York, 1966.

[41]. J. Holland, Adaptation in Natural and Artificial Systems. The University of Michigan Press, 1975.

[42]. J.D. Schaffer, Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, Ph.D. thesis, Vanderbilt University, 1984.

[43]. K.A. De Jong. The Analysis of the Behavior of a Class of Genetic Adaptative Systems. PhD tesis, University of Michigan, 1975.

[44]. J.E. Baker, “Reducing bias and inefficiency in the selection algorithm”, In J.J. Grefenstette (ed.), *Proceedings of the 2nd. International Conference on Genetic Algorithms*, pp. 14-21. Laurence Erlbaum Associates, 1987.

[45]. D. Whitley, “The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best”, *International Conference on Genetic Algorithms*, Fairfax, Virginia, pp. 116-121, 1989.

[46]. G. Syswerda, “Uniform Crossover in Genetic Algorithms”, In J.D. Schaffer (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 2-9. Morgan Kaufmann Publishers, San Mateo, California, 1989.

[47]. H. Bremermann, M. Rogson, S. Salaff, “Global properties of evolution processes”. In Pattee, H., Edlsack, E., Fein, L., and Callahan, A. (eds.), *Natural Automata and Useful Simulations*, pp. 3-41, Spartan Books, 1966.

[48]. K. Deb, R. Agrawal, “Simulated binary crossover for continuous search space”, *Complex Systems*, Vol. 9, Nº2, pp. 115-148, 1995.

[49]. D.E. Goldberg, J. Richardson, "Genetic algorithm with sharing for multimodal function optimization," In John J. Grefenstette (ed.), *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41-49, Hillsdale, New Jersey, Lawrence Erlbaum, 1987.

[50]. E. Zitzler, L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, Nº 4, pp. 257–271, 1999, doi:10.1109/4235.797969.

[51]. F. Kursawe, "A variant of evolution strategies for vector optimization", In: Schwefel HP., Männer R. (eds), *Parallel Problem Solving from Nature, PPSN 1990, Lecture Notes in Computer Science*, vol. 496, Springer, Berlin, Heidelberg, 1991, doi.org/10.1007/BFb0029752.

[52]. N. Srinivas, K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms", *Evolutionary Computation*, Vol. 2, Nº3, pp. 221-248, 1994, doi:10.1162/evco.1994.2.3.221.

[53]. J. Horn, N. Nafpliotis, D.E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Orlando, FL, Vol.1, pp. 82-87, 1994, doi:10.1109/ICEC.1994.350037.

[54]. C.M. Fonseca, P.J. Fleming, "Genetic algorithms for multiobjective Optimization: Formulation, discussion and generalization", In Stephanie Forrest (ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416–423, San Mateo, California, University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers, 1993.

[55]. C.A. Coello, "Evolutionary Multi-Objective Optimization: A Historical View of the Field", *IEEE Computational Intelligence Magazine,* Vol. 1, Nº1, pp. 28-36, 2016, doi: 10.1109/MCI.2006.1597059.

[56]. J.D. Knowles, D.W. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000, doi:10.1162/106365600568167.

[57]. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, Nº2, pp. 182-197, 2002, doi:10.1109/4235.996017.

[58]. Corne D.W., Knowles J.D., Oates M.J., "The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization", In: Schoenauer M. et al. (eds.), *Parallel Problem Solving from Nature PPSN VI. PPSN 2000. Lecture Notes in Computer Science*, Vol. 1917, Springer, Berlin, Heidelberg, 2000. https://doi.org/10.1007/3-540-45356-3_82.

[59]. E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," In K. Giannakoglou et al. (eds.), *EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pp. 95-100, Athens, Greece, 2002, doi.org/10.3929/ethz-a-004284029.

[60]. K. Deb, S. Jain, "Running performance metrics for evolutionary multi-objective optimization", *Tech. Rep., Kanpur Genetic Algorithms Laboratory, ITT Kanpur*, 2002.

[61]. M. Emmerich, A. Deutz, "A tutorial on multiobjective optimization: fundamentals and evolutionary methods", *Natural Computing,* Vol. 17, pp. 585-609, 2018, doi:10.1007/s11047-018-9685-y.

[62]. D. Greiner, J. Periaux, J. Emperador, B. Galván, G. Winter, "Game Theory Based Evolutionary Algorithms: A Review with Nash Applications in Structural Engineering Optimization Problems", *Archives of Computational Methods in Engineering*, Vol. 24, pp. 703-750, 2017. doi:10.1007/s11831-016-9187-y.

[63]. N. Beume, B. Naujoks, M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume", *European Journal of Operational Research*, Vol. 181, pp. 1653-1669, 2007, doi:10.1016/j.ejor.2006.08.008.

[64]. S. Jiang, J. Zhang, Y.S. Ong, A.N. Zhang, P.S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Transactions on Cybernetics*, Vol. 45, Nº10, pp. 2202-2213, 2015. doi:10.1109/TCYB.2014.2367526.

[65]. J. Bader, E. Zitzler, “HypE: An algorithm for fast hypervolume-based many-objective optimization*”, Evolutionary Computation*, Vol. 19, Nº1, pp. 45-76, 2011, doi:10.1162/EVCO_a_00009.

[66]. Q. Zhang, H. Li, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition”, *IEEE Transactions on Evolutionary Computation*, Vol. 11, Nº6, pp. 712-731, 2007, doi:10.1109/TEVC.2007.892759.

[67]. K. Deb, H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints”, *IEEE Transactions on Evolutionary Computation*, Vol. 18, Nº4, pp. 577-601, 2014, doi:10.1109/TEVC.2013.2281535.

[68]. H. Ishibuchi, T. Murata, “A multi-objective genetic local search algorithm and its application to flowshop scheduling”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 28, Nº3, pp. 392-403, 1998, doi:10.1109/5326.704576.

[69]. S. Kukkonen, J. Lampinen, “GDE3: The third evolution step of generalized differential evolution”, *IEEE Congress on Evolutionary Computation, Edinburgh, Scotland*, Vol. 1, pp. 443-450, 2005, doi:10.1109/CEC.2005.1554717.

[70]. R. Storn, K. Price, “Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces”, *Journal of Global Optimization*, Vol. 11, Nº4, pp. 341-359, 1997.

[71]. H. Li, Q. Zhang, “Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II”, *IEEE Transactions on Evolutionary Computation*, Vol. 13, Nº2, pp. 284-302, 2009. doi:10.1109/TEVC.2008.925798.

[72]. E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, V. D. Fonseca, “Performance assessment of multiobjective optimizers: Analysis and review”, *IEEE Transactions on Evolutionary Computation*, Vol. 7, Nº2, pp. 117-132, 2003, doi:10.1109/TEVC.2003.810758.

[73]. M. Fleischer, “The Measure of Pareto Optima Applications to Multi-objective Metaheuristics”, In: Fonseca C.M., Fleming P.J., Zitzler E., Thiele L., Deb K.

(eds.), *Evolutionary Multi-Criterion Optimization. EMO 2003. Lecture Notes in Computer Science*, Vol. 2632, Springer, Berlin, Heidelberg, 2003, doi.org/10.1007/3-540-36970-8_37.

[74]. N. Riquelme, C. Von Lücken, B. Baran, "Performance metrics in multi-objective optimization," *2015 Latin American Computing Conference (CLEI)*, pp. 1-11, Arequipa, 2015, doi: 10.1109/CLEI.2015.7360024.

[75]. T. Okabe, Y. Jin, B. Sendhoff, "A critical survey of performance indices for multi-objective optimisation", *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, Canberra, ACT, Australia, 2003, Vol. 2, pp. 878-885, doi:10.1109/CEC.2003.1299759.

[76]. D.A. Van Veldhuizen, “Multiobjective evolutionary algorithms: classifications, analyses, and new innovations,” *DTIC Document, Tech. Rep.*, 1999.

[77]. J.B. Kollat, P.M. Reed, “The Value of Online Adaptive Search: A Performance Comparison of NSGAII, ε-NSGAII and εMOEA”, In Coello Coello C.A., Hernández Aguirre A., Zitzler E. (eds.), *Evolutionary Multi-Criterion Optimization. EMO 2005. Lecture Notes in Computer Science*, Vol. 3410, Springer, Berlin, Heidelberg, 2005, doi.org/10.1007/978-3-540-31880-4_27.

[78]. E. Zitzler, D. Brockhoff, L. Thiele, “The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration”, In: Obayashi S., Deb K., Poloni C., Hiroyasu T., Murata T. (eds.), *Evolutionary Multi-Criterion Optimization, EMO 2007, Lecture Notes in Computer Science*, Vol. 4403, Springer, Berlin, Heidelberg, 2007, doi.org/10.1007/978-3-540-70928-2_64.

[79]. D. Coit, E. Zio, “The evolution of system reliability optimization”, *Reliability Engineering and System Safety*, 192, 2019, 106259, ISSN 0951-8320, doi:10.1016/j.ress.2018.09.008.

[80]. R.E. Bellman, “The theory of dynamic programming”, RAND Corporation, Santa Monica, CA., 1954.

[81]. G.B. Dantzig, A. Orden, P. Wolfe, “The generalized simplex method for minimizing a linear form under linear inequality restraints”, *Pacific Journal of Mathematics,* Vol. 5, Nº 2, 1955.

[82]. F.A. Tillman, C. Hwang, W. Kuo, "Optimization Techniques for System Reliability with Redundacy - A Review", *IEEE Transactions on Reliability*, Vol. R-26, Nº 3, pp. 148-155, 1977, doi: 10.1109/TR.1977.5220100.

[83]. J.H. Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence", MIT press, 1992.

[84]. G. Levitin, A. Lisnianski, H. Ben-Haim, D. Elmakis, "Redundancy optimization for series-parallel multi-state systems", *IEEE Transactions on Reliability,* Vol. 47, Nº 2, pp. 165-72, 1998, doi: 10.1109/24.722283.

[85]. R.Y. Rubinstein, G. Levitin, A. Lisnianski, H. Ben-Haim, "Redundancy optimization of static series-parallel reliability models under uncertainty", *IEEE Transactions on Reliability*, Vol. 46, Nº 4, pp. 503-511, 1997, doi:10.1109/24.693783.

[86]. D.W. Coit, "Maximization of system reliability with a choice of redundancy strategies", *IIE Transactions*, Vol. 35, Nº 6, pp. 535-543, 2003, doi:10.1080/07408170304420.

[87]. M. Yildirim, X.A. Sun, N.Z. Gebraeel, "Sensor-driven condition-based generator maintenance scheduling-part II: incorporating operations", *IEEE Transactions on Power Systems*, Vol. 31, Nº 6, pp. 4263-4271, 2016, doi:10.1109/TPWRS.2015.2506604.

[88]. P. Ramanan, M. Yildirim, E. Chow, N. Gebraeel, "Asynchronous decentralized framework for unit commitment in power systems", *Procedia Computer Science*, Vol. 108, pp. 665-674, 2017, ISSN 1877-0509, doi.org/10.1016/j.procs.2017.05.038.

[89]. W.L. Oberkampf, S.M. DeLand, B.M. Rutherford, K.V. Diegert, K.F. Alvin, "Error and uncertainty in modeling and simulation", *Reliability Engineering & System Safety*, Vol. 75, Issue 3, pp. 333-357, 2002, ISSN 0951-8320, doi.org/10.1016/S0951-8320(01)00120-X.

[90]. OREDA, OREDA – "Offshore Reliability Data Handbook", 5th Edition, OREDA participants, prepared by: SINTEF, Distributed by: Det Norske Veritas (DNV), 2009.

[91]. C. for Chemical Process Safety, “Guidelines for Process Equipment Reliability data with data tables”, Centre for Chemical Process Safety of the American Institute of Chemical Engineers, 1998.

[92]. B. de Jonge, P.A. Scarf, “A review on maintenance optimization”, *European Journal of Operational Research*, Vol. 285, Issue 3, pp. 805-824, 2020, ISSN 0377-2217, doi.org/10.1016/j.ejor.2019.09.047.

[93]. R. Bellman, S. Dreyfus, “Dynamic Programming and the Reliability of Multicomponent Devices”, *Operations Research,* Vol. 6, Nº2, pp. 200-206, 1958. Retrieved January 15, 2021, from http://www.jstor.org/stable/167613.

[94]. P. Ghare, R. Taylor, “Optimal Redundancy for Reliability in Series Systems”, *Operations Research*, Vol. 17, Nº 5, pp. 838-847, 1969. Retrieved January 15, 2021, from http://www.jstor.org/stable/168358.

[95]. L. Painton, J. Campbell, "Genetic algorithms in optimization of system reliability", *IEEE Transactions on Reliability*, Vol. 44, Nº. 2, pp. 172-178, 1995, doi: 10.1109/24.387368.

[96]. D.W. Coit, A.E. Smith, “Reliability optimization of series-parallel systems using a genetic algorithm”, *IEEE Transactions on Reliability*, Vol. 45, Nº 2, pp. 254-260, 1996, doi: 10.1109/24.510811.

[97]. M. Cantoni, M. Marzaguerra, E. Zio, “Genetic algorithms and Monte Carlo simulation for optimal plant design”, Reliability Engineering and System Safety, Vol. 68, pp. 29-38, 2000, doi:10.1016/S0951-8320(99)00080-0.

[98]. M. Ouzineb, M. Nourelfath, M. Gendreau, “Availability optimization of series-parallel multi-state systems using a tabu search meta-heuristic”, *Proceedings ICSSSM’06 2006 Int. Conf. Serv. Syst. Serv. Manag.*, Vol. 2, pp. 953-958, 2006.

[99]. O. Bendjeghaba, D. Ouahdi, “Multi-agent ant system for redundancy allocation problem of multi states power system”, *2008 IEEE 2nd International Power and Energy Conference*, Johor Bahru, pp. 1270-1274, 2008, doi: 10.1109/PECON.2008.4762662.

[100]. Ying-Shen Juang, Shui-Shun Lin, Hsing-Pei Kao, “A knowledge management system for series-parallel availability optimization and design”,

Expert Systems with Applications, Vol. 34, Issue 1, pp. 181-193, 2008, ISSN 0957-4174, doi.org/10.1016/j.eswa.2006.08.023.

[101]. D. Zou, L. Gao, S. Li, J.Wu, “An effective global harmony search algorithm for reliability problems”, *Expert Systems with Applications*, Vol. 38, Issue 4, pp. 4642-4648, 2011, ISSN 0957-4174, doi.org/10.1016/j.eswa.2010.09.120.

[102]. Y. Wang, L. Li, "Heterogeneous Redundancy Allocation for Series-Parallel Multi-State Systems Using Hybrid Particle Swarm Optimization and Local Search", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 42, Nº 2, pp. 464-474, 2012, doi:10.1109/TSMCA.2011.2159585.

[103]. E. Valian, S. Tavakoli, S. Mohanna, A. Haghi, “Improved cuckoo search for reliability optimization problems”, *Computers & Industrial Engineering*, Vol. 64, Issue 3, pp. 459-468, 2013, ISSN 0360-8352, doi.org/10.1016/j.cie.2012.07.011.

[104]. P. Pourkarim Guilani, M. Sharifi, S.T.A. Niaki, A. Zaretalab, “Redundancy Allocation Problem of a System with Three-state Components : A Genetic Algorithm, International Journal of Engineering, Vol. 27, Nº 11, pp. 1663-1672, 2014.

[105]. Tsung-Jung Hsieh, “Hierarchical redundancy allocation for multi-level reliability systems employing a bacterial-inspired evolutionary algorithm”, *Information Sciences*, Vol. 288, pp. 174-193, 2014, ISSN 0020-0255, doi.org/10.1016/j.ins.2014.07.055.

[106]. Y. Xu, D. Pi, S. Yang, Y. Chen, “A novel discrete bat algorithm for heterogeneous redundancy allocation of multi-state systems subject to probabilistic common-cause failure”, *Reliability Engineering & System Safety*, Vol. 208, 107338, 2021, ISSN 0951-8320, doi.org/10.1016/j.ress.2020.107338.

[107]. C. Elegbede, K. Adjallah, “Availability allocation to repairable systems with genetic algorithms: a multi-objective formulation”, *Reliability Engineering & System Safety*, Vol. 82, Issue 3, pp. 319-330, 2003, ISSN 0951-8320, doi.org/10.1016/j.ress.2003.08.001.

[108]. L. Sahoo, A.K. Bhunia, P.K. Kapur, "Genetic algorithm based multi-objective reliability optimization in interval environment", *Computers & Industrial Engineering*, Vol. 62, Issue 1, pp. 152-160, 2012, ISSN 0360-8352, doi.org/10.1016/j.cie.2011.09.003.

[109]. Z. Li, H. Liao, D.W. Coit, "A two-stage approach for multi-objective decision making with applications to system reliability optimization", *Reliability Engineering & System Safety*, Vol. 94, Issue 10, pp. 1585-1592, 2009, ISSN 0951-8320, doi.org/10.1016/j.ress.2009.02.022.

[110]. A. Azadeh, B. Maleki Shoja, S. Ghanei, M. Sheikhalishahi, "A multi-objective optimization problem for multi-state series-parallel systems: A two-stage flow-shop manufacturing system", *Reliability Engineering & System Safety*, Vol. 136, pp. 62-74, 2015, ISSN 0951-8320, doi.org/10.1016/j.ress.2014.11.009.

[111]. P.G. Busacca, M. Marseguerra, E. Zio, "Multiobjective optimization by genetic algorithms: application to safety systems", *Reliability Engineering & System Safety*, Vol. 72, Issue 1, pp. 59-74, 2001, ISSN 0951-8320, doi.org/10.1016/S0951-8320(00)00109-5.

[112]. M. Marseguerra, E. Zio, L. Podofillini, W. Coit, "Optimal Design of Reliable Network Systems in Presence of Uncertainty", *IEEE Transactions on Reliability*, Vol. 54, Nº 2, pp. 243-253, 2005, doi.org/10.1109/TR.2005.847279.

[113]. Z. Tian, M.J. Zuo, "Redundancy allocation for multi-state systems using physical programming and genetic algorithms", *Reliability Engineering & System Safety*, Vol. 91, Issue 9, pp. 1049-1056, 2006, ISSN 0951-8320, doi.org/10.1016/j.ress.2005.11.039.

[114]. J.H. Zhao, Z. Liu, M.T. Dao, "Reliability optimization using multiobjective ant colony system approaches", *Reliability Engineering & System Safety*, Vol. 92, Issue 1, pp. 109-120, 2007, ISSN 0951-8320, doi.org/10.1016/j.ress.2005.12.001.

[115]. H.A. Taboada, F. Baheranwala, D.W. Coit, N. Wattanapongsakorn, "Practical solutions for multi-objective optimization: An application to system reliability

design problems”, *Reliability Engineering & System Safety*, Vol. 92, Issue 3, pp. 314-322, 2007, ISSN 0951-8320, doi.org/10.1016/j.ress.2006.04.014.

[116]. C.H. Chiang, L.H. Chen, “Availability allocation and multi-objective optimization for parallel–series systems”, *European Journal of Operational Research*, Vol. 180, Issue 3, pp. 1231-1244, 2007, ISSN 0377-2217, doi.org/10.1016/j.ejor.2006.04.037.

[117]. P. Limbourg, H.D. Kochs, “Multi-objective optimization of generalized reliability design problems using feature models—A concept for early design stages”, *Reliability Engineering & System Safety*, Vol. 93, Issue 6, pp. 815-828, 2008, ISSN 0951-8320, doi.org/10.1016/j.ress.2007.03.032.

[118]. H.A. Taboada, J.F. Espiritu, D.W. Coit, "MOMS-GA: A Multi-Objective Multi-State Genetic Algorithm for System Reliability Optimization Design Problems", *IEEE Transactions on Reliability*, Vol. 57, Nº1, pp. 182-191, March 2008, doi.org/10.1109/TR.2008.916874.

[119]. R. Kumar, K. Izui, M. Yoshimura, S. Nishiwaki, “Multi-objective hierarchical genetic algorithms for multilevel redundancy allocation optimization”, *Reliability Engineering & System Safety*, Vol. 94, Issue 4, pp. 891-904, 2009, ISSN 0951-8320, doi.org/10.1016/j.ress.2008.10.002.

[120]. H.Z. Huang, J. Qu, M.J. Zuo, “Genetic-algorithm-based optimal apportionment of reliability and redundancy under multiple objectives”, *IIE Transactions*, Vol. 41, Nº4, pp. 287-298, 2009, doi:10.1080/07408170802322994.

[121]. I.D. Lins, E.A.L. Droguett, “Multiobjective optimization of availability and cost in repairable systems design via genetic algorithms and discrete event simulation”, *Pesquisa Operacional*, Vol. 29, pp. 43-66, 2009. Doi:10.1590/S0101-74382009000100003.

[122]. I.D. Lins, E. López, “Redundancy allocation problems considering systems with imperfect repairs using multi-objective genetic algorithms and discrete event simulation”, *Simulation Modelling Practice and Theory*, Vol. 19, Is. 1, pp. 362-381, 2011, ISSN 1569-190X, doi.org/10.1016/j.simpat.2010.07.010.

[123]. A. Chambari, S.H.A. Rahmati, A.A. Najafi, A. Karimi, “A bi-objective model to optimize reliability and cost of system with a choice of redundancy

strategies", *Computers & Industrial Engineering*, Vol. 63, Issue 1, pp. 109-119, 2012, ISSN 0360-8352, doi.org/10.1016/j.cie.2012.02.004.

[124]. J. Safari, "Multi-objective reliability optimization of series-parallel systems with a choice of redundancy strategies", *Reliability Engineering & System Safety*, Vol. 108, pp. 10-20, 2012, ISSN 0951-8320, doi.org/10.1016/j.ress.2012.06.001.

[125]. K. Khalili-Damghani, A.R. Abtahi, M. Tavana, "A new multi-objective particle swarm optimization method for solving reliability redundancy allocation problems", *Reliability Engineering & System Safety*, Vol. 111, pp. 58-75, 2013, ISSN 0951-8320, doi.org/10.1016/j.ress.2012.10.009.

[126]. G. Jiansheng, W. Zutong, Z. Mingfa, W. Ying, "Uncertain multiobjective redundancy allocation problem of repairable systems based on artificial bee colony algorithm", *Chinese J. Aeronaut.*, Vol. 27, Nº 6, pp. 1477-1487, 2014.

[127]. M.A. Ardakan, A.Z. Hamadani, M. Alinaghian, "Optimizing bi-objective redundancy allocation problem with a mixed redundancy strategy", *ISA Transactions*, Vol. 55, pp. 116-128, 2015, ISSN 0019-0578, doi.org/10.1016/j.isatra.2014.10.002.

[128]. M.K. Ghorabaee, M. Amiri, P. Azimi, "Genetic algorithm for solving bi-objective redundancy allocation problem with k-out-of-n subsystems", *Applied Mathematical Modelling*, Vol. 39, Issue 20, pp. 6396-6409, 2015, ISSN 0307-904X, doi.org/10.1016/j.apm.2015.01.070.

[129]. M. Amiri, M. Khajeh, "Developing a bi-objective optimization model for solving the availability allocation problem in repairable series–parallel systems by NSGA-II", *Journal of Industrial Engineering International,* Vol. 12, Nº 1, pp. 61-69, 2016, doi.org/10.1007/s40092-015-0128-4.

[130]. A.E. Jahromi, M. Feizabadi, "Optimization of multi-objective redundancy allocation problem with non-homogeneous components", *Computers & Industrial Engineering*, Vol. 108, pp. 111-123, 2017, ISSN 0360-8352, doi.org/10.1016/j.cie.2017.04.009.

[131]. F. Kayedpour, M. Amiri, M. Rafizadeh, A.S. Nia, "Multi-objective redundancy allocation problem for a system with repairable components considering instantaneous availability and strategy selection", *Reliability Engineering &*

System Safety, Vol. 160, pp. 11-20, 2017, ISSN 0951-8320, doi.org/10.1016/j.ress.2016.10.009.

[132]. A. Samanta, K. Basu, “An attraction-based particle swarm optimization for solving multi-objective availability allocation problem under uncertain environment”, *Journal of Intelligent & Fuzzy Systems*, Vol. 35, Nº 1, pp. 1169-1178, 2018.

[133]. M. Sharifi, T.A. Moghaddam, M. Shahriari, “Multi-objective Redundancy Allocation Problem with weighted-k-out-of-n subsystems”, Heliyon, Vol. 5, Issue 12, e02346, 2019, ISSN 2405-8440, doi.org/10.1016/j.heliyon.2019.e02346.

[134]. C.P. de Paula, L.B. Visnadi, H.F. de Castro, “Multi-objective optimization in redundant system considering load sharing”, *Reliability Engineering & System Safety*, Vol. 181, p.p. 17-27, 2019, ISSN 0951-8320, doi.org/10.1016/j.ress.2018.08.012.

[135]. A. Chambari, P. Azimi, A.A. Najafi, “A bi-objective simulation-based optimization algorithm for redundancy allocation problem in series-parallel systems”, *Expert Systems with Applications*, Vol. 173, 114745, 2021, ISSN 0957-4174, doi.org/10.1016/j.eswa.2021.114745.

[136]. B. Lin, J. Wu, R. Lin, J. Wang, H. Wang, X. Zhang, “Optimization of high-level preventive maintenance scheduling for high-speed trains”, *Reliability Engineering & System Safety*, Vol. 183, pp. 261-275, 2019, ISSN 0951-8320, doi.org/10.1016/j.ress.2018.11.028.

[137]. B.L. Kralj, R. Petrovic, “A multiobjective optimization approach to thermal generating units maintenance scheduling”, *European Journal of Operational Research*, Vol. 84, Issue 2, pp. 481-493, 1995, ISSN 0377-2217, doi.org/10.1016/0377-2217(93)E0316-P.

[138]. M. Charest, J.A. Ferland, “Preventive maintenance scheduling of power generating units”. *Annals of Operations Research*, Vol. 41, pp. 185-206, 1993 doi.org/10.1007/BF02023074.

[139]. M. Marseguerra, E. Zio, “Optimizing maintenance and repair policies via a combination of genetic algorithms and Monte Carlo simulation”, *Reliability*

Engineering & System Safety, Vol. 68, Issue 1, pp. 69-83, 2000, ISSN 0951-8320, doi.org/10.1016/S0951-8320(00)00007-7.

[140]. Y.T. Tsai, K.S. Wang, H.Y. Teng, “Optimizing preventive maintenance for mechanical components using genetic algorithms”, *Reliability Engineering & System Safety*, Volume 74, Issue 1, pp. 89-97, 2001, ISSN 0951-8320, doi.org/10.1016/S0951-8320(01)00065-5.

[141]. R. Bris, E. Châtelet, F. Yalaoui, “New method to minimize the preventive maintenance cost of series-parallel systems”, *Reliability Engineering and System Safety*, Vol. 82, Issue 3, pp. 247-255, 2003, ISSN 0951-8320, doi.org/10.1016/S0951-8320(03)00166-2.

[142]. M. Samrout, F. Yalaoui, E. Châtelet, N. Chebbo, “New methods to minimize the preventive maintenance cost of series-parallel systems using ant colony optimization”, *Reliability Engineering and System Safety*, Vol. 89, Issue 3, pp. 346-354, 2005, ISSN 0951-8320, doi.org/10.1016/j.ress.2004.09.005.

[143]. C.M.F. Lapa, C.M.N.A. Pereira, P.F. Frutuoso e Melo, “Surveillance test policy optimization through genetic algorithms using non-periodic intervention frequencies and considering seasonal constraints”, *Reliability Engineering & System Safety*, Vol. 81, Issue 1, pp. 103-109, 2003, ISSN 0951-8320, doi.org/10.1016/S0951-8320(03)00085-1.

[144]. C.M.F. Lapa, C.M. Pereira, M.P De Barros, “A model for preventive maintenance planning by genetic algorithms based in cost and reliability”, *Reliability Engineering and System Safety*, Vol. 91, Issue 2, pp. 233-240, 2006, SSN 0951-8320, doi.org/10.1016/j.ress.2005.01.004.

[145]. S.M.H. Hadavi, “Risk-Based, genetic algorithm approach to optimize outage maintenance schedule”, *Annals of Nuclear Energy*, Vol. 35, Issue 4, pp. 601-609, 2008, ISSN 0306-4549, doi.org/10.1016/j.anucene.2007.08.011.

[146]. C.H. Wang, T.W. Lin, “Improved Genetic Algorithm for Minimizing Periodic Preventive Maintenance Costs in Series-Parallel Systems”, In: Hsu CH., Yang L.T., Park J.H., Yeo SS. (eds.), *Algorithms and Architectures for Parallel Processing. ICA3PP 2010. Lecture Notes in Computer Science*, Vol. 6081, 2010, Springer, Berlin, Heidelberg, doi.org/10.1007/978-3-642-13119-6_8.

[147]. C.H. Wang, T.W. Lin, "Improved particle swarm optimization to minimize periodic preventive maintenance cost for series-parallel systems", *Expert Systems with Applications*, Vol. 38, Issue 7, pp. 8963-8969, 2011, ISSN 0957-4174, doi.org/10.1016/j.eswa.2011.01.113.

[148]. T.W. Lin, C.H. Wang, "A hybrid genetic algorithm to minimize the periodic preventive maintenance cost in a series-parallel system", *J Intell Manuf*, Vol. 23, pp. 1225-1236, 2012, doi.org/10.1007/s10845-010-0406-3.

[149]. C.S. Wong, F.T.S. Chan, S.H. Chung, "A joint production scheduling approach considering multiple resources and preventive maintenance tasks", *International Journal of Production Research*, Vol. 51, Nº3, pp. 883-896, 2013, doi:10.1080/00207543.2012.677070.

[150]. A. Ebrahimy Zade, M.B. Fakhrzad, "A Dynamic Genetic Algorithm for Solving a Single Machine Scheduling Problem with Periodic Maintenance," *ISRN Industrial Engineering*, Vol. 2013, pp. 11, 2013, doi: 10.1155/2013/936814.

[151]. Y. Zheng, K. Mesghouni, S.C. Dutilleul, "Condition based Maintenance applied to Reduce Unavailability of Machines in Flexible Job Shop Scheduling Problem", *IFAC Proceedings Volumes*, Vol. 46, Issue 9, pp. 1405-1410, 2013, ISSN 1474-6670, ISBN 9783902823359, doi.org/10.3182/20130619-3-RU-3018.00566.

[152]. H. Canh Vu, P. Do, A. Barros, C. Bérenguer, "Maintenance grouping strategy for multi-component systems with dynamic contexts", *Reliability Engineering and System Safety*, Vol. 132, pp. 233-249, 2014, ISSN 0951-8320, doi.org/10.1016/j.ress.2014.08.002.

[153]. H. Yin, G. Zhang, H. Zhu, Y. Deng, F. He, "An integrated model of statistical process control and maintenance based on the delayed monitoring". *Reliability Engineering and System Safety*, Vol. 133, pp. 323-333, 2015, ISSN 0951-8320, doi.org/10.1016/j.ress.2014.09.020.

[154]. L. Xiao, S. Song, X. Chen, D.W. Coit, "Joint optimization of production scheduling and machine group preventive maintenance", *Reliability Engineering and System Safety*, Vol. 146, pp. 68-78, 2016, ISSN 0951-8320, doi.org/10.1016/j.ress.2015.10.013.

[155]. I. Maatouk, N. Chebbo, I. Jarkass, E. Chatelet, “Maintenance Optimization using Combined Fuzzy Genetic Algorithm and Local Search”, *IFAC-PapersOnLine*, Vol. 49, Issue 12, pp. 757-762, 2016, ISSN 2405-8963, doi.org/10.1016/j.ifacol.2016.07.865.

[156]. X. Zhang, J. Zeng, “Joint optimization of condition-based opportunistic maintenance and spare parts provisioning policy in multiunit systems”, *European Journal of Operational Research*, Vol. 262, Nº2, pp. 479-498, 2017, ISSN 0377-2217, doi.org/10.1016/j.ejor.2017.03.019.

[157]. A. Yahyatabar, A.A. Najafi, “A quadratic reproduction based Invasive Weed Optimization algorithm to minimize periodic preventive maintenance cost for series-parallel systems”, *Computers & Industrial Engineering*, Vol. 110, pp. 436-461, 2017, ISSN 0360-8352, doi.org/10.1016/j.cie.2017.06.024.

[158]. S.H.A. Rahmati, A. Ahmadi, K. Govindan, “A novel integrated condition-based maintenance and stochastic flexible job shop scheduling problem: simulation-based optimization approach”. *Ann Oper Res*, Vol. 269, pp. 583-621, 2018, doi.org/10.1007/s10479-017-2594-0.

[159]. Z. Dahia, A. Bellaouar, S. Billel, “Optimization of the Preventive Maintenance for a Multi-component System Using Genetic Algorithm”, *Renewable Energy for Smart and Sustainable Cities*, Vol. 62, 2019, ISBN: 978-3-030-04788-7.

[160]. G. Quan, G.W. Greenwood, D. Liu, S. Hu, “Searching for multiobjective preventive maintenance schedules: Combining preferences with evolutionary algorithms”, *European Journal of Operational Research*, Vol. 177, Issue 3, pp. 1969-1984, 2007, ISSN 0377-2217, doi.org/10.1016/j.ejor.2005.12.015.

[161]. S. Carlos, A. Sanchez, S. Martorell, J.-F. Villanueva, “Particle Swarm Optimization of safety components and systems of nuclear power plants under uncertain maintenance planning”, *Advances in Engineering Software*, Vol. 50, pp. 12-18, 2012, ISSN 0965-9978, doi.org/10.1016/j.advengsoft.2012.04.004.

[162]. G. Balaji, R. Balamurugan, L. Lakshminarasimman, “Mathematical approach assisted differential evolution for generator maintenance scheduling”, *International Journal of Electrical Power & Energy Systems*, Vol. 82, pp. 508-518, 2016, ISSN 0142-0615, doi.org/10.1016/j.ijepes.2016.04.033.

[163]. C. Zhang, H. Zhu, J. Wu, Y. Cheng, Y. Deng, C. Liu, "An Economical Optimization Model of Non-Periodic Maintenance Decision for Deteriorating System," *IEEE Access*, Vol. 6, pp. 55149-55161, 2018, doi:10.1109/ACCESS.2018.2872348.

[164]. A. Muñoz, S. Martorell, V. Serradell, "Genetic algorithms in optimizing surveillance and maintenance of components", *Reliability Engineering & System Safety*, Vol. 57, Issue 2, 1997, pp. 107-120, ISSN 0951-8320, doi.org/10.1016/S0951-8320(97)00031-8.

[165]. M. Marseguerra, E. Zio, L. Podofillini, "Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation", *Reliability Engineering and System Safety*, Vol. 77, Issue 2, pp. 151-166, 2002, ISSN 0951-8320, doi.org/10.1016/S0951-8320(02)00043-1.

[166]. S. Martorell, J.F. Villanueva, S. Carlos, Y. Nebot, A. Sánchez, J.L. Pitarch, V. Serradell, "RAMS+C informed decision-making with application to multi-objective optimization of technical specifications and maintenance using genetic algorithms", *Reliability Engineering & System Safety*, Vol. 87, Issue 1, pp. 65-75, 2005, ISSN 0951-8320, doi.org/10.1016/j.ress.2004.04.009.

[167]. J. Gao, M. Gen, L. Sun, "Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm", *Journal of Intelligent Manufacturing,* Vol. 17, pp. 493–507, 2006, doi.org/10.1007/s10845-005-0021-x.

[168]. A. Oyarbide-Zubillaga, A. Goti, A. Sanchez, "Preventive maintenance optimisation of multi-equipment manufacturing systems by combining discrete event simulation and multi-objective evolutionary algorithms", *Production Planning & Control*, Vol. 19, Nº4, pp. 342-355, 2008, doi.org/10.1080/09537280802034091.

[169]. A. Berrichi, L. Amodeo, F. Yalaoui, E. Châtalet, M. Mezghiche, "Bi-objective optimization algorithms for joint production and maintenance scheduling: application to the parallel machine problem", *Journal of Intelligent Manufacturing*, Vol. 20, pp. 389-400, 2009, doi.org/10.1007/s10845-008-0113-5.

[170]. A. Sánchez, S. Carlos, S. Martorell, J.F. Villanueva, "Addressing imperfect maintenance modelling uncertainty in unavailability and cost-based

optimization", *Reliability Engineering and System Safety*, Vol. 94, Issue 1, pp. 22-32, 2009, ISSN 0951-8320, doi.org/10.1016/j.ress.2007.03.022.

[171]. A. Berrichi, F. Yalaoui, L. Amodeo, M. Mezghiche, "Bi-Objective Ant Colony Optimization approach to optimize production and maintenance scheduling", *Computers & Operations Research*, Vol. 37, Issue 9, 2010, pp. 1584-1596, ISSN 0305-0548, doi.org/10.1016/j.cor.2009.11.017.

[172]. E. Moradi, S.M.T. Fatemi Ghomi, M. Zandieh, "Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem", *Expert Systems with Applications*, Vol. 38, Issue 6, pp. 7169-7178, 2011, ISSN 0957-4174, doi.org/10.1016/j.eswa.2010.12.043.

[173]. Y. Wang, H. Pham, "A multi-objective optimization of imperfect preventive maintenance policy for dependent competing risk systems with hidden failure", *IEEE Transactions on Reliability*, Vol. 60, Nº4, pp. 770-781, 2011, doi.org/10.1109/TR.2011.2167779.

[174]. M. Ben Ali, M. Sassi, M. Gossab, Y. Harrath, "Simultaneous scheduling of production and maintenance tasks in the job shop", *International Journal of Production Research,* Vol. 49(13), pp. 3891-3918, 2011, doi.org/10.1080/00207543.2010.492405.

[175]. F. Hnaien, F. Yalaoui, "A bi-criteria flow-shop scheduling with preventive maintenance", *IFAC Proceedings Volumes*, Vol. 46, Issue 9, pp. 1387-1392, 2013, ISSN 1474-6670, ISBN 9783902823359, doi.org/10.3182/20130619-3-RU-3018.00206.

[176]. K. Suresh, N. Kumarappan, "Hybrid improved binary particle swarm optimization approach for generation maintenance scheduling problem", *Swarm and Evolutionary Computation*, Vol. 9, pp. 69-89, 2013, ISSN 2210-6502, doi.org/10.1016/j.swevo.2012.11.003.

[177]. J.Q. Li, Q.K. Pan, M. F. Tasgetiren, "A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities", *Applied Mathematical Modelling*, Vol. 38, Issue 3, pp. 1111-1132, 2014, ISSN 0307-904X, doi.org/10.1016/j.apm.2013.07.038.

[178]. Y. Gao, Y. Feng, Z. Zhang, J. Tan, "An optimal dynamic interval preventive maintenance scheduling for series systems", *Reliability Engineering & System Safety*, Vol. 142, pp. 19-30, 2015, ISSN 0951-8320, doi.org/10.1016/j.ress.2015.03.032.

[179]. S. Wang, M. Liu, "Multi-objective optimization of parallel machine scheduling integrated with multi-resources preventive maintenance planning", *Journal of Manufacturing Systems*, Vol. 37, Part 1, pp. 182-192, 2015, ISSN 0278-6125, doi.org/10.1016/j.jmsy.2015.07.002.

[180]. D. Piasson, A.A.P. Bíscaro, F.B. Leão, J.R. Sanches Mantovani, "A new approach for reliability-centered maintenance programs in electric power distribution systems based on a multiobjective genetic algorithm", *Electric Power Systems Research*, Vol. 137, pp. 41-50, 2016, ISSN 0378-7796, doi.org/10.1016/j.epsr.2016.03.040.

[181]. M. Sheikhalishahi, N. Eskandari, A. Mashayekhi, A. Azadeh, "Multi-objective open shop scheduling by considering human error and preventive maintenance", *Applied Mathematical Modelling*, Vol. 67, pp. 573-587, 2019, ISSN 0307-904X, doi.org/10.1016/j.apm.2018.11.015.

[182]. Y. An, X. Chen, J. Zhang, Y. Li, "A hybrid multi-objective evolutionary algorithm to integrate optimization of the production scheduling and imperfect cutting tool maintenance considering total energy consumption", *Journal of Cleaner Production*, Vol. 268, 121540, 2020, ISSN 0959-6526, doi.org/10.1016/j.jclepro.2020.121540.

[183]. R. Boufellouh, F. Belkaid, "Bi-objective optimization algorithms for joint production and maintenance scheduling under a global resource constraint: Application to the permutation flow shop problem", *Computers & Operations Research*, Vol. 122, 104943, 2020, ISSN 0305-0548, doi.org/10.1016/j.cor.2020.104943.

[184]. S. Bressi, J. Santos, M. Losa, "Optimization of maintenance strategies for railway track-bed considering probabilistic degradation models and different reliability levels", *Reliability Engineering & System Safety*, Vol. 207, 107359, 2021, ISSN 0951-8320, doi.org/10.1016/j.ress.2020.107359.

[185]. C. Zhang, T. Yang, “Optimal maintenance planning and resource allocation for wind farms based on non-dominated sorting genetic algorithm-II”, *Renewable Energy*, Vol. 164, pp. 1540-1549, 2021, ISSN 0960-1481, doi.org/10.1016/j.renene.2020.10.125.

[186]. X. Zhu, X. Bei, N. Chatwattanasiri, D.W. Coit, "Optimal System Design and Sequential Preventive Maintenance Under Uncertain Aperiodic-Changing Stresses," *IEEE Transactions on Reliability*, Vol. 67, Nº3, pp. 907-919, 2018, doi: 10.1109/TR.2018.2798298.

[187]. G. Levitin, A. Lisnianski, Joint redundancy and maintenance optimization for multistate series–parallel systems, Reliability Engineering & System Safety, Vol. 64, Issue 1, 1999, pp. 33-42, ISSN 0951-8320, doi.org/10.1016/S0951-8320(98)00052-0.

[188]. A. Monga, M.J. Zuo, “Optimal design of series-parallel systems considering maintenance and salvage value”, *Computers & Industrial Engineering*, Vol. 40, Issue 4, pp. 323-337, 2001, ISSN 0360-8352, doi.org/10.1016/S0360-8352(01)00032-8.

[189]. M. Nourelfath, E. Chatelet, N. Nahas, “Joint redundancy and imperfect preventive maintenance optimization for series–parallel multi-state degraded systems”, *Reliability Engineering and System Safety*, Vol. 103, pp. 51-60, 2012, ISSN 0951-8320, doi.org/10.1016/j.ress.2012.03.004.

[190]. N.M. Okasha, D.M. Frangopol, “Lifetime-oriented multi-objective optimization of structural maintenance considering system reliability, redundancy and life-cycle cost using GA”, Structural Safety, Vol. 31, Issue 6, pp. 460-474, 2009, ISSN 0167-4730, doi.org/10.1016/j.strusafe.2009.06.005.

[191]. O. Adjoul, K. Benfriha, C. El Zant, A. Aoussat, “Algorithmic Strategy for Simultaneous Optimization of Design and Maintenance of Multi-Component Industrial Systems”, *Reliability Engineering & System Safety*, Vol. 208, 2021, 107364, ISSN 0951-8320, doi.org/10.1016/j.ress.2020.107364.

[192]. C. Segura, C.A.C, Coello, G. Miranda et al, “Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization”. *Annals of Operations Research*, Vol. 240, pp. 217-250, 2016. https://doi.org/10.1007/s10479-015-2017-z.

[193]. S.J. Louis, G. Rawlins, “Pareto optimality, GA-easiness and deception”, In *Proceedings of the fifth international conference on genetic algorithms*, pp. 118-123, San Francisco: Morgan Kaufmann, 1993.

[194]. J. Knowles, R.A. Watson, D. Corne, “Reducing Local Optima in Single-Objective Problems by Multi-objectivization”, In *Proceedings of the first international conference on evolutionary multi-criterion optimization*, pp. 269-283, Springer, London, UK, EMO ’01, 2001.

[195]. S. Bleuler, M. Brack, L. Thiele, E. Zitzler, “Multiobjective genetic programming: reducing bloat using SPEA2”, In *Proceedings of the 2001 IEEE congress on evolutionary computation*, Vol. 1, pp. 536-543, 2001.

[196]. R.O. Day, J.B. Zydallis, G.B. Lamont, R. Pachter, “Solving the protein structure prediction problem through a multiobjective genetic algorithm”, *Nanotechnology*, 2, pp. 32-35, 2002.

[197]. D. Greiner, J. Emperador, G. Winter, B. Galvan, “Improving computational mechanics optimum design using helper objectives: An application in frame bar structures,” in *Evolutionary Multi-Criterion Optimization*, *Lecture Notes in Computer Science, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds*, vol. 4403, pp. 575-589, Springer Berlin Heidelberg, 2007.

[198]. J. Jacques, J. Taillard, D. Delerue, L. Jourdan, C. Dhaenens, “The benefits of using multiobjectivization for mining Pittsburgh partial classification rules in imbalanced and discrete data”, In *Proceedings of the 15th annual conference on genetic and evolutionary computation GECCO’13*, pp. 543-550, New York, NY: ACM, 2013.

[199]. H. Ishibuchi, Y. Hitotsuyanagi, Y. Nakashima, Y. Nojima, “Multiobjectivization from two objectives to four objectives in evolutionary multi-objective optimization algorithms”, In *2010 Second world congress on Nature and biologically inspired computing (NaBIC)*, pp. 502-507, 2010.

[200]. Y. Zheng, Z. Zhu, Y. Qi, L. Wang, X. Ma, "Multi-objective multifactorial evolutionary algorithm enhanced with the weighting helper-task", *2020 2nd International Conference on Industrial Artificial Intelligence (IAI)*, pp. 1-6, 2020, doi: 10.1109/IAI50351.2020.9262200.

[201]. J. Knezevic, Mantenibilidad, 1st Edition, ISDEFE – Ingeniería de Sistemas, 1996.

[202]. R. Doran, "The Gray Code", *Journal of Universal Computer Science*, Vol. 13, Nº 11, pp. 1573-1597, 2007.

[203]. D. Greiner, G. Winter, J.M. Emperador, B. Galván, "Gray Coding in Evolutionary Multicriteria Optimization: Application in Frame Structural Optimum Design", in: Carlos A. Coello Coello, Arturo Hernández Aguirre, Eckart Zitzler (Eds.), *Evolutionary Multi-Criterion Optimization, Third International Conference, EMO 2005*, Proceedings, Guanajuato, Mexico, March 9-11, 2005, 576-591.

[204]. R. Tanabe, H. Ishibuchi, "An Analysis of Control Parameters of MOEA/D Under Two Different Optimization Scenarios", *Applied Soft Computing*, Vol. 70, pp. 22-40, 2018, doi.org/10.1016/j.asoc.2018.05.014.

[205]. B. Liu, F. Fernández, Q. Zhang, M. Pak, S. Sipahi, G. Gielen, "An enhanced MOEA/D-DE and its application to multiobjective analog cell sizing", in: *IEEE Congress on Evolutionary Computation*, pp. 1-7, 2010, doi.org/10.1109/CEC.2010.5585957.

[206]. Q. Zhang, W. Liu, H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances", *IEEE Congress on Evolutionary Computation*, pp. 203-208, 2009, doi.org/10.1109/CEC.2009.4982949.

[207]. Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs. Artificial Intelligence", Springer-Verlag, New York, 1996.

[208]. F. González, D. Greiner, V. Mena, R. Souto, J. Santana, J. Aznárez, "Fitting procedure based on Differential Evolution to evaluate impedance parameters of metal coating systems", *Engineering Computations*, Vol. 36 (9), pp. 2960-2982, 2019, doi.org/10.1108/EC-11-2018-0513.

[209]. S. García, F. Herrera, "An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons", *Journal of Machine Learning Research*, Vol. 9, pp. 2677-2694, 2008.

[210]. A. Benavoli, G. Corani, F. Mangili, "Should We Really Use Post-Hoc Tests Based on Mean-Ranks?", *Journal of Machine Learning Research*, Vol. 17(5), pp. 1-10, 2016.

[211]. C. Fonseca, L. Paquete, M. López-Ibáñez, “An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator”, in: *IEEE Congress on Evolutionary Computation*, Vancouver, BC, pp. 1157-1163, 2006, doi.org/10.1109/CEC.2006.1688440.

[212]. Y. Tian, R. Cheng, X. Zhang, Y. Jin, “PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization”, [Educational Forum], *IEEE Computational Intelligence Magazine*, Vol. 12(4), pp. 73-87, 2017, doi.org/10.1109/MCI.2017.2742868.

[213]. A. Cacereño, B. Galván, D. Greiner, “Solving Multi-objective Optimal Design and Maintenance for Systems Based on Calendar Times Using NSGA-II”. In: Gaspar-Cunha A, Periaux J, Giannakoglou K, et al (eds) *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*. Springer Nature Switzerland AG, p 245-259, 2021, https://doi.org/10.1007/978-3-030-57422-2.

[214]. M. G. Kanabar, T. S. Sidhu, Reliability and availability analysis of IEC 61850 based substation communication architectures, in: *2009 IEEE Power & Energy Society General Meeting*, 2009, pp. 1-8, 2009.

[215]. X. Yang, N. Das, S. Islam, Analysis of IEC 61850 for a reliable communication system between substations, in: *2013 Australasian Universities Power Engineering Conference (AUPEC)*, pp. 1-6, 2014, https://doi.org/10.1109/aupec.2013.6725482.

[216]. M. Natham, D. James, Protection System Coordination, Testing, and Maintenance to Comply with NERC Requirements, Bureau of Reclamation, Denver, CO, 2017.

[217]. J. Tarquínio, Reliability Analysis of the Control and Automation System in Electrical Substations, M.Sc. Thesis, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal, 2021.

[218]. G. Scheer, D. Dolezilek, Comparing the Reliability of Ethernet Network Topologies in Substation Control and Monitoring Networks, in: *2nd Annual Western Power Delivery Automation Conference*, 2000.

[219]. Network availability: How much do you need? How do you get it?, Cisco Systems.

[220]. Protection System Maintenance – A Technical Reference, System Protection and Controls Task Force of the NERC Planning Committee, 2007.