

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

DEPARTAMENTO DE FÍSICA



TESIS DOCTORAL

**CONTRIBUCIONES A LA EVALUACIÓN
CUANTITATIVA DE ÁRBOLES DE FALLOS**

BLAS JOSÉ GALVÁN GONZÁLEZ

Las Palmas de Gran Canaria, septiembre de 1999

74/1998-99

**UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
UNIDAD DE TERCER CICLO Y POSTGRADO**

Reunido el día de la fecha, el Tribunal nombrado por el Excmo. Sr. Rector Magfco. de esta Universidad, el/a aspirante expuso esta **TESIS DOCTORAL**.

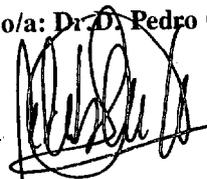
Terminada la lectura y contestadas por el/a Doctorando/a las objeciones formuladas por los señores miembros del Tribunal, éste calificó dicho trabajo con la nota de **SOBRESALIENTE**
CUM LAUDE (POR UNANIMIDAD)

Las Palmas de Gran Canaria, a 20 de septiembre de 1999.

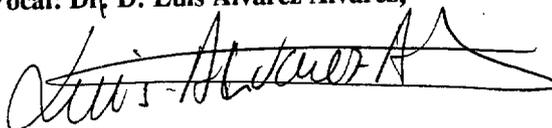
El/a Presidente/a: Dr. D. Gabriel Winter Althaus,



El/a Secretario/a: Dr. D. Pedro Cuesta Moreno,



El/a Vocal: Dr. D. Luis Álvarez Álvarez,



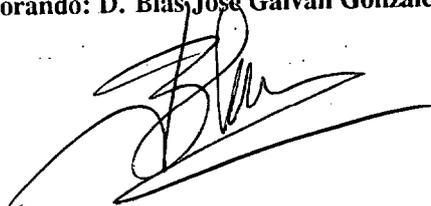
El/a Vocal: Dr. D. Sebastian Martorell ~~Mora~~ ALSINA



El/a Vocal: Dr. D. Jacques Periaux,



El Doctorando: D. Blas José Galván González,





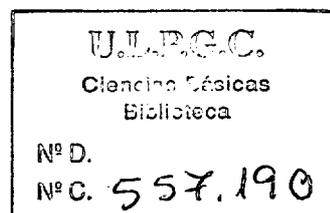
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Departamento de Física

Programa de Doctorado: Física Fundamental y Aplicada

**Contribuciones a la Evaluación Cuantitativa de
Árboles de Fallos**

© Universidad de Las Palmas de Gran Canaria. Biblioteca Digital. 2003



Por:
Blas José Galván González
Ingeniero Industrial

- Septiembre 1999 -

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Departamento de Física

Programa de Doctorado: Física Fundamental y Aplicada

**Contribuciones a la Evaluación Cuantitativa de
Árboles de Fallos**

Presentada por Blas José Galván González

Dirigida por Dr. D. Lorenzo Doreste Suárez

El Director



El Doctorando



Las Palmas de Gran Canaria, a 1 de Septiembre de 1999

**A mi esposa Fátima e hija Sara, a mis padres Nino y Pilar, a mi
hermana Encarna y su esposo Pedro.**

AGRADECIMIENTOS

Mi más sincero agradecimiento al Director de este trabajo el Catedrático de la U.L.P.G.C. D. Lorenzo Doreste Suárez por el maravilloso magisterio ejercido sobre mí a lo largo de la realización, así como por sus sabios consejos, dedicación, paciencia y confianza sin reservas.

Agradezco a los Doctores D. Luis González Sánchez y D. Diego García Vera su estímulo, ayuda y consejos de gran valor a lo largo de la realización de los capítulos, las demostraciones teóricas y la optimización de algoritmos.

La recopilación bibliográfica para la revisión del estado del arte ha sido posible gracias al generoso esfuerzo del Departamento de Matemáticas de la Universidad de Las Palmas de Gran Canaria, que ha proporcionado los recursos económicos necesarios a lo largo de los años de realización del trabajo.

La exploración del ámbito de aplicación práctica real de los métodos presentados en este trabajo no hubiese sido posible sin la utilización de los recursos informáticos del Centro de Aplicaciones Numéricas en Ingeniería (C.E.A.N.I.). Ha sido muy importante el estímulo, ayuda y apoyo recibidos del Catedrático D. Gabriel Winter Althaus y del Dr. D. Pedro Cuesta Moreno en los últimos años, gracias a ellos los resultados obtenidos son aplicables en procesos reales complejos de optimización multicriterio.

Mi agradecimiento al Colegio Oficial de Ingenieros Industriales de Canarias por su soporte económico para la realización de esta obra.

RESUMEN

El análisis mediante árboles de fallos permite estudiar la Fiabilidad y la No Disponibilidad de sistemas técnicos mediante el uso de puertas lógicas organizadas en una estructura que conduce desde unos sucesos no deseados hasta sus causas fundamentales o sucesos básicos. Las aplicaciones incluyen Optimización del Diseño, Análisis Probabilístico de Seguridad y desarrollo de Software y Sistemas Tolerantes a Fallos. El principal desafío de los árboles de fallos de sistemas reales, una vez obtenidos, es la evaluación cuantitativa de los mismos para obtener el valor del parámetro de Fiabilidad que se estudie, ya que el tamaño o la complejidad lógica de los mismos suele convertir dicha tarea en un problema de cálculo formidable. En esta tesis se desarrollan nuevos métodos directos de evaluación cuantitativa que amplían el tamaño abordable de árboles hasta valores prácticos comparables con los que alcanzan otros métodos en uso, con la ventaja sobre ellos de ser seguros y susceptibles de contemplar modernas puertas lógicas incorporadas en los últimos años a este tipo de análisis.

Se comienza con un estudio sistemático de las ventajas e inconvenientes de los métodos utilizados hasta la actualidad, seguido de una recopilación de las conclusiones que diversos investigadores han obtenido a cerca de un método ideal de evaluación cuantitativa, con ello se obtiene un marco general concensuado de búsqueda del mismo. En base a ese marco general se desarrollan las contribuciones prácticas de este trabajo. La primera contribución práctica consiste en un nuevo método determinista de evaluación al que hemos denominado *método de los Pesos* que comprende expansiones incompletas de Shannon de la función de estructura del árbol de fallos priorizadas mediante un teorema de Ordenación Intrínseca.

Se sigue con una contribución basada en simulación por el método de Monte Carlo a la que hemos llamado *muestreo Daga Generalizado* que optimiza la forma de muestrear mediante el uso de una alta correlación negativa y el mecanismo de generación de muestras mediante un algoritmo eficiente de proximidad.

Finalmente se enlazan los métodos anteriores en un nuevo método combinado determinista-simulación, al que hemos denominado *método PRD*, usando para ello una derivación e interpretación teórica de un estimador general de muestreo restringido. Su principal propiedad es el carácter cooperativo de los métodos que lo componen por lo que aumenta la precisión y disminuyen los tiempos de CPU en comparación con los de cada método considerados de forma individual. El estudio factorial de este método permite recomendar un procedimiento de trabajo con árboles de fallos que facilita, manteniendo las condiciones de realismo, saber a priori, antes incluso de la construcción del mismo, si este va a ser evaluable cuantitativamente con precisión aceptable. Este procedimiento permite también asegurar la viabilidad, en cuanto a tiempos de CPU, de procesos de optimización multicriterio de tipo evolutivo, en los que se usa la No Disponibilidad como parámetro de Fiabilidad, mediante el uso de la Función de Estructura de un árbol de fallos, para definir una de las funciones a optimizar.

De todos los métodos desarrollados se aportan algoritmos optimizados y se ha realizado una validación práctica con ejemplos conocidos, un estudio de los factores que influyen en los mismos, una determinación mediante ensayos de los rangos de aplicación, así como una estimación los tiempos de CPU esperados para su empleo en diferentes procesos.

ÍNDICE

	<u>Pag.</u>
 <u>CAPÍTULO 1.- INTRODUCCIÓN</u>	
1.1.- INTRODUCCIÓN	1
1.2.- RESUMEN DE CONTRIBUCIONES ORIGINALES	2
1.3.- ORGANIZACIÓN DE LA TESIS	4
1.4.- MÉTODOLOGÍAS DE ANÁLISIS DE FIABILIDAD	5
 <u>CAPÍTULO 2.- ANÁLISIS DE ÁRBOL DE FALLOS</u>	
2.1.- ANÁLISIS DE ÁRBOL DE FALLOS	13
2.2.- CONCEPTOS FUNDAMENTALES	16
2.2.1.- El espacio de los fallos como entorno de análisis	16
2.2.2.- Terminología	18
2.2.3.- Símbolos Lógicos	20
2.3.- ETAPAS DEL PROCESO DE ANÁLISIS	22
2.4.- DEFINICIÓN DEL SISTEMA	25
2.5.- CONSTRUCCIÓN DEL ÁRBOL DE FALLOS	26
2.5.1.- Descripción del proceso de construcción	26
2.5.2.- Metodologías para la construcción	30
2.5.3.- Uso de Tablas de Decisión	32
2.5.4.- Uso de Digrafos	34
2.5.5.- Uso de Redes Semánticas	38
2.6.- ANÁLISIS CUALITATIVO	42
2.6.1.- Introducción	42
2.6.2.- Representación Matemática del Árbol de Fallos	44
2.6.3.- Métodos Directos para obtener PIs/GCM	45
2.6.4.- Métodos Indirectos. Diagramas de Decisión Binarios	60
2.6.5.- Causas Comunes de fallos	68

2.7.- ANÁLISIS CUANTITATIVO	76
2.7.1.- Introducción	76
2.7.2.- Clasificación de los Métodos de Análisis Cuantitativo	78
2.7.3.- Métodos Directos	80
2.7.4.-Métodos basados en GCM/PIs	96
2.7.5.- Métodos basados en Diagramas de Decisión Binarios	101
2.7.6.-Teoría Cinética del Árbol	104
2.7.7.- Simulación del Árbol de Fallos	112
2.7.8.- Importancia	132
2.7.9.- Los fallos con Causa Común en el análisis cuantitativo	134
2.8.- APLICACIONES Y NUEVAS TENDENCIAS	136

**CAPÍTULO 3.- CONTRIBUCIONES A LA EVALUACIÓN DIRECTA DE
ÁRBOLES DE FALLOS**

3.1.- INTRODUCCIÓN	145
3.2.- ANÁLISIS COMPARATIVO DE LOS MÉTODOS DE EVALUACIÓN EXISTENTES	147
3.2.1.- Métodos de simulación	147
3.2.2.- Métodos analíticos	148
3.2.3.- Conclusiones del análisis comparativo	149
3.3.- OBJETIVOS DEL ESTUDIO	151
3.4.- ESTUDIO GENERAL DEL PROBLEMA DE LA EVALUACIÓN DE UN ÁRBOL DE FALLOS	153
3.4.1.- Formulación general del problema	154
3.4.2.- Particiones de U	155
3.4.3.- Ordenamientos de U. Teorema de Ordenación Intrínseca	159
3.4.4.- Combinación de Particiones y Ordenamientos	160
3.5.- EL MÉTODO DE LOS PESOS	161
3.5.1.- Descripción del método	161
3.5.2.- Algoritmo del método de los Pesos	164

3.6.- SIMULACIÓN POR MONTE CARLO CRUDO O DIRECTO	165
3.7.- MUESTREO DAGA GENERALIZADO	166
3.7.1.- Introducción	166
3.7.2.- Generalización del muestreo Daga	167
3.7.3.- Forma de realizar el muestreo Daga	170
3.7.4.- Algoritmo del muestreo Daga Generalizado	174
3.8.- MUESTREO RESTRINGIDO	175
3.8.1.- Descripción del muestreo	175
3.8.2.- Forma de obtener las muestras para el estimador Restringido	180
3.8.3.- Algoritmo del muestreo Restringido	181
3.9.- MÉTODO PESOS-RESTRINGIDO-DAGA (PRD)	183
3.9.1.- Descripción del método	183
3.9.2.- Características del método PRD	184
3.9.3.- Algoritmo del método PRD	186
3.10.- UN NUEVO SISTEMA PARA EVALUAR ÁRBOLES DE FALLOS	188

CAPÍTULO 4.- APLICACIONES, CÁLCULOS Y DISCUSIÓN

4.1.- INTRODUCCIÓN	189
4.2.- VALIDACIÓN DE ALGORITMOS Y APLICACIONES EN INGENIERÍA	190
4.2.1.- Árbol propuesto por Kumamoto et al.	190
4.2.2.- Sistema de mezcla de Amoniaco y Aire de una planta para la fabricación de Ácido Nítrico.	194
4.2.3.- Sistema de inyección Spray en la contención de una Central Nuclear (CSIS).	207
4.2.4.- Sistema de Suministro Eléctrico de una gran Central	213
4.2.5.- Estudio de aplicaciones a grandes árboles de fallos	220

CAPÍTULO V.- CONCLUSIONES y LÍNEAS FUTURAS

5.1.- PRÓLOGO	227
5.2.- CONCLUSIONES GENERALES	228
5.3.- CONCLUSIONES DEL ANÁLISIS FACTORIAL DEL MÉTODO PRD	233
5.4.- CONSECUENCIAS	235
5.5.- LÍNEAS FUTURAS	236
 <u>BIBLIOGRAFÍA</u>	 239

CAPÍTULO I
INTRODUCCIÓN

1.1.- INTRODUCCIÓN

El objetivo de este trabajo es desarrollar métodos eficientes y seguros para evaluar cuantitativamente grandes árboles de fallos con sucesos repetidos y lógica estática. Para alcanzar este objetivo se utilizarán métodos deterministas basados en formulaciones analíticas, métodos probabilistas basados en Simulación por Monte Carlo y métodos combinados determinista-simulación, todos ellos desarrollados según las conclusiones obtenidas por diversos investigadores sobre el comportamiento de un método ideal de evaluación cuantitativa. Por eficientes entenderemos métodos capaces de evaluar en tiempo de CPU admisible y con precisión aceptable árboles de gran tamaño, y por seguros a métodos capaces de dar siempre una respuesta sea cual sea la complejidad del árbol o el tipo de puertas lógicas estáticas que incorporen. Se aportan los desarrollos teóricos, las pruebas de validación y los resultados del estudio del ámbito real de aplicación de cada método.

A pesar del gran esfuerzo realizado en investigación a lo largo de los últimos treinta años y debido fundamentalmente al desarrollo tecnológico acelerado de las últimas décadas, los métodos de evaluación cuantitativa de grandes árboles de fallos presentan importantes limitaciones. Entre los métodos deterministas cabe destacar el crecimiento exponencial del número de combinaciones de sucesos básicos (Grupos de Corte Mínimos o Primeros Implicantes), la incapacidad para manejar modernas puertas lógicas tanto estáticas como dinámicas y la no existencia de fundamentos teóricos demostrados para decidir el adecuado orden de las variables en el caso de convertir el árbol de fallos a su Diagrama de Decisión Binario equivalente para usar Metaproductos. Entre los métodos de simulación el principal inconveniente es el elevado tiempo de CPU necesario para obtener respuestas con precisión aceptable, derivado fundamentalmente de la carencia de métodos de Reducción de Varianza eficaces.

La evaluación cuantitativa de árboles de fallos se usa en estudios de Fiabilidad, en Análisis Probabilístico de Riesgos, en el desarrollo de Software o Sistemas Tolerantes a Fallos y en la Optimización Multicriterio de sistemas técnicos, tanto en la etapa de diseño como en la de operación. Es importante encontrar nuevas técnicas que minimicen inconvenientes como los resaltados anteriormente, y otros que se mencionarán, para abordar con éxito el estudio y diseño de modernos sistemas altamente integrados que incorporen lógicas complejas y sistemas computerizados de control. En particular, los inconvenientes descritos de las técnicas de evaluación

actuales se vuelven críticos cuando se pretende realizar una optimización multicriterio del diseño de un sistema técnico complejo. En estos procesos es necesario evaluar cientos o miles de árboles de fallos de diversos tamaños correspondientes a configuraciones potencialmente muy diferentes del sistema y, por tanto, con combinaciones lógicas y sucesos básicos distintos. En estos casos, si la evaluación no es segura y eficiente, el proceso de optimización se puede volver inviable. Esto es lo que frecuentemente ocurre cuando se pretende adaptar códigos "clásicos" de evaluación para ser parte de uno de estos procesos, en la práctica el proceso está limitado no por la falta de nuevos elementos a incorporar en el diseño, sino por la inviabilidad de modelar estos con dichos códigos o la inseguridad de obtener una solución en tiempo de CPU aceptable.

1.2.- RESUMEN DE CONTRIBUCIONES ORIGINALES

La primera contribución comienza con un estudio sistemático de las ventajas e inconvenientes de los métodos utilizados hasta la actualidad, en el que se tienen en cuenta tanto consideraciones teóricas de capacidad de modelado como prácticas de potencial de aplicación. Se sigue con una recopilación de las conclusiones que diversos investigadores han obtenido a cerca de un método ideal de evaluación cuantitativa, obtenidas fundamentalmente de sus experiencias a la hora de implementar prácticamente los diversos métodos existentes, haciendo énfasis en los motivos que impiden ampliar el rango de aplicación. Todo lo anterior permite obtener un marco general de búsqueda de un método de evaluación cuantitativa lo más amplio y concensuado posible, pero con criterios claros y aplicables. En base a ese marco general se desarrollan las contribuciones prácticas de este trabajo.

La segunda contribución es un nuevo método determinista de evaluación al que hemos denominado *método de los Pesos* que comprende expansiones incompletas de Shannon de la función de estructura del árbol de fallos priorizadas mediante un teorema de Ordenación Intrínseca. El método es independiente de la lógica del árbol y su eficiencia sólo depende del número de sucesos básicos y de la probabilidad de los mismos.

Se sigue con una contribución basada en simulación por el método de Monte Carlo a la que hemos llamado *muestreo Daga Generalizado* que optimiza la forma de muestrear mediante el uso de una alta correlación negativa y el mecanismo de generación de muestras mediante un algoritmo eficiente de proximidad. El método permite evaluar con gran precisión y tiempos de CPU aceptables árboles de gran tamaño, siendo su principal inconveniente la presencia de probabilidades muy bajas para lo que se usa truncamiento en función de la CPU disponible.

La contribución final enlaza los métodos anteriores en un nuevo método combinado determinista-simulación, al que hemos denominado *método PRD*, usando para ello una derivación e interpretación teórica de un estimador general de muestreo restringido. Se demuestran sus propiedades, se verifica que su precisión es mayor que la de sus componentes y que cumple con la mayoría de las características deseables establecidas en el marco general deducido inicialmente. Su principal propiedad es el carácter cooperativo de los métodos que lo componen, lo cual se obtiene mediante la realización de una partición estructural simple del conjunto de combinaciones de sucesos básicos, gobernada por el número de bits significativos de las mismas. Este carácter aumenta la precisión y disminuye los tiempos de CPU, en comparación con los de cada método considerados de forma individual, pero mantiene la independencia de la lógica del árbol e incrementa el rango práctico de aplicación. El estudio factorial de este método permite recomendar un procedimiento de trabajo con árboles de fallos que facilita, manteniendo las condiciones de realismo, saber a priori, antes incluso de la construcción del mismo, si este va a ser evaluable cuantitativamente con precisión aceptable. Para ello solo será necesario controlar el valor de las probabilidades de los sucesos básicos presentes en las bases de datos de control de fallos, para mantenerlas entre ciertos rangos mediante métodos de agregación o desagregación de sucesos. Este procedimiento permite también asegurar la viabilidad, en cuanto a tiempos de CPU, de procesos de optimización multicriterio de tipo evolutivo, en los que se usa la No Disponibilidad como parámetro de Fiabilidad, mediante el uso de la Función de Estructura de un árbol de fallos, para definir una de las funciones a optimizar.

De todos los métodos desarrollados se aportan algoritmos optimizados y se ha realizado una validación práctica con ejemplos conocidos, un estudio de los factores que influyen en los mismos, una determinación mediante ensayos los rangos de aplicación, así como una estimación los tiempos de CPU esperados para su empleo en diferentes procesos.

1.3.- ORGANIZACIÓN DE LA TESIS

La presente Tesis Doctoral está dividida en 5 capítulos y complementada con una bibliografía seleccionada.

El Capítulo I define los objetivos del trabajo fundamentados en la necesidad de métodos de evaluación eficientes y seguros, resume las contribuciones originales, describe la organización de la obra escrita y presenta una descripción de los métodos Inductivos y Deductivos más importantes para la realización de un Análisis de Fiabilidad y Seguridad de los sistemas técnicos actuales.

El Capítulo II contiene una revisión del Estado del Arte del Análisis de Árbol de Fallos con especial énfasis en la Construcción, Análisis Cualitativo y Cuantitativo, Aplicaciones y Nuevas tendencias del Análisis. En la Construcción del Árbol de Fallos se diferencia entre los métodos procedurales, imperantes durante una gran parte de los aproximadamente 35 años de existencia de la metodología, y los sistemas expertos que constituyen y constituirán el futuro de este tipo de tareas. En el Análisis Cualitativo se revisan los métodos de obtención de los Primeros Implicantes/Grupos de Corte Mínimos agrupándolos según empleen simulación, procedimientos deterministas u otras técnicas de diversa clasificación. El Análisis Cuantitativo se aborda diferenciando los métodos entre aquellos que emplean para la evaluación los Primeros Implicantes/Grupos de Corte Mínimos (Análisis Cualitativo previo) y aquellos que evalúan el árbol Directamente. Se hace especial énfasis en los Métodos Directos ya que son la base de partida para el desarrollo del capítulo III. En cuanto a las Aplicaciones del Árbol de Fallos se realiza una clasificación de las referencias más significativas según su campo de aplicación. El apartado Nuevas Tendencias recoge los trabajos de tipo reciente que permiten establecer las líneas actuales de investigación que además de las mencionadas (Uso de Sistemas Expertos y Evaluación Directa) están siendo tratadas, clasificadas en tres grupos: Análisis de Incertidumbre, Sistemas Dinámicos y Técnicas Combinadas. En el primero se engloban todas aquellas técnicas que abordan la Incertidumbre y Vaguedad en los datos y los cálculos (Lógica Difusa, Teoría de Dempster-Shafer etc.), en el segundo los trabajos para estudiar los sistemas dinámicos complejos actuales y en el tercero las combinaciones con otras disciplinas del Algebra Discreta (Redes de Petri, Cadenas de Markov, etc.).

El Capítulo III contiene un análisis de ventajas e inconvenientes de los métodos de valuación cuantitativa, la deducción de un marco general de derivación de un método ideal de evaluación, *El método de los Pesos (Analítico)*, *muestreo Daga Generalizado* (Simulación por Monte Carlo) y la combinación de ambos en el *método PRD* deducida a partir de un estimador clásico de muestreo restringido. Se finaliza con la presentación de un Sistema Conjunto para Evaluación de Árboles de Fallos.

El Capítulo IV contiene las aplicaciones y cálculos realizados utilizando los métodos propuestos en el capítulo anterior. En el se puede encontrar una aplicación a una industria química, una Central Eléctrica y un estudio de grandes árboles.

El Capítulo V presenta las conclusiones derivadas de la aplicación práctica de los métodos propuestos y se especifican las líneas abiertas para futuros estudios.

1.4.- MÉTODOS DE ANÁLISIS DE FIABILIDAD

La Fiabilidad es una medida de la capacidad que tiene un sistema para funcionar durante el tiempo que dure su misión bajo las condiciones en las que fue diseñado. El proceso conocido como Análisis de Fiabilidad de Sistemas está compuesto por una serie de etapas que conducen no solo a la evaluación precisa de determinados parámetros sino a un profundo conocimiento del sistema por parte de quien lo realiza. Antes de comenzar cualquiera de estos análisis será necesario definir clara y explícitamente el alcance y el grado de detalle. En la práctica estos parámetros se suelen fijar de forma iterativa, ya que, con el avance del estudio se llega a una comprensión y definición más exacta de los mismos. La correcta elección es de gran importancia, ya que se trata de incluir todo aquello que tenga importancia para el análisis pretendido y que sea abordable tanto por disponer de información sobre aspectos cualitativos y/o datos técnicos como por disponer de conocimientos sobre su comportamiento y métodos de resolución adecuados. Tanto el alcance como el grado de detalle deben ser complementados con el horizonte temporal del análisis así como con la definición de los objetivos del mismo y medios disponibles para la realización. Con todo ello se llega a

una primera aproximación válida y realista al contexto general del análisis y se puede plantear una estrategia global de resolución.

Finalizada la etapa anterior, se dispone de una serie de procedimientos cuya clasificación puede realizarse, de acuerdo con Hauptmans (1986), atendiendo a los siguientes conceptos:

CONCEPTO	MÉTODO
Según el tipo de razonamiento	Métodos Inductivos Métodos Deductivos
Según el Alcance	Métodos Cualitativos Métodos Cuantitativos
Según el Objetivo	Métodos para identificar posibles potenciales de peligro Métodos para evaluar potenciales de peligro

Normalmente los métodos de análisis poseen varios de los atributos señalados con el objeto de generalizar y facilitar su aplicación. Dado su carácter más general utilizaremos como marco de referencia la clasificación según el tipo de razonamiento, o sea, Métodos Inductivos y Métodos Deductivos e incluiremos los demás atributos de forma explícita en los diferentes métodos que se describan.

Los Métodos Inductivos comienzan con hechos individuales para llegar a conclusiones de tipo general. Los hechos individuales serán, por lo general, sucesos o fallos ocurridos a elementos, del sistema y las conclusiones serán efectos tanto individuales sobre los elementos como globales sobre el sistema. Entre los métodos inductivos más referenciados destacan: Análisis de los Modos de Fallo y sus Efectos (FMEA), Análisis de Riesgo y Operabilidad (HAZOP), Análisis de Markov (MA) y Secuencias de Sucesos (ET).

El Análisis de los Modos de Fallo y sus Efectos (FMEA) es un potente y detallado método cualitativo que analiza sistemáticamente los modos de fallo de los elementos y subsistemas e identifica su efecto sobre el sistema. Su objetivo básico es identificar las zonas de diseño de un nuevo sistema que necesitan ser mejoradas para asegurar que será fiable y seguro para la misión que se le asigna. Existen diversas modalidades de este método tales como FMEA, FMECA etc. . En el cuadro siguiente se

ha clasificado y esquematizado el conocimiento básico que se necesita tener del sistema y sus componentes para poder este método.

INFORMACIÓN MÍNIMA NECESARIA PARA UN FMEA	
DEL SISTEMA	DE LOS ELEMENTOS
<ul style="list-style-type: none"> - Descripción física - Misión a cumplir - Descripción funcional - Condiciones del entorno - Descripción de categorías de fallos 	<ul style="list-style-type: none"> - Descripción física - Descripción funcional - Misión a cumplir - Condiciones y entorno de funcionamiento - Modos de fallo - Mecanismos de fallo - Efecto del fallo sobre el sistema - Otros efectos a contemplar - Forma(s) de detectar el fallo - Anotaciones

Existen diversos estudios como los de Stamanis (1995) o Bowles & Bonnell (1996) donde se analizan en profundidad y detalle estos métodos. La automatización de este proceso es ciertamente compleja ya que, por un lado la cantidad de datos a almacenar y tratar, según se observa en la tabla anterior, es enorme, y por otro lado la naturaleza de esos datos es, normalmente, de tipo semántico ("alto", "medio", ... etc.). Dado que el razonamiento simbólico es normalmente más adecuado que los métodos numéricos, al proporcionar una interpretación del modelo del sistema más próxima a la mente humana. Normalmente no se dispone de datos numéricos durante la mayor parte del tiempo de realización de un FMEA. La necesidad de trabajar con términos semánticos y razonamiento simbólico hace que los métodos, tanto cualitativos como cuantitativos, capaces de contemplar y manejar la incertidumbre sean muy adecuados (Fragola, 1993), (Pelaez y Bowles, 1995).

La característica más sobresaliente de este método es que trata todos los elementos del sistema proporcionando todos los modos de fallo a considerar. Es posible cuantificar los acontecimientos mediante el uso de probabilidades o frecuencias esperadas para los diferentes modos de fallo y en muchas ocasiones se utiliza un FMEA como paso previo para la aplicación de otros métodos tales como el de Árbol de Fallos o el de Secuencias de Sucesos.

El Análisis Crítico de los Modos de Fallo y Efectos (FMECA) extiende el FMEA para incluir la probabilidad de cada modo de fallo y priorizar sus acciones correctivas. Tradicionalmente las formas de llevarlo pasan por el desarrollo del Número de Riesgo Prioritario (RPN), y por el cálculo de un Número de Criticidad por modo de fallo.

El desarrollo del RPN utiliza términos semánticos para clasificar las probabilidades de ocurrencia de los modos de fallo, la severidad de los mismos en una escala de 1 a 10, y su probabilidad de detección. Esas clasificaciones multiplicadas dan un RPN. Los Modos de fallo con un mayor RPN serán los más importantes y se les asignará mayor prioridad que a los demás. Este método es muy utilizado en la industria automovilística.

El cálculo del Número de Criticidad comienza categorizando la severidad del efecto del modo de fallo y desarrolla a continuación un rango de criticidad, que no es otra cosa que la probabilidad de que un fallo ocurra con la severidad prevista. El procedimiento permite determinar: La probabilidad del efecto del fallo, la tasa de fallo del elemento o parte y su tiempo de operación, y usar esos valores para calcular el Número Crítico por cada severidad de efecto del modo de fallo. Esta variante se utiliza más frecuentemente en la industria Nuclear y Aeroespacial. El principal inconveniente de estos métodos es la vaguedad o imprecisión en los datos, lo cual hace imprescindible la aplicación de algún método para abordar la incertidumbre, de entre los cuales la Lógica Difusa es aplicado/a con frecuencia.

El Análisis de Riesgo y Operabilidad (HAZOP) también denominado Estudio del Potencial de Peligro y Operabilidad, es un método cualitativo cuyo objetivo fundamental es identificar potenciales de peligro de un sistema técnico. Se parte de una descripción física y funcional del sistema y se intenta, mediante el uso de diagramas de flujo, conocimiento técnico sobre el comportamiento y palabras guía, detectar causas y consecuencias de desviaciones de los parámetros de sus valores nominales. Han sido empleados ampliamente en la industria química para detectar sucesos no deseados que se han de utilizar en el análisis de árbol de fallos posteriores (Hauptmans, 1986).

El Análisis de Markov (MA) es un método inductivo y cuantitativo caracterizado por conducir su análisis en múltiples direcciones o estados. Cuando las características estocásticas de los elementos de un sistema dependen del estado del

sistema, los diagramas lógicos (Árbol de Fallos, Secuencias de Sucesos) han de complementarse con técnicas especializadas para la evaluación cuantitativa de las medidas de fiabilidad. En particular, cuando esas características dependen de parejas de valores de estados inicial y final del sistema, la técnica indicada es el Análisis de Markov. Algunas de las circunstancias más comunes que generan dependencias del estado del sistema son:

- Sistemas con componentes reparables
- Sistemas con diferentes políticas de mantenimiento
- Operabilidad de sistemas en espera del fallo
- Redundancia de sistemas en espera del fallo
- Condiciones ambiente extremas
- Cargas comunes compartidas por diversos elementos
- Causas y modos comunes de fallo para varios elementos

La capacidad de los modelos Markovianos ha sido ampliamente reconocida y por ello muy utilizados. Su principal inconveniente estriba en el estudio de grandes sistemas en los cuales el número de estados y transiciones puede ser inabordable en la práctica, aún para los más potentes ordenadores.

El análisis de Secuencias de Sucesos (ET), también llamado Diagrama de Sucesos ó Árbol de Sucesos, es una técnica inductiva que ha sido empleada con profusión, en especial combinada con la de Árbol de Fallos. Describe de forma cualitativa las respuestas de un sistema técnico frente a un hecho inicial, normalmente un fallo, usando un encadenamiento hacia atrás que despliega tantas ramas como consecuencias tenga dicho hecho, contemplando tanto opciones de funcionamiento como de fallo de todos los elementos del sistema ya sean operacionales o de control. El resultado del análisis es una estructura ramificada que se representa mediante un diagrama. Cada rama del diagrama es la representación estática de un proceso dinámico. El diagrama es cuantificable asignando probabilidades a sus nodos, lo cual se puede hacer usando observaciones estadísticas ó con un análisis de Árbol de Fallos en el cual los fallos contemplados en el diagrama serán los sucesos no deseados a cuantificar. La aplicación más común de esta técnica se encuentra en la realización de Análisis Probabilístico de Riesgos (PRA) (Henley y Kumamoto, 1993) también denominados

Análisis Probabilístico de Seguridad (PSA), que son de especial relevancia para determinar efectos sobre la población de accidentes graves tales como los de Centrales Nucleares, Superpetroleros, Plantas Químicas e incendios de edificios con gran densidad de población o gran importancia socio/cultural.

Los Métodos Deductivos por su parte comienzan con la definición de un hecho o suceso de interés al nivel de sistema y proceden al estudio de las causas del mismo hasta el grado de detalle predefinido para el análisis. El Análisis de Árbol de Fallos es la técnica deductiva más extendida y en ella se enmarca el contenido del presente trabajo.

El Análisis de Árbol de Fallos comienza con la definición de un suceso no deseado al que se denominará "Suceso Top", el objetivo será analizar las causas de dicho suceso y cuantificar su probabilidad de ocurrencia. El Análisis de las causas se realiza mediante un diagrama lógico, el Árbol de Fallos, que muestra como las combinaciones de diversos sucesos más elementales, usando operadores Booleanos, conducen a la aparición del suceso Top.

Esta técnica es aplicable a sistemas estáticos y dinámicos complejos para cuyo estudio es una completa y versátil herramienta. Provee bases objetivas para analizar el diseño de un sistema, sus causas comunes de fallo y permite comprobar el cumplimiento de los requisitos de seguridad, así como justificar cambios y añadidos. Como pone de manifiesto Lee (1982), el Análisis de Árbol de Fallos tiene gran utilidad entre otras para:

- Conducir al analista en forma deductiva a poner de manifiesto los posibles fallos del sistema.
- Destacar aspectos del sistema que son importantes desde el punto de vista de los fallos.
- Proporcionar una ayuda gráfica que haga visibles los cambios de diseño.
- Acceder al Análisis de Fiabilidad tanto cualitativo como cuantitativo.
- Facilitar a los analistas concentrarse en un fallo determinado de interés de forma rápida y ordenada
- Incrementar el conocimiento del analista sobre el sistema en estudio.

La técnica de Análisis de Árbol de Fallos fue concebida en los laboratorios Bell Telephone en 1961 para ayudar en el análisis de seguridad del sistema de lanzamiento de los misiles Minuteman (Bell Telephone, 1961). Fue posteriormente desarrollada por la compañía Boeing (Haasl, 1965) y a partir de entonces fue aplicada en multitud de disciplinas científicas tales como tecnología aeroespacial, fiabilidad de reactores nucleares, industrias químicas, sistemas tolerantes a fallos, sistemas de seguridad, sistemas de potencia y como base para la elaboración de las reglas que componen parte de la base de datos de conocimientos de un sistema experto.

En los primeros años de desarrollo se emplearon métodos de Monte Carlo para es estudio cualitativo y cuantitativo para pasar rápidamente al empleo de métodos deterministas tales como el Algoritmo de Test de Combinaciones implementado en los conocidos códigos PREP-KITT (Vesely y Narum, 1970), el algoritmo descendente de Fussell y Vesely (1972) implementado en el código MOCUS, el Sistema de Transformación de Ecuaciones SETS (Worrell, 1974) y el algoritmo eficiente de Kumamoto et al. (1978). Una gran aportación fue la aparición de la Teoría Cinética del Árbol (Vesely, 1969) que continua siendo referente obligado hoy día. En al década de los 80 se amplían y perfeccionan los métodos basados en Grupos de Corte Mínimos con profusión de códigos que los implementan, pero también comienzan a evaluarse los árboles de forma directa usando algoritmos recursivos (Page y Perry, 1986) y programación orientada a objetos (Koen, 1989). Es también en esta época cuando se estudian nuevos reductores de varianza para simular el árbol comparados por Fishman (1986) y se comienza con las primeras experiencias con sistemas expertos para la construcción. La década de los 90 a representado un gran incremento en el potencial de aplicación de la metodología, comenzando por el congreso de Copenhage (1992) donde se analizan nuevas tendencias y por la presentación de los métodos implícitos vía Diagramas de Decisión Binarios (Coudert y Madre, 1993, 1994). Nuevas y prometedoras aplicaciones se derivan del uso de modernas puertas lógicas (Dugan et. al. ,1992) así como del empleo de sofisticados sistemas expertos (Geymair y Ebequen, 1995). El desarrollo en esta década de sofisticados métodos de reducción de varianza en otras áreas de la ciencia permite el empleo de simulación con eficiencia competitiva con los métodos antes citados. Por último señalar las excelentes y completas revisiones bibliográficas realizadas por Vesely et al. (1981), Lee (1982) y Misra (1993), así como las aplicaciones en Análisis Probabilístico de Riesgos cuyo exponente magistral fue la gran obra conocida como WASH 1400 (1975).

CAPÍTULO II
ANÁLISIS DE ÁRBOL DE FALLOS



2.1.- ANÁLISIS DE ÁRBOL DE FALLOS

El Análisis de Árbol de Fallos es un método deductivo que permite obtener información cualitativa y cuantitativa de un sistema bajo estudio. Este análisis se realiza sobre el denominado Árbol de Fallos, entendiendo por ello, la representación gráfica de un modelo lógico con combinaciones de eventos que conducen a la aparición de un suceso predefinido y no deseado de un sistema. La información cualitativa más importante está constituida por la identificación de aquellos grupos mínimos de fallos que conducen a la aparición del fallo no deseado. La información cuantitativa más relevante es la No Disponibilidad y la Fiabilidad del sistema. Este tipo de análisis se puede aplicar tanto en la etapa de diseño como en la de explotación de un sistema técnico, pudiendo proporcionar en ambos casos mejoras substanciales en los procesos involucrados.

La introducción de nuevas metodologías ha llegado en la última década de la mano de los sistemas expertos, ha supuesto un notable avance cuyas últimas realizaciones incorporan flexibilidad y capacidad de resolución mediante la integración de herramientas y conocimientos usando interfaces gráficas de última generación (Huang, 1992) (Elliott, 1994) (Geymayr y Ebecken, 1995)(Kócza y Vossche, 1997). Tradicionalmente el concepto de Árbol de Fallos se ajustaba a la definición que de él hacía Hauptmans (1986) como representación estática de procesos que normalmente son dinámicos y por tanto, necesitan resultados de cálculos dinámicos. Los modernos sistemas, sobre todo de control, incluyen en casos de alta complejidad, ordenadores de propósito específico embebidos en sus diseños operacionales. Sus especiales requerimientos incluyen cierta tolerancia a los fallos, lo que conduce a determinados tipos de fallos que no son modelables con la lógica convencional (operación (1) - fallo (0)) debiendo permitir múltiples estados intermedios. En muchas aplicaciones actuales los nodos de un árbol de fallos (puertas y sucesos básicos) pueden tener más de un nodo "padre" y por tanto, la consideración como "árbol" pierde su sentido, pareciendo más apropiada otra denominación tal como "estructura de fallos". En las figuras siguientes se pueden apreciar dos árboles de fallos, el primero usa lógica clásica para modelar un sistema eléctrico, mientras que el segundo usa lógica de nueva generación para modelar un sistema tolerante a fallos.

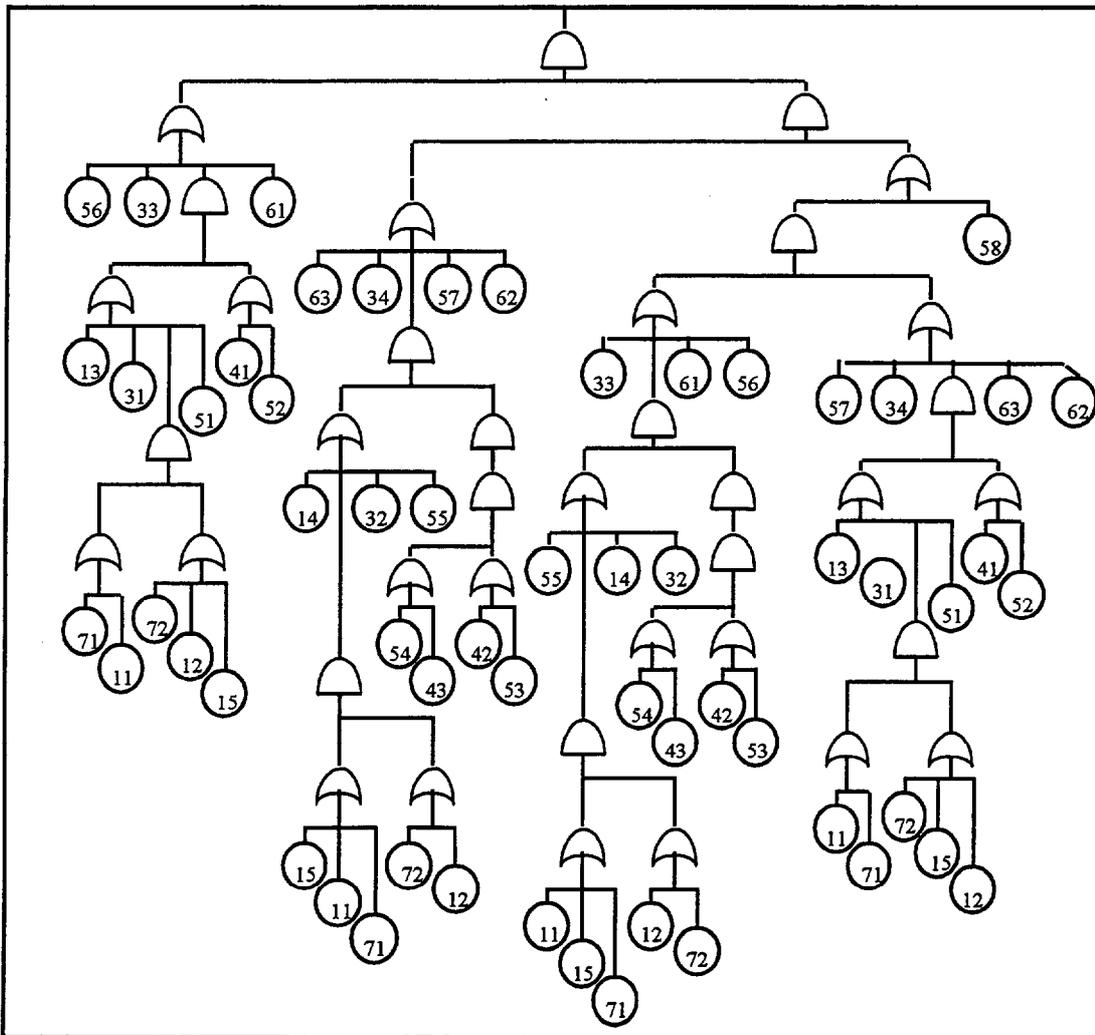
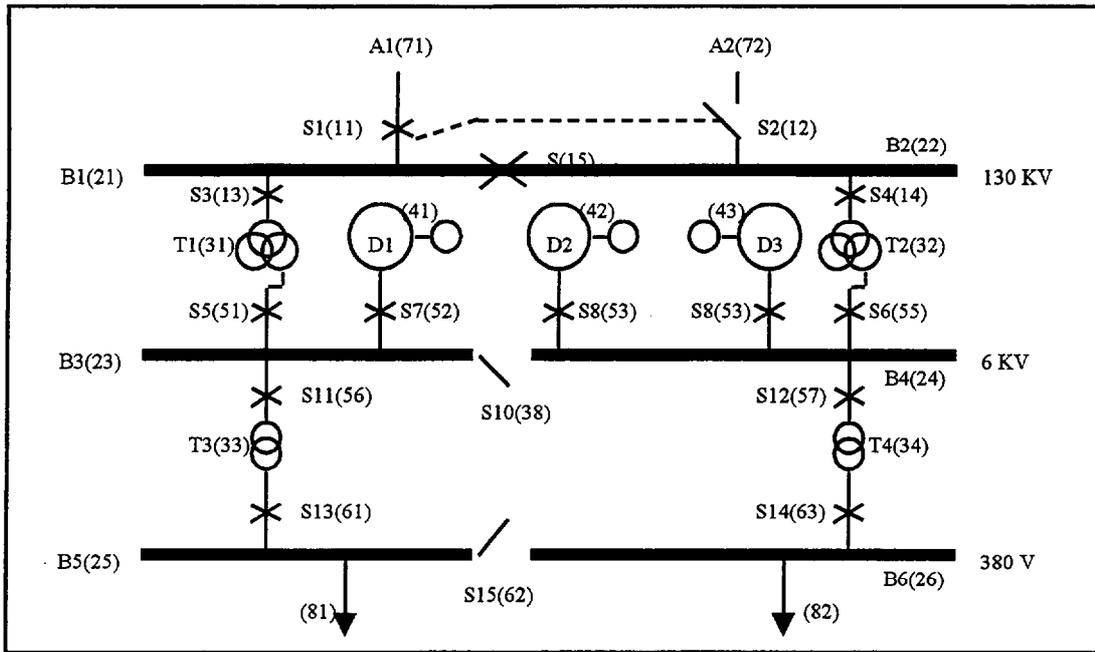


Fig. 2.1.- ÁRBOL DE FALLOS PARA EL SUCESO "Sin suministro eléctrico en (81)"
(Camarda et al., 1978)

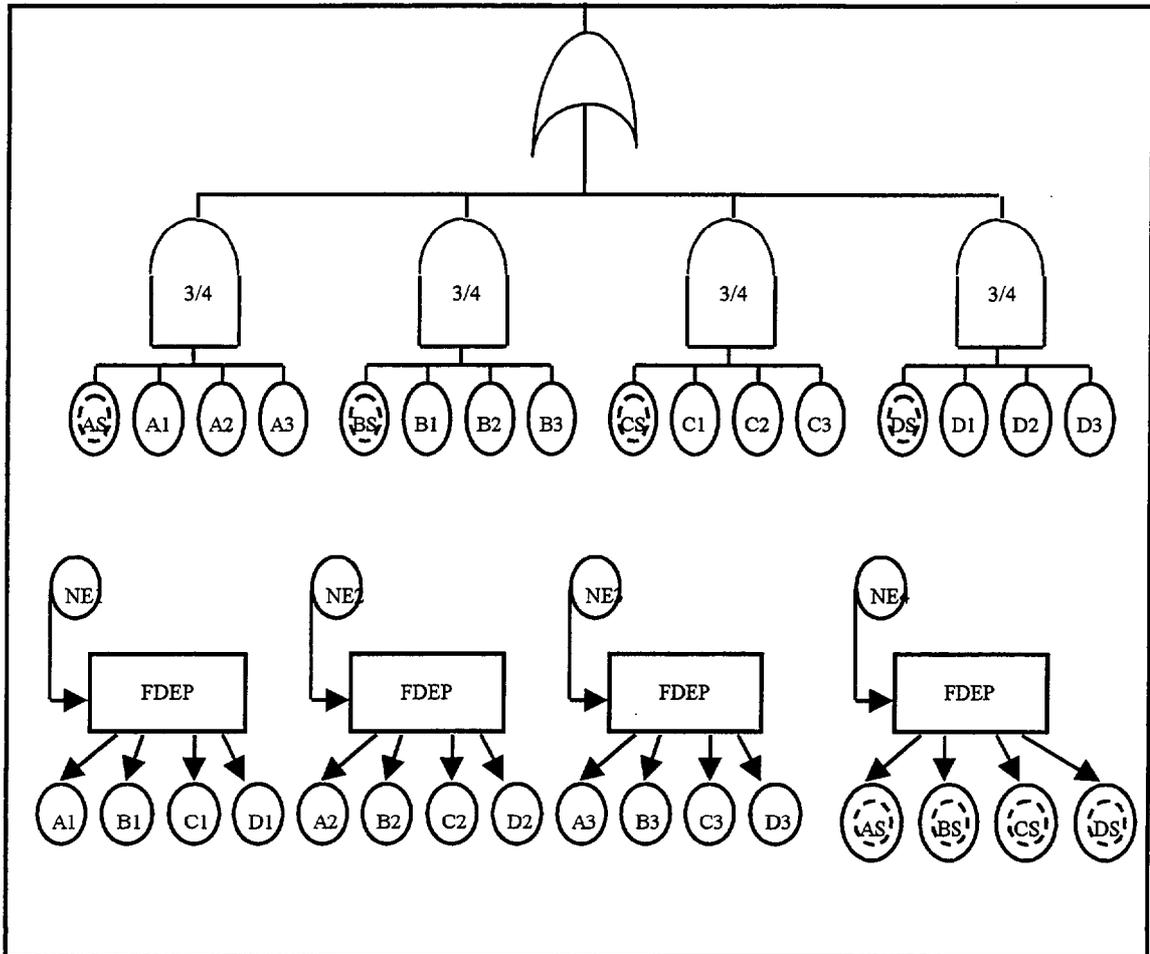
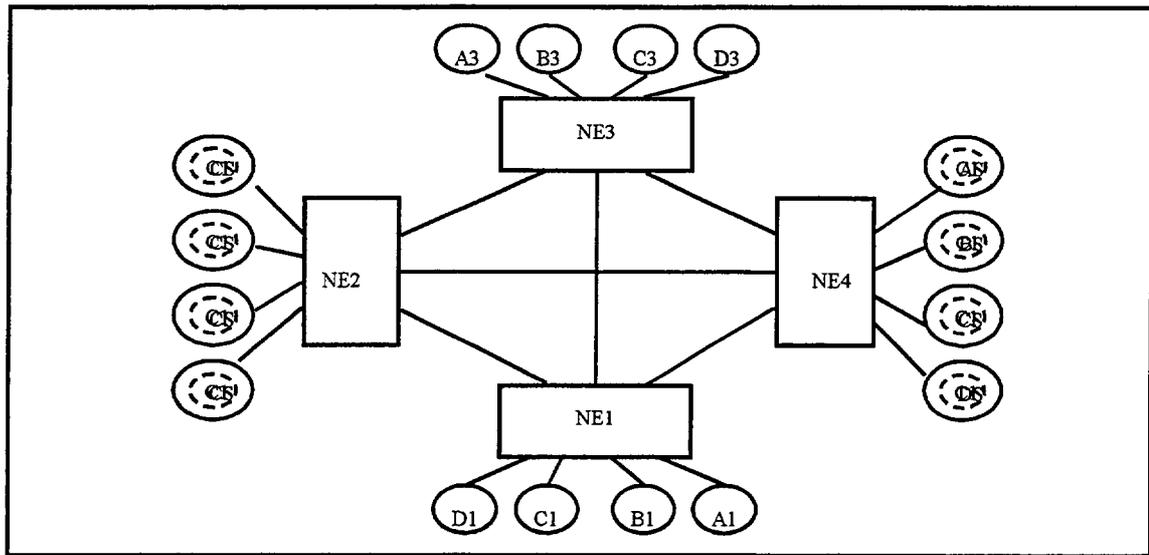


Fig. 2.2.- ÁRBOL DE FALLOS CON PUERTAS DINÁMICAS PARA PROCESADORES PARALELOS TOLERANTES A FALLOS (Dugan y Doyle, 1996)

2.2.- CONCEPTOS FUNDAMENTALES

2.2.1.- El espacio de los fallos como entorno de análisis

La operación de un sistema puede considerarse desde dos puntos de vista (Vesely et al., 1981): Podemos enumerar los diferentes caminos para que el sistema funcione (**ESPACIO DEL FUNCIONAMIENTO**) o podemos enumerar los diferentes caminos para que falle (**ESPACIO DE LOS FALLOS**).

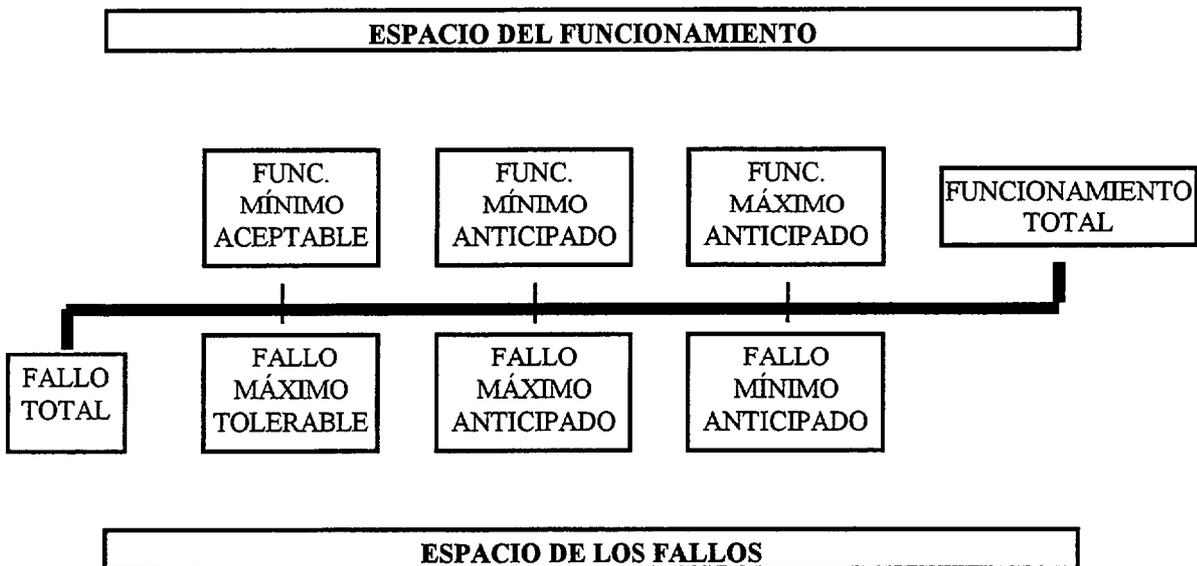


Fig. 2.3.- ESPACIO DEL FUNCIONAMIENTO Y DE LOS FALLOS COMO ENTORNO DEL ANÁLISIS DE ÁRBOL DE FALLOS

El trabajo en el espacio de los fallos requiere de una formalización de sus elementos teniendo en cuenta las diferentes tareas inherentes a una buena administración de los fallos, las fundamentales son: Detección, Identificación, Categorización, Evaluación, Estudio, Almacenado, Efectos y Prevención.

Para la detección de los fallos es necesario disponer de fuentes, generalmente On-line, de información rápida y fiable sobre el estado del sistema técnico que se administra. Generalmente existe una red de sensores que alimentan dispositivos de alarma e

intervención (tanto inmediata como diferida). Dependiendo de la especialización del sistema de detección de fallos es posible identificar un fallo con mayor o menor rapidez. Como mínimo deben ser identificables de forma inmediata aquellos fallos que puedan afectar a la integridad del sistema técnico y/o las personas de su entorno. Una vez detectado e identificado, el fallo deberá ser categorizado (primario, secundario, etc.) y evaluado, llegando a cualificadores como "Severo", "Importante" etc. El estudio de un fallo consiste básicamente en su descomposición en causas más elementales si se pudiese. Esto podría estar ya hecho si existiese algún tipo de estudio previo como FMEA, FMECA, etc. El almacenado de un fallo es equivalente al "aprendizaje" de la existencia del mismo y requiere una codificación apropiada para su utilización en cualquier otra circunstancia.

Los efectos de un fallo de modo conocido deberán estar analizados en un FMEA o FMECA previo, de no ser así deberán evaluarse éstos utilizando la experiencia y el conocimiento que tengan sobre el sistema los ingenieros de explotación.

La prevención está compuesta por aquel conjunto de acciones que permita corregir y/o prevenir el fallo tanto de forma inmediata como diferida mediante su inclusión en los planes de Mantenimiento Preventivo y Programado.

En los últimos años se han realizado aplicaciones de sistemas expertos basados en árbol de fallos para la detección y el diagnóstico de fallos (Kim & Modarres, 1987), (Chen & Modarres, 1990), (Patton & Clark, 1989), (Dhurjati, 1992) (Kneale, 1997). La representación del conocimiento basada en reglas ha sido el modelo empleado y por el momento carecen de generalidad y transparencia suficiente, además de presentar un comportamiento pobre para analizar las situaciones novedosas (Geymayr, 1995). Algunos investigadores en sistemas expertos para el diagnóstico de fallos en procesos, fundamentalmente de industrias químicas, están de acuerdo en que la mejor estrategia está en emplear una cierta combinación de sistemas basados en reglas y empleo de Programación Orientada a Objetos, como una solución de compromiso entre la eficiencia del proceso y el comportamiento cualitativo (Teixeira et al., 1993). Existen múltiples razones que sugieren la necesidad de analizar un sistema desde el punto de vista de los fallos y más concretamente utilizando análisis de Árbol de Fallos. Entre ellas cabe destacar:

- El número de situaciones de fallo es mucho menor que el de situaciones de funcionamiento.
- Un árbol de fallos no considera todas las formas de fallo de un sistema, sólo aquellas que están relacionadas con un fallo no deseado (suceso top) y dentro de estas sólo las que son relevantes para el mismo desde el punto de vista del analista.
- El árbol de fallos es una vía para orientar la formación y ordenar la información de que se dispone para el estudio de las causas de sucesos no deseados.

2.2.2.- Terminología

Para la correcta interpretación de un árbol de fallos es necesario establecer una terminología cuyas definiciones se presentan seguidamente:

SUCESO FINAL o TOP.- Suceso no deseado que se quiere estudiar y que es el situado más alto en un árbol de fallos.

COHERENCIA.- Un árbol de fallos lo llamaremos "coherente" si no aparecen al mismo tiempo un suceso y su negación como sucesos básicos, en caso contrario se denominará " no coherente ".

SUCESO DE ORDEN SUPERIOR.- Es aquel que puede ser descompuesto en sucesos más elementales, representándose en el árbol por una puerta lógica.

SUCESO DE ORDEN BÁSICO.- Es aquel que no puede ser descompuesto en sucesos más elementales

FALLO.- Es un suceso no deseado que puede estar ocasionado por:

El buen funcionamiento de un elemento o sistema fuera de tiempo y forma
(Orden Superior)

El mal funcionamiento de un elemento en tiempo y forma (Orden básico, Avería)

NODO.- Todo suceso básico o puerta lógica de un árbol de fallos

OCURRENCIA Y EXISTENCIA DE LOS FALLOS.- En los sistemas no reparables los fallos ocurren y continúan existiendo indefinidamente, mientras que en los reparables un fallo ocurre y existe sólo hasta que finalice su reparación.

CATEGORIAS DE LOS FALLOS.- Los fallos pueden agruparse dentro de tres categorías:

Fallo Primario.- Fallo de un componente que ocurre en un entorno para el cual está cualificado (diseñado), también llamado Avería Primaria

Fallo Secundario.- Fallo de un componente que ocurre en un entorno para el cual no está cualificado (diseñado), también llamado Avería Secundaria.

Fallo de Orden.- Fallo producido por un buen funcionamiento en tiempo y forma/lugar equivocados

MECANISMO, MODO Y EFECTO DE LAS AVERÍAS.- Por Efecto de una avería entenderemos las consecuencias que ésta puede causar, si las hubiere, en el sistema. Por Modo de la avería entendemos la forma física en que ésta ocurre. Por Mecanismo de una avería entendemos como se produce un determinado modo de avería

2.2.3.- Símbolos lógicos

Algunas símbolos lógicos empleados se pueden apreciar en las siguientes tablas:

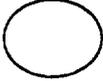
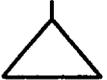
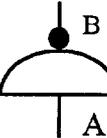
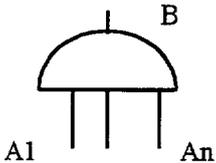
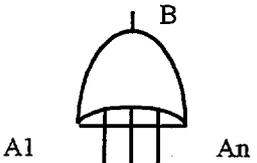
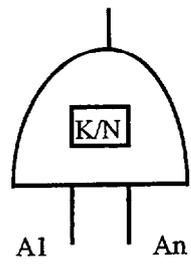
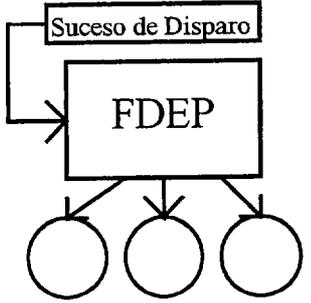
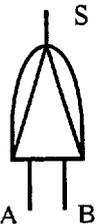
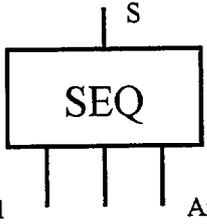
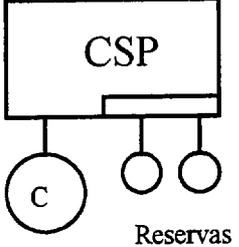
SÍMBOLO	DENOMINACIÓN	SIGNIFICADO
	Rectángulo	Describe una variable
	Círculo	Variable primaria que pertenece a un componente independiente
	Octógono	Variable primaria que pertenece a un componente dependiente
	Rombo	Variable no primaria que debiera ser descompuesta en variables más básicas pero que por alguna razón no lo ha sido.
	Casa	Una variable cuyo espacio muestral contiene sólo un elemento. Es constante y tomará siempre uno de los valores 1 ó 0. Este símbolo se usa sólo como entrada a una puerta AND.
	Transferencia de Entrada	Es un símbolo de transferencia o conexión. Indica que una variable entra al árbol de fallos
	Transferencia de Salida	Es un símbolo de transferencia ó conexión. Indica que una variable sale del árbol de fallos
	NOT, $B = \bar{A}$	La salida toma el valor "1" si la entrada tiene el valor "0" y viceversa
	AND, $B = \prod_{i=1}^n A_i$	La salida toma el valor "1" si, y sólo si, todas las entradas toman el valor "1"
	OR, $B = \prod_{i=1}^n A_i$	La salida toma el valor "1" si, al menos, una de las entradas toma el valor "1".

Fig. 2.4.- SÍMBOLOS LÓGICOS PARA ÁRBOLES DE FALLOS

	<p>K ocurren de N</p>	<p>Se expande en K sobre N posibilidades</p>
	<p>Dependencia Funcional</p>	<p>Los Sucesos Básicos dependientes son forzados a ocurrir cuando ocurre el suceso de disparo El suceso de disparo debe ser un subárbol</p>
	<p>AND prioritaria</p>	<p>La salida S ocurre si ocurren A y B pero además A ocurre antes que B Todas la entradas deben ser subárboles Se pueden poner en cascada</p>
	<p>puerta de secuencia obligatoria</p>	<p>La salida S ocurre si ocurren todos los sucesos A1...An en secuencia 1...n A1... An deben ser subárboles</p>
	<p>puerta de reserva fria</p>	<p>El componente C será reemplazado , si falla, por los reservas, los cuales se supone que no fallan durante su puesta en funcionamiento</p>

© Universidad de Las Palmas de Gran Canaria. Biblioteca Digital. 2003

Fig. 2.4 SÍMBOLOS LÓGICOS PARA ÁRBOLES DE FALLOS (Continuación)

2.3.- ETAPAS DEL PROCESO DE ANÁLISIS

Siguiendo los trabajos de Vesely et al. (1981) el análisis mediante árbol de fallos se puede descomponer generalmente en cuatro etapas:

Definición del sistema.- Compuesta por una descripción detallada del sistema y su entorno hasta el nivel de desagregación que se desee. Se incluirán diagramas de flujo, planos y esquemas funcionales de interés para el estudio que se quiera realizar.

Construcción del Árbol de Fallos.- En el que se desarrolla un árbol de fallos para un suceso no deseado, empleando ciertas reglas, un conjunto de símbolos lógicos, la descripción del sistema y el conocimiento (experiencia) que se tenga del comportamiento del mismo.

Análisis Cualitativo.- Con el que se obtiene información relevante acerca de las interrelaciones entre los fallos de los componentes y el fallo no deseado, susceptible de ser empleada para modificar el diseño y/o el proceso de operación del sistema.

Análisis Cuantitativo.- En el que se asignan valores a los sucesos básicos y se procede a evaluar el árbol para obtener magnitudes asociadas al suceso final no deseado que se estudia.

No todos los autores conceden igual importancia a estas etapas. Así, por ejemplo, para **Hauptmans (1986) el análisis de árboles de fallos es un método deductivo que se utiliza casi exclusivamente de forma cuantitativa**, y en cuanto a las etapas del proceso, propone el esquema que se refleja en el diagrama de bloques que sigue.

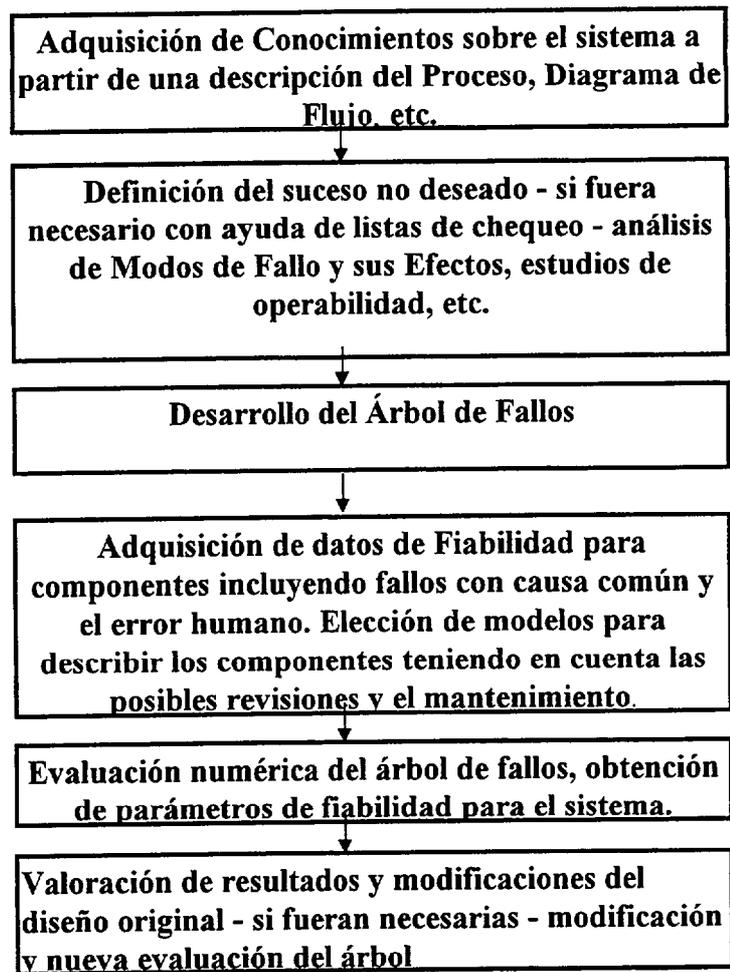


Fig. 2.5. - ESQUEMA DE FLUJO PARA LA ELABORACIÓN DE UN ANÁLISIS DE ÁRBOL DE FALLOS (Haptmans, 1986)

El diagrama de flujo anterior es representativo de los denominados métodos Procedurales (Procesos lineales con bucles de realimentación) característicos de los esquemas tradicionales del Análisis de Árbol de Fallos. Los nuevos métodos basados en Sistemas Expertos con reglas, constituyen una aproximación diferente con un alto grado de conectividad y comunicación con el usuario. En la figura siguiente se pueden apreciar los elementos fundamentales de un sistema experto para la realización de un análisis de este tipo, presentado por Geymayr y Ebecken (1995) que será detallado más adelante. Con este tipo de sistemas ya no es tan fácil y correcto hablar de "etapas" del proceso de análisis, parece más lógico hablar de "sesiones" de análisis. En cada sesión el usuario genera, compara, evalúa y estudia diversos árboles de fallos, ayudado por un entorno

gráfico que facilite la edición y modificación de árboles y por unas fuentes de conocimientos que le proporcionan herramientas especializadas para todas las tareas que se realizan, siempre de forma interactiva a través de un panel de control global ó "pizarra". En este sistema se utiliza programación Orientada a Objetos, usando las conocidas propiedades de Encapsulado, Herencia y Polimorfismo para representar los árboles de fallos en el dominio de los objetos. Permite también contemplar en el análisis la Incertidumbre y Vaguedad en los datos mediante el uso de la Lógica Difusa, así como integrar múltiples fuentes de conocimientos en un sólo análisis. Si el usuario no es experto sus resultados serán notablemente superiores usando este sistema que con un método convencional.

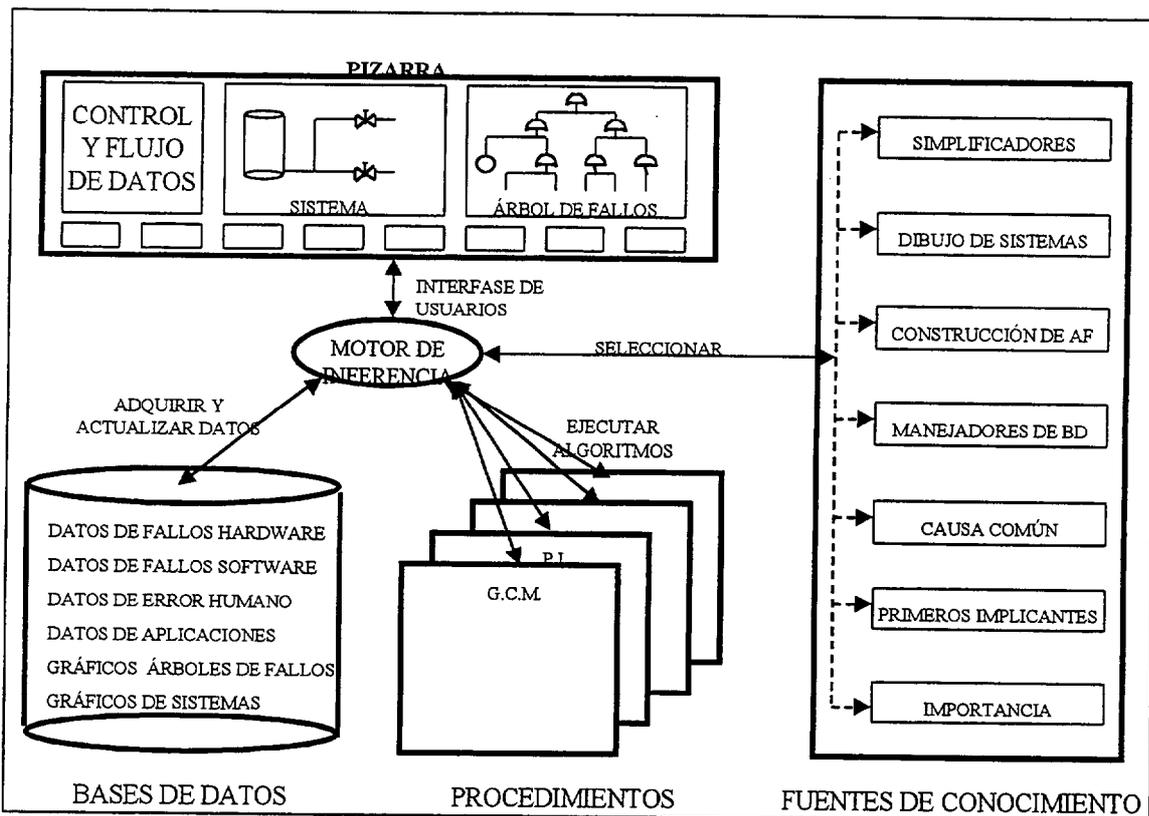


Figura 2.6.- SISTEMA EXPERTO DE SEGUNDA GENERACIÓN PARA ANÁLISIS DE ÁRBOL DE FALLOS(Geymayr y Ebecken,1995)

2.4.- DEFINICIÓN DEL SISTEMA

La definición del sistema al que se va a aplicar el análisis de árbol de fallos ha sido una materia muy controvertida en la literatura especializada, esto ha derivado principalmente del hecho de que las herramientas de construcción del árbol han sido obtenidas en base a formas muy concretas de definir los sistemas a estudiar. Una forma muy general de definir un sistema es decir que es *toda entidad determinista compuesta por un conjunto de elementos discretos que interaccionan entre si*, definición que se debe a Vesely et al. (1981). En orden a conseguir un análisis eficiente de árbol de fallos son necesarios tres tipos de características:

- Componentes, modos de operar y modos de fallo. Lo cual implica una descripción de cómo los estados de salida están influidos por los estados de entrada y modos de operación interna de tales componentes.
- Diagramas del sistema. Que incluye las relaciones internas entre los diversos componentes, es decir, como las entradas y salidas de cada componente están conectadas con las demás.
- Establecer las condiciones de contorno del sistema. Que definen claramente la situación para la cual se representa el árbol de fallos, que son fundamentalmente: condiciones externas, condiciones internas y límite de resolución. Las condiciones externas determinan la amplitud del análisis y el límite de resolución el detalle con que se realiza el mismo.

Especial atención deberán recibir las descripciones tanto de los componentes individuales como los diversos subsistemas presentes, describiendo con todo detalle sus modos de operar en todas las situaciones normales y anormales en que pueda considerarse dividida la actividad del mismo, y también todos los modos posibles de fallar y sus consecuencias sobre componentes, subsistemas y proceso. Es altamente recomendable antes de iniciar la construcción de árboles de fallos sobre un sistema la realización de un exhaustivo análisis de Los Modos de Fallo y sus Efectos, con la ayuda de un software específico que recoja con amplitud los conocimientos y experiencias operativas pasadas de operarios y expertos.

2.5.- CONSTRUCCIÓN DEL ÁRBOL DE FALLOS

La construcción de un árbol de fallos se convierte a menudo en una tarea titánica ya que la necesidad de los analistas de ofrecer resultados aceptables (fiables) obliga, con frecuencia, a fijar unos límites tanto interiores (nivel de desagregación) como exteriores (alcance) de cierta amplitud. Todo ello desemboca en cientos o miles de fallos de todo tipo y, por tanto, en árboles de cientos o miles de nodos lo que hace que hoy siga siendo el cuello de botella de este tipo de análisis. La lucha por mantener la coherencia y el control sobre el contenido del árbol ha ido inevitablemente unida al ritmo del progreso técnico en cuanto al desarrollo de la informática, la presentación del primer ordenador personal "XT" en 1981 abre una etapa de expansión en la que se incrementan notablemente las aportaciones desde todas partes del mundo. Todo lo anterior ha provocado una vertiginosa carrera de logros aparejados a las prestaciones cada vez más potentes (y baratas) de los ordenadores de sobremesa. Hasta mediados de la década de los ochenta, la construcción de los árboles de fallos estuvo dominada por los procedimientos procedurales que se basan en diagramas de flujo lineales con bucles de realimentación, clásicos en la algorítmica empleada para programar los ordenadores disponibles (Mainframes o Miniordenadores). A partir de esa fecha comienza la aplicación progresiva de Sistemas Expertos, la Programación Orientada a Objetos, la Lógica Difusa y los Entornos Gráficos avanzados, cuyo nivel de desarrollo e integración actual es alto, pero de los que se espera mucho más en los próximos años.

2.5.1.- Descripción del Proceso de Construcción

El proceso de construcción comienza con la definición del suceso no deseado para el cual se construirá el árbol de fallos (Top). Una vez definido este se procede a determinar las causas inmediatas del mismo, las cuales se pueden definir (Vesely et al., 1981) como las causas necesarias y suficientes para que aparezca. Estas causas serán consideradas, a continuación, de forma independiente como sucesos "sub-top" y se determinan las causas necesarias y suficientes para que ocurra cada una de ellas; y así sucesivamente hasta alcanzar el nivel de desagregación elegido.

La complicación del proceso de construcción del árbol de fallos radica en la diversidad de elementos que intervienen en el proceso y en la falta de capacidad de los conocimientos actuales para reflejar todas las particularidades de cada elemento al mismo tiempo que los unifica en un proceso común que debe desembocar en el árbol de fallos.

Analizando los elementos que intervienen en el proceso de construcción, podemos distinguir los siguientes:

a) **Un modelo** que describe de la manera más adecuada posible a un sistema técnico, cuantos más detalles, características y especificaciones, relevantes para la construcción del árbol, contenga el modelo más preciso podrá ser el árbol construido, pero si el número de ellos es abrumador, harán imposible la construcción. Lo que significa que es necesario elegir "que características" del sistema reflejará el modelo, tanto en cantidad como en calidad de forma previa al comienzo de la tarea de obtener del A.F.

b) **Uno o varios "expertos"** encargados de la construcción del árbol de fallos con cierto grado de experiencia en los siguientes aspectos:

- Conocimiento físico del sistema (Experiencia del diseño), Conocimiento operacional del sistema (Experiencia de Operación).
- Comportamiento del sistema fuera de los límites nominales de Operación.
- Formas de fallo del sistema (nominales y fuera de límites).
- Comportamiento de las secuencias ó cadenas de fallos, tolerancias y márgenes (operativos y de tiempos)
- Comportamiento de otros sistemas y equipos parecidos.
- Proceso de Construcción de un árbol de fallos.
- Criterios válidos y contrastados sobre la relevancia de los eventos y/o fallos en la explotación del sistema.
- Hábito de desenvolverse en el entorno de los fallos.

c) **Un procedimiento marco** manual o informatizado, que garantice el seguimiento de las reglas básicas de construcción del árbol de fallos mientras estas son aplicadas al modelo del sistema.

Finaliza este proceso con la comprobación de que el árbol generado es completo (se han desarrollado todas las puertas lógicas) y conexo (todas las puertas están conectadas de alguna forma al árbol resultante). Todos estos elementos componen un proceso de construcción que habrá de repetirse tantas veces como sea necesario, por ejemplo si se está en la etapa de diseño de un sistema habrá que estudiar el árbol de fallos para un mismo suceso top en todos los modelos de los diferentes diseños alternativos.

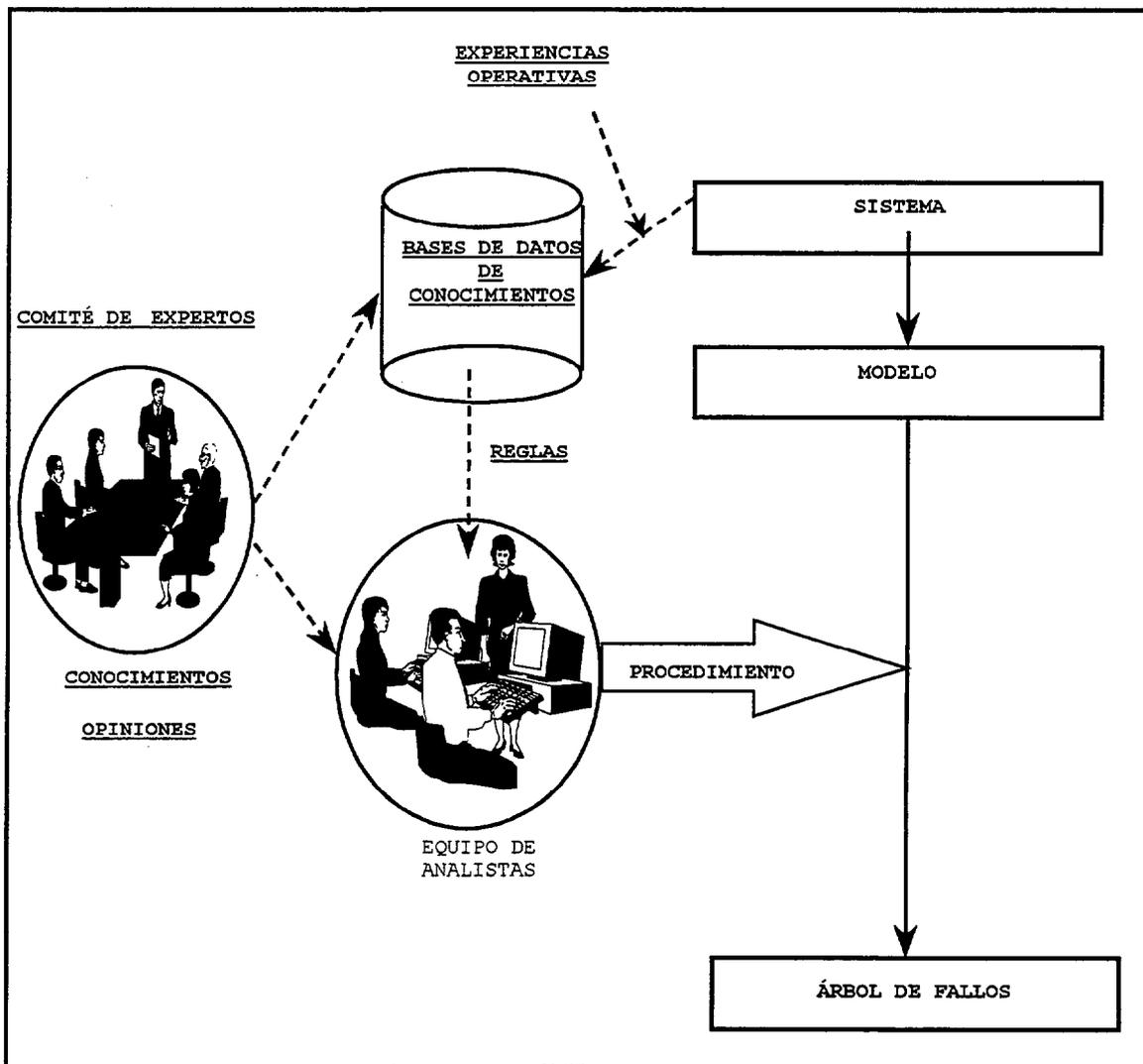


Fig. 2.7.- ELEMENTOS DEL PROCESO DE CONSTRUCCIÓN DE ÁRBOLES DE FALLOS

El procedimiento para construir un árbol de fallos debiera ser diferente según los objetivos que se pretenda alcanzar. Así si lo que se pretende es obtener un árbol de fallos en un sistema muy grande, en un tiempo relativamente corto y que refleje en todo lo posible los conocimientos de diferentes expertos sobre el sistema, lo mejor será emplear un sistema experto. Si, por el contrario, lo que se pretende es adquirir un conocimiento profundo sobre las causas de fallo de un sistema de un tamaño no muy grande, lo mejor es construir manualmente el árbol de fallos.

No es fácil conciliar ambos objetivos y, por supuesto, ambos métodos, no existe la solución universal, todo dependerá pues del problema a estudiar y de los condicionantes de contorno (tiempo, medios y expertos) de que se disponga, para adoptar una decisión óptima. Lo que si es evidente, es la evolución en los sistemas de construcción de árboles de fallos asistidos por ordenador, lo cual ha sido parejo a la evolución técnica de los Ordenadores Personales, de los Sistemas Operativos y de ciertas especialidades de la Inteligencia Artificial. A pesar de los evidentes avances todavía se confía la última revisión "visto bueno" del árbol de fallos al ser humano experto. Para que el proceso anterior conduzca a un árbol de fallos que represente correctamente la lógica que conduce al suceso no deseado, es necesario observar una serie de reglas básicas (Vesely et al., 1981):

Regla I : Escribir las descripciones de los sucesos como fallos, precisando de manera escueta y clara ¿Qué fallos es? y ¿Cuándo ocurre?. Abrevie las palabras pero no las ideas.

Regla II : Realícese para cada fallo la pregunta ¿Es este fallo una avería de componente?. Si la respuesta es afirmativa clasifique el fallo como avería de componente, si la respuesta es no clasifíquelo como fallo de sistema.

Regla III : Si el funcionamiento normal de un componente ayuda a propagar un fallo, entonces dicho componente funciona con normalidad (No ha fallado)

Regla IV : Todas las entradas de una puerta deben ser definidas completamente antes de emprender el análisis individual de cada una de ellas.

Regla V : Las entradas de una puerta no deben proceder directamente de otra puerta, ya que, toda puerta debe conducir a un suceso que deberá estar perfectamente categorizado como fallo según la regla I.

2.5.2.- Metodologías para la Construcción

Según se expuso en el apartado anterior las metodologías para la construcción pueden ser diferenciadas atendiendo a las siguientes características:

- a) Forma de Modelar el sistema a estudiar.
- b) Tipo de algoritmo para obtener el árbol a partir del modelo.
- c) Estrategia, si la hay, para utilizar el conocimiento.

Se debe a Haals (1965) la primera formalización del proceso completo de desarrollo de un árbol de fallos. Describió un proceso estructurado en el que se establecen reglas para determinar el tipo de puertas a usar y las entradas y salidas de la misma. Haals divide el árbol en tres partes : Estructura Superior, que incluye es suceso no deseado (top) y sus causas inmediatas, en cuya definición no interviene el proceso estructurado. Nivel Intermedio, en el que se divide el sistema en fases y subfases, hasta que las características del sistema no cambian el entorno ambiental del fallo. El analista examina el sistema desde un punto de vista funcional. Nivel Inferior, en al que el proceso estructurado conduce a un nivel detallado de fallos de forma deductiva hasta el nivel de desagregación elegido.

Fussell (1973) presentó su Modelado de Árbol Sintético (STM), que es un modelo sistemático de construcción de árbol de fallos basado en la elaboración de una función de transferencia del fallo de cada componente del sistema , las cuales deben ser definidas de forma independiente del sistema bajo estudio. Seguidamente se procede al trazado (coalición) de funciones de transferencia a través del esquema del sistema para obtener el árbol de fallos. En la figura siguiente se puede apreciar el diagrama de flujo del proceso, que fue implementado en el código DRAFT y aplicado para sistemas eléctricos.

Power y Tompkins (1975) y **Taylor (1982)** emplean mini árboles de fallos de componentes y/o subsistemas conocidos para componer el árbol para un suceso top específico. El método de Power y Tompkins permite contemplar, además de las clásicas interacciones de tipo mecánico, complejas interacciones de tipo químico.

Camarda , Corsi y Trentadue (1978) desarrollaron un algoritmo para la síntesis automática del árbol de fallos partiendo del grafo probabilístico (grafo que muestra todas las posibles formas de funcionar correctamente que posee un sistema), a partir del cual se obtienen los caminos mínimos para que el sistema funcione. Dichos caminos proporcionan, mediante una transformación adecuada los grupos de corte mínimos, que constituyen una forma válida de expresar un árbol de fallos, y pueden ser utilizados para una posterior evaluación numérica. En la figura siguiente se presenta una recopilación de referencias bibliográficas atendiendo a las características del proceso de construcción empleado.

METODOLOGÍA	REFERENCIAS BIBLIOGRÁFICAS
Proceso Estructurado	Haals, 1965
Uso de Tablas de Decisión	Salem et al, 1979
Uso de función de transferencia de fallos	Fussell, 1973
Uso de Mini árboles de fallos	Taylor, 1982 Power y Tompkins, 1973, 1975
Uso de Grafo Probabilístico	Camatda et al, 1978
Uso de Digrafos	Lapp y Powers, 1977 Lambert, 1979 Lapp y Powers, 1979 Lambert et al, 1980 Allen y Rao, 1980 Cummings et al., 1983 Kumamoto y Henley, 1986 Andrews y Morgan, 1986 Andrews y Brenman, 1990 Iverson, 1992 Boyd e Iverson, 1993
Uso de Árboles Causales	Bossche, 1985, 1991
Uso de Digrafos + Árboles Causales + Conocimiento	Kohda y Henley, 1988 Kohda et al., 1989 Kohda e Inoue, 1993
Retroalimentación	Schneeweiss, 1992
Sistema experto	Huang y Li, 1992
Uso de Redes Semánticas y Conocimiento	Kumamoto, 1993
Método orientado a objetos	Cepin y Mavko, 1998
Modelos previos	Kócza y Bossche, 1997
Estados finitos de máquinas	Liggsmeyer y Rothfelder, 1998

Fig. 2.8.- REFERENCIAS BIBLIOGRÁFICAS SOBRE CONSTRUCCIÓN DE ÁRBOLES DE FALLOS SEGÚN LA METODOLOGÍA EMPLEADA

Las Metodologías de construcción basadas en **Tablas de Decisión, Digrafos y Redes Semánticas** se expondrán básicamente en los próximos apartados.

2.5.3.- Uso de Tablas de Decisión

Salem, Apostolakis y Okrent (1978), pusieron a punto el complejo código CAT. Este código es un conjunto integrado para el modelado de sistemas tales como los nucleares y otros muy complejos que comprenden interacciones mecánicas, eléctricas, hidráulicas y humanas, así como el análisis de fallos por causa común. Se basa en el uso de tablas de decisión para el modelado de los componentes (1979), seguido de un proceso paso a paso de edición para acoplar los diversos componentes, y un seguimiento a través del sistema para llegar a construir el árbol. Si se tiene suficiente información sobre el sistema que va a ser analizado y una buena librería de modelos de componentes, este sistema permite la construcción rápida y ordenada del árbol de fallos de una manera estandarizada.

Una tabla de decisión es una extensión de una tabla de verdad en la cual se pueden usar un número de estados por cada entrada de la misma, ya que la lógica 0-1 es, normalmente, insuficiente para representar elementos de sistemas complejos. Por tanto en tabla de unidad descubrirá cada posible estado de salida como un conjunto completo de estados de entrada e entornos tanto operacionales como de fallos. Las tablas de decisión, una vez filtrados para no contener filas equivalentes o situaciones indiferentes, se usan para generar árboles de fallos.

Dado un estado de salida a estudiar se procede a buscar todas aquellas filas que produzcan dicho estado las cuales alimentarán una puerta OR. A continuación se estudian los componentes de cada fila por separado para determinar los elementos de cada rama. En la figura siguiente se puede apreciar una tabla de verdad para un fusible (elemento de sistema) así como el árbol de fallos obtenido para el suceso no deseado "sin señal a la salida del fusible" (o). Este método de modelado de componentes se complementa con un algoritmo que permite ir acoplando unos a otros todos los elementos del sistema siguiendo la pista, hacia sus causas, al suceso no deseado a través de todo el sistema, dicho algoritmo de generación se vuelve muy complicado cuando se encuentran negaciones o lazos, siendo los árboles obtenidos en estos casos poco parecidos a los obtenidos de forma manual.

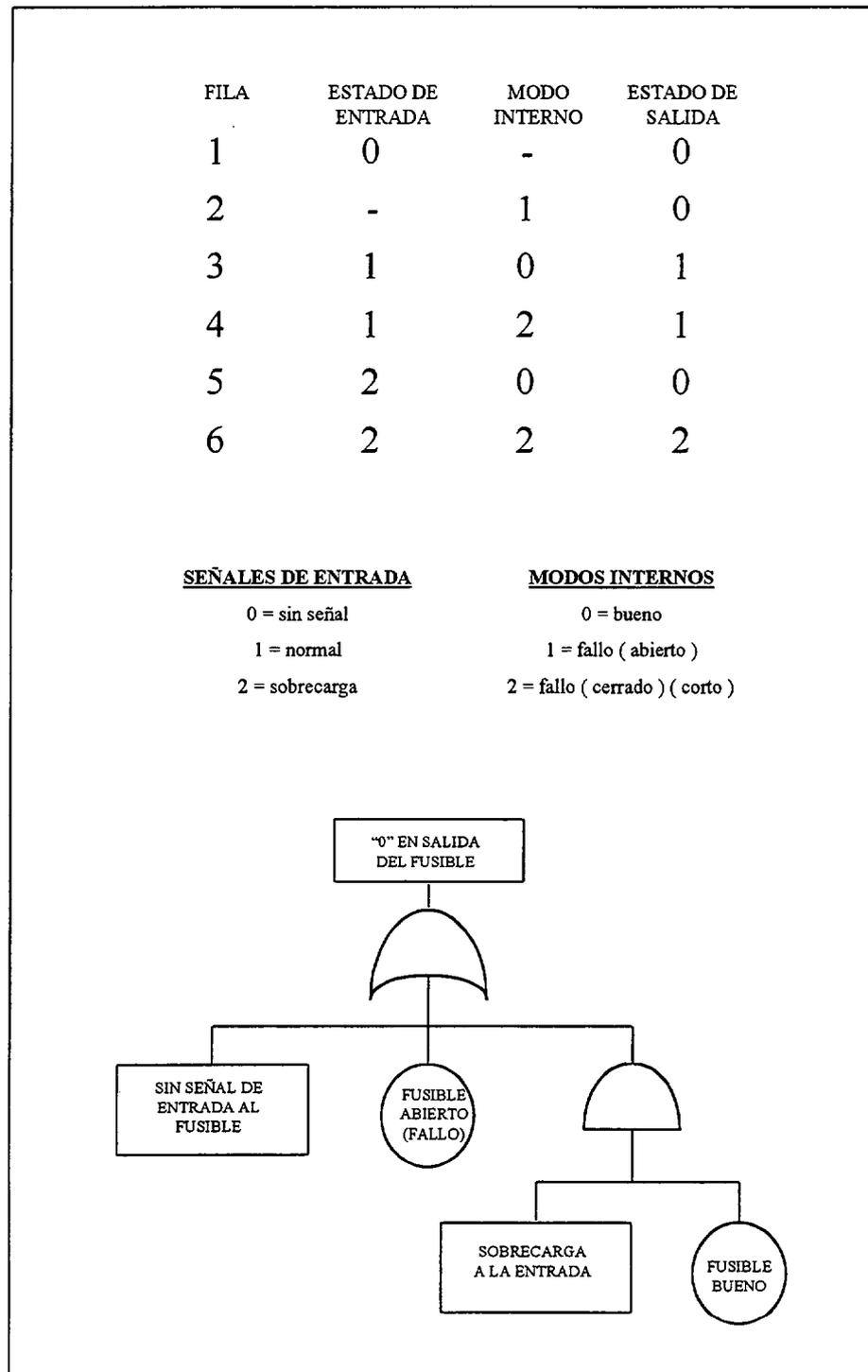


Fig. 2.9.- TABLA DE DECISIÓN Y ÁRBOL DE FALLOS PARA UN FUSIBLE.

2.5.4.-Uso de Digrafos.

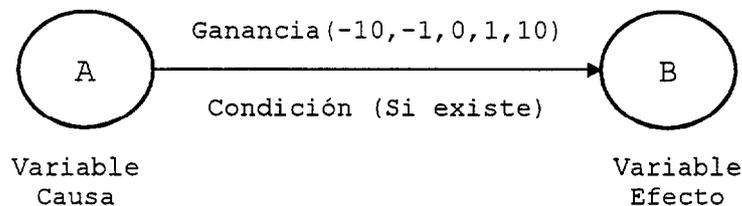
Un Digrafo (Grafo Dirigido) es un modelo del sistema compuesto Nodos y Ramas. Los Nodos representan variables del proceso tales como: Temperatura, Presión, Flujo u otros de sucesos anormales, mientras que las Ramas representan relaciones entre las desviaciones de las variables de los Nodos (Andrews y Brennan, 1990). El uso de Digrafos para modelar el sistema, como paso previo para la obtención del árbol de fallos, ha sido una técnica ampliamente aceptada y se han presentado múltiples aportaciones que la usan (Lapp y Powers, 1977), (Boyd e Iverson, 1993).

En una Rama hay que considerar tres elementos:

Dirección. - La flecha en la rama va desde la variable causa hasta la variable efecto.

Ganancia. - Representa la desviación entre las variables de los nodos y pueden tomar los valores 10,1,0,-1,-10, donde 1 indica ganancia moderada y 10 ganancia fuerte.

Signo. - El signo de la ganancia indica el sentido en el que cambia la variable efecto cuando se incrementa la variable causa.



Una rama podrá estar condicionada a ciertas situaciones de las variables lo cual se expresará también en la misma. En los nodos las desviaciones se representan también con 10,1,0,-1,-10, el signo indica la dirección de una desviación comparado con el valor normal, o indica una desviación, 1 indica desviación moderada y 10 gran desviación. Si un nodo representa a un fallo el valor 0 indica que no ha ocurrido y el valor 1 que si ha ocurrido. En la tabla siguiente se pueden observar las relaciones entre las variables de los Nodos causa, las ganancias de las ramas y las variables de los nodos efecto.

Ganancia de la Rama	Variable del nodo Origen	Variable del nodo Destino
+10(-10)	+10 +1 -1 -10	+10(-10) +10(-10) -10(+10) -10(+10)
+1(-1)	+10 +1 -1 -10	+10(-10) +1(-1) -1(+1) -10(-10)
0	+10 +10 -1 -10	* * * *

Fig. 2.10.- RELACIONES ENTRE VARIABLES DE NODOS

Para la Construcción del digrafo del Sistema se seguirán los siguientes pasos.

- 1) Indicar la desviación de la variable del Proceso que representa el suceso top.
- 2) Localizar las variables relacionadas con la variable anterior ya sea a través de acciones componentes o de leyes de la física.
- 3) Añadir las nuevas variables al digrafo conectadas a la variable del suceso top mediante ramas apropiadas.
- 4) Repetir para cada nueva variable a partir del paso 2.
- 5) Terminar cuando se llegue al nivel de desagregación elegido.

Para la construcción del árbol de fallos a partir del Digrafo habrá que distinguir entre sistemas sencillos (sin lazos) ó complejos (con lazos). En los sistemas simples se procederá siguiendo las causas potenciales de desviación de la variable del suceso top; en un proceso hacia atrás encadenando con puertas OR hasta que se encuentran ramas condicionales para

las que se empleará una puerta AND ya que solo ocurrirá cuando se producen simultáneamente la desviación de la variable y la condición. Para árboles complejos será necesaria la identificación de todos los lazos de retro y post alimentación para poder proceder a la construcción del árbol según el esquema anterior, ello obliga a la definición de un conjunto de operaciones relacionadas con tales identificaciones, que varía según el algoritmo desarrollado al efecto. Los principales inconvenientes son: la falta de claridad sobre quién dominará cuando dos flechas de ganancias opuestas fluyen al mismo nodo y la complicación del algoritmo de generación del árbol (Kumamoto, 1993).

Iverson (1992) presentó una técnica de traducción automática de Digrafo a Árbol de Fallos que incluye aquellas que contienen lazos. El programa que realiza dicha tarea utiliza Programación Orientada a Objetos para representación de los modelos y resuelve los lazos utilizando sucesos repetitivos en el árbol de fallos

Para resolver los lazos del digrafo cada nodo repetido en el árbol de fallos corresponde a un nodo en el digrafo y debe tener los mismos grupos de corte mínimos que éste. Esto se realiza construyendo un subárbol a partir de los grupos de corte Mínimo del nodo del digrafo. El nodo repetido se transforma en una puerta OR con una entrada por cada Grupo de Corte Mínimo del nodo del digrafo, si el grupo de corte tiene un sólo elemento se añadirá un nodo de avería y si tiene varios se añadirá una puerta AND con un sólo nodo de avería por cada miembro del grupo de corte. Los Grupos de Corte Mínimo para los nodos del digrafo se calculará de forma muy similar a la descrita anteriormente, durante su cálculo cada nodo del digrafo que es visitado se marca como tal, si uno de esos nodos se encuentra de nuevo la recursión se para.

Para completar la conversión, los grupos de Corte mínimos para cada nodo del digrafo que corresponde a un nodo repetido del árbol de fallos se convierten en nodos avería y se construye un subárbol para ellos siguiendo el proceso ya descrito. El resultado será un árbol de fallos en el que cada nodo tiene los mismos grupos de corte que su correspondiente nodo en digrafo del sistema. El algoritmo ha sido implementado en LISP y G.

En las figura siguiente se puede apreciar un sistema técnico y su correspondiente digrafo para el suceso top especificado.

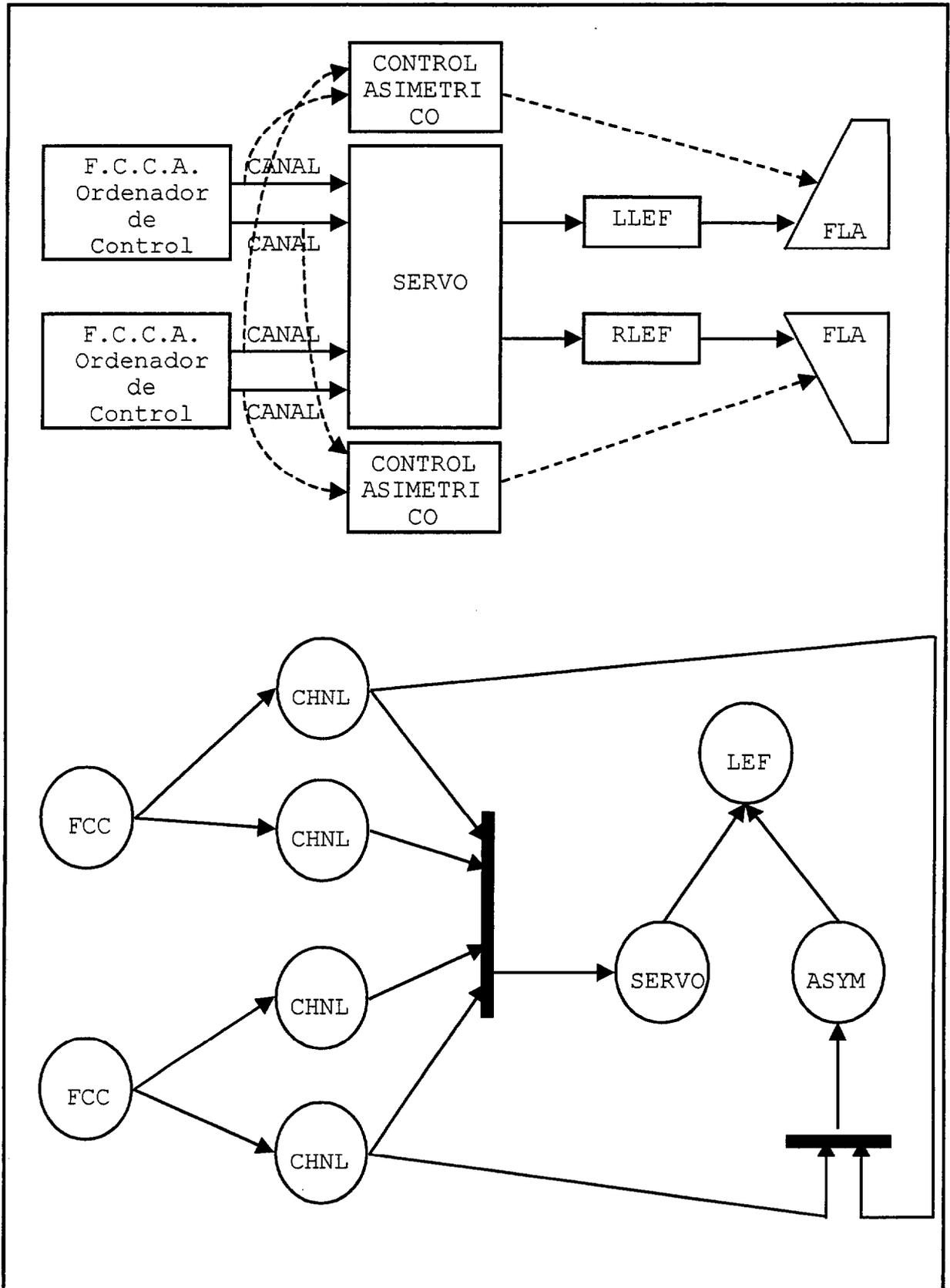
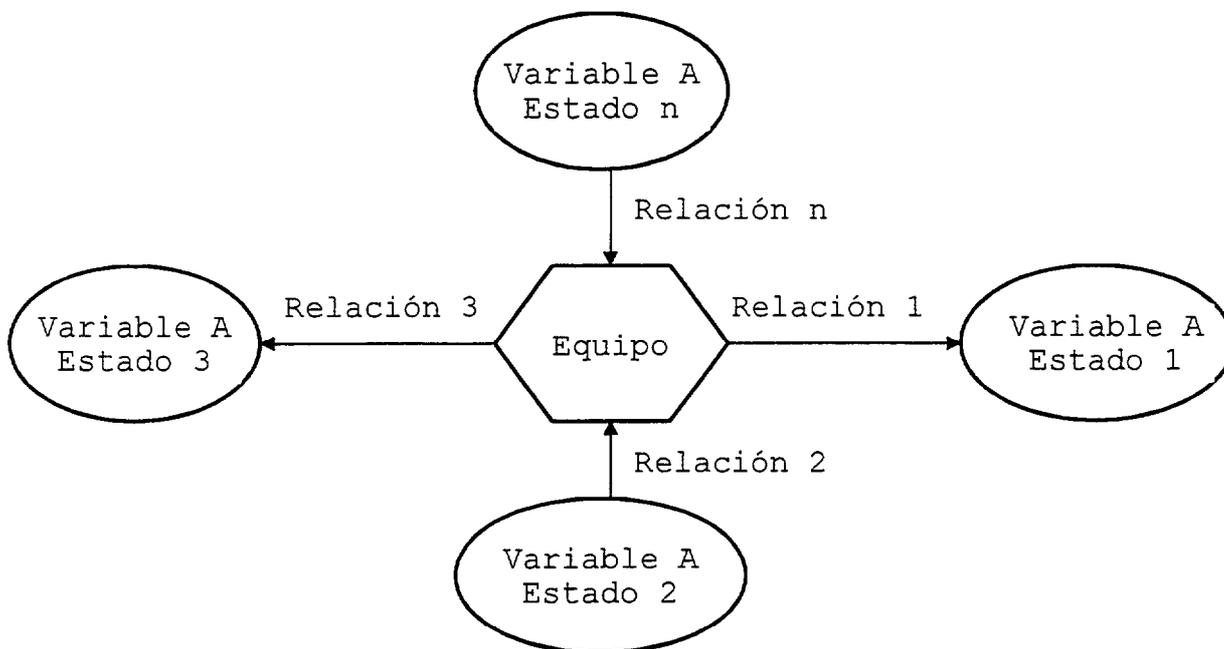


Fig. 2.11.- DIGRAFO PARA EL SISTEMA DE CONTROL DEL BORDE DE ATAQUE DE LOS FLAPS DEL F-18 (Doyle et al., 1995)

2.5.5.- Uso de Redes Semánticas

El uso de Redes Semánticas para modelar el sistema unido a un conjunto de reglas "If-Then" para localizar las causas de los eventos y representar el conocimiento de los expertos, es una reciente y novedosa aportación (Henley y Kumamoto, 1993), ya que las dos estrategias de representar el conocimiento mencionadas son muy importantes en el desarrollo de sistemas expertos.

Un sistema puede ser modelado usando una red semántica haciendo corresponder a cada elemento del mismo una relación simple ó múltiple con una variable significativa a través de ramas con etiquetas que identifica la relación. Para facilitar la construcción de la red semántica se utiliza una librería que contiene las redes semánticas de los equipos (Piezas) básicas del sistema, con una librería adecuada pueden ser posibles análisis detallados de sistemas complejos.



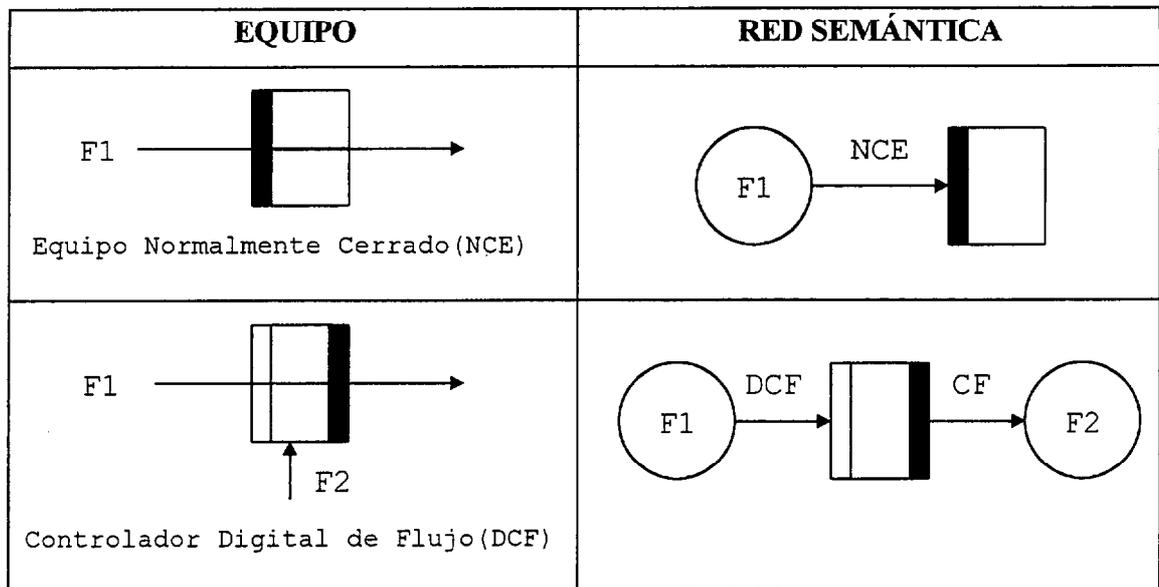


Fig. 2.12.- EJEMPLOS DE REDES SEMÁNTICAS DE EQUIPOS
(Henley y Kumamoto, 1993)

El método de construcción de árboles de fallos propuesto por Henley y Kumamoto (1993) incorpora :

- a) Representación mediante redes semánticas del sistema a analizar (modelo del sistema)
- b) Uso de reglas para localizar las causas de los eventos. (uso del conocimiento)
- c) Un algoritmo recursivo para generar el árbol de fallos utilizando la red semántica y las reglas, (procedimiento marco)

Para éste método existen tres tipos de eventos ligados por reglas de relación:

- a) Ternas de flujo (Flujo, Atributo, valor)
- b) Equipo sospechoso, indica que una pieza específica del equipo es sospechosa de causar un evento.
- c) Equipo averiado, es una avería básica de un equipo (son con frecuencia los Sucesos básicos).

Ternas (Flujo, Atributo, valor)**A) Los Flujos se pueden clasificar en:**

- (1) Flujos de Material (Líquido, gas, y vapor)
- (2) Flujos de Información (Señales, datos, comandos, alarmas)
- (3) Flujos de Energía (Luz, calor, sonido, vibraciones).
- (4) Otros (Fuego, chispa, etc.)

B) Los atributos de un flujo son controlados o determinados por cada pieza del equipamiento del sistema. Los atributos más comunes son: Tasa de Flujo, tasa de generación, apertura, Temperatura, presión, concentración etc.

C) Cada atributo tiene diversos valores representativos:

Apertura. - Fully-closed (F-CI) Increased (Inc) Constant (Cons) etc.

Tasa de Generacion. - Zero, Inc, Cons, Dec, Max, Positive, Not-max.

Tasa de Flujo. - Idem que los de generación.

Utilizando todas estas definiciones las ternas (Flujo, Atributo, Valor) definen la existencia u ocurrencia de un determinado evento.

Finalmente un algoritmo recursivo que retorna un valor compuesto por una parte para inclusión y exclusión de eventos y otro de inclusión y exclusión de puertas.

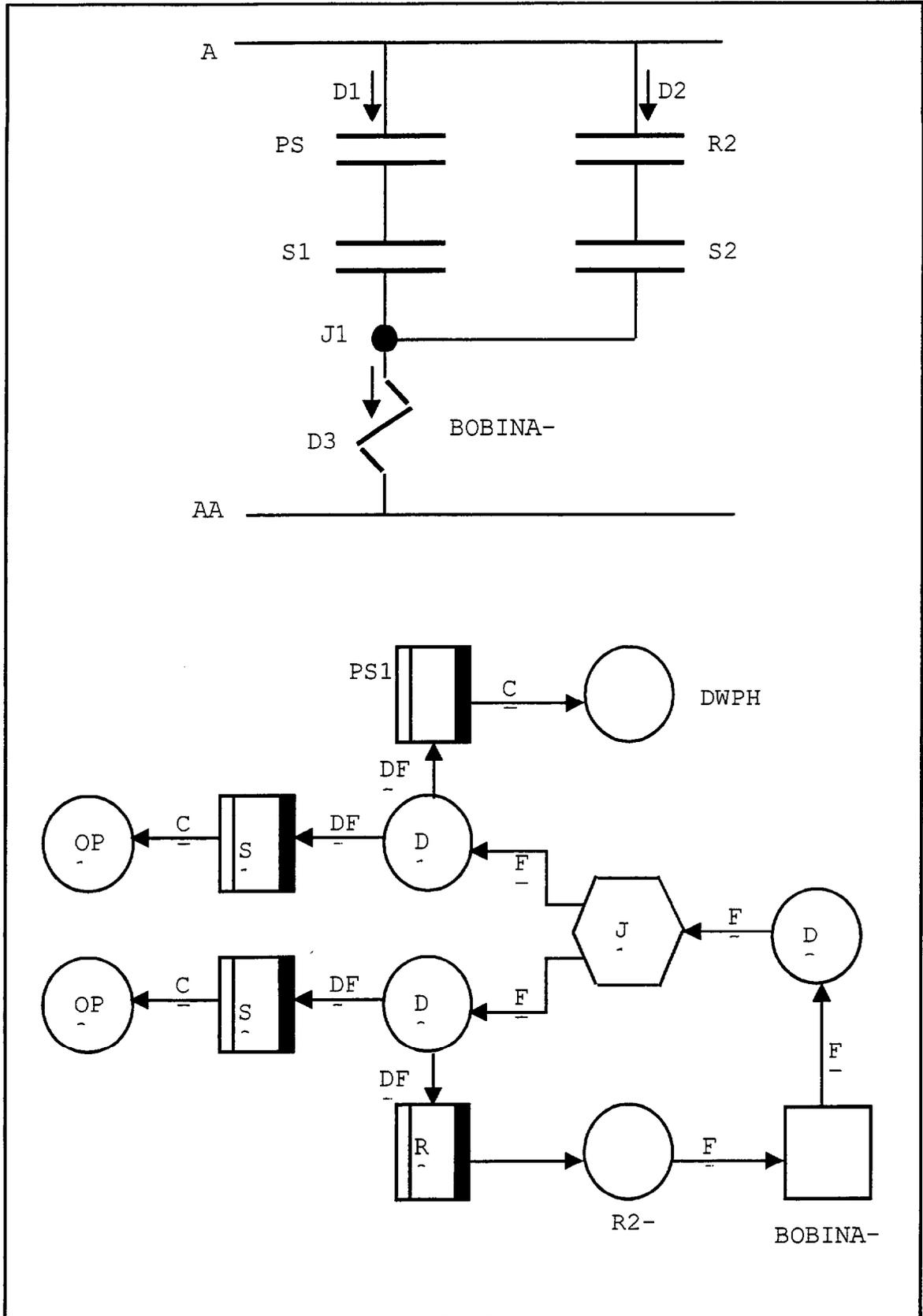


Fig. 2.13.- REPRESENTACIÓN SEMÁNTICA DE UN CIRCUITO DE RELÉS
(Henley y Kumamoto, 1993)

2.6.- ANÁLISIS CUALITATIVO

2.6.1.- Introducción

El resultado más frecuente de la etapa de construcción del árbol de fallos es el grafo del mismo, o sea, la representación gráfica de las relaciones lógicas entre las variables básicas, intermedias y Top. A partir de este grafo se puede obtener la función lógica que lo representa, que será en nuestro caso una FUNCION BOOLEANA DE n VARIABLES INDICADORAS BINARIAS del suceso Top y a la que denominaremos FUNCION DE ESTRUCTURA. Este tipo de funciones Booleanas posee múltiples representaciones y será necesario encontrar aquellas que son adecuadas para analizar el sistema representado, desde un punto de vista ingenieril.

La mejor manera de comprender como falla un sistema, o sea, como se presenta el suceso top no deseado, es encontrar aquellas combinaciones mínimas de sucesos básicos que son suficientes para ello. Estas combinaciones son las componentes de una de las representaciones posibles de la Función de Estructura, que se denomina Forma Disyuntiva Normal (FDN)(obtenida como la suma de productos de las variables que componen cada una de éstas combinaciones), y se llaman Primeros Implicantes (PIs). Un caso especial de función de estructura es aquella en la que no aparecen como sucesos básicos negaciones de otros sucesos básicos, denominada Coherente, para la cual los PIs se denominan Grupos de Corte Mínimos (GCM).

La determinación de los PIs (o de los GCM) no es una tarea trivial dado que pueden ser billones de ellos los que compongan la FDN de una función de estructura medianamente compleja. Para afrontarla será necesario disponer de:

- Algoritmos que generen de forma eficiente los PIs
- Usar estrategias adecuadas para representar y almacenar los PIs que empleen la menor cantidad de recursos (memoria y operaciones) posibles.

La obtención de los PIs como elemento fundamental para comprender como falla un sistema es la piedra angular del denominado **Análisis Cualitativo del Árbol de Fallos**. Este análisis se puede abordar mediante diversos métodos que son agrupables en dos clases:

- **Directos**, que abordan la función de estructura tal cual se ha obtenido del árbol de fallos. Esta es la opción clásica, desarrollada durante muchos años y con logros bien establecidos hasta la actualidad, pero con limitaciones que obligan a procesos de truncamiento y aproximaciones en el caso de árboles muy grandes o complicados.

- **Indirectos**, que se basan en transformar el árbol de fallos en otro tipo de grafo llamado DIAGRAMA DE DECISIÓN BINARIO (DDB) antes de proceder a la determinación de los PIs. Esta es una opción reciente y novedosa que presenta importantes ventajas que están siendo puestas de manifiesto con los trabajos actuales. Entre ellas cabe citar la posibilidad de abordar grandes árboles y de realizar la posterior evaluación de la probabilidad del suceso top de forma exacta e implícita. Sus inconvenientes más importantes son:

- La necesidad de escoger un orden determinado de las variables, que representan a los sucesos básicos, para obtener un DDB adecuado. Cuestión para la que no se han desarrollado criterios demostrados matemáticamente.
- La imposibilidad de aplicar los DDB a procesos dinámicos.

El Análisis Cualitativo suele ser complementado con la determinación de la importancia relativa de unos sucesos básicos frente a otros a efectos de su contribución a la No disponibilidad del sistema y con la determinación de Causas Comunes de fallos.

En este capítulo se describirá someramente la teoría de Funciones Booleanas de Variables Indicadoras necesaria para el estudio de la Función de Estructura, ya que ésta ha sido estudiada en profundidad (Schneeweiss, 1989) y se hará más énfasis en los métodos directos e indirectos mencionados, así como en las estrategias de representación y almacenamiento más importantes que permiten abordar la obtención de los PIs para sistemas reales complejos.

2.6.2.-Representación matemática del árbol de fallos

Una vez construido el árbol de fallos se procede a obtener una función booleana que lo represente. Si los sucesos básicos del árbol son $X_1, X_2, X_3, \dots, X_n$, la función que describe dicho árbol:

$$\phi(X_1, X_2, X_3, \dots, X_n) \quad X_i \in \{0,1\} \quad \phi\{0,1\}^n \rightarrow \{0,1\}$$

se denomina Función de estructura, que no es más que una función Booleana de variable indicadora binaria. Esta función contiene cualitativa y cuantitativamente toda la información que encierra el árbol de fallos acerca del suceso top no deseado. Las funciones de estructura son de dos tipos:

No coherentes.- Cuando aparecen sucesos básicos y sus negaciones.

Coherentes.- Cuando no aparecen negaciones de sucesos básicos.

Estas funciones Booleanas tienen múltiples representaciones. Desde el punto de vista cualitativo aquellas combinaciones suficientes para que ocurra el suceso top son de evidente interés. Estas combinaciones están presentes en una de las expresiones de la función que se denomina Forma Disyuntiva Normal (FDN). En el caso de funciones No Coherentes estas combinaciones se denominan Primeros Implicantes (PIs), sin son Coherentes se denominarán Grupos de Corte Mínimos (GCM).

Los PIs (o GCM) son únicos para el suceso top e independientes de las diferentes formas equivalentes de la función de estructura. Para obtener los PIs se empleara algún método matemático de transformación de la función de estructura.

Si $P_1, P_2, P_3, \dots, P_m$ son los primeros implicantes, la función de estructura tendrá la forma

$$\phi(X_1, X_2, X_3, \dots, X_n) = \sum_{j=1}^m P_j \quad P_j = \prod_{i=1}^j X_i \quad j = 1, \dots, m$$

La función de estructura puede ser expresada en función de elementos disjuntos aplicando la expansión de Shannon, que es una técnica general aplicable a todo tipo de

función Booleana y que goza de un gran número de aplicaciones interesantes. Según el teorema de Shannon una función Booleana de n variables puede ser expandida sobre una, dos o todas sus variables. Si la función anterior se expande por la variable X_1 nos daría:

$$\phi(X_1, X_2, X_3, \dots, X_n) = X_1\phi(1, X_2, X_3, \dots, X_n) + \bar{X}_1\phi(0, X_2, X_3, \dots, X_n)$$

donde los símbolos “.” y “+” corresponden a las operaciones de intersección y unión respectivamente. La ecuación anterior puede volver a ser expandida por otras variables y, cuando se ha llevado a cabo la expansión por todas las variables, las expresiones obtenidas se llaman Minitérminos y habrá 2^n de ellos. Si se intercambian “.” y “+” así como “0” y “1”, nos queda:

$$\phi(X_1, X_2, X_3, \dots, X_n) = (X_1 + \phi(0, X_2, X_3, \dots, X_n))(\bar{X}_1 + \phi(1, X_2, X_3, \dots, X_n))$$

esta expansión cuando se ha llevado a cabo en todas las variables se dice en Maxitérminos, y es complementaria de la anterior.

2.6.3.- Métodos directos para obtener los PI's / GCM

Introducción

Cuando se utiliza el árbol de fallos tal cual, con su función de estructura en cualquiera de sus formas, se está empleando un proceso directo de evaluación cualitativa. La determinación de los PIs (ó GCM) se puede abordar desde dos métodos bien diferentes: **Simulación**.- Se generan aleatoriamente combinaciones de estados de los sucesos básicos y se comprueba si producen el suceso top. En caso de producirse el suceso top, se almacenan las combinaciones para proceder más tarde a su reducción a mínimos mediante la aplicación de las leyes de Idempotencia y Absorción, o bien, mediante un algoritmo de eliminación de las variables no necesarias para la ocurrencia del suceso top.

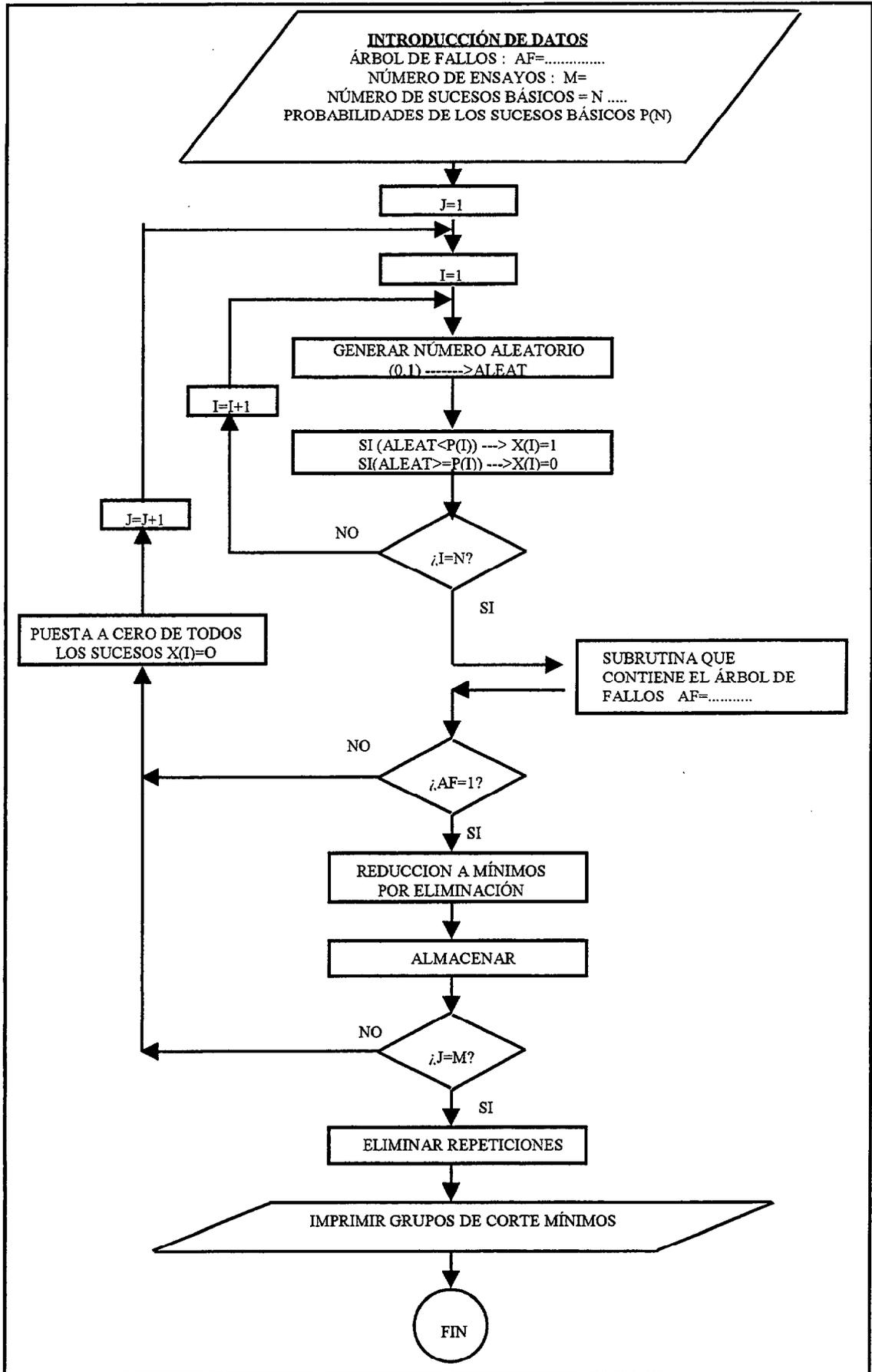
Deterministas.- Se aplican las leyes del álgebra Booleana y diversos teoremas demostrados para reducir la función de estructura a una FDN. Estos procesos suelen constar de dos fases: una en la que se sustituyen las puertas por sus entradas, que puede

desencadenarse ascendentemente (Bottom-Up) o descendientemente (Top-Down), y otra de reducción y simplificación a mínimos.

Los métodos deterministas gozan hoy en día de gran aceptación ya que se han conseguido algoritmos muy eficientes, sin embargo el gran incremento en las prestaciones de los ordenadores actuales así como el desarrollo de métodos de Reducción de Varianza eficientes permite aplicar métodos de simulación con éxito hasta árboles cada vez mayores. Además la simulación sigue siendo el método apropiado cuando se quiere incluir en el árbol nuevas puertas lógicas o estudiar dependencias de difícil modelado lógico o la evolución temporal de modelos complejos.

Métodos de Simulación:

Los métodos de simulación emplean los generadores de números aleatorios para obtener estados de cada suceso básico del árbol de fallos, componiendo con ellos conjuntos de estados denominados vectores de estado. Si la salida del árbol de fallos ocurre para un determinado vector de estado, este es almacenado como candidato a Primer Implicante (o GCM). El paso siguiente es la reducción a mínimos antes comentada. El generador de vectores de estado puede considerar situaciones temporales incluyendo tiempos de reparación de las averías o puede ser atemporal basándose sólo en las probabilidades de los sucesos básicos. En la figura siguiente se puede apreciar un diagrama de flujo clásico de este tipo de método. Las principales ventajas de los métodos de simulación son la sencillez de los códigos y la versatilidad para simular todo tipo de puertas lógicas. Sus principales inconvenientes son la gran cantidad de tiempo que se puede emplear en generar los PIs y la carencia de seguridad de haber encontrado todos los PIs de la Función de Estructura (Vesely, 1975) (Hauptmans, 1986) (Henley & Kumamoto, 1993).



© Universidad de Las Palmas de Gran Canaria. Biblioteca Digital. 2003

Fig. 2.14.- OBTENCIÓN DE LOS GRUPOS DE CORTE MÍNIMOS POR SIMULACIÓN

Métodos Deterministas:

Aplicar un método determinista consiste en transformar la función de estructura mediante la aplicación de Leyes, Teoremas y Propiedades de las funciones Booleanas (ó aplicables a ellas) para obtener una Forma Disyuntiva Normal (FDN) cuyos componentes serán los PIs (o GCM). El problema de encontrar los PIs de una función Booleana es bien conocido y ha sido abordado desde hace tiempo. Son de aplicabilidad para ello los procesos de Quine-McCluskey, Consenso de Quine y Doble Negación de Nelson (Schneeweiss, 1989). Sin embargo es más útil el empleo de algoritmos eficaces que encuentren la FDN en tiempo adecuado. En este sentido, la referencia anterior cita el algoritmo de Abraham (1979), un algoritmo basado en la expansión de Shannon ya descrita en apartados anteriores, cuya complejidad depende del número de elementos que tenga la FDN y del tamaño de éstos.

A lo largo de los años han sido elaborados muchos algoritmos por muy diferentes autores desde distintos puntos de vista. Una primera clasificación bien conocida diferencia si los algoritmos son Ascendentes (Bottom-Up) o Descendentes (Top-Down). Estos algoritmos recorren el árbol de abajo a arriba o de arriba a abajo respectivamente, descomponiendo la función Booleana, teniendo en cuenta que una puerta AND incrementa el número de sucesos básicos de un conjunto de implicantes (ó grupos de corte en árboles coherentes), y una puerta OR incrementa el número de implicantes (o grupos de corte). El trabajo de Lee (1985) contiene una exhaustiva recopilación de estas aportaciones. Los algoritmos se diferencian unos de otros por la forma de almacenar y representar los candidatos a PIs y por como desencadenan el proceso de recorrer el árbol y de simplificar-reducir los candidatos para obtener el conjunto final de PIs.

Otros algoritmos proceden de forma muy diferente y potente, tal es el caso de la Transformación de Ecuaciones empleada por el código SETS (Worrell, 1974), en la cual se obtienen los PIs desarrollando un algoritmo de transformación de las ecuaciones originales mediante un proceso directo de sustitución en las mismas.

Algoritmos para obtener PIs o GCM

En la figura 2.15 se presentan algunas aportaciones en la determinación de los Primero Implicantes o los Grupos de Corte Mínimos, en ella se pueden apreciar diversas características de interés en especial las relativas a la forma de recorrer árbol (ascendente o

descendente), forma de almacenar los candidatos y procedimiento de simplificación-reducción para obtener los mínimos. Se han realizado varias revisiones exhaustivas del estado del arte en la materia (Lee, 1982), (Lee et al., 1985), (Misra, 1992). Es importante destacar la estrategia empleada para almacenar los candidatos a PIs y la de reducción-simplificación para obtenerlos. Destacan en este sentido la representación de los grupos de corte mediante Números Primos (Semanderes, 1971), el uso de bit más el test OR-EOR desarrollado por Wheeler et al. (1977), el uso de Matrices de Adyacencia Dispersas y su descomposición mediante el teorema de Shannon (Heger et al, 1995) y la representación multinivel del árbol más simplificación-reducción por cada nivel (Hennings & Kuznetsov, 1995). A continuación se describirán más en detalle estas referencias así como el algoritmo de Kumamoto y Henley (1978) que ha sido uno de los más referenciados por su eficiencia y aplicabilidad.

Semanderes (1971) presentó el programa ELRAFT para el estudio de árboles de fallos mediante un algoritmo ascendente (Abajo-Arriba). La principal aportación consistió en el empleo de números primos para representar a los sucesos básicos, lo que simplifica notablemente el almacenamiento de los grupos de corte y la obtención de los Grupos de Corte Mínimos (GCM). Se comienza por asignar a cada suceso básico o variable Booleana un número primo diferente por orden creciente desde el menor primo. Entonces, los Grupos de corte se obtienen multiplicando los números primos que representan a los miembros del grupo de variables Booleanas que lo conforman. Con posterioridad se podrá averiguar que variables básicas componen un determinado grupo de corte sin más que factorizar en producto de primos el número que lo representa (El Teorema Fundamental de la Aritmética establece la existencia y unicidad de factorización de cualquier número natural en producto de primos).

La simplificación se realiza descomponiendo los grupos de corte en sus elementos y suprimiendo los repetidos identificados en la factorización como potencias de primos con exponente superior a la unidad., seguido de la eliminación de aquellos que sean divisibles por otros ; pues si el número que representa al grupo de corte es múltiplo del número representativo del grupo de corte, obviamente será grupo de corte mínimo. El inconveniente más importante es que en caso de grupos de corte de orden alto (con gran números de sucesos) ó, simplemente, conteniendo una variable correspondiente a un número primo alto, el representante del mismo puede ser un número prohibitivo. En la figura 2.16 se puede apreciar un ejemplo sencillo extraído de Lee (1982).

AUTOR(ES) AÑO	A/D	C/NC	FORMA DE ALMACENAR CANDIDATOS	MÉTODO	COMENTARIOS	NOMBRE DEL PROGRAMA
Vesely y Narum, 1970		C			Algoritmo de Test de Combinaciones COMBO	PREP-KITT
Semanderes, 1971	A	C	Números Primos	Factorización- División		ELRAFT
Fussell y Vesely, 1972	D	C				MOCUS
Worrell, 1974		NC			Transformación de Ecuaciones.	SETS
Pande et AL, 1975	A	C				MICSUP
Wheeler et al., 1977	D	C	Bits	Suma lógica y test OR-EOR		FAULTRAN
Kumamoto y Henley, 1978	D	NC				
Rasmuson y Marshall, 1978		C			Truncamiento a partir de cierto orden	FATRAM
Nakashima y Hattory, 1979	A	NC				BUP-CUTS
Bojadhiev, 1984	D	NC	Byte-word	Rutinas Assembler	Descripción dual del árbol, algoritmo de K. y H. mejorado	FTANS
Limnios y Ziani, 1986	D	C		Reducción de los repetidos	Mejoras a MOCUS, partición de los grupos de corte en dos familias: Repetidos y No Repetidos	
Hennings y Kuznetsov, 1995	A	NC		En cada nivel del árbol	Ascendente en cada subárbol encontrado. Usa Modularización	FAMOCUTN
Heger et al., 1995		C			Truncamiento a partir de un cierto orden	Sigma-Pi PATREC

Fig. 2.15.- REFERENCIAS SIGNIFICATIVAS Y CARACTERÍSTICAS DE ALGORITMOS PARA DETERMINAR LOS PIs (o GCM)

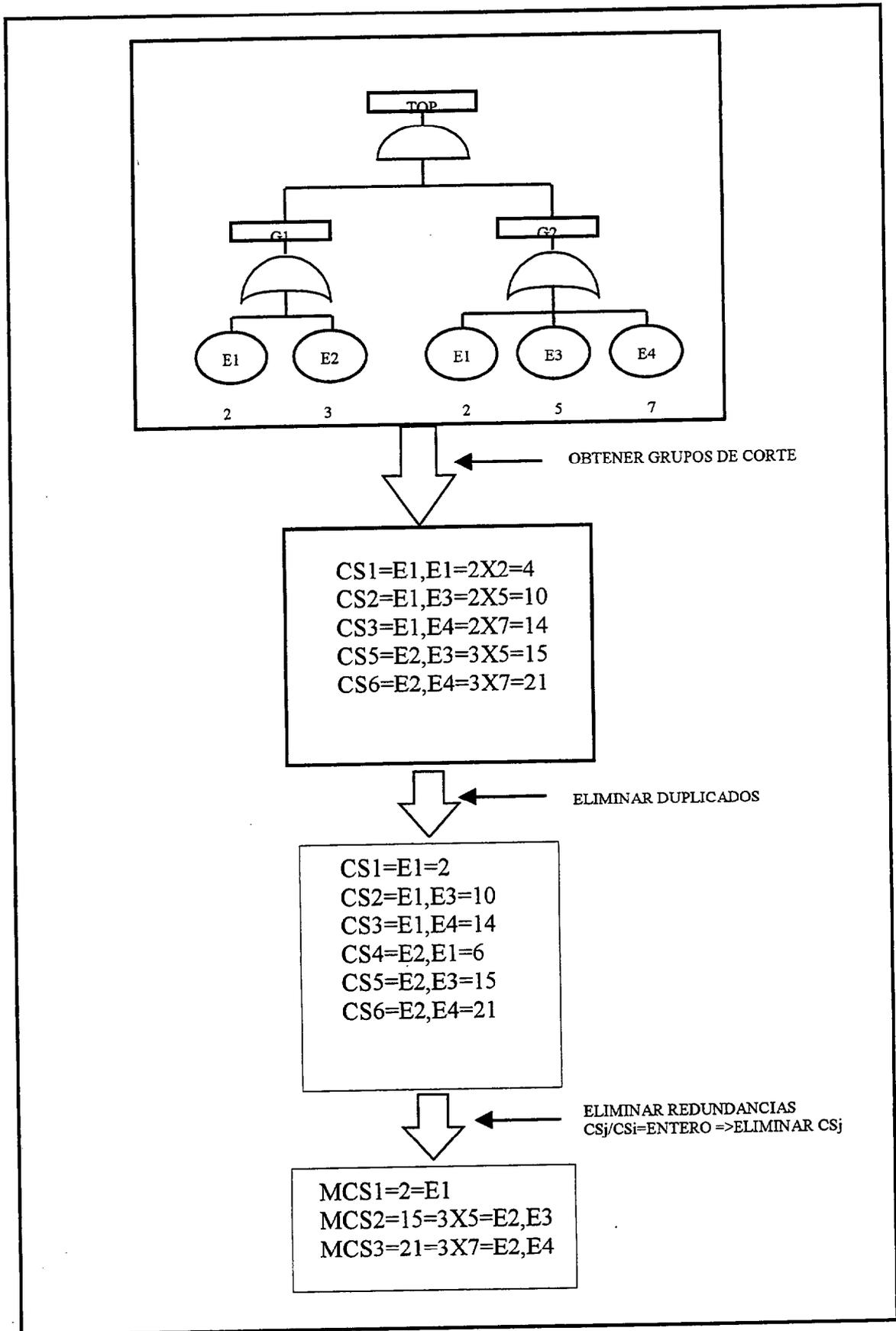


Fig. 2.16.- CÁLCULO DE GRUPOS DE CORTE MÍNIMOS USANDO NÚMEROS PRIMOS.

Kumamoto y Henley (1978) presentaron un método muy eficiente para determinar los PIs mediante un proceso de tipo Arriba-Abajo (Top-Down).

Si ψ es la función de estructura y ψ^* una expresión en suma de productos del dual del árbol de fallos, el algoritmo usa el hecho de que la condición necesaria y suficiente para que un grupo de términos sea un grupo de implicantes es que todos los productos en ψ^* son eliminados cuando los elementos en un grupo toman el valor cero. El conjunto de las clases de los conjuntos de implicantes se dividen en subclases más pequeñas de forma progresiva y cada conjunto de primeros implicantes se obtiene como un grupo de implicantes mínimo en una subclase.

El algoritmo consta de tres partes:

PARTE 1

Obtener la expresión local y el dominio para cada puerta G_i en el árbol de fallos dual.

PARTE 2

Obtener el operador $L_k^{i_k} \dots L_1^{i_1}$ que satisface la identidad

$$L_k^{i_k} \dots L_1^{i_1} \psi^* \equiv 0 \quad (1)$$

o

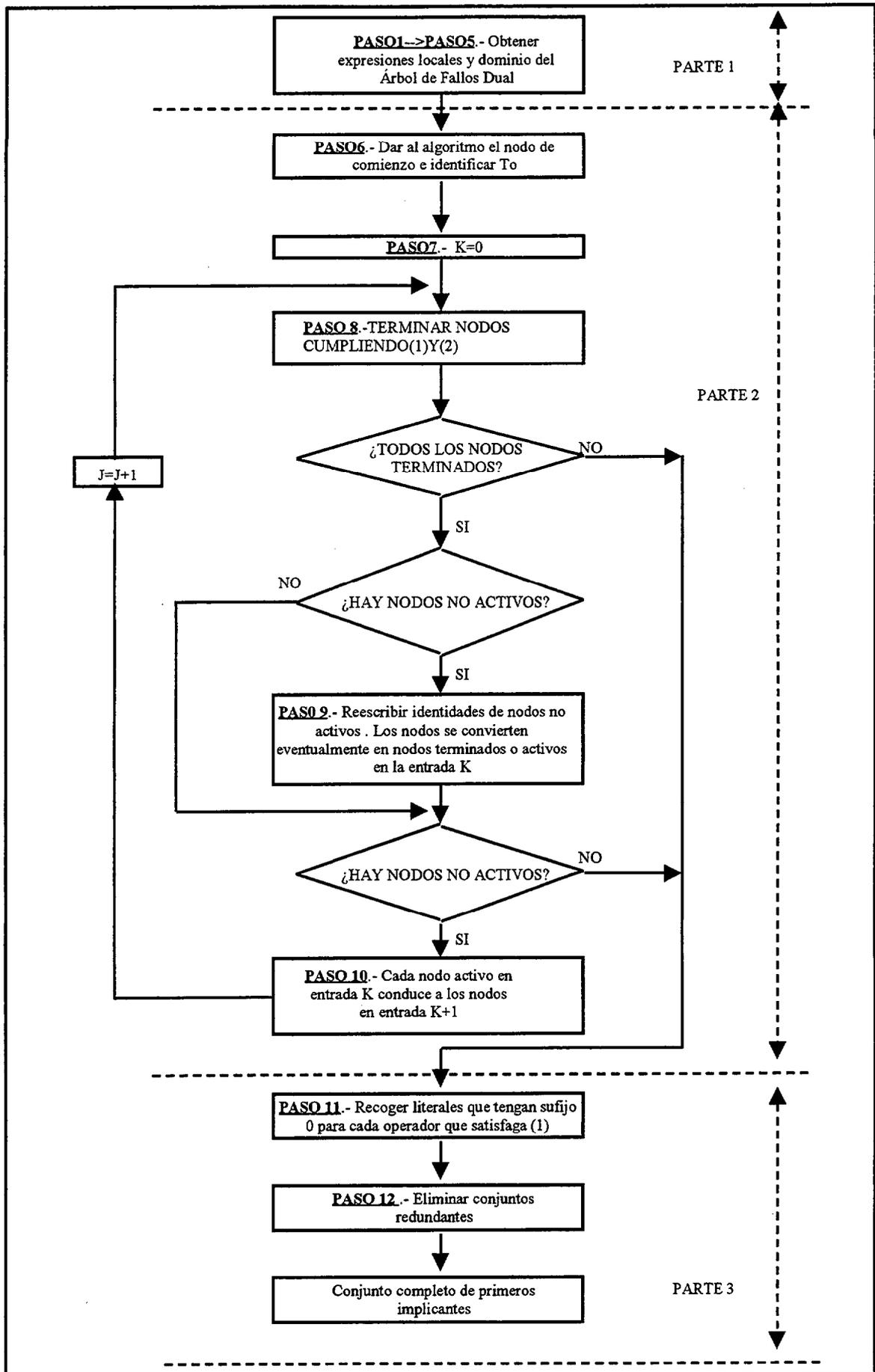
$$L_k^{i_k} \dots L_1^{i_1} \psi^* \equiv 1 \quad (2)$$

El operador (1) corresponde a las subclases que tienen cero ψ^*

El operador (2) corresponde a las subclases que tienen uno ψ^*

PARTE 3

Escoger los literales L_j que tengan por superíndice $i_j = 0$ para cada operador que satisfaga (1). Cada grupo de literales así obtenidos es candidato a primer implicante. La eliminación de los conjuntos redundantes conduce a los primeros implicantes buscados.



© Universidad de Las Palmas de Gran Canaria. Biblioteca Digital. 2003

Fig. 2.17.- ALGORITMO PARA OBTENER LOS PRIMEROS IMPLICANTES
(Kumamoto y Henley, 1978)

Wheeler et al (1977) presentaron un algoritmo de tipo Abajo-Arriba clásico en este sentido pero que incorporaba una importante reducción en los requerimientos de memoria necesaria para procesar la información intermedia en el proceso de determinación de los GCM, en base al empleo de bits para representar los estados de los sucesos básicos. De esta forma en lugar de emplear "palabras" de 8,16 ó 32 bits para representar un solo suceso básico se pueden representar tantos casos por palabra como tamaño tenga ésta (en función del ordenador disponible). Suponiendo una función de 28 variables, representadas por las letras del abecedario, las representaciones de algunos vectores de estado serán:

Vector de Estado Bits de Representación

A C	-----	1 0 1 0 0 0 0.....
B	-----	0 1 0 0 0 0 0.....
A C E	-----	1 0 1 0 1 0 0.....

El algoritmo de asignación de bits es simple y fácil de implementar:

NARRAY = 0	NARRAY=NUMERO DE VECTORES
DO 2 N=1,NEVENT	BIT=VALOR BINARIO (0-1)
IF (MOD(N-1,35).NE.0) GOTO 1	NEVENT=N. DE SUCEOS BASICOS
BIT = 1	MOD(a,b)=MODULO(a,b)
NARRAY = NARRAY + 1	EVENT=MATRIZ DE SUCEOS
1 EVENT(NARRAY,N) = BIT	CODIFICADOS EN BINARIO
BIT=BIT+BIT	
2 CONTINUE	

La obtención del número total de grupos de corte se convierte también en muy simple y eficiente, ya que la salida de una puerta AND es la suma lógica (bit a bit) de todas la entradas (lo cual se puede lograr con la función OR intrínseca del sistema), mientras que la salida de una puerta OR es la lista de sus entradas (eliminando las duplicadas), (lo cual elimina de paso las redundancias del tipo (XAA=A).

Una vez que se han obtenido los grupos de corte se procede a eliminar las redundancias de los tipos: $A+AB=A$, $AB+BA=AB$ mediante el empleo de las funciones intrínseca OR y Exclusiva OR (EOR).

El proceso para comparar un grupo de corte de referencia con otro cualquiera (de test) es:

- a) Calcular el resultado de la operación OR (referencia, Test) = Result.
- b) Calcular la operación EOR (Result, test)
- c) Si el resultado es cero el grupo de test es redundante y deberá ser eliminado.

COMPARACION

REFERENCIA = 1 0 0 0

TEST = 1 1 0 0

$(1\ 0\ 0\ 0) \text{ OR } (1\ 1\ 0\ 0) = (1\ 1\ 0\ 0)$

$(1\ 1\ 0\ 0) \text{ EOR } (1\ 1\ 0\ 0) = (0\ 0\ 0\ 0) \implies 1\ 1\ 0\ 0 \text{ será eliminado}$

Este proceso aplicado a la comparación de “todos con todos” conduce a la determinación de los GCM. Wheeler et al aplicaron este método en su código FAULTRAN con buenos resultados. Esta técnica se sigue empleando en algunos códigos actuales (Geymayr y Ebecken, 1995).

Hennings y Kuznetsov (1995) han presentado un método basado en representar el árbol mediante niveles y encontrar y simplificar los grupos de corte en cada nivel. La secuencia lógica es Transformar, Modularizar y Reducir el árbol antes de encontrar y simplificar los grupos de corte. Los niveles del árbol de fallos se establecen en función de las puertas repetidas siendo el nivel 1 el árbol original. Para establecer el nivel 2 se buscan las puertas repetidas del nivel 1 y se marcan las puertas que están entre éstas y el suceso top. Sólo las puertas marcadas formarán parte del nivel 2 de representación. Este proceso se repite para establecer los siguientes niveles hasta que se llega a un nivel cuyo árbol de fallos contiene sólo puertas no repetidas. El árbol de fallos con puertas no repetidas es desarrollado con un algoritmo ascendente clásico, pero la especial construcción de los árboles de cada nivel produce muchos menos grupos de corte que en el caso de afrontar el árbol original. Este método es muy potente para evaluar grandes árboles de fallos tanto cualitativa como cuantitativamente, a partir de los grupos de corte mínimos encontrados. Para calcular los grupos de corte mínimos se procede a calcular en primer lugar los del último nivel encontrado; a continuación se retrocede un nivel desagrupando los elementos correspondientes y volviendo a reducir a mínimos. El proceso finalizará cuando alcancemos el nivel 1 (árbol de fallos original). El resultado será el conjunto de grupos de corte mínimos, obtenidos con mucho menos esfuerzo computacional. En la figura 2.18 se puede apreciar la secuencia de obtención de árboles de fallos para diferentes niveles aplicando este método.

Heger et al (1995) presentaron un método de obtener los PIs basado en la expansión de Shannon del árbol de fallos visto éste como una matriz dispersa de grupos de corte. Como entrada, su algoritmo (y programa) llamado $\sum \prod -PATREC$, recibe los grupos de corte calculados mediante otro programa, los cuales se representan en forma matricial dispersa. A continuación se procede a la factorización de dicha matriz según determinados sucesos básicos y posteriormente a su reducción a mínimos aplicando idempotencia y absorción. El principal inconveniente es la gran sensibilidad al orden en que se escogen los implicantes (o grupos de corte) y al pivote elegido para descomponer el problema, lo que puede conducir a un problema de complejidad similar a la del original. Para evitarlo se recurre a la técnica de "memoización" similar al reconocimiento de patrones de Koen y Carnino usando Programación Dinámica. Se utilizan técnicas de reconocimiento de patrones para evitar cálculos repetitivos (fig. 2.19).

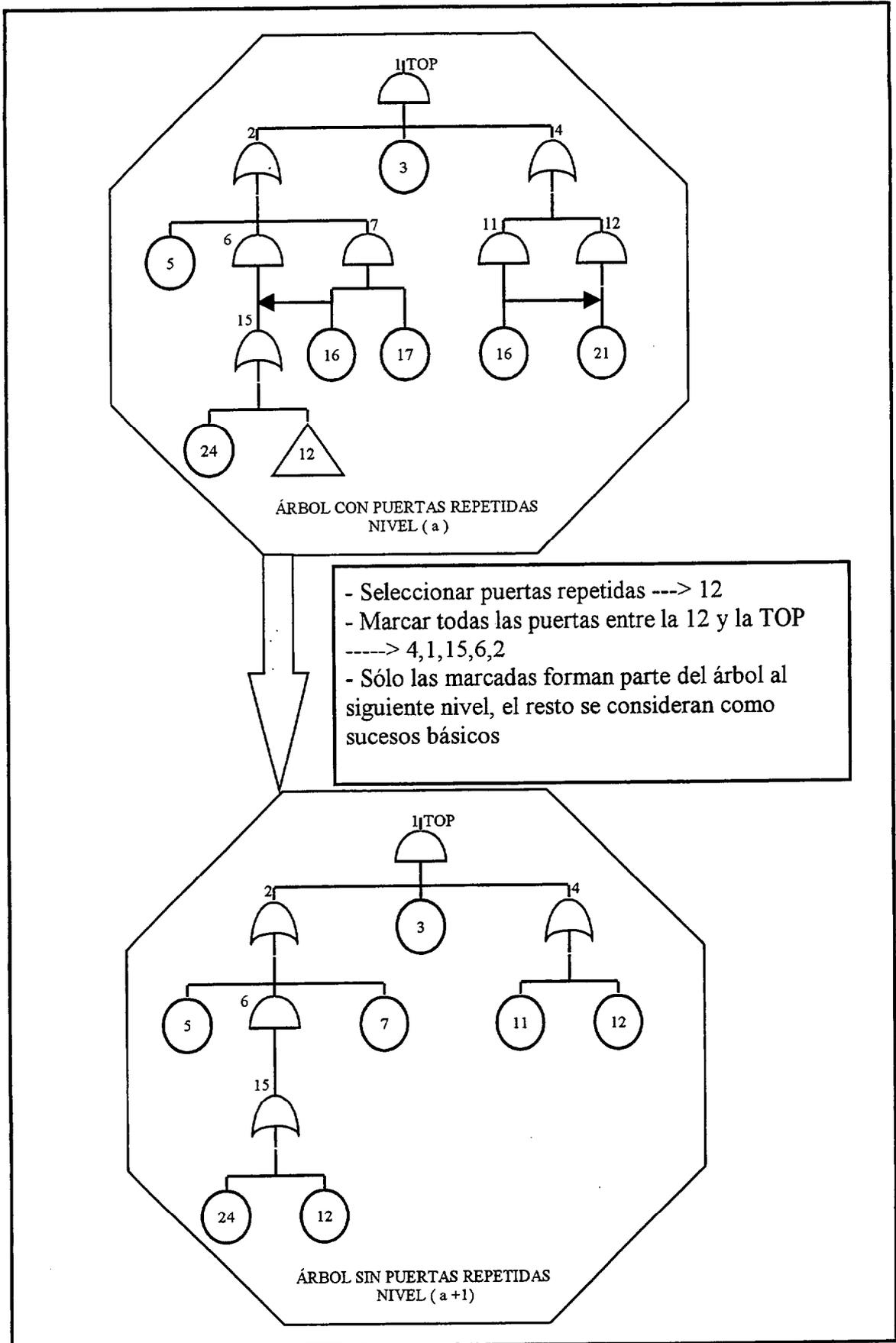
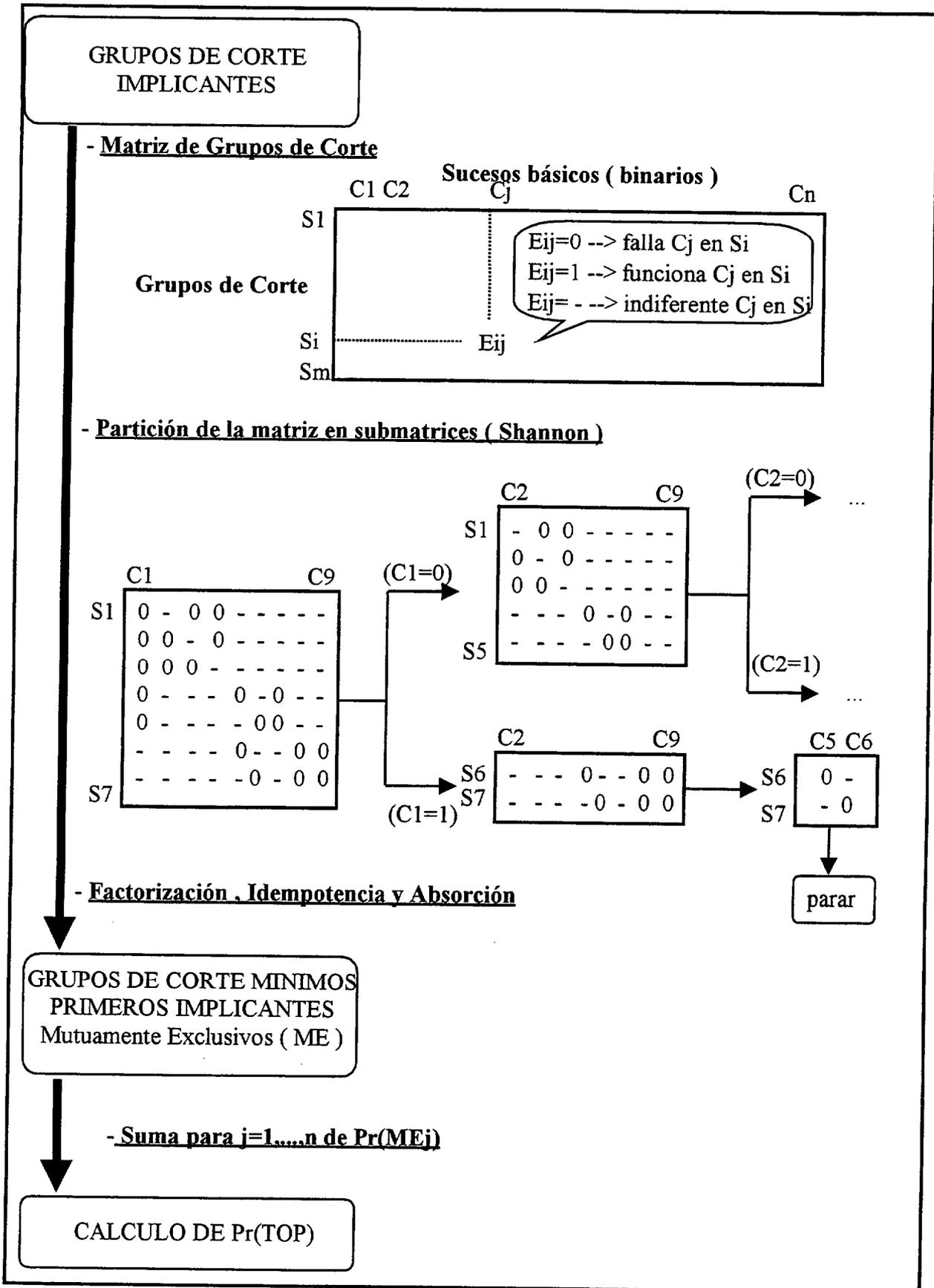


Fig. 2.18.- OBTENCIÓN DE ÁRBOLES DE FALLOS A DISTINTOS NIVELES DE REPRESENTACIÓN (Hennings y Kuznetsov, 1995)



© Universidad de Las Palmas de Gran Canaria. Biblioteca Digital. 2003

Fig. 2.19.- OBTENCIÓN DE PIs MEDIANTE MATRICES DE GRUPOS DE CORTE (Heger et al, 1995)

Módulos

La obtención de módulos comenzó como una alternativa para el análisis de árboles cuya lógica conducía a un número de grupos de corte intratable por un ordenador (Chatterjee, 1975) y se ha convertido, con el tiempo, en un recurso permanente para la simplificación de árboles de gran tamaño ó complejidad (Gulati, 1996). La idea central del empleo de módulos es la de simplificar la representación del sistema identificando subsistemas que intervienen sólo en unos pocos puntos muy localizados del proceso. Los subárboles correspondientes a esos subsistemas se denominarán Módulos, cuya definición desde el punto de vista del árbol de fallos (Rosenthal, 1980) es: Módulo es un conjunto de al menos dos sucesos que tienen sólo una salida al resto del árbol y ninguna entrada desde el resto del árbol".

El suceso de salida de un módulo se denominará "**top del Módulo**" y los módulos del árbol de fallos lo son de la función de estructura representada por éste. El recíproco no tiene porqué ser cierto. Una vez que un módulo ha sido identificado se le puede estudiar independientemente (calcular sus PIs o la probabilidad de su suceso top) y el subárbol correspondiente se sustituye por un suceso simple. Si el número de módulos encontrado es significativo (o su tamaño adecuado), el estudio de un árbol grande puede convertirse en muy accesible para un ordenador de formato convencional. El grupo de PIs que deberá encontrarse se reduce considerablemente ya que solo se deberán obtener los PIs del árbol principal después de sustituir los módulos por sus sucesos simples representantes, más los de cada módulo. Con todo ello la información cualitativa se vuelve además mucho más clara, sintetizada y representativa del sistema real. La clave del análisis modular está en la eficiencia del algoritmo que busca los sucesos del árbol que son "top de Módulos". Muchos esfuerzos se han realizado en este sentido, con excelentes resultados (Olmos y Wolf, 1977), (Kodha et al, 1988), (Patterson-Hine y Dugan, 1992), (Gulati y Dugan, 1997), (Manjan et al., 1998).

2.6.4.-Métodos indirectos. Diagramas de Decisión Binarios

Introducción

Si en lugar de abordar directamente el estudio de la función de estructura del suceso top del árbol de fallos, se procede a través del uso de un Diagrama de Decisión Binario (DDB), estamos empleando un Método Indirecto para estudiarlo. Los DDBs han sido introducidos recientemente en el análisis de árboles de fallos (Bryant, 1986), (Rauzy, 1993) y constituyen una poderosa herramienta que ha permitido estudiar de forma determinista árboles inaccesibles hasta ahora (Bouissou, 1996). El análisis que permiten realizar es tanto de tipo cualitativo como cuantitativo. El aspecto clave de este método es el proceso de obtención del DDB y más concretamente el orden de las variables que hay que adoptar para obtener un DDB minimizado que conduzca directamente a los PIs o GCM (Sinnamon y Andrews, 1997), (Rauzi y Dutuit, 1997). Otro aspecto importante es comprobar el alcance del método, o sea, comprobar el tamaño de los árboles de fallos que pueden ser tratados (Coudert y Madre, 1994 y 1994), (Aralia Group, 1995).

Descripción y Obtención de DDBs

Un Diagrama de Decisión Binario es un grafo obtenido a partir del árbol de fallos para un determinado orden de las variables básicas, codificado según una estructura **If-Then-Else** (ITE) y derivado de la expansión de Shannon de la función de estructura del suceso top. Constituye una representación canónica de la función de estructura y es aplicable tanto para árboles coherentes como no coherentes.

Para determinar los Grupos de Corte sólo hay que seguir los caminos que conducen desde el vértice raíz hasta un uno, e incluyendo para cada camino solamente las variables que van acompañadas de un uno (indicador de que el fallo ha ocurrido) se finalizará sometiendo al conjunto de caminos a las leyes de Idempotencia y Absorción, siendo los que queden los Primeros Implicantes. Si el orden inicial elegido para las variables es el adecuado, entonces el DDB será mínimo y el resultado de los caminos que conducen hasta los unos será directamente el conjunto de PIs, no siendo necesaria la aplicación de las leyes de simplificación mencionadas, lo que significa un importantísimo ahorro de coste

computacional y justifica la importancia de realizar una adecuada selección del orden de las variables.

Sea $\phi(X_1, X_2, X_3, \dots, X_n)$ la función de estructura del suceso top ϕ_{top}

Expandiendo según Shannon por la variable X_1 quedará :

$$\phi(X_1, X_2, X_3, \dots, X_n) = X_1\phi(1, X_2, X_3, \dots, X_n) + \bar{X}_1\phi(0, X_2, X_3, \dots, X_n)$$

simplicadamente podemos reescribir la ecuación anterior como

$$\phi_{top} = X_1\phi_1 + \bar{X}_1\phi_2$$

Lo que se puede interpretar como una condición If -Then-Else como sigue

$$if - (X_1) - then - (\phi_1) - else - (\phi_2) = ite(X_1, \phi_1, \phi_2)$$

Fig. 2.20.- ESTRUCTURA IF-THEN-ELSE SOBRE LA EXPANSIÓN DE SHANNON DE ϕ_{TOP}

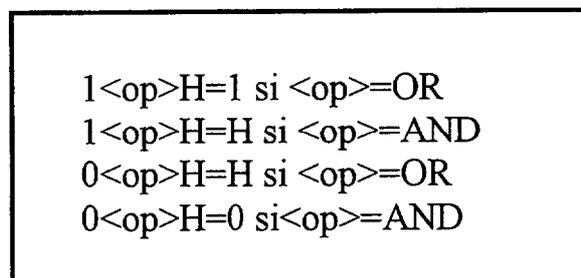
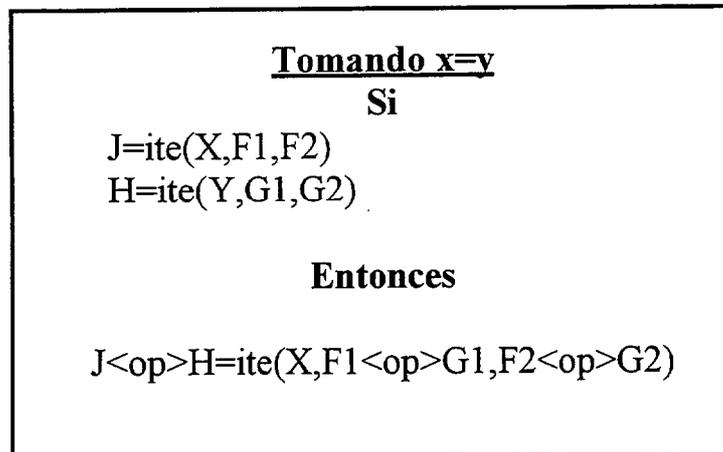
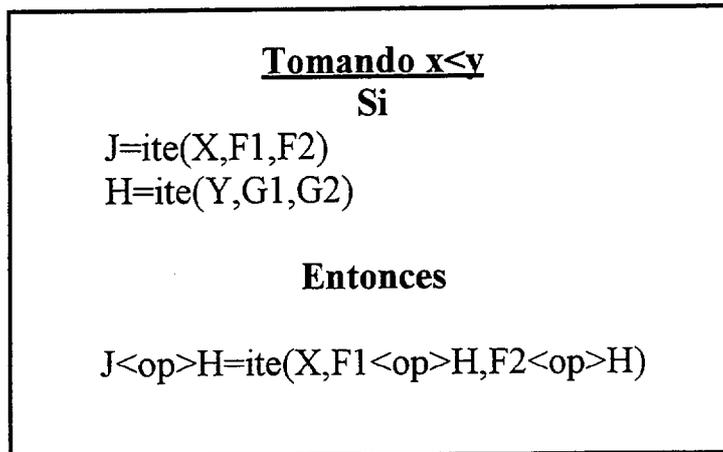
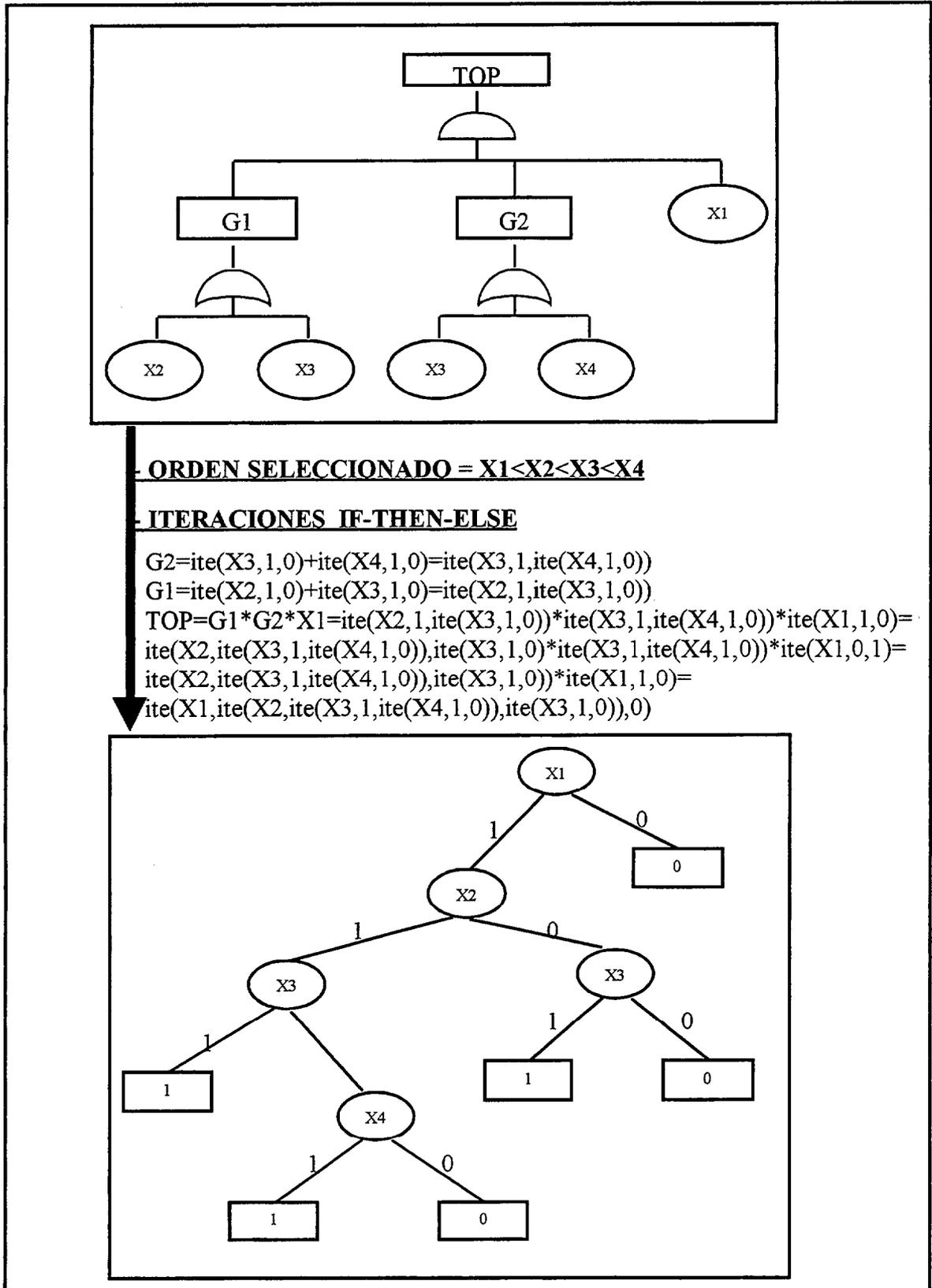
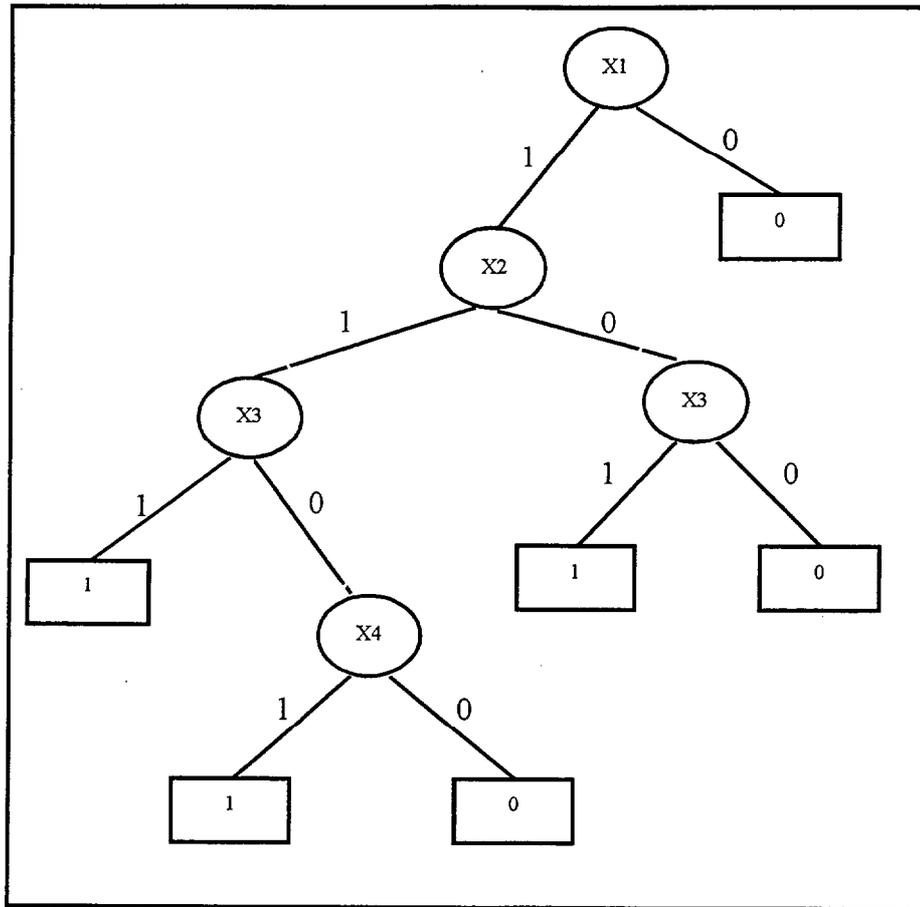


Fig. 2.21.- OPERADORES IF-THEN-ELSE EN PUERTAS LÓGICAS
 (Sinnamon y Andrews, 1996)



© Universidad de Las Palmas de Gran Canaria. Biblioteca Digital. 2003

Fig. 2.22.- OBTENCIÓN DE UN DIAGRAMA DE DECISIÓN BINARIO MEDIANTE ITERACIONES IF-THEN-ELSE(ite)(Sinnamon y Andrews, 1996)



Caminos hacia el uno

GRUPOS DE CORTE

- a) ---> $X1 * X3$
- b) ---> $X1 * X2 * X4$
- c) ---> $X1 * X2 * X3$

Idempotencia y Absorción

GRUPOS DE CORTE MÍNIMOS

$$GCM1 = X1 * X3$$

$$GCM1 = X1 * X2 * X4$$

Fig. 2.23.- OBTENCIÓN DE LOS GRUPOS DE CORTE MÍNIMOS VÍA DIAGRAMA DE DECISIÓN BINARIO

Selección del Orden de las Variables

El problema de la selección del orden adecuado de las variables para obtener el DDB mínimo es intratable para problemas reales y está siendo objeto de gran atención actualmente, dado que es el único cuello de botella aparente del método (Bouissou, 1996), (Andrews y Bartlet, 1998). Para abordarlo se han propuesto diferentes métodos, la mayoría basados en la intuición o la experiencia:

- 1.- Explorar el árbol en profundidad y hacia la izquierda primero (PI), colocando las variables en el orden en que son encontradas.
- 2.- Explorar el árbol a lo ancho y a la izquierda primero (AI), colocando las variables en el orden en que son encontradas.
- 3.- Colocar las variables en orden creciente de nivel (NIV), siendo el nivel de una variable el número de puertas que hay hasta el suceso top del árbol.
- 4.- Asignar peso 1 a cada rama del árbol. Calcular el peso de cada puerta sumando los pesos de sus entradas. Explorar el árbol en profundidad, una vez pesadas todas las puertas, seleccionando las entradas de las puertas según los pesos crecientes de éstas en cada nivel. Las variables se colocan según el orden en que son encontradas.
- 5.- Uso de Módulos y Sub-Módulos
- 6.- Ordenar según el número de repeticiones de las variables, escogiéndolas en orden decreciente según el número de repeticiones.

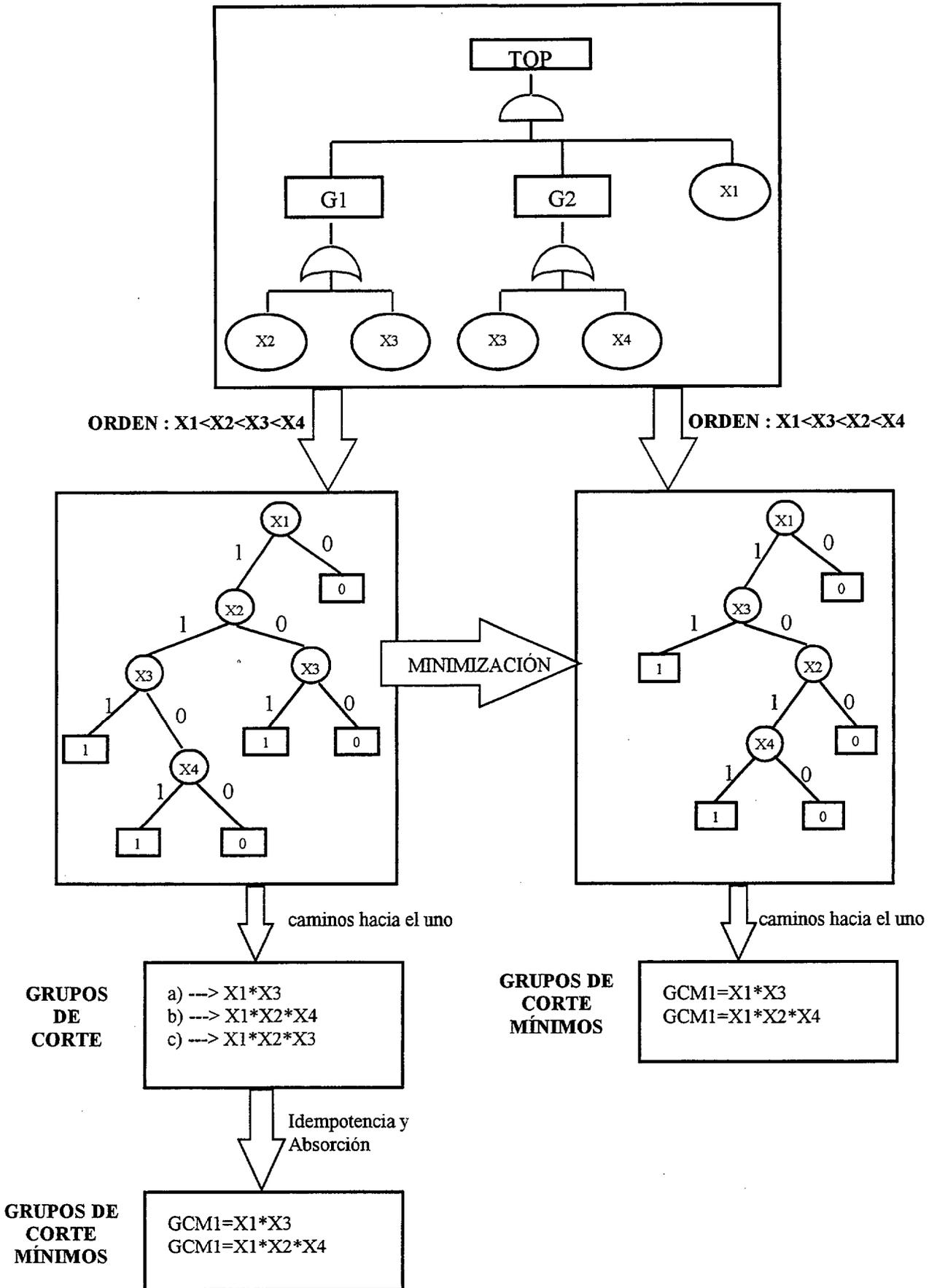


FIG. 2.24.- INFLUENCIA DEL ORDEN DE LAS VARIABLES EN EL DIAGRAMA DE DECISIÓN BINARIO

Tamaño de los DDBs y alcances del método

El tamaño mínimo del DDB es el número n de variables de que depende la función booleana del árbol. El tamaño máximo del Diagrama de Decisión Binario, medido por el número de sus nodos no terminales, no puede exceder de $2^n - 1$ que es el del correspondiente árbol de Shannon. Lo que significa un intervalo de tamaños posibles de extraordinaria amplitud: $2^n - n - 1$. Los resultados obtenidos hasta el momento varían considerablemente de unos autores a otros, así Bryant (1992) para una función de $2n$ variables obtiene como mínimo $2n + 2$ nodos en el DDB, mientras que Ross (1990) obtiene $\frac{(n+1)^2}{3}$ o $\frac{n(n+2)}{3}$ para funciones simétricas de n variables. Con todo, constituyen reducciones importantes, pero existen dificultades todavía no solventadas para deducir matemáticamente los tamaños mínimos de los DDBs mediante un orden adecuado de las variables. En cuanto al tamaño de los árboles tratados mediante DDBs, Bouissou (1996) indica que ha podido tratar de forma eficiente árboles de fallos de hasta 1000 nodos (variables ó puertas lógicas).

2.6.5.- Causas Comunes de fallos

Introducción

Una Causa o Modo común de fallo es todo suceso que provoque una pérdida de múltiples elementos o defensas de un sistema que no pueda ser explicada por la simple coincidencia de fallos aleatorios independientes. (Améndola, 1988). (Wagner et al, 1977). Existen múltiples motivos para que aparezcan fallos debido a causas comunes, entre ellos cabe destacar:

- Estructuras dependientes no identificadas
- Aparición aleatoria de condiciones medioambientales anormales
- Deficiencias en el diseño o la construcción.
- Operación inapropiada del sistema.

En los gráficos siguientes se puede encontrar una clasificación de las Causas Comunes de Fallos, que se pueden usar con lista de chequeo a la hora de buscarlas en el sistema. Están clasificadas en dos grandes tipos:

- Originadas en la Ingeniería del Sistema, ya sea en la etapa de diseño o en la de construcción.
- Originadas en la Operación del Sistema, ya sean provenientes de los procedimientos involucrados en la explotación o de las condiciones medioambientales.

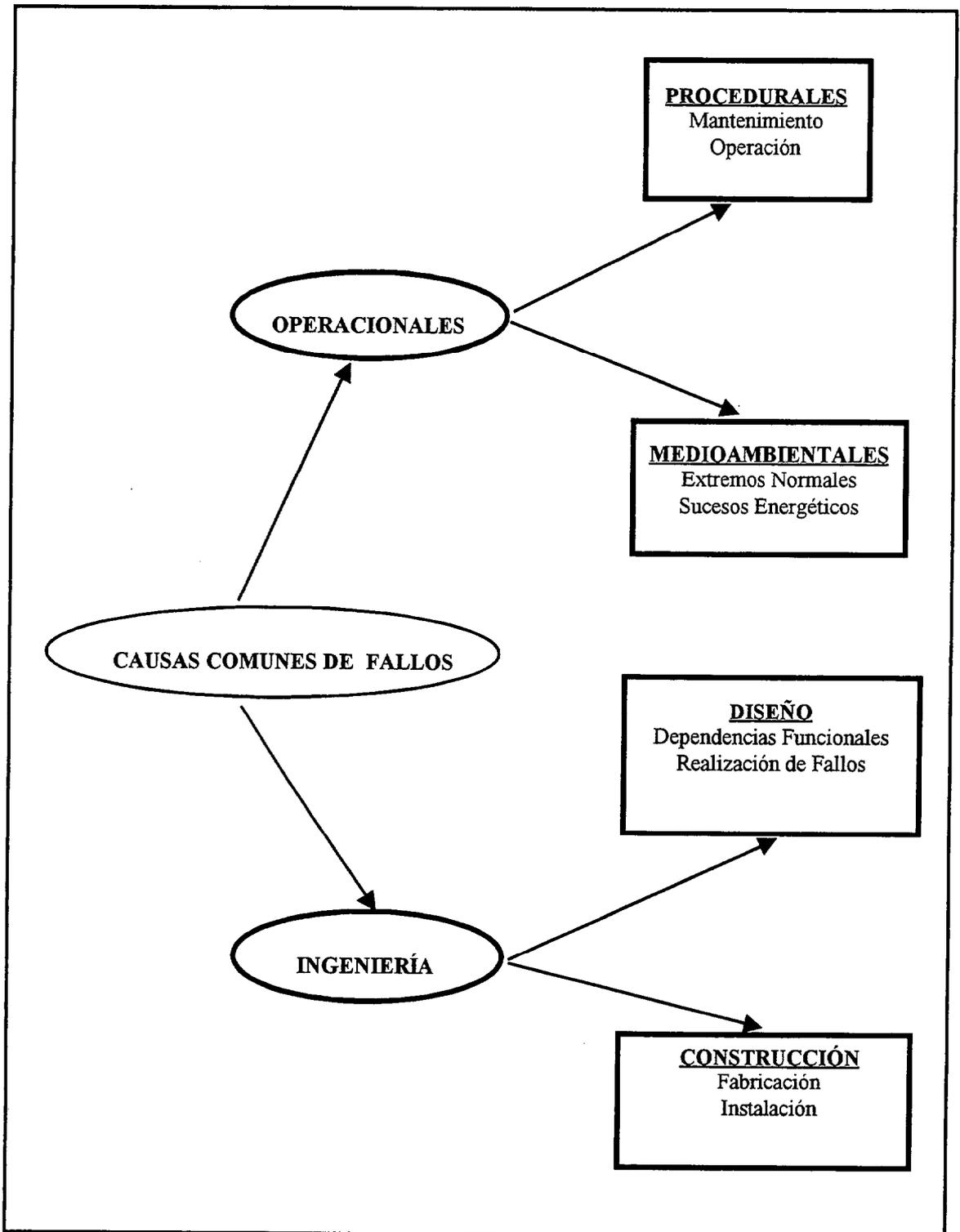


Fig. 2.25.- CLASIFICACIÓN DE LAS CAUSAS COMUNES DE FALLOS

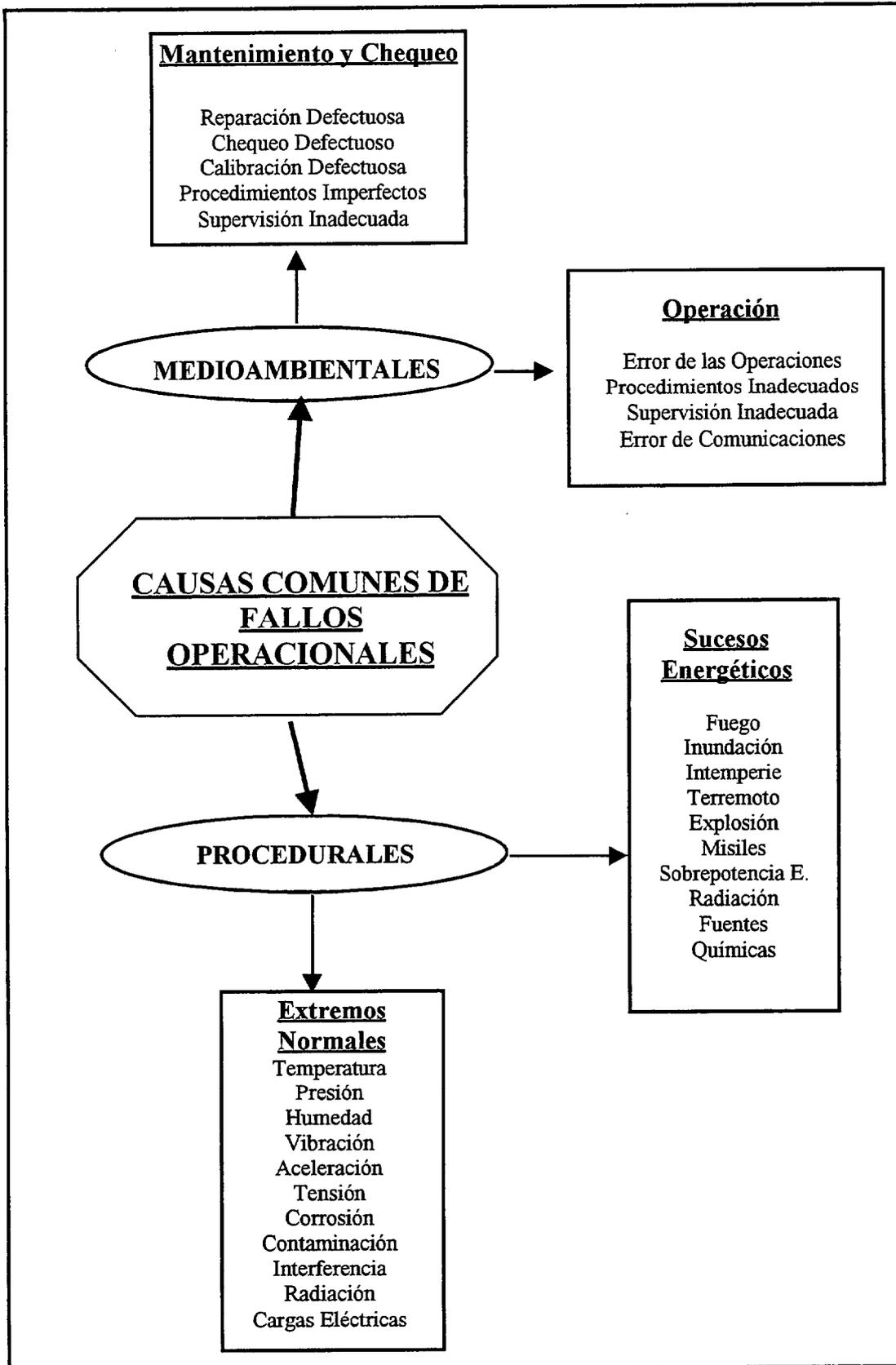


Fig. 2.26.- CAUSAS COMUNES DE FALLOS OPERACIONALES

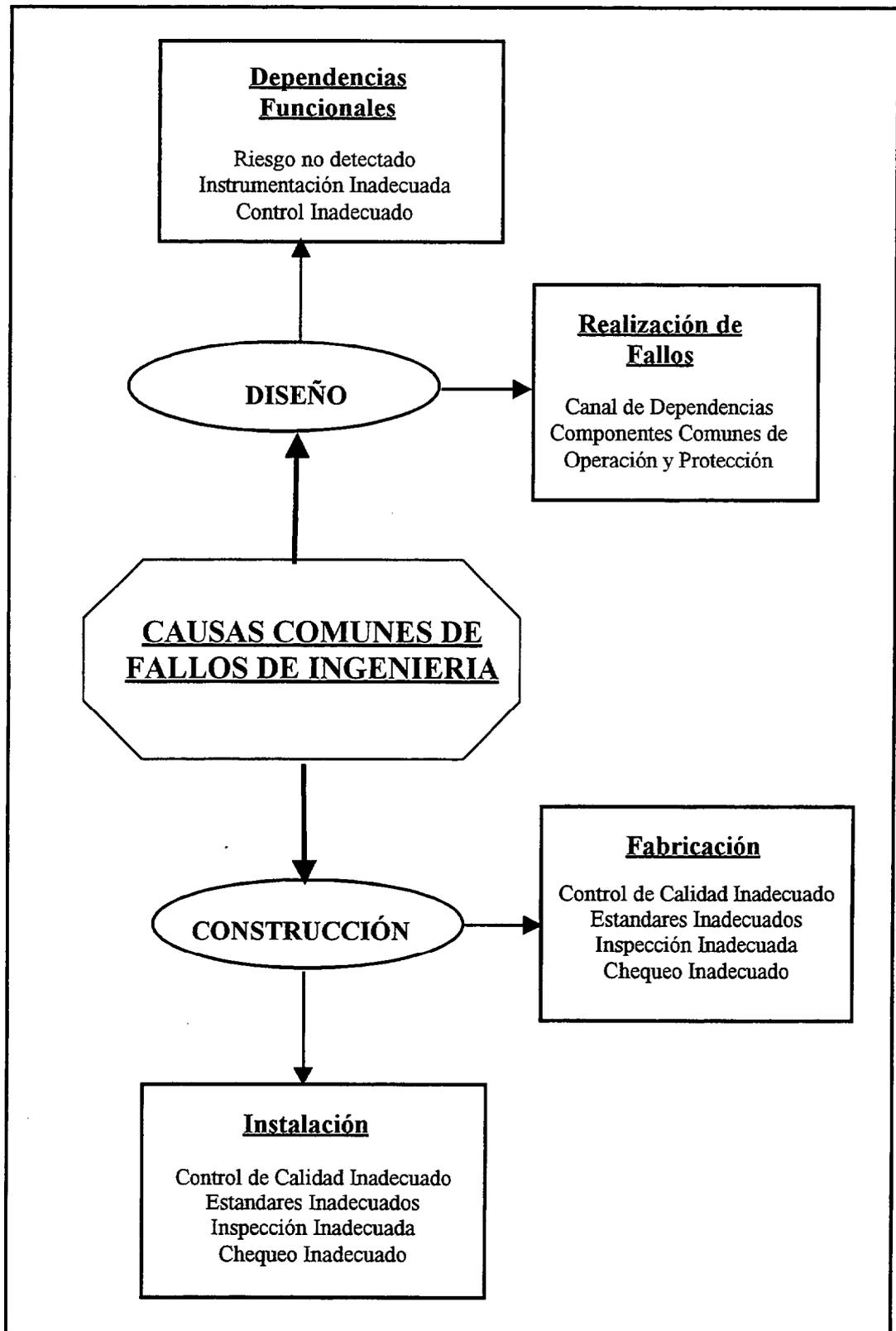


Fig. 2.27.- CAUSAS COMUNES DE FALLOS DE INGENIERÍA

Clases de estructuras dependientes

La dependencia entre sucesos diferentes puede deberse a razones funcionales (normalmente de naturaleza determinista) o a sucesos aleatorios (de naturaleza estocástica). Las principales clases de estructuras dependientes son:

- 1.- No Disponibilidades Funcionales con un mismo sistema. - Aquel caso en el que el fallo de un elemento impide a otros elementos, que no han fallado, realizar su trabajo.

- 2.- No Disponibilidades Funcionales entre diferentes sistemas. - Aquel caso en el que varios sistemas diferentes, localizados en diferentes lugares, fallan debido a que comparten algún subsistema. Ejemplo: la misma fuente de Energía Eléctrica.

- 3.- Cascada de Fallos. - La no disponibilidad de un componente (o sistema) puede provocar la no disponibilidad de otros componentes (o sistemas) relacionados con él, por ejemplo: el fallo de una protección eléctrica puede provocar el fallo de otros elementos por exceso de tensión.

- 4.- Restricciones Procedurales. - Pueden aparecer No Disponibilidades o fallos debidos a restricciones en los procesos causadas por procedimientos administrativos o por la realización de mantenimientos.

- 5.- Causas Externas. - Condiciones medioambientales externas pueden provocar el fallo de múltiples elementos o sistemas (Tormenta, Accidentes, etc.)

- 6.- Estructuras Dependientes unidas a Factores Humanos. El comportamiento humano puede potenciar notablemente dependencias funcionales o de fallos, especialmente en caso de tensión o de crisis operacional.

- 7.- Otros. - Profundizando más en este tipo de causas se pueden identificar otras, tales como: correlaciones estadísticas entre tasas de fallos y clases residuales de dependencias potenciales (Apostolakist y Moieni, 1986), (Bourne et al., 1981).

Para describir la ocurrencia de Causas Comunes de Fallos en los modelos de sistemas se emplean diversos modelos estocásticos paramétricos, entre los que cabe mencionar:

- Modelo Paramétrico Básico (Marshall y Olkin, 1967)
- Modelo Paramétrico Mixto (Kvam, 1998)
- Uso del Factor Beta (Fleming et al, 1986)
- El Método de Múltiples letras Griegas (Fleming y Kalinowski, 1983)
- Tasa de Fallos Binomial (Vesely, 1977)
- Método implícito (Vaurio, 1998)

Causas Comunes de Fallos y Análisis Cualitativo

Es muy importante completar y corregir los modelos de sistemas incorporando las Causas Comunes de Fallos en sistemas complejos, para que dichos modelos sean realmente representativos. Paso previo para ello es realizar un análisis cualitativo cuyos objetivos fundamentales han de ser:

- Entender los mecanismos y factores que determinan las dependencias entre elementos de sistemas.
- Identificar potenciales "no disponibilidades" o fallos dependientes.
- Identificar grupos de componentes que pudieran ser afectados por causas dependientes genéricas.
- Incrementar la efectividad de las defensas internas del sistema que previenen o limitan la ocurrencia de fallos dependientes.
- Clasificar y poner de manifiesto las posibles dependencias potenciales, de tal forma que pudieran ser incorporadas a los modelos del sistema de forma efectiva.

En lo que respecta a los árboles de fallos, las dependencias provocadas por relaciones funcionales o propagación de fallos se pueden incluir en el mismo de manera sencilla. Existen dos vías para ello (ver figuras siguientes):

- Modificar directamente el Arbol de Fallos.- Junto a las causas independientes se agregarán las causas dependientes, según la estrategia lógica que se muestra en la figura siguiente. (Fleming et al 1986). Este camino es el lógicamente correcto pero presenta el

inconveniente de que, en sistemas complejos, incrementa notablemente el tamaño de los árboles. Se suele denominar Proceso Explícito.

- Modificar los PIs ó GCM. - Se determinan primero los conjuntos de PIs del árbol de fallos compuesto solo por sucesos independientes y se añaden a continuación nuevos elementos que representan las causas comunes de fallos. Este método puede incurrir en graves omisiones de PIs representativos y debe ser aplicado con extremo cuidado. Se suele denominar Proceso Implícito.

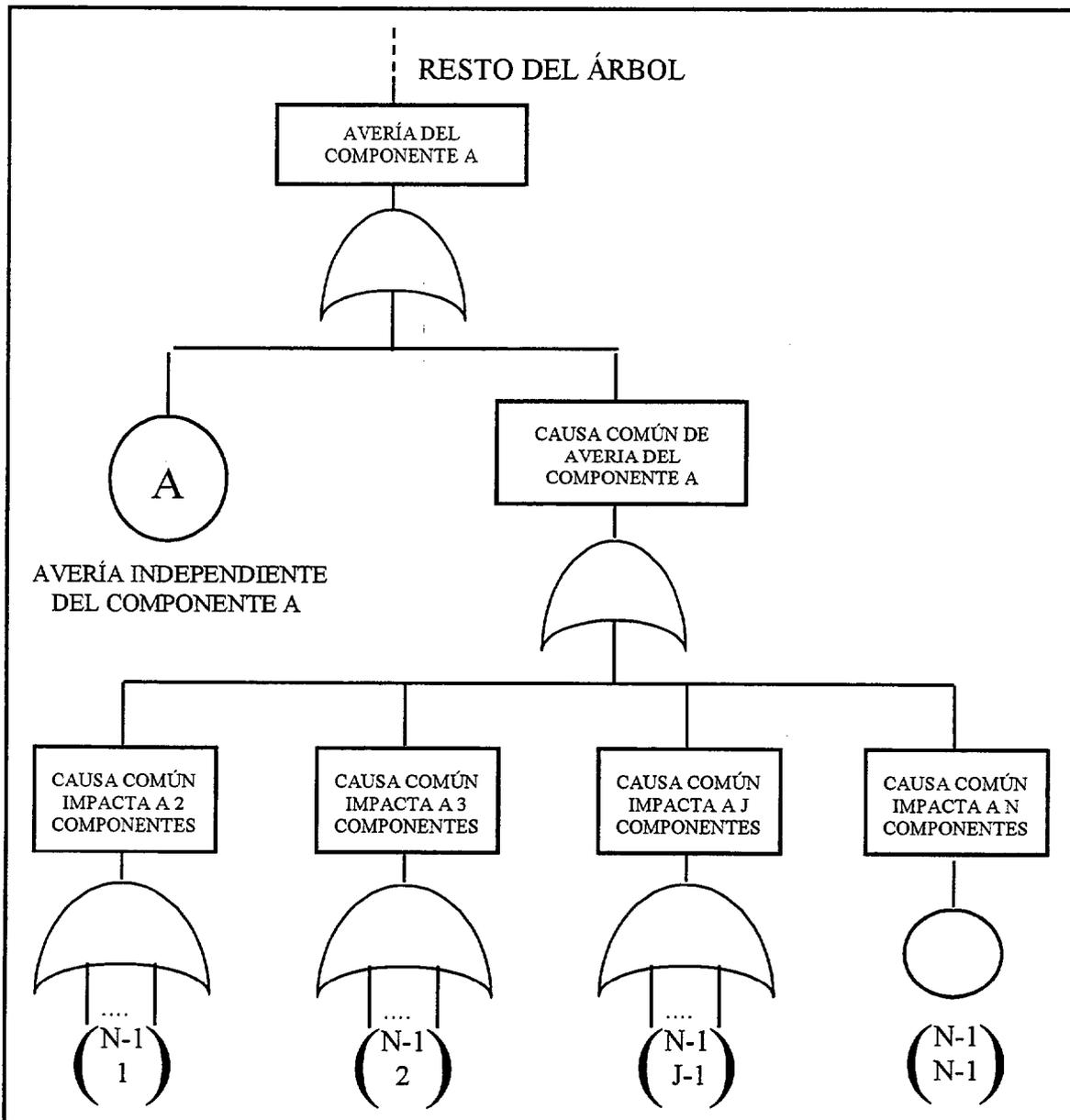


Fig. 2.28.- SUBÁRBOL DE FALLOS DE CAUSA COMÚN PARA UN COMPONENTE "A" EN UN GRUPO DE CAUSA COMÚN DE "N" COMPONENTES

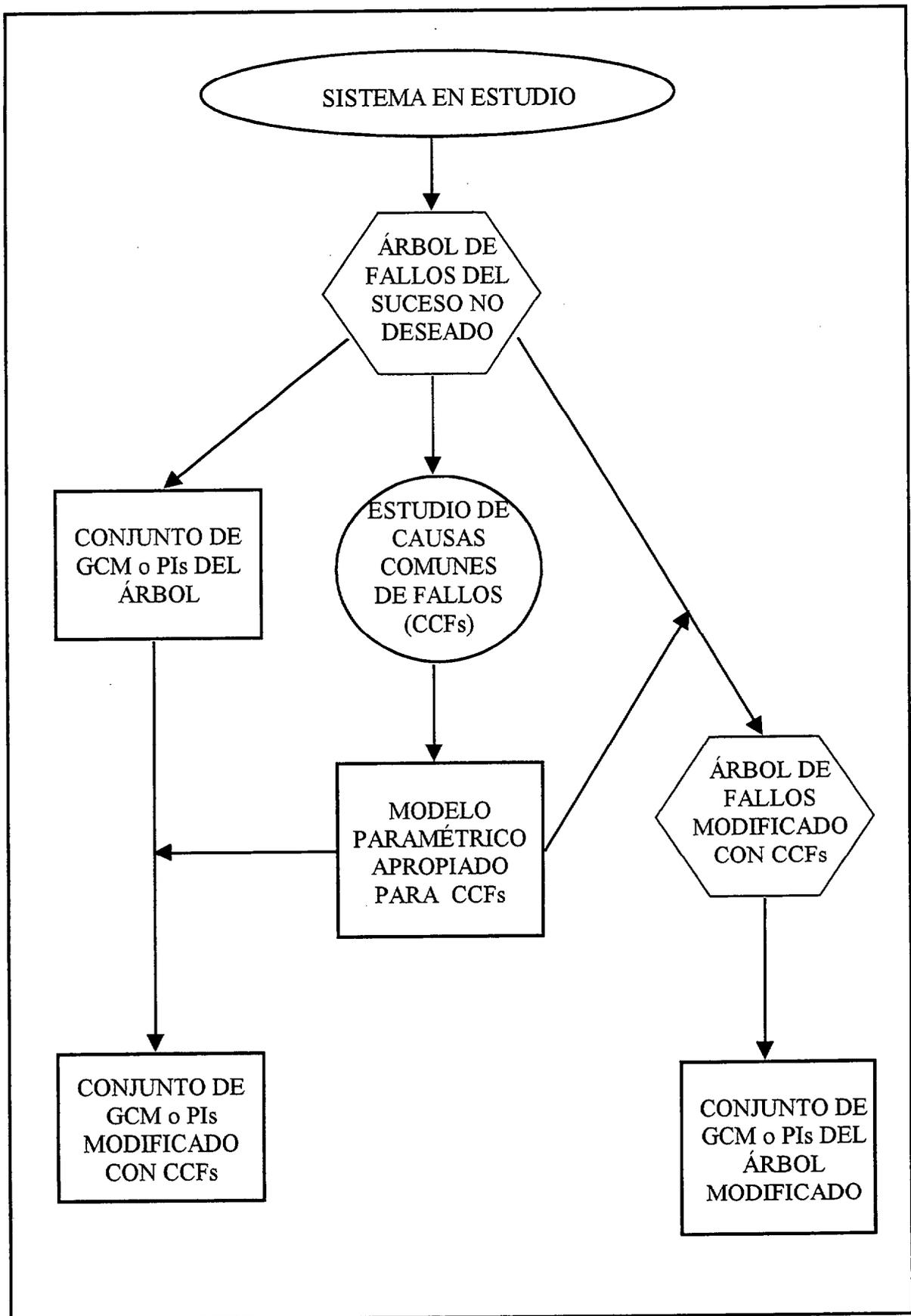


Fig. 2.29.- ESTUDIO DE CAUSAS COMUNES Y FORMA DE INCORPORARLAS AL ANÁLISIS DE ÁRBOLES DE FALLOS

2.7.- ANÁLISIS CUANTITATIVO DEL ÁRBOL DE FALLOS

2.7.1.- Introducción

La cuantificación es una etapa esencial en el Análisis de Árbol de Fallos, ya que puede proporcionar diversas magnitudes de gran interés desde el punto de vista de la fiabilidad del sistema. En la figura 2.10 se pueden apreciar las diversas magnitudes a medir a la salida (suceso top) del modelo (árbol de fallos). Gran cantidad de aportaciones se han producido desde principios de los 60 a la cuantificación de los árboles de fallos, todas estas ellas clasificadas desde diversos puntos de vista (apartado siguiente), sin embargo es de destacar que también se han producido grandes cambios debido al ritmo del progreso técnico en el desarrollo de ordenadores cada vez más rápidos. Con frecuencia, métodos que eran desechados por no poder evaluar árboles de determinado tamaño, vuelven a ser considerados como aceptables pocos años después a causa de nuevas generaciones de ordenadores.

Históricamente se ha evolucionado a partir de la Simulación por Monte-Carlo (1975) a la Teoría Cinética del Árbol (1979) puramente determinista, así como a métodos basados en los Grupos de Corte Mínimos (exactos y aproximados), a partir de 1986 comienza el interés por la evaluación directa por métodos exactos y en la década de los 90 se ha puesto gran énfasis en los métodos directos aproximados usando truncamiento y recientemente se ha complementado con la evaluación usando Diagramas de Decisión Binarios (DDBs) mediante productos implícitos (Metaproductos). En la actualidad, si el árbol tiene solo puertas lógicas básicas, el método que permite evaluar mayores árboles de fallos en un tiempo menor de CPU es el de conversión a DDB seguido de Metaproductos, mientras que si el árbol incorpora modernas puertas lógicas, será necesario tratarlo vía Cadenas de Markov o emplear simulación por Monte Carlo si fuese de grandes dimensiones. La organización del capítulo consta de una clasificación de los métodos seguida de una explicación metodológica. Se hace especial énfasis en los métodos directos y en la simulación por Monte-Carlo (a la que se dedica un apartado), ya que son sustento del resto del trabajo así como a la Teoría Cinética del Árbol como aglutinante de todos los conocimientos sobre las magnitudes a evaluar.

NOMBRE	DESCRIPCIÓN	REPRESENTACIÓN
Disponibilidad	Probabilidad de que el suceso top no exista al tiempo t	$A_s(t)$
No Disponibilidad	Probabilidad de que el suceso top exista al tiempo t	$Q_s(t)=1-A_s(t)$
Fiabilidad	Probabilidad de que el suceso top no ocurra en $(0,t]$	$R_s(t) \leq A_s(t)$
No Fiabilidad	Probabilidad de que el suceso top ocurra antes de t	$F_s(t)=1-R_s(t)$ $F_s(t) \geq Q_s(t)$
Densidad de Fallos	Derivada primera de la distribución de fallos $F_s(t)$	$f_s(t)=dF_s(t)/dt$
Intensidad condicional de fallos	Probabilidad de que el suceso top ocurra por unidad de tiempo t	$\lambda s(t)$
Número diferencial esperado de ocurrencias del suceso top	Número de veces que se espera que ocurra el suceso top durante $(t,t+dt)$	$W_s(t,t+dt)=w_s(t)dt$
Número esperado de ocurrencias del suceso top	Número de veces que se espera que ocurra el suceso top durante $[t_1,t_2)$	$W_s(t_1,t_2) = \int_{t_1}^{t_2} w_s(t)dt$
Tiempo Medio al Primer Fallo	Tiempo esperado que transcurrirá hasta la primera ocurrencia del suceso top	$MTTF_s = \int_0^{\infty} f_s(t)dt$

Figura 2.30.- DIFERENTES MAGNITUDES MEDIBLES EN EL SUCESO TOP DE UN ÁRBOL DE FALLOS

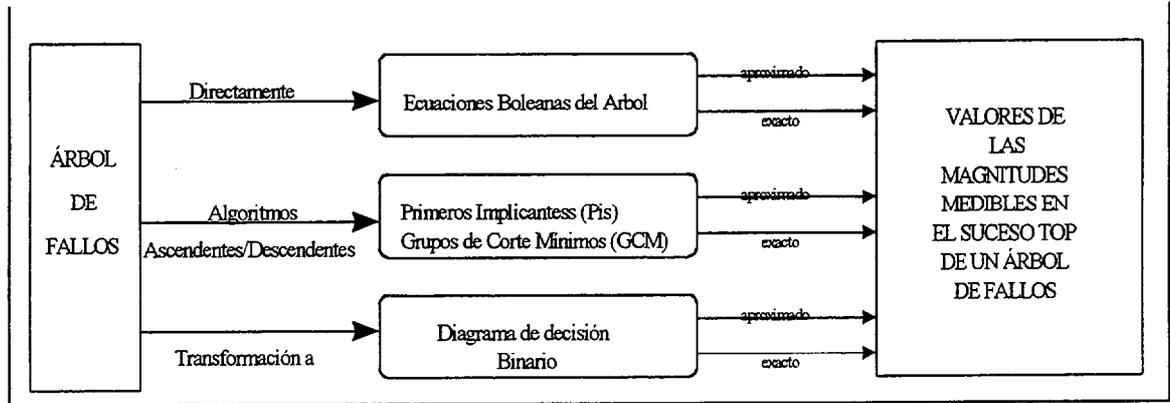
2.7.2.- Clasificación de los métodos de Análisis Cuantitativo

Los métodos de cuantificación del árbol de fallos se pueden clasificar de acuerdo al camino seguido para abordarla. Una primera clase de métodos es aquella que emplea Primeros Implicantes (PIs) (ó GCM), que puede ser definida como el procedimiento clásico de evaluación ya que se ha empleado con profusión durante muchos años. En estos métodos antes de realizar la cuantificación se realiza un análisis cualitativo y se determinan las combinaciones mínimas de sucesos básicos que causan la aparición del suceso top del árbol, denominados Primeros Implicantes (PIs) en el caso de estructuras no coherentes o Grupos de Corte Mínimos en el caso de estructuras coherentes. Una vez obtenidos estos conjuntos mínimos es posible abordar mediante cálculos probabilísticos la evaluación.

Otra clase está constituida por los métodos que evalúan directamente a partir de su Función de Estructura y que tienen dos importantes subdivisiones: procedimientos exactos capaces de evaluar el árbol exactamente, empleables para árboles de hasta mediano tamaño, y procedimientos aproximados, empleables para árboles de gran tamaño. Estos métodos evitan las dificultades inherentes al hecho de realizar el análisis cualitativo previo, pero se enfrentan a los derivados de afrontar directamente la función de estructura.

La última clase y muy recientemente perfeccionada, es la que evalúa el árbol mediante una transformación previa a Diagrama de Decisión Binario seguida del empleo de productos implícitos, que ha permitido evaluar exactamente árboles que hasta ahora sólo se evaluaban de forma aproximada. El principal inconveniente de estos métodos viene del hecho de no existir criterios matemáticamente demostrados que permitan definir el orden adecuado de los sucesos básicos que conduce al DDB óptimo, lo que puede llegar a anular las ventajas a conseguir en comparación con los otros métodos.

En las figuras siguientes se pueden apreciar el detalle de ésta clasificación, se ha procurado por coherencia del cuadro dividir los métodos entre exactos y aproximados y a continuación, para cada uno de ellos, diferenciar en cuanto a la metodología subyacente en las diferentes aportaciones. Ante el gran número de aportaciones a los diferentes métodos, se ha elegido la estrategia de realizar una clasificación bibliográfica previa para encuadrar trabajos significativos, seguida de una presentación de los métodos relevantes intentando reflejar gráficamente un resumen del contenido de las mismas, allí donde era posible.



MÉTODOS DE CUANTIFICACIÓN DEL ÁRBOL DE FALLOS	Directos	Exactos	Uso de Patrones y List Processing Uso de Recursividad Uso eficiente de Algoritmos ascendentes Uso del Teorema de Factorización Uso de Programación Orientada a Objetos
		Aproximados	Uso de la expansión de Shannon Uso del procesado paralelo Uso de Cortes Heurísticos Uso de Simulación
	Basados en PI/GCM	Exactos	Basados en la función de estructura
		Aproximados	Extremos de Inclusión-Exclusión Extremos de Esary y Proschan Extremos Mín.-Max. Uso de Módulos Uso de Truncamiento
	Basados en DDBs	Exactos	Uso de Productos implícitos
		Aproximados	Vía GCM

Fig. 2.31.- CLASIFICACIÓN DE LOS MÉTODOS DE CUANTIFICACIÓN DE ÁRBOLES DE FALLO

2.7.3.- Métodos Directos

Introducción

Los métodos directos emplean la función de estructura en la forma en que se obtiene del proceso de construcción y abordan su evaluación sin ningún tipo de transformación previa. Dependiendo de la metodología empleada se podrá llegar a resultados exactos o aproximados. En el cuadro siguiente se puede encontrar una selección de aportaciones para cada metodología. En ella se puede apreciar que las aportaciones se concentran en los últimos 15 años excepto las que emplean simulación que comienzan mucho antes. La importancia de estos métodos radica en que permiten tratar muy grandes árboles de fallos generalmente con un corte computacional aceptable, susceptibles de paralización ó evaluar todo tipo de características cuando se procede por simulación. En lo que sigue se hará una exposición siguiendo el cuadro mencionado atendiendo a las características más sobresalientes de la metodología empleada así como al tamaño de los árboles tratados en las referencias.

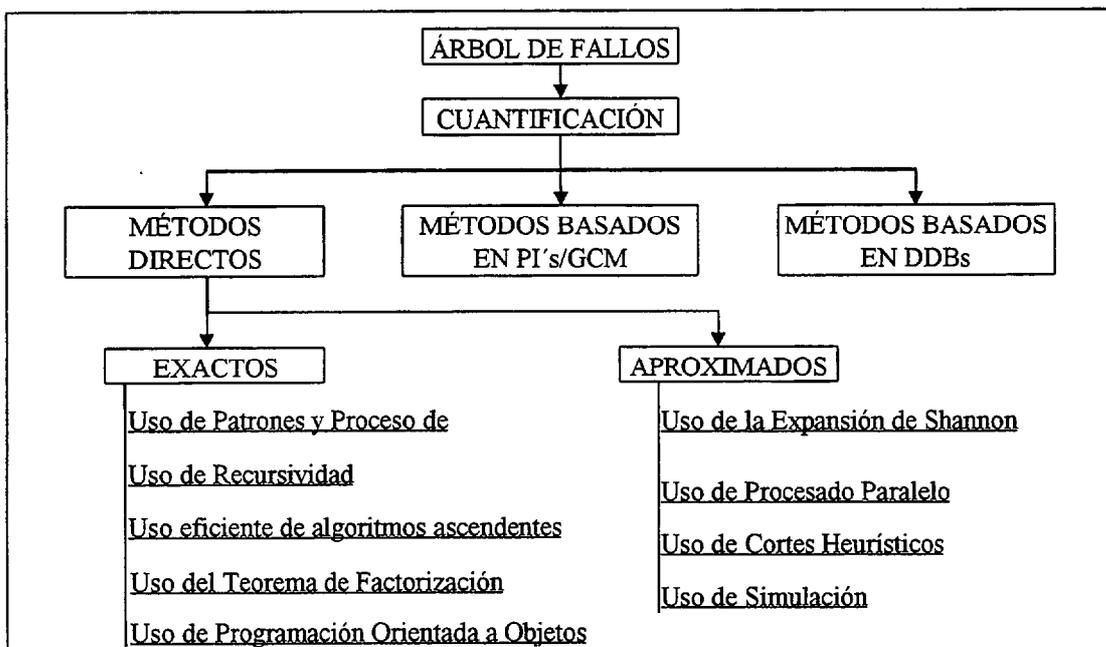


Fig. 2.32.- CLASIFICACIÓN DE LOS MÉTODOS DIRECTOS DE CUANTIFICACIÓN DE ÁRBOLES DE FALLOS

MÉTODOS APROXIMADOS	REFERENCIAS
Uso de la Expansión de Shannon	(Braun, 1991) (Schneeweiss, 1992)
Uso de Procesado Paralelo	(Heger et al., 1993)
Uso de Cortes Heurísticos	(Rai, 1994)
Uso de Simulación	(Vesely, 1975) (Kumamoto et al., 1980) (Fishman, 1987) (Kumamoto, 1987) (Henley y Kumamoto, 1993)

MÉTODOS EXACTOS	REFERENCIAS
Uso de Patrones y Proceso de Listas	(Koen, 1973) (Blin et al., 1978)
Uso de Recursividad	(Page & Perry, 1986 a,b), (Patterson-Hine & Koen, 1987) (Radke y Evanoff, 1994) (Page & Perry, 1991)
Uso eficiente de algoritmos ascendentes	(Stecher, 1986) (Feo, 1986)
Uso del Teorema de Factorización	(Page & Perry, 1987) (Page & Perry, 1991)
Uso de Programación Orientada a Objetos	(Patterson-Hine y Koen, 1989)

Fig. 2.33.- APORTACIONES SIGNIFICATIVAS A LOS MÉTODOS DIRECTOS DE CUANTIFICACIÓN DE ÁRBOLES DE FALLOS

Métodos Exactos

El **uso de patrones** (ó reconocimiento de formas) fue introducido por Koen (1972) mediante el empleo de un algoritmo para localizar subárboles dentro del árbol en estudio cuya estructura fuese igual a alguna de las almacenadas en la librería de patrones. Cuando se localiza un subárbol de ésta forma, se sustituye éste por la probabilidad de su suceso top, cuya fórmula de cálculo está almacenada en la librería de patrones de búsqueda. Si se logra encontrar patrones es posible lograr una reducción considerable en la lógica del árbol original. Este proceso continua hasta que el árbol entero es reducido a un simple suceso con su probabilidad de ocurrencia (ver figura 2.35).

Este proceso fue implementado en el código PATREC (Blin et al, 1978) escrito en PL/1. Este código ha sido durante muchos años el estándar para la evaluación de árbol de fallos de la Comisión de Energía Atómica de Francia. Versiones posteriores del programa se implementaron usando técnicas de **procesamiento de listas** con lenguaje LISP (lenguaje de procesamiento simbólico) (Patterson-Hine y Koen, 1985) y en Prolog (Koen & Koen 1987). Deberá disponerse de una abundante librería de formas preevaluadas así como de un algoritmo eficiente de búsqueda de patrones para que éste método sea aplicable en la práctica.

La Recursividad es una característica de sólo algunos lenguajes de programación de ordenadores, no fue empleada con frecuencia a la hora de analizar árboles de fallo debido fundamentalmente a que el FORTRAN no la posee y éste fue durante muchos años el lenguaje predilecto de los ingenieros. Sólo algunas aplicaciones como PATREC ya mencionado en el apartado anterior empleaban cierta recursividad.

En 1986 Page y Perry reiniciaron el interés por la evaluación directa de los árboles y emplearon para ello la posibilidad de empleo de subrutinas recursivas del Pascal. Su trabajo despertó gran interés pues permitía abordar árboles de fallos para evaluación directa en ordenadores de pequeño formato. Hasta ese entonces el camino que se seguía para la evaluación era usando los Grupos de Corte Mínimos. Empleando para ello Mainframes ó Miniordenadores. El algoritmo de Page y Perry emplean las probabilidades básicas de los sucesos y una subrutina recursiva para ir recorriendo el árbol, sustituyendo las puertas por su evaluación mediante sucesivas llamadas a ésta. En la figura siguiente se puede apreciar una de las subrutinas recursivas presentadas así como el número de llamadas a la misma al evaluar un árbol ejemplo. Un punto importante para la eficiencia de

esta metodología es optimizar el número de llamadas necesarias a la subrutina recursiva. En un posterior trabajo Page y Perry (1991) optimizaron sus algoritmos logrando el estudio con éxito de Árboles de hasta 256 nodos (sucesos básicos) más puertas lógicas contados tantas veces como aparezcan los árboles tratados pueden ser coherentes y no coherentes aunque en árboles grandes se limitaron al primer caso. La forma de proceder es descendente (Top-Down). Según (Rai,1994) éstos métodos no se pueden aplicar a árboles con puertas XOR y NOR ni otras puertas no estándar (figuras 2.34 y 2.36).

```

FUNCTION PROB
(actúa sobre un conjunto S de etiquetas de nodo pasadas como un parámetro)

IF S contiene sólo etiquetas de nodos primarios THEN

    IF (S contiene un par de sucesos básicos complementarios) THEN
        PROB = 0
    ELSE
        PROB = Producto de las probabilidades de los sucesos básicos
    ENDIF

ELSE

    IF ( S contiene una etiqueta n correspondiente a una puerta AND ) THEN

        PROB = PROB (  $S \cup \{ LEFT[n], RIGHT[n] \} - \{n\}$  )
    ELSE seleccionar una puerta OR etiqueta n

    PROB = PROB(  $S \cup \{ LEFT[n] \} - \{n\}$  ) + PROB(  $S \cup U \{ RIGHT[n] \} - \{n\}$  ) -
    PROB(  $S \cup \{ LEFT[n], RIGHT[n] \} - \{N\}$  )

    ENDIF

ENDIF

```

Fig. 2.34.- SUBROUTINA RECURSIVA PROPUESTA POR Page y Perry (1986) PARA EVALUAR ÁRBOLES DE FALLOS

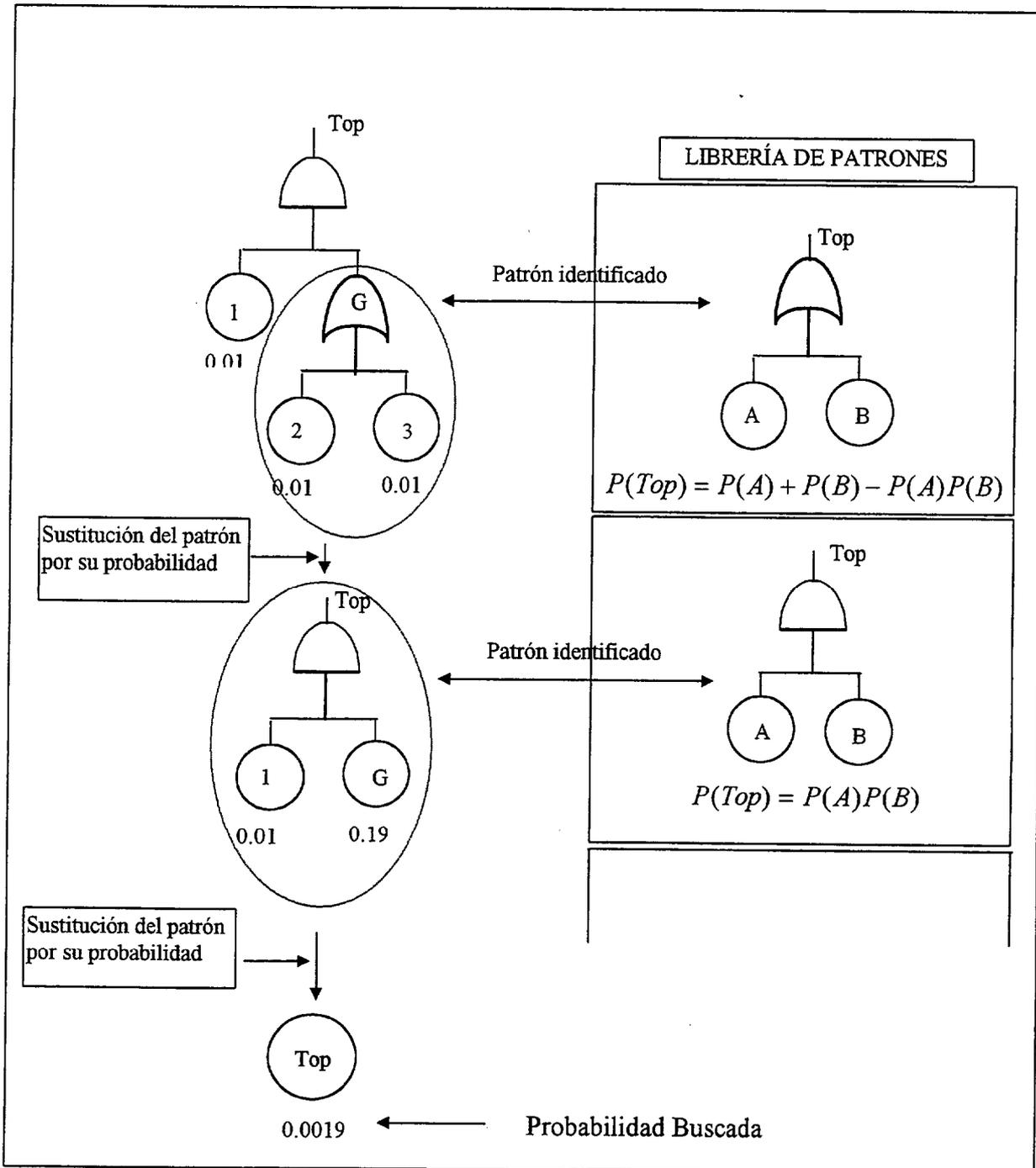


Fig 2.35.- CUANTIFICACIÓN DIRECTA DEL ÁRBOL USANDO RECONOCIMIENTO DE PATRONES

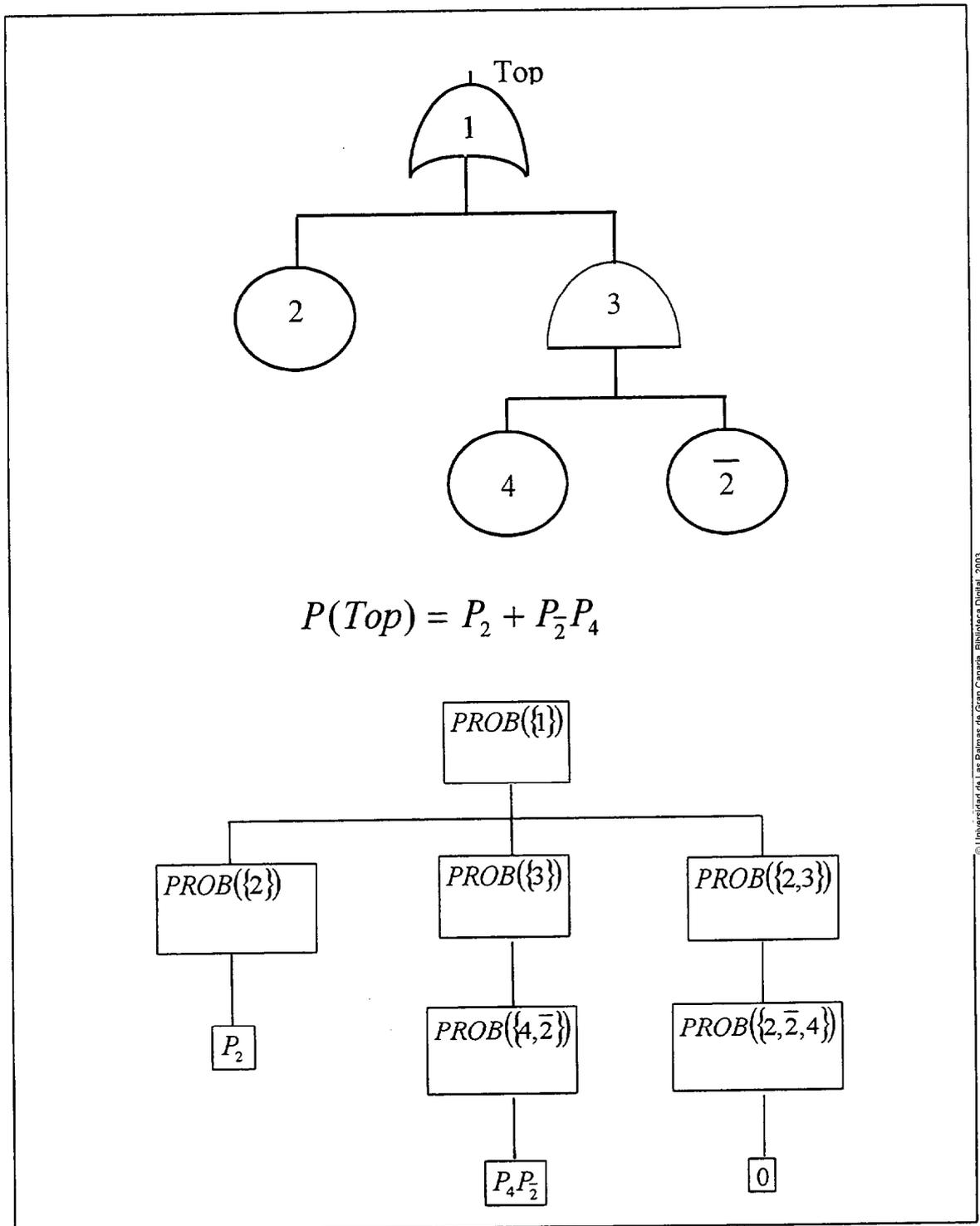


Fig. 2.36.- LLAMADAS RECURSIVAS A LA SUBROUTINA DE EVALUACIÓN PARA UN ÁRBOL DE FALLOS SENCILLO

El **uso del Teorema de Factorización** fue introducido por Page & Perry (1987) mediante una implementación para miniordenadores y ha mostrado una eficacia interesante para árboles de hasta tamaño medio. Se basa en la propiedad elemental de las probabilidades .

$$P(F) = P(E)P(F / E) + P(\bar{E})P(F / \bar{E})$$

También desconocida a veces Ley multiplicativa y que se usa como precursor del Teorema de Bayes. La esencia de la idea de Page & Perry es aprovechar aquellos árboles que tienen sucesos y sus negaciones para descomponer su evaluación en la de árboles más pequeños, obtenidos del original, empleando la fórmula anterior. Supóngase que un determinado árbol tiene un suceso A y su negación \bar{A} empleando la fórmula anterior obtendremos:

$$P(TOP) = P(A)P(TOP / A) + P(\bar{A})P(TOP / \bar{A})$$

donde:

$P(TOP / A)$ = Probabilidad del suceso top del subárbol obtenido suponiendo que ha ocurrido A y no ha ocurrido \bar{A}

$P(TOP / \bar{A})$ = Probabilidad del suceso top del subárbol obtenido suponiendo que no ha ocurrido A y que ha ocurrido \bar{A}

Ello implica que el árbol es descompuesto en nuevos árboles mediante un proceso de reconstrucción que supone la existencia o no de los diferentes sucesos básicos, mientras se aplica la fórmula de evaluación. Este proceso es el talón de Aquiles de ésta metodología que ha producido, sin embargo resultados notables. En un trabajo más reciente (Page & Perry, 1991) los autores han evaluado con éxito de forma exacta árboles cercanos a los 800 nodos (sucesos básicos más puertas contadas tantas veces como aparecen). Además comparan los resultados con los obtenidos mediante el empleo de recursividad, siendo superior en prestaciones los algoritmos que emplean Factorización. En la figura siguiente se muestra un ejemplo sencillo que ilustra esta forma de proceder en esencia.

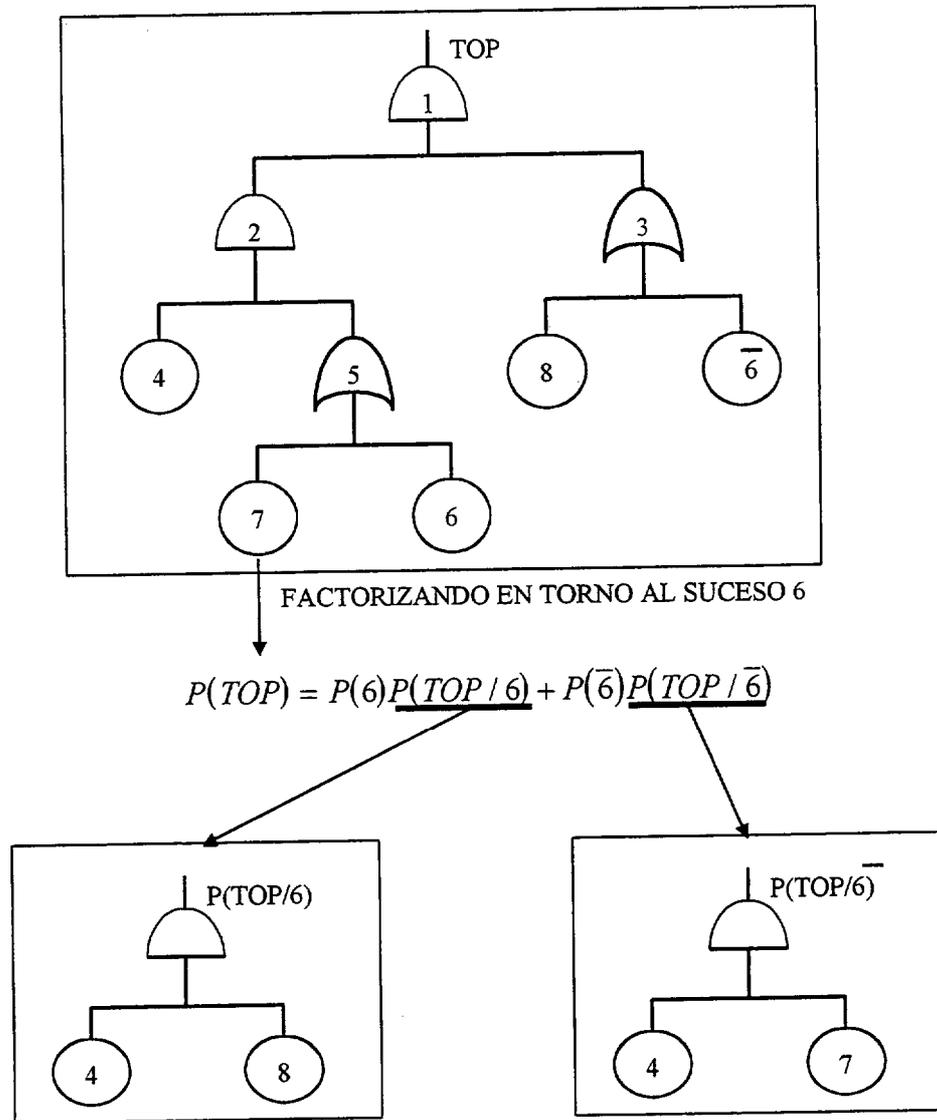


Fig. 2.37.- APLICACIÓN DEL TEOREMA DE FACTORIZACIÓN PARA CUANTIFICAR ÁRBOL DE FALLOS

La **Programación Orientada a Objetos (POO)** para la evaluación de árboles de fallos fue introducida por Paherson-Hine y Koen (1989). Su aplicación es ciertamente una combinación de técnicas bien conocidas en la evaluación de árboles de fallos ya que incorpora además un proceder ascendente (Bottom-Up) para árboles sin sucesos repetidos y un proceso descendente (top-down) y recursivo para árboles con sucesos repetidos. Usan además LIST (List Processing y capacidad de tratamiento simbólico) y calculan tanto la probabilidad del suceso top como la de los sucesos intermedios. La Programación Orientada a objetos es una filosofía diferente para programar y usar ordenadores, incorpora tres propiedades básicas denominadas Encapsulado, Herencia y Polimorfismo que permiten desarrollar un software modular, extensible, flexible y conciso. La entidad básica en POO es el objeto que es un tipo de dato abstracto. Los objetos se agrupan en clases que combinan comportamiento y estado de los objetos. Cuando se analiza un árbol de fallos con objetos tienen la capacidad de almacenar los resultados de las reducciones lo que incrementa la eficiencia del proceso descendente empleando para evaluar el árbol. En la figura siguiente se puede apreciar el algoritmo recursivo descendente empleado cuyos resultados fueron buenos para árboles de nivel medio. No fue implementado para árboles no coherentes pero los autores afirman que la extensión a este tipo de árboles es sencilla y segura.

```

PROBABILIDAD DEL SUCESO S

IF ( S contiene sucesos independientes ) THEN
    Marcar los sucesos independientes para que no sean expandidos
IF ( Todos los sucesos en S son independientes ) THEN
    PROB-OF-EVENT(S) = producto de las no disponibilidades de cada
                        elemento de S
ELSE IF ( S contiene puertas AND ) THEN
    S' = S con todas las puertas AND reemplazadas por sus entradas
    IF ( S' contiene puertas AND ) THEN
        Marcar las puertas AND independientes o expandirlas
    ELSE
        PROB-OF-EVENT =  $PROB - OF - EVENT(\{j\} \cup S - \{n\}) +$ 
             $(1 - Unavailability(\{j\})) * PROB - OF - EVENT(S \cup \{k\} - \{n\})$ 
    ELSE
        PROB-OF-EVENT =  $PROB - OF - EVENT(S - \{n\} \cup \{j\}) +$ 
             $PROB - OF - EVENT(S - \{n\} \cup \{k\}) -$ 
             $PROB - OF - EVENT(S - \{n\} \cup \{j, k\})$ 

```

Fig. 2.38.- SUBROUTINA RECURSIVA (Patterson-Hine y Koen (1989)

Métodos Aproximados

La evaluación directa del árbol de fallos puede ser abordada mediante métodos aproximados. Una opción interesante para ello es el **uso de la Expansión de Shannon** (ya introducida en el apartado 3.2.), bien mediante una expansión incompleta (Braun, 1991) ó mediante cierta selección de los elementos a considerar (Schneeweiss, 1992) de tal manera que se asegure un determinado error.

Dada una función de estructura $\phi(X_1, X_2, \dots, X_n)$ la expansión de Shannon respecto a cada variable aplicada sobre un determinado orden de las mismas produce una función fácilmente expresable en forma de diagrama (árbol de Shannon de la función) (árbol binario), dado que para valores de "n" considerablemente grandes (100 ó más), es imposible en la práctica obtener la descomposición completa se debe proceder a interrumpir el proceso de expansión en un punto tal que el error cometido sea admisible (menor que uno seleccionado a priori) Para un determinado árbol de Shannon la forma de interrumpir el desarrollo es sustituir un nodo por la probabilidad igual a la unidad (despreciando el subárbol que termina en él) y el error cometido será, al menos, la suma de las probabilidades de las ramas que conducen desde el suceso top hasta cada nodo sustituido.

Este proceso deberá implementarse mediante un algoritmo computacional eficiente capaz de explorar las diversas expansiones posibles para una función de estructura y determinar cual es la más apropiada para obtener el resultado con una precisión prefijada. Schneeweiss (1992) propuso el siguiente algoritmo para implementar este proceso:

- 1) Expandir la función de estructura $\phi(X_1, X_2, \dots, X_n)$ mediante Shannon algunos pasos a un binario (árboles de Shannon) en el cual se pueda suponer que un determinado valor U'_s de la no disponibilidad del sistema sobrepasa a la no disponibilidad exacta en, al menos, ε_{max} (máxima diferencia entre los extremos superior e inferior de la no disponibilidad).
- 2) Si $\varepsilon \leq \varepsilon_{max}$ ir al paso 4
- 3) Si $\varepsilon > \varepsilon_{max}$ volver al paso 1
- 4) Evaluar el árbol de Shannon truncado reemplazando su descripción algebraica de literales por las esperanzas matemáticas correspondientes, considerando independencia estadística.

(Si el componente no está negado se reemplaza por su no disponibilidad).

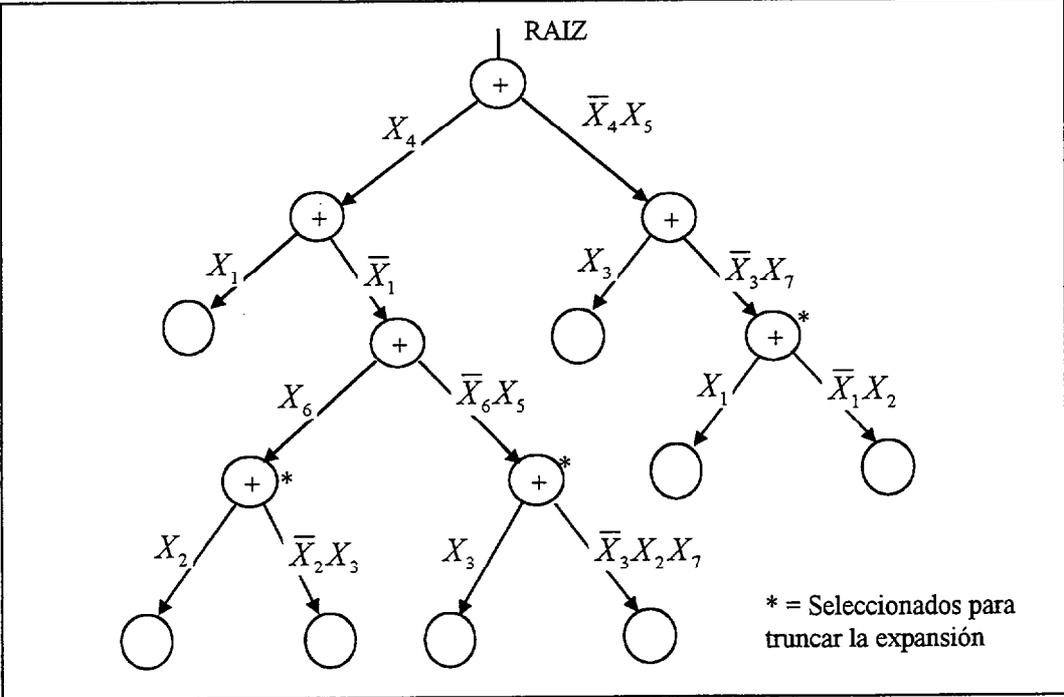
(Si el componente está negado, por su disponibilidad).

5) Imprimir los valores numéricos de ε y aproximado U'_s de U_s . Téngase en cuenta que:

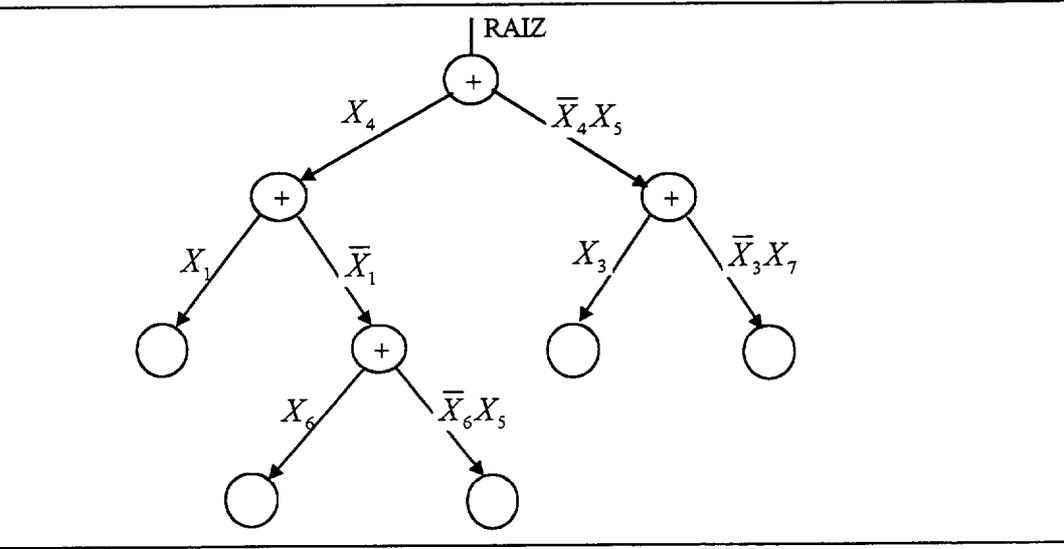
$$U'_s - \varepsilon < U_s \leq U'_s \quad \varepsilon < \varepsilon_{max}$$

En la figura siguiente se puede apreciar un ejemplo de función de estructura, árbol de Shannon correspondiente, árbol simplificado eliminando (cortando la expansión de Shannon) determinados subárboles y error cometido con esa eliminación. No se han comunicado resultados de este algoritmo para el caso de árboles de fallos de mediano o gran tamaño, lo cual hace difícil su comparación con otros métodos.

$$\phi(X_1, X_2, \dots, X_7) = X_4 \left\{ X_1 + \bar{X}_1 \left[X_6 (X_2 + \bar{X}_2 X_3) + \bar{X}_6 X_5 (X_3 + \bar{X}_3 X_2 X_7) \right] \right\} + \bar{X}_4 X_5 \left\{ X_3 + \bar{X}_3 X_7 [X_1 + \bar{X}_1 X_3] \right\}$$



ÁRBOL SIMPLIFICADO POR TRUNCAMIENTO



$$ERROR_COMETIDO = \varepsilon = U_4 A_1 (U_6 + A_6 U_5) + A_4 U_5 A_3 U_7$$

→ PROB=1 A = Disponibilidad U = No Disponibilidad

Fig. 2.39.- ERROR COMETIDO AL TRUNCAR LA EXPANSIÓN DE SHANNON según SCHNEEWEISS(1992)

El uso de **Cortes Heurísticos** para obtener extremos de la probabilidad del suceso top del árbol ha sido introducido por Rai (1994). El método desarrollado es aplicable a grandes árboles de fallos con sucesos repetidos y todo tipo de puertas lógicas convencionales (AND, OR, XOR, NOT), el proceso es ascendente (bottom-up). El concepto de Corte Heurístico convierte un árbol de fallos en un árbol equivalente con sucesos no repetidos mediante la asignación a una sola de las ramas de un suceso repetido de su valor de probabilidad exacto y al resto un intervalo de probabilidad que será función de tres cosas:

- El Tipo de Reconvergencia
- La Paridad de las ramas
- El Tipo de Puerta en el que reconverge

El Tipo de Reconvergencia puede ser positivo o negativo. Será positivo cuando todas las ramas en las que reconverge un suceso repetido tienen igual (todas "par o todas impar") y negativo en caso de ser paridades desiguales (unas "par" otras "impar"). La impar de una rama es el número de cambios de signo (o a 1 ó viceversa), que hay en esa rama entre el punto de disipación del repetido y el punto de reconvergencia. Los pasos fundamentales de éste método son:

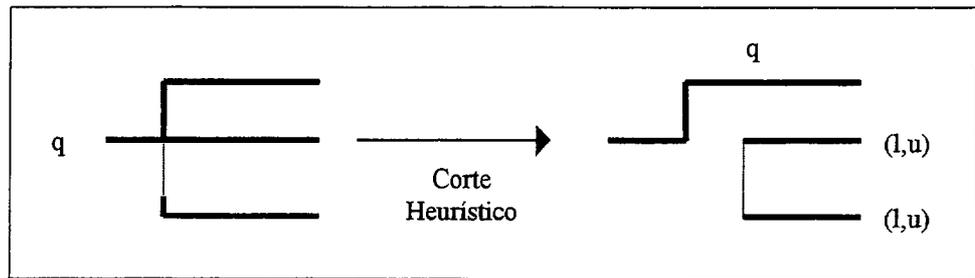
- 1) Asignar probabilidades a los sucesos básicos (q_i).
- 2) Procediendo de forma ascendente calcular las probabilidades de todas las líneas del árbol que no contienen sucesos repetidos usando las fórmulas siguientes en función del tipo de puertas.

$$\text{NOT : } \Pr(y) = 1 - q$$

$$\text{AND : } \Pr(y) = \prod_{i=1}^n q_i$$

$$\text{OR : } \Pr(y) = 1 - \prod_{i=1}^n (1 - q_i)$$

- 3) Modificar el árbol cortando heurísticamente todas las ramas menos una de los sucesos repetidos de acuerdo con la figura siguiente.



4) Obtener los extremos de probabilidad para todas las ramas restantes utilizando los intervalos mediante las fórmulas siguientes.

TIPO DE RECONVERGENCIA	PARIDAD	INTERVALO PARA	INTERVALO PARA
		AND	OR
Positiva	PAR	[g,1]	[0,g]
	IMPAR	[0,g]	[g,1]
Negativa	PAR	[0,g]	[g,1]
	IMPAR	[g,1]	[0,g]

PUERTA	EXTREMO INFERIOR	EXTREMO SUPERIOR
NOT	$1 - u$	$1 - l$
AND	$\prod_1^n l_i$	$\prod_1^n u_i$
OR	$1 - \prod_1^n (1 - l_i)$	$1 - \prod_1^n (1 - u_i)$
XOR	$L_n = L_{n-1}(1 - 2l_n) + l_n$	$U_n = U_{n-1}(1 - 2u_n) + u_n$

En la figura siguiente se puede apreciar un ejemplo de aplicación de este método a un árbol de fallos sencillo.

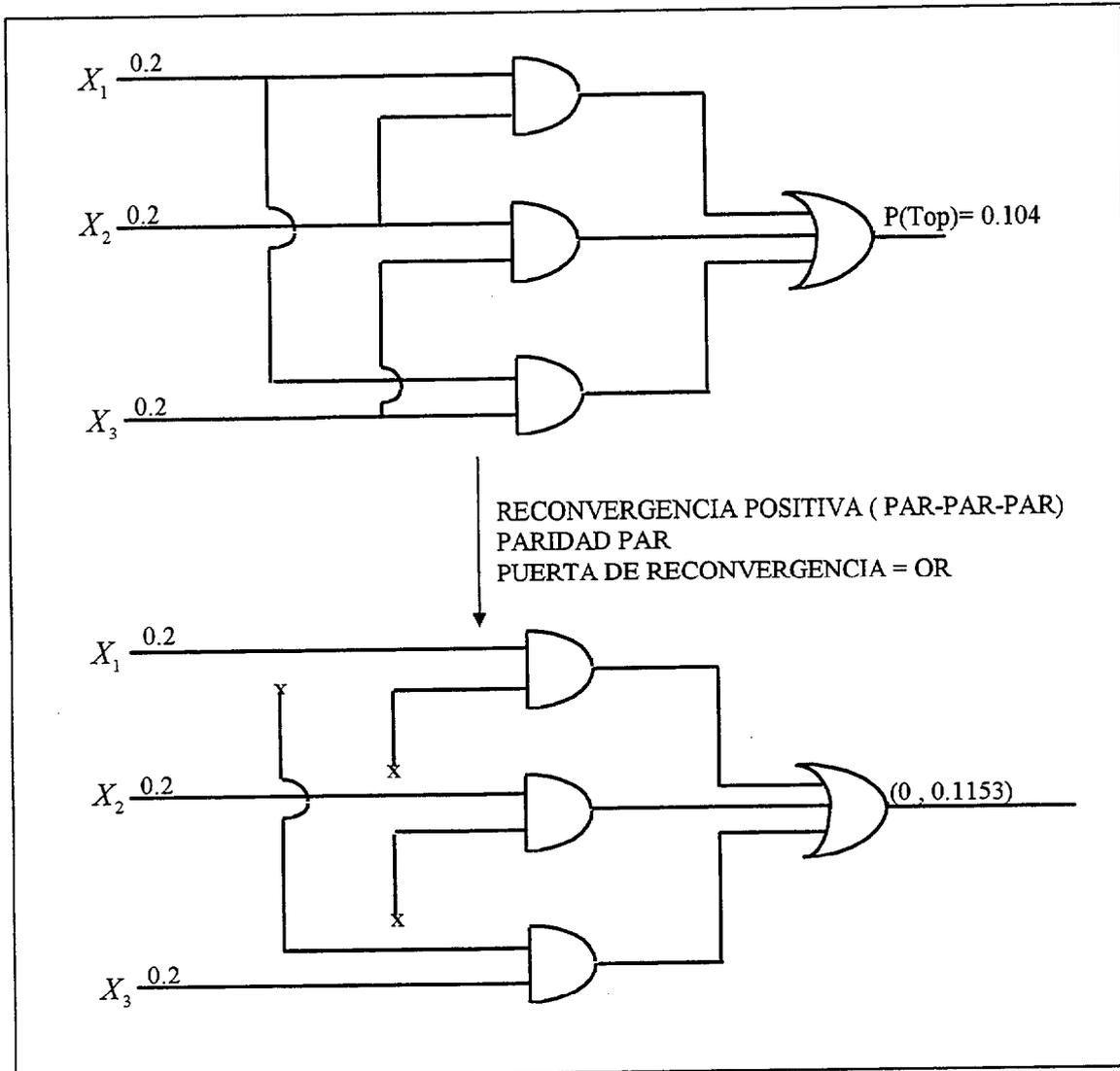


Fig. 2.40.- SIMPLIFICACIÓN Y EVALUACIÓN DEL ÁRBOL DE FALLOS MEDIANTE CORTES HEURÍSTICOS

La simulación es una poderosa herramienta para resolver muchos y muy diferentes problemas, entre ellos la cuantificación del árbol de fallos. Dada su sencillez de programación y versatilidad para incorporar todo tipo de lógica se usa simulación siempre que se desea explorar un árbol de fallos de características no contempladas por los procedimientos deterministas disponibles. También es indispensable cuando la lógica del árbol es demasiado compleja o esté demasiado grande para los mismos.

El proceso de simular un árbol de fallos se compone de dos etapas: Una en la que se obtienen estados binarios para todos los sucesos básicos (vector de estado, un "0" ó un "1" para cada suceso básico), y otra en la que se comprueba si se produce el suceso top para cada vector de estado, contando aquellas ocasiones en que así ocurre. El resultado será la probabilidad del suceso top y un intervalo de confianza para dicha magnitud.

El principal inconveniente de la simulación es el elevado número de casos (vectores de estado + comprobación del árbol + resultado caso de ocurrir el top) (historias) que es necesario ensayar para obtener un valor de la probabilidad buscada con suficiente precisión. Cuando se conoce algo sobre el proceso que se está simulando se puede intentar reflejar ese conocimiento en el proceso de simulación con el fin de obtener una respuesta más precisa con muchos menos ensayos, este proceso se denomina Reducción de Varianza. En el caso de árboles muy grandes o complejos es necesario recurrir a la reducción de varianza para que la respuesta obtenida sea aceptable. A lo largo de los años han sido desarrollados diversos procedimientos de este tipo, algunos de ellos de gran eficacia. Dada la importancia de este procedimiento para el Análisis de Árbol de Fallos se ha desarrollado en un epígrafe a parte de éste capítulo. (ver epígrafe 2.7.7).

2.7.4 - Métodos basados en los GCM/PIs.

Introducción

Los Primeros Implicantes ó Grupos de Corte Mínimos (para sistemas coherentes) constituyen en sí mismos una valiosa información, tal como se ha visto en el apartado de Análisis Cualitativo. Por otra parte consideramos en conjunto como representación de la función de estructura ϕ del árbol de fallos permita además el cálculo de la probabilidad del suceso top mediante el empleo de las propiedades estocásticas de los mismos. Si los grupos de corte son disjuntos entre sí, o sea, no existen sucesos básicos repetidos en el árbol de fallos, cálculo de P (top) es directo y sencillo, en caso contrario existirá dependencia entre los mismos y el cálculo se puede complicar enormemente, máxima si existen muchos GCM como es el caso de la mayoría de los árboles de fallos reales. Durante muchos años los métodos basados en los PIs (ó GCM)) han sido la opción estándar a la hora de proceder al cálculo de la probabilidad del suceso top .

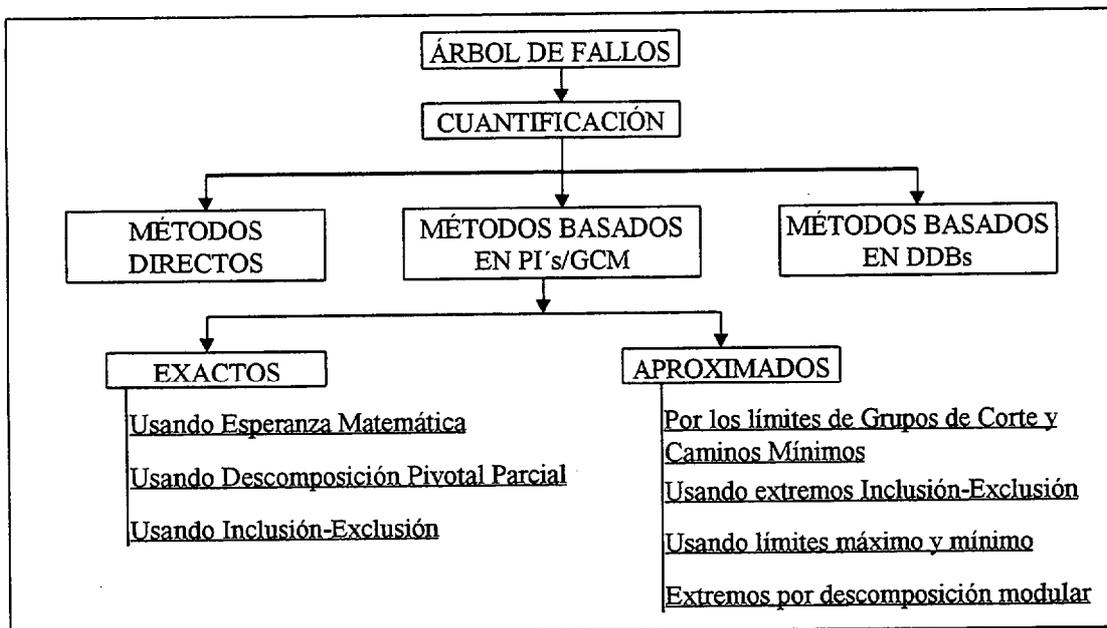


Fig. 2.41.- MÉTODOS DE CUANTIFICACIÓN BASADOS EN GCM/PI's

Métodos Exactos

Una vez que han sido determinados los Primeros Implicantes PIs o los Grupos de Corte Mínimos (GCM) es posible determinar la probabilidad del suceso top exactamente aunque, dado el crecimiento exponencial con el número de sucesos básicos esto sólo es posible para árboles de pequeño tamaño. Henley & Kumamoto (1993) exponen tres métodos para ello.

Si los sucesos básicos aparecen sólo en un grupo de corte mínimo, la función de estructura puede escribirse como:

$$\phi(Y) = \bigcap_{j=1}^m \left[\bigcup_{i=1}^{n_j} Y_{ij} \right] = 1 - \prod_{j=1}^m \left[1 - \prod_{i=1}^{n_j} Y_{ij} \right] = 1 - \prod_{j=1}^m [1 - k_j(Y)]$$

$$Y_{ij} = GCM_{ij} \quad m = \text{número de GCM}$$

que da la función de estructura del árbol de fallos en términos de las estructuras de los cortes mínimos $K_j(Y)$. Expandiendo y sometiendo a la ley de absorción ésta expresión se obtiene una forma polinomial simple de la que se puede obtener $\Pr(top) = E(\phi(Y))$

La expresión (1) no es cierta en el caso en que alguno(s) de los sucesos básicos aparece(n) en más de un grupo de corte mínimo. Una forma de obtener $E(\phi(Y))$ es expandir $\phi(Y)$ y simplificar los resultados aplicando la ley de absorción, lo que puede ser una tarea imposible dado que para árboles medianos o grandes puede haber millones de términos. Es posible hacer una descomposición parcial pivotando sobre las variables indicadoras binarias repetidas. La función de estructura se puede reescribir.

$$\phi(Y) = Y_i \phi(1_i, Y) + (1 - Y_i) \phi(0_i, Y)$$

donde $\phi(1_i, Y)$ y $\phi(0_i, Y)$ son funciones binarias obtenidas poniendo a "1" y a "0" la variable i ésima respectivamente. Ambas funciones pueden ser pivotadas sobre sus variables repetidas en un proceso reiterado hasta que todas las funciones binarias obtenidas consten de términos independientes solamente.

Otra posibilidad es calcular la No disponibilidad del sistema usando la expresión siguiente basada en el llamado Principio de Inclusión - Exclusión.

$$Q_s(t) = \Pr\left(\bigcup_{i=1}^{N_c}\right) = \sum_{i=1}^{N_c} \Pr(d_i) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} \Pr(d_i \cap d_j) + \dots + (-1)^{N_c-1} \Pr(d_1 \cap d_2 \cap \dots \cap d_{N_c})$$

donde

d_i = suceso compuesto por todos los sucesos básicos que ocurren en el i ésimo grupo de corte mínimo al tiempo t

N_c = número total de cortes mínimos.

Métodos Aproximados

Por los límites de Grupos de Corte y Caminos Mínimos

Si los sucesos básicos son estadísticamente independientes pero se encuentran repetidos en las expresiones de los grupos de corte mínimos y caminos mínimos, la probabilidad del suceso final se mantendrá entre los valores de ambos.

$$\prod_{r=1}^P \bigcup_{i \in P_r} F_i \leq P(\text{Suceso}_\text{Final}) \leq \bigcup_{s=1}^k \prod_{i \in K_s} F_i$$

Por el método de Inclusión-Exclusión

Este método nos da límites sucesivos tanto superior como inferior para la probabilidad del suceso final no deseado. La única condición que se debe cumplir es que todos los sucesos sean estadísticamente independientes.

Sea :

$$P(E_s) = \prod_{i \in K_s} F_i$$

Si el suceso final es la unión de sucesos $\bigcup_{s=1}^k E_s$ para un árbol de fallos que contenga k grupos de corte mínimos. Entonces

$$P(\text{Suceso_Final}) = P\left(\bigcup_s^k E_s\right)$$

Sea

$$\sum_{1 \leq i_1 \leq i_2 \leq \dots \leq i_r \leq k} P(E_{i_1} \cap E_{i_2} \dots \cap E_{i_r})$$

Donde aplicando los principios de Inclusión-Exclusión se obtiene la secuencia :

$$P(\text{Suceso_Final}) = \sum_{r=1}^k (-1)^{r-1} S_r$$

$$P(\text{Suceso_Final}) \leq S_1 = \sum_{s=1}^k \prod_{i \in K_s} F_i$$

$$P(\text{Suceso_Final}) \geq S_1 - S_2$$

$$P(\text{Suceso_Final}) \leq S_1 - S_2 + S_3$$

Por los límites Máximo y Mínimo

Si los sucesos básicos no son estadísticamente independientes, no son aplicables los métodos anteriores, situación nada rara en el análisis de fiabilidad de situaciones reales. Barlow y Proschan demostraron que en el caso de variables asociadas, la probabilidad del suceso final se encuentra entre los valores:

$$\text{MAX} \prod_{i \in K_s} F_i \leq P(\text{Suceso_Final}) \leq \text{MIN} \bigcap_{i \in P_{i \in Pr}} F_i$$

Extremos por descomposición modular

En un árbol de fallos una puerta lógica cuyos sucesos básicos no se repitan en ninguna otra parte del mismo forma con dichos sucesos un conjunto disjunto respecto al resto del árbol, este conjunto se denomina modulo.

Barlow y Proschan (1975) probaron que en el caso de existir independencia estadística el extremo superior de corte mínimo dado por la expresión

$$\prod_{r=1}^P \prod_{i \in P_r} F_i \leq P[\text{SucesoFinal}] \leq \prod_{s=1}^k \prod_{i \in K_s} F_i$$

es el mejor extremo cuando el árbol de fallos se descompone en módulos. Chatterjee (1975) y Olmos & Wolf (1978) desarrollaron algoritmos para calcular la probabilidad del suceso final y de los módulos en los que se puede descomponer un árbol de fallos. El primero basándose en la teoría de juegos y cuyo principal inconveniente es que necesita como dato de partida los grupos de corte mínimos. Los segundos realizan la descomposición directamente del diagrama del proceso y que fue implementado mediante el código PL-MOD

Se ha probado que todos los métodos aplicables a estructuras coherentes son aplicables a las no coherentes con excepción de los extremos de los grupos de corte (camino) mínimos (Chu y Apostolakis, 1980). Alesso y Benson (1980) definieron conceptos que permiten la descomposición de sistemas no coherentes en subsistemas coherentes.

2.7.5.- Métodos basados en Diagramas de Decisión Binarios

Introducción.

El reciente y gran interés por los Diagrama de Decisión Binarios radica en la posibilidad de evaluar el suceso top utilizando fórmulas implícitas, camino novedoso y diametralmente opuesto a todos los seguidos hasta ahora que se basaban en la evaluación explícita. Aunque los métodos implícitos se aplican a los grupos de Primeros Implicantes (PIs) se ha preferido no incluirlos en el apartado dedicado a los métodos de evaluación basados en ellos, para remarcar de ésta forma la diferencia drástica en su concepción. La formulación implícita evita el crecimiento exponencial del número de Primeros Implicantes para el cual el único remedio era el truncamiento ya sea por tamaño o por probabilidad de los mismos, con la consiguiente pérdida de precisión (error de truncamiento). La obtención de los DDBs y el análisis de la problemática selección del orden de las variables ya fue abordada en el apartado 2.6.4, por tanto nos limitaremos en lo que sigue a la definición de los productos implícitos y su empleo para cuantificar un árbol de fallos.

Cuantificación de los DDBs. Metaproductos

Los metaproductos han sido definidos y utilizados por Coudert et al (1993) para la implantación de los cálculos implícitos sobre los DDBs obtenidos del árbol de fallos. Su formulación será como sigue:

$$\phi(X_1, X_2, \dots, X_n) = \text{Función de estructura de un árbol de fallos.}$$

el conjunto de productos P_n está definido por:

$$\{l_1 \cap \dots \cap l_n / l_k \in \{X_k, \bar{X}_{k,1}\} \text{ para } 1 \leq k \leq n\}$$

$$l_1 \cap \dots \cap l_n \equiv l_1 \dots l_n$$

el orden de un producto es el número de los literales (X_i) que ocurren en el $P \in P_n$ es un implicante de ϕ si sólo si $S'_p \leq S_f$ es un primer implicante de ϕ y no existe otro implicante q de ϕ tal que $S_p \subset S_q$.

Existen 3^n elementos en el conjunto P_n , por tanto es suficiente un espacio Booleano de dimensión $n \lg_2(3)$ para definir todos sus elementos de forma no ambigua, sin embargo se usará uno de dimensión 2^n (mayor de lo necesario) pero que permite un manejo más adecuado en la implementación y facilidad del análisis.

Sea un conjunto σ definido en P_n desde $\{0,1\}^n \times \{0,1\}^n$ de la forma

$$\sigma([o_1, \dots, o_n], [S_1, \dots, S_n]) = l_1 \dots l_n$$

con

$$l_k = \begin{cases} 1 - si - o_k = 0 \\ \bar{X}_k - si - o_k = 1 - y - S_k = 0 \\ X_k - si - o_k = 1 - y - S_k = 1 \end{cases}$$

El Metaproducto P del conjunto de productos p es la función característica del conjunto $\left(U_{p \in P} \sigma^{-1}(p) \right)$. Dado que el conjunto $\{\sigma^{-1}(p) / p \in P_n\}$ es una partición de $\{0,1\}^n \times \{0,1\}^n$, se puede afirmar que los metaproductos son representaciones canónicas de los subconjuntos de P_n y el conjunto de operaciones sobre los conjuntos de productos corresponden a operaciones lógicas entre los metaproductos. Cada conjunto de productos P (y su metaproducto P) es la unión de tres subconjuntos disjuntos:

a) Elementos de P en los que la variable X_k no ocurre, cuyo metaproducto es

$$\bar{o}_k \cap P_{\bar{o}_k}$$

b) Elementos de P en los que la variable \bar{X}_k ocurre, cuyo metaproducto es

$$o_k \cap \bar{S}_k \cap P_{o_k, \bar{S}_k}$$

c) Elementos de P en los cuales la variable X_k ocurre, cuyo metaproducto es

$$o_k \cap \bar{S}_k \cap P_{o_k, S_k}$$

La participación del conjunto de productos P en $P_{\bar{o}_k}, P_{o_k, \bar{S}_k}, P_{o_k, S_k}$ hace que muchas operaciones sobre los conjuntos de productos sean lineales con respecto al Diagrama de Decisión Binario de los metaproductos de esos conjuntos, si estos son construidos según el orden de las variables definido por

$$O_{\pi(1)} < S_{\pi(1)} < O_{\pi(2)} < S_{\pi(2)} < \dots < O_{\pi(n)} < S_{\pi(n)}$$

donde π es una permutación de los enteros $\{1, \dots, n\}$

El cálculo de los Metaproductos correspondientes al conjunto de los Primeros Implicantes de la función ϕ se basa en el siguiente teorema.

Teorema

Sea ϕ una función Booleana de $\{0,1\}^n$ en $\{0,1\}$ una de cuyas variables es x_k . El Metaproducto $\text{Prime}(\phi)$ del conjunto de primeros implicantes de ϕ es:

$$\text{Prime}(\phi) = \left(\bar{o}_k \wedge \text{Prime}(\phi_{\bar{x}_k} \wedge \phi_{x_k}) \right) \vee \left(o_k \wedge \bar{s}_k \text{Prime}(\phi_{\bar{x}_k}) \wedge \neg \text{Prime}(\phi_{\bar{x}_k} \wedge \phi_{x_k}) \right) \vee \left(o_k \wedge \bar{s}_k \wedge \text{Prime}(\phi_{x_k}) \wedge \neg \text{Prime}(\phi_{\bar{x}_k} \wedge \phi_{x_k}) \right)$$

Este teorema muestra que el coste de cálculo del DDB del metaproducto $\text{Prime}(\phi)$ a partir del DDB de los metaproductos $\text{Prime}(\phi_{\bar{x}_k})$, $\text{Prime}(\phi_{x_k})$ y $\text{Prime}(\phi_{\bar{x}_k} \wedge \phi_{x_k})$ es polinomial con respecto al tamaño de dichos DDBs.

Aportaciones Significativas

Lo reciente de estos desarrollos hace que no sean muchas las aportaciones prácticas presentadas, existan hasta el momento dos paquetes; Metaprime (Coudert et al 1993) y Aralia (Aralia Group, 1995) que han servido para demostrar la gran potencia de estos métodos y para despertar un enorme interés por su aplicación y por los Diagramas de Decisión Binarios.

El talón de Aquiles de ésta metodología sigue siendo la carencia de criterios matemáticamente demostrados y computacionalmente eficientes para elegir el orden de las variables idóneo que conduce desde el árbol de fallos hasta el Diagrama de Decisión Binario más simple posible.

Los árboles tratados han sido tanto coherentes como no-coherentes y algunos no habían podido ser tratados por vía analítica exacta nunca antes.

2.7.6.-Teoría Cinética del Árbol

Esta teoría fue presentada por Vesely (1969) y parte de dos supuestos básicos:

- 1) Los sucesos básicos del árbol son independientes entre si
- 2) Se conocen los Grupos de Corte Mínimos

Con los dos supuestos anteriores mas las probabilidades de fallo y reparación este método permite obtener una información muy completa para los sucesos básicos, los grupos de corte y el suceso final no deseado.

Información sobre los sucesos básicos

Sean los siguientes:

$Z^P(t)dt$ = La probabilidad de que el fallo ocurra en el intervalo desde t a $t + dt$ y que el fallo no existe en el tiempo t

$u^P(t)dt$ = La probabilidad de que el fallo sea reparado en el intervalo desde t a $t + dt$ y dado que el fallo existe en el tiempo t

Los dos datos definidos son conocidos para todos los sucesos básicos del árbol. A partir de ellos se pueden deducir otros datos de interés. La probabilidad de que el primer fallo en un suceso básico ocurra en un determinado intervalo $t, t + dt$ suponiendo que no se ha producido en el tiempo t es:

$$a(t', t)dt = \exp\left(-\int_{t'}^t Z^P(t'')dt''\right)Z^P(t)dt \quad t' \leq t$$

En lo que respecta a la reparación, la probabilidad de que un fallo en un suceso básico (fallo primario) sea reparado en un tiempo $t, t + dt$ suponiendo que existe fallo en el instante t es:

$$b(t', t)dt = \exp\left(-\int_{t'}^t u^P(t'')dt''\right)u^P(t)dt \quad t' \leq t$$

También es muy importante en todo estudio de Fiabilidad la función de densidad de fallos $f^P(t)$ que se define como:

$f^P(t)$ = El numero esperado de veces que un fallo primario ocurre hasta un tiempo t por unidad de tiempo

Se puede obtener una ecuación para $f^P(t)$ en función de los datos sobre los fallos primarios tal como:

$$f^P(t) = \alpha(0, t) + \int_0^t f^P(t'')dt'' \int_{t''}^t b(t'', t')\alpha(t', t)dt'$$

El primer termino del lado derecho de la ecuación es la contribución al valor de $f^P(t)$ de la primera ocurrencia de un fallo primario, el segundo factor es la contribución a $f^P(t)$ de los fallos que ocurren hasta t'' siendo reparados hasta t' , y entonces volviendo a ocurrir hasta t. En el caso en que los fallos sean no-reparables, por ejemplo $b(t', t)=0$ la ecuación anterior se convierte en:

$$f^P(t) = \alpha(0, t) \quad \text{fallo primario no reparable}$$

Estas ecuaciones se pueden resolver por Métodos Numéricos o por Transformadas de Laplace.

Otra característica importante a conocer es la probabilidad de existencia de un fallo primario $F^P(t)$ definida como:

$$F^P(t) = \text{Probabilidad de que exista un fallo primario a un tiempo t}$$

La probabilidad de no existencia de fallo primario a un tiempo t es $1 - F^P(t)$, y de las anteriores definiciones se desprende que:

$$Z^P(t) = f^P(t) / (1 - F^P(t))$$

ó

$$F^P(t) = 1 - f^P(t) / Z^P(t)$$

Las cantidades $f^P(t)$ y $F^P(t)$ conjuntamente con las $Z^P(t)$, $u^P(t)$, $a(t',t)$ y $b(t',t)$ son las que definen la evolución probabilística de los fallos primarios a lo largo de todo el tiempo, obtenidas de cada fallo primario son importantes en si mismas ya que muestran los efectos de la reparación, mantenimiento y cambios en el medio ambiente de dichos sucesos, mostrando dichos efectos en función del tiempo y además permiten obtener las características en términos probabilistas de los grupos de corte y el suceso final.

Información sobre los Grupos de Corte Mínimos

Tal y como se ha definido un Grupo de Corte Mínimo es una combinación mínima de sucesos básicos (Fallos Primarios) que conducen a la aparición del suceso final no deseado. Consideremos un Grupo de Corte Mínimo compuesto de n fallos primarios que serán indicados con subíndices de 1 a n , en tal situación podemos definir:

$F^m(t)$ = Probabilidad de que exista el Grupo de Corte Mínimo al tiempo t

El Grupo de Corte Mínimo existirá en el tiempo t si y solo si existen todos los fallos primarios que lo componen en el tiempo t , o sea:

$$F^m(t) = \prod_{j=1}^n F_j^P(t)$$

donde $F_j^p(t)$ es la probabilidad de existencia del fallo primario j del Grupo de Corte Mínimo. La probabilidad de no- existencia del Grupo de Corte Mínimo será por tanto $-F^m(t)$ y es la probabilidad de que el Grupo de Corte Mínimo no exista al tiempo t . Por otra parte también podemos definir:

$f^m(t)$ = Numero esperado de veces que el Grupo de Corte Mínimo ocurrirá en el tiempo t por unidad de tiempo

Para que se produzca el Grupo de Corte Mínimo, uno o mas de los sucesos básicos (fallos primarios) que lo componen deberá no existir al tiempo t , el conjunto de todos estos fallos primarios no existentes en t deberán ocurrir en el intervalo $(t, t+dt)$. Es valido despreciar ordenes iguales o superiores a dos, por lo que resulta:

$$f^m(t) = \sum_{j=1}^n f_j^p(t) \prod_{i=1, i \neq j}^n F_i^p(t)$$

También podemos definir:

$Z^m(t)$ = Probabilidad de que la ocurrencia de los fallos del Grupo de Corte Mínimo en el intervalo $(t, t+dt)$ si no existen en el tiempo t

Por supuesto, el riesgo del Grupo de Corte Mínimo se convierte en:

$$Z^m(t) = \frac{\sum_{j=1}^n f_j^p(t) \prod_{i=1, i \neq j}^n F_i^p(t)}{1 - F^m(t)}$$

Todas estas características de los Grupos de Corte Mínimos son calculables a partir de las características de los sucesos básicos (fallos primarios) que los componen.

Información sobre el suceso top

Suponiendo conocidos los N Grupos de Corte Mínimos existentes para un determinado árbol de fallos, así como que los sucesos básicos (fallos primarios) no han ocurrido en $t=0$, podemos definir:

$F^T(t)$ = Probabilidad de que el suceso final exista en el tiempo t

Por supuesto que el suceso final existirá si y solo si uno o mas de los Grupos de Corte Mínimos existe, o sea:

$$F^T(t) = \bigcup_{i=1}^N F_i^m(t)$$

Por expansión tenemos:

$$F^T(t) = \sum_{i=1}^N F_i^m(t) - \sum_{i=2}^N \sum_{j=1}^{i-1} \prod_{+i,j} F^P(t) + \dots + (-1)^{N-1} \prod_{+i,\dots,N} F^P(t)$$

Donde el símbolo de producto significa que:

$\prod_{+1,\dots,N}$ = Producto de los valores de sucesos básicos cuando dicho suceso ocurre al menos en uno de los Grupos de Corte Mínimos 1,.....,N

El valor exacto de $F^T(t)$ se puede determinar directamente usando la ecuación anterior. Con frecuencia los Árboles de fallos son demasiado grandes y el numero de Grupos de Corte también, por lo que el calculo se puede volver demasiado costoso en términos de tiempo de CPU . Existe una forma de proceder que permite obtener sucesivos valores de $F^T(t)$ cada vez mas ajustados al valor real, la cual es aplicable para valores de probabilidad de los sucesos básicos mucho menores que la unidad, lo que por otra parte constituye el caso mas frecuente a encontrar en la realidad de los sistemas físicos de interés.

$$F^T(t) \leq \sum_{i=1}^N F_i^m(t)$$

$$F^T(t) \geq \sum_{i=1}^N F_i^m(t) - \sum_{i=2}^N \sum_{j=2}^{i-1} \prod_{+i,j} F^p(t)$$

etc.

En aquellos casos en que nos baste con una aproximación simple pero fiable de $F^T(t)$, los extremos de probabilidad desarrollados por Easary y Proschan (1963) para estructuras coherentes son aplicables y se traducen en la siguiente ecuación:

$$F^T(t) \leq 1 - \prod_{i=1}^N (1 - F_i^m(t))$$

Esta ecuación estima un extremo superior, seguro y conservador del valor de $F^T(t)$. También se puede definir:

$f^T(t)$ = número esperado de veces que ocurrirá el suceso final no deseado al tiempo t por unidad de tiempo

Para que el suceso no deseado ocurra en el intervalo desde t hasta $t+dt$, los Grupos de Corte Mínimos no existirán en el tiempo t y al menos uno de ellos se producirá entre t y $t+dt$. Por consiguiente podemos definir la siguiente ecuación:

$$f^T(t)dt = f^{T(1)}(t)dt - f^{T(2)}(t)dt$$

La primera contribución a $f^T(t)dt$ es la ocurrencia de uno o más Grupos de Corte Mínimos. Aplicando la teoría de expansión probabilista a los dos términos de la ecuación anterior obtenemos:

$$f^{T(1)}(t) = \sum_{i=1}^N f_i^m(t) - \sum_{i=2}^N \sum_{j=1}^{i-1} f(t; i, j) \prod_{+i,j} F^p(t) + \sum_{i=1}^N \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} f(t; i, j, k) \prod_{+i,j,k} F^p(t) + \dots$$

donde $f(t;1,\dots,m)$ es la intensidad de fallos para un modo de fallo que tiene sus sucesos básicos (fallos primarios) que son miembros comunes a todos los Grupos de Corte Mínimos $1,\dots,m$.

siendo:

$\prod_{+1,\dots,m}$ = El producto de las cantidades de cada suceso básico que ocurre al menos una vez en el Grupo de Corte Mínimo $1,\dots,m$, pero no es un miembro común de todos ellos.

mientras

$$f^{T(2)}(t) = \sum_{i=1}^N f_B(t;i) - \sum_{i=2}^N \sum_{j=1}^{i-1} f_B(t;i,j) + \dots$$

donde

$$f_B(t;i_1,\dots,i_n) = \sum_{i'=1}^N f(t;i_1,\dots,i_n-i') \prod_{i_1 \dots i_n}^{i'} F^P(t) - \sum_{i'=2}^N \sum_{j'=1}^{i'-1} f(t;i_1 \dots i_n - i', j') \prod_{i_1 \dots i_n}^{i',j'} F^P(t)$$

siendo

$f(t;1,\dots,m-1,\dots,n)$ = la intensidad de fallos para un modo de fallo el cual tiene como sucesos básicos (fallos primarios) los fallos primarios comunes a todos los GCM $1,\dots,m$ borrados aquellos que tienen fallos primarios también en algunos de los GCM $1,\dots,n$, mientras que

$\prod_{1,\dots,m}^{1,\dots,n}$ = El producto solo de aquellas cantidades de los fallos primarios que son

miembros de uno de los GCM $1,\dots,n$ ó son miembros de los GCM $1,\dots,m$ pero no es miembro común de esos m GCM.

También como en casos anteriores en casos de un gran numero de GCM se pueden obtener estimados de un extremo superior, seguro y conservador utilizando para estimar $f^T(t)$:

$$f^T(t) \leq \sum_{i=1}^N f_i^m(t)$$

también definimos, de igual forma que anteriormente:

$Z^T(t)dt$ = Probabilidad de que ocurra el suceso final no deseado en el intervalo $t, t+dt$, si no existía en el tiempo t

siendo por definición:

$$Z^T(t) = f^T(t) / (1 - F^T(t))$$

también:

$$Z^T(t) < \frac{\sum_{i=1}^N f_i^m(t)}{\prod_{i=1}^N (1 - F_i^m(t))}$$

Fue implementada en los programas PREP-KITT (Vesely y Narum, 1970), escritos en Fortran IV para un IBM 360/75. El árbol puede contener elementos reparables y no reparables. Pueden tratar los problemas en una o varias FASES, en las cuales las proporciones de fallo y los tiempos de reparación pueden permanecer constantes o variar de una fase a otra. Un componente puede ser reparable en una fase y en otras no, así como emplear probabilidades dependientes del tiempo.



2.7.7.- Simulación del Árbol de Fallos

Introducción al método de Monte Carlo

Fue en 1944 cuando se denominó por tal nombre al método de Monte Carlo, a partir de ese año comenzó el agrupamiento y desarrollo sistemático de las hasta entonces dispersas técnicas de simulación cuya naturaleza común sentaba las bases necesarias para la nueva técnica. El primer uso de la técnica como herramienta de investigación ocurrió con la simulación directa de los problemas probabilísticos relacionados con la difusión aleatoria de los neutrones en los materiales fisibles para la construcción de la primera bomba atómica. Von Neumann y Ulam refinaron la simulación empleando técnicas de reducción de varianza, en particular las llamadas "Ruleta Rusa" y "Splitting" (Curtiss,1951), el desarrollo sistemático de estas ideas comenzó con Harris y Herman Kahn en 1948. La posibilidad de aplicar Monte Carlo a problemas deterministas fue comunicada por Fermi, Von Neumann y Ulam, siendo popularizadas por los mismos sus dichas aplicaciones en los años posteriores a la segunda guerra mundial. Durante la década de los 50 los diversos intentos de aplicación no contribuyeron, en general, a consolidar la técnica ya que no se alcanzaba mayor precisión que con los métodos deterministas. A partir de la década de los 60, sin embargo, se multiplicaron las aplicaciones con resultados admisibles, se perfeccionaron y ampliaron las técnicas de reducción de varianza, por todo lo cual desde hace bastantes años es una técnica muy popular y admitida entre la comunidad científica.

El método de Monte Carlo tiene carácter inferencial y se basa en la estadística. Los tipos de tareas que de forma general se realizan para la determinación de una característica de un problema que se desea estudiar, se agrupan en el siguiente procedimiento:

- 1.- Definir un problema a estudiar y determinar que variable(s) es necesario estudiar para resolverlo
- 2.- Desarrollar un modelo probabilístico del problema definido, obteniendo las funciones de densidad de probabilidad de las variables que intervienen.

3.- Obtener valores aleatorios de las variables del modelo, sustituirlos en el modelo y observar el valor que toma la variable a estudiar. Almacenar los valores de la variable a estudio hasta que se tenga un número suficiente de las mismas que permita concluir de forma estadísticamente aceptable cual es su ley de comportamiento y su valor más probable.

La realización correcta de los pasos anteriores requiere el conocimiento de una serie de conceptos y técnicas estadísticas de forma profunda, básicamente podemos pensar en que es necesario tener conocimientos de:

- *Sucesos aleatorios y Probabilidad*
- *Variabes Aleatorias, Distribuciones y Esperanzas*
- *Estimación y Métodos de Muestreo*
- *Eficiencia de estimadores*
- *Números aleatorios*
- *Análisis de la Varianza*

No esta dentro de los objetivos de esta memoria realizar un tratado sobre el método abarcando todas las áreas anteriores, del que existen además excelentes obras escritas, podemos citar entre otras Hammersley y Handscomb, (1964), McGrath e Irving (1975), Rubinstein (1981) y Kalos y Whitlock (1986). Los conceptos necesarios sobre estadística y probabilidad pueden aprenderse en Sobol (1994). Bibliografía sobre métodos de reducción de varianza, además de los ya mencionados, se aportará más adelante en esta memoria. Mención importante es la de la continua aportación de las Ciencias Físicas al desarrollo de métodos eficientes de reducción de varianza, conocimientos que luego son exportados a otras áreas del conocimiento, es imposible mencionar todas las aportaciones, baste como muestra Spanier y Gelbard (1969) como referencia clásica de aplicaciones al transporte de neutrones, o también Miller (1984) como excelentes e instructivas aportaciones. Muy interesantes avances se han producido recientemente en otras áreas entre las que hemos seleccionado la de iluminación artificial de imágenes generadas por ordenador, en la que buenos exponentes del nivel de desarrollo alcanzado son Ohbuchi y Aono (1996), Veach (1997).

Principios generales del método

La idea esencial del método es evaluar la integral

$$I = \int_{\Omega} f(x) d\mu(x)$$

usando muestreo aleatorio, lo cual en principio se realiza tomando N muestras independientes X_1, X_2, \dots, X_N de acuerdo con cierta función de densidad de probabilidad p , y a continuación calcular el estimado

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)} \quad (1)$$

en el que es importante destacar que el resultado es una variable aleatoria y que, por tanto, sus propiedades dependen del número de muestras tomadas. Este estimador fue usado primeramente en muestreo y es conocido como el estimador Horvitz-Thompson (Horvitz y Thompson, 1952) y a los procesos que lo emplean se les denomina frecuentemente *Crudos* o *Directos*.

Por ejemplo, supóngase que el dominio es $\Omega = [0,1]^s$ y que las muestras X_i son tomadas independientes y uniformemente distribuidas. En este caso el estimador (1) se reduce a

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N f(X_i)$$

que tiene la misma forma que una regla de Cuadratura excepto que la localización de las muestras es aleatoria. Es directo demostrar que el estimador \hat{I} da el resultado correcto en promedio, ya que su Esperanza Matemática es

$$E[\hat{I}] = E\left[\frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}\right] = \frac{1}{N} \sum_{i=1}^N \int_{\Omega} \frac{f(x)}{p(x)} p(x) d\mu(x) = \int_{\Omega} f(x) d\mu(x) = I$$

supuesto que $\frac{f(x)}{p(x)}$ es finito y $f(x) \neq 0$.

Las ventajas de la integración Monte Carlo son:

1. Converge según $O(N^{-1/2})$ en espacios de cualquier dimensión, sin tener en cuenta la regularidad o suavidad del integrando.
2. La integración es simple, ya que solo se requieren dos operaciones básicas como son: Muestreo y Evaluación Puntual.
3. Es general, lo cual se deriva del hecho de que se basa en muestreo aleatorio. El muestreo puede realizarse también sobre dominios que no tienen correspondencia natural con el $[0,1]^s$ y que, por tanto, no son adecuados para cuadratura numérica.
4. Son más adecuados que los métodos de Cuadratura para integrandos con singularidades. El Muestreo por Importancia se puede emplear para manejar estos integrandos de forma adecuada, incluyendo aquellos casos en que no existe transformación analítica para evitar la singularidad.

Razón de Convergencia

Consideremos, por simplicidad de la notación, que $Y_i = f(X_i) / p(X_i)$, con lo que el estimador (1) será

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N Y_i$$

sea también $Y = Y_i$. Por lo tanto tenemos que la varianza del estimador \hat{I} será:

$$V[Y] = E[Y^2] - E[Y]^2 = \int_{\Omega} \frac{f^2(x)}{p^2(x)} p(x) d\mu(x) - I^2 = \int_{\Omega} \frac{f^2(x)}{p(x)} d\mu(x) - I^2$$

asumiendo que esta cantidad es finita, se deduce la ley de decrecimiento de esta varianza

$$V[\hat{I}] = V\left[\frac{1}{N} \sum_{i=1}^N Y_i\right] = \frac{1}{N^2} V\left[\sum_{i=1}^N Y_i\right] = \frac{1}{N^2} \sum_{i=1}^N V[Y_i] = \frac{1}{N} V[Y] \quad (2)$$

donde se ha usado $V[\alpha Y] = \alpha^2 V[Y]$ y el hecho de que Y_i son muestras independientes. Con ello, la desviación típica será:

$$\sigma[\hat{I}] = \frac{1}{\sqrt{N}} \sigma[Y]$$

que nos muestra que la tasa de Convergencia es $O(N^{-1/2})$

Acotación del error cometido

Es posible obtener también cotas o extremos probabilísticos del error absoluto usando la desigualdad de Chebychev

$$\Pr \left\{ |F - E[F]| \geq \left(\frac{V[F]}{\delta} \right)^{1/2} \right\} \leq \delta$$

la cual es cierta siempre que, para la variable aleatoria F se cumpla que $V[F] < \infty$. Aplicando esta desigualdad a la varianza dada por la ec. (2) se obtiene:

$$\Pr \left\{ |\hat{I} - I| \geq N^{-1/2} \left(\frac{V[I]}{\delta} \right)^{1/2} \right\} \leq \delta$$

y por tanto, para un cierto umbral fijo δ , el error absoluto decrece con la proporción $O(N^{-1/2})$. Cotitas seguras del error absoluto se pueden obtener usando el Teorema central del Límite, que establece que \hat{I} converge a una distribución Normal en el límite $N \rightarrow \infty$.

$$\Pr \left\{ \frac{1}{N} \sum_{i=1}^N Y_i - E[Y] \leq t \frac{\sigma[Y]}{\sqrt{N}} \right\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-x^2/2} dx$$

donde la expresión del lado derecho de la ecuación es la función de Distribución de la Ley Normal. Reorganizando la ecuación obtenemos:

$$\Pr\{|\hat{I} - I| \leq t\sigma[\hat{I}]\} = \sqrt{\frac{2}{\pi}} \int_t^\infty e^{-x^2/2} dx$$

La integral del lado derecho decrece muy rápidamente con los valores de t . Por ejemplo, cuando $t=3$, el lado derecho vale aproximadamente 0.003. Por tanto existe una probabilidad del 0.3% de que \hat{I} difiera de su media en más de tres desviaciones estándar, supuesto que N es suficientemente grande para poder aplicar el Teorema Central del Límite.

Además la integración de Monte Carlo convergerá aunque la varianza $V[Y]$ sea infinita, supuesto que la esperanza $E[Y]$ exista (aunque dicha convergencia será lenta). Esto está garantizado por la Ley de Los Grandes Números:

$$\Pr\left\{\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N Y_i = E[Y]\right\} = 1$$

Estimadores

Para evaluar la integral de Monte Carlo ya se ha mencionado la técnica estándar basada en el estimador Horvitz-Thompson, ec. (1):

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}$$

En la actualidad existe una gran variedad de técnicas disponibles para obtener estimadores eficientes para simular muchos tipos de procesos, todas estas técnicas están comprendidas en el concepto de estimador de Monte Carlo. La característica común de todos los estimadores es que han sido derivados con objeto de reducir la varianza, por ello hemos dedicado un apartado posterior específico para mostrarlos, en dicho apartado se encuentra la figura 2.42 bis en la que se muestran algunos estimadores de interés así como la idea básica de reducción de varianza que los origina.

El propósito de un estimador de Monte Carlo es aproximar el valor de cierta magnitud de interés Q (también llamada *estimando*). Normalmente definiremos Q como el valor de una integral dada, aunque son posibles situaciones más generales (por ejemplo Q puede ser la razón de dos integrales). Un estimador es definido como una función de la forma:

$$\hat{I} = \hat{I}(X_1, X_2, \dots, X_N) \quad (3)$$

donde las X_i son variables aleatorias. Un valor numérico particular de \hat{I} es llamado un *estimado*, además las X_i no tienen que ser necesariamente independientes, y pueden tener diferentes distribuciones. En estadística cada X_i se denomina *observación*, el vector (X_1, X_2, \dots, X_N) se llama *muestra* y N se denomina *tamaño muestral*.

Propiedades de los estimadores

Entre las propiedades más importantes de los estimadores cabe citar:

- La cantidad $\hat{I} - Q$ se llama *error* y su valor esperado se denomina *sesgo*: *sesgo de \hat{I}* $= \beta[\hat{I}] = E[\hat{I} - Q]$
- Un estimador se denomina *no sesgado* si $\beta[\hat{I}] = 0$ para todos los tamaños muestrales N , o en otras palabras: $E[\hat{I}] = Q \quad \forall N \geq 1$

Por ejemplo la variable aleatoria (estimador) $\hat{I} = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}$ es un estimador no sesgado de la integral $I = \int_{\Omega} f(x) d\mu(x)$ porque $E[\hat{I}] = Q \Rightarrow \beta[\hat{I}] = E[\hat{I} - Q] = 0$

- Un estimador se denomina *consistente* si el error $\hat{I} - Q$ tiende a cero con probabilidad 1, o sea: $\Pr\left\{\lim_{N \rightarrow \infty} \hat{I} = Q\right\} = 1$. Para que un estimador sea consistente, una condición suficiente es que tanto el sesgo como la varianza tiendan a cero si N se incrementa $\lim_{N \rightarrow \infty} \beta[\hat{I}] = \lim_{N \rightarrow \infty} V[\hat{I}] = 0$. En particular, un estimador no sesgado es consistente mientras su varianza decrece si N tiende a infinito.

Error cometido por los estimadores no sesgados

La principal razón de preferir estimadores no sesgados frente a otros es que para estos es más fácil estimar el error. Típicamente nuestro objetivo es minimizar el error cuadrático medio ECM, definido por:

$$ECM[\hat{I}] = E\left[(\hat{I} - Q)^2\right]$$

que en general se puede convertir como sigue:

$$ECM[\hat{I}] = E\left[(\hat{I} - Q)^2\right] = E\left[(\hat{I} - E[\hat{I}])^2\right] + 2E[\hat{I} - E[\hat{I}]](E[\hat{I}] - Q) + (E[\hat{I}] - Q)^2 = V[\hat{I}] + \beta[\hat{I}]$$

pero en estimadores no sesgados también se cumple que $\beta[\hat{I}] = 0$, y por tanto

$$ECM[\hat{I}] = V[\hat{I}] = E\left[(\hat{I} - E[\hat{I}])^2\right]$$

o sea que el error cuadrático medio de un estimador no sesgado es idéntico a la varianza. Lo cual simplifica la obtención de estimados del error, simplemente tomando muchas muestras independientes. Sean Y_1, Y_2, \dots, Y_N muestras independientes de un estimador no sesgado Y , y sea:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N Y_i$$

según se ha demostrado (Kalos y Whitlock, 1986) la varianza muestral se puede estimar mediante el estimador no sesgado:

$$V[\hat{I}] = \frac{1}{N-1} \left\{ \left(\frac{1}{N} \sum_{i=1}^N Y_i^2 \right) - \left(\frac{1}{N} \sum_{i=1}^N Y_i \right)^2 \right\}$$

lo que permite estimar el error. Por otro lado para disminuir el error de estos estimadores basta con incrementar el número de muestras independientes.

Eficiencia de un estimador

Con frecuencia, incrementar la simulación en N muestras significativas incrementará en el mismo factor el tiempo de CPU necesario lo cual no suele ser aceptable. De forma ideal nos gustaría encontrar estimadores para los cuales tanto la varianza como el tiempo de ejecución sean pequeños. Este equilibrio está contemplado en la denominada *eficiencia* absoluta de un estimador de Monte Carlo:

$$e[\hat{I}] = \frac{1}{V[\hat{I}]T[\hat{I}]} \quad T[\hat{I}] \equiv \text{Tiempo de CPU}$$

por lo que un estimador es tanto más eficiente cuanto menor es su varianza para un número fijo de iteraciones. Si se comparan dos métodos de reducción de varianza, la eficiencia relativa de uno frente a otro vendrá dada por:

$$e = \frac{V(\hat{I}_1)T(\hat{I}_1)}{V(\hat{I}_2)T(\hat{I}_2)}$$

es frecuente que como primer estimado se adopte el estándar de Horvitz-Thompson.

Métodos de reducción de varianza

Como su nombre indica, las técnicas de reducción de varianza se emplean para aumentar la precisión de la estimación de parámetros por el método de Monte Carlo. El método de referencia con el cual se contrastan las técnicas de reducción de varianza, ya sean aplicadas de forma individual ó combinadas entre si, es el Monte Carlo Crudo. Por otro lado se puede considerar que las técnicas de reducción de varianza intentan incrementar la efectividad de la simulación por medio de alguno de los siguientes sistemas:

- *Modificando el procedimiento de simulación*
- *Utilizando aproximaciones ó información analítica*
- *Estudiando el sistema en un contexto diferente ó representación abstracta .*

Basándose en estos criterios se presenta en la figura siguiente una clasificación de las distintas técnicas (McGrath & Irving, 1975), (Veach, 1997). Se agrupan según relaciones más que por diferencias ya que es extremadamente complejo encontrar clasificaciones que expresen todas las características de forma diferenciada.

MODIFICACIÓN DEL SISTEMA DE MUESTREO	Muestreo por Importancia	Clark 1961, McGrath e Irving 1975 Bavuso y Hampton 1997, Gray et al. 1998, Smith 1997 Keramat y Kilbasa 1998 Veach y Guibas 1995
	Ruleta Rusa y Splitting	Hammersley y Handscomb, 1964 Veach 1997
	Muestreo Sistemático	Ehrenfeld y Ben-Tuvia 1962 Lassila y Virtamo 1998
	Muestreo Estratificado	Bracken et at 1968, Mitchell, 1996 Keramat y Kilbasa 1997
	Hipercubo Latino	McKay et al. 1979, Stein 1987 Keramat y Kilbasa 1997
	Métodos Casi Monte Carlo	Niederreiter 1992, Owen 1994 y 1997
	Muestreo con Arreglos Otogonales	Owen 1992, Tang 1993
EQUIVALENCIA ANALÍTICA	Valores Esperados	Kahn y Marshall 1953
	Estimación Estadística	Kahn 1954
	Muestreo Correlacionado	Evans 1963
	Reanálisis Histórico	Kahn y Marshall 1953
	Variables de Control	Gaver 1969
	Variable Antitética	Fishman 1968
	Uso de Regresión	Ehrenfeld y Ben-Tuvia 1962
TÉCNICAS ESPECIALIZADAS	Muestreo Secuencial o Adaptativo	Hamm. y Han. 1964, Kirk y Arvo 1991 Billington y Jonnavithula 1997
	Formulación adjunta	Hartley y Rao 1967
	Transformaciones	Clark 1966
	Funciones Ortonormales	Hammersley y Handscomb, 1964
	Monte Carlo Condicional	Burt y Garman 1971

Fig. 2.42.- CLASIFICACIÓN DE LAS TÉCNICAS DE REDUCCIÓN DE VARIANZA

Cuando se emplean técnicas de modificación del proceso de muestreo se suelen utilizar estrategias de muestreo más efectivas ó se alteran las distribuciones de las que se muestrea. Si se emplean técnicas de equivalencia analítica es porque alguna parte del proceso simulado puede tener una resolución analítica completa y conocida ó porque todo el proceso, ó parte de él, puede tener una solución analítica aproximada, en ambos casos se aprovecha dicha circunstancia para "conducir" la simulación obligando a que se cumplan las soluciones conocidas ó que este cercana a la solución analítica aproximada. Cuando se emplean técnicas especializadas lo que se puede encontrar son poderosas combinaciones de todas las anteriores que suelen conducir a soluciones muy eficaces de problemas complejos. La clave para la correcta elección del método de reducción de varianza a emplear es tener perfectamente claro que:

La Reducción de Varianza es un método de usar lo que se sabe acerca del modelo (sistema) que se está simulando para aumentar la eficiencia del proceso de simulación.

El caso más común es conocer información cualitativa y/o cuantitativa, y tener que usarla de forma explícita y cuantitativa. Es importante resaltar que, por tanto, si no se conoce nada acerca del proceso que se simula no es posible obtener una reducción de varianza. La elección del método de reducción de varianza depende pues del proceso que se esté simulando, requiere cierto entrenamiento y experiencia por parte del investigador para su eficaz utilización y, por supuesto, ciertas dosis de "inspiración creadora". La búsqueda del método apropiado para el proceso a estudiar, fuerza al investigador a responder a dos cuestiones básicas:

- *¿ Qué respuestas debe proporcionar la simulación ?*
- *¿ Qué se conoce acerca del comportamiento del proceso ?*

La definición exacta del problema es, por tanto, de importancia capital y antes de seleccionar el método de reducción de varianza se deben poner de manifiesto todos aquellos aspectos, ya mencionados, que permiten una buena elección. En todo caso, y como mínimo, se deberá tener muy claro lo siguiente:

- *Lista de todos los parámetros que se estimarán por simulación*
- *Relación de lo conocido sobre la marcha interna del proceso*

NOMBRE	IDEA	ESTIMADOR
Uso de Valores esperados o Reducción de Dimensionalidad	Integrando analíticamente con respecto a una o más variables del dominio	$\hat{I}' = E_y \left[\frac{f(X, Y)}{p(X, Y)} \right] =$ $= \int \frac{f(X, y)}{p(X, y)} p(y X) dy$
Muestreo por Importancia	Encontrando una función de densidad p similar al integrando f	$p(x) = cf(x)$ $c = \frac{1}{\int_{\Omega} f(y) d\mu(y)}$ $\hat{I} = \frac{f(X)}{p(X)} = \frac{1}{c}$
Muestreo estratificado	Dividir el dominio en subdominios no solapados entre si (estratos) Tomar un número de muestras fijo en cada estrato	$\Omega = [0,1]^s \quad p_i = k \text{ en } \Omega_i$ $\hat{I}' = \sum_{i=1}^n v_i F_i$ $F_i = \frac{1}{n_i} \sum_{j=1}^{n_i} f(X_{i,j})$
Hipercubo Latino	Si se van a tomar N muestras, subdividir el dominio en N subintervalos en cada dimensión y tomar una muestra por intervalo	$s \equiv \text{permut.}(\pi_1, \dots, \pi_s) \text{ de } \{1, \dots, N\}$ $\text{posic. de muestra} \rightarrow X_i^j = \frac{\pi_j(i) - U_{i,j}}{N}$ $X_i^j = \text{Coord. } j \text{ de la muestra } X_i$ $U_i^j = \text{Aleatorios independientes en } [0,1]$
Variable de Control	Encontrar una función g similar al integrando y que pueda ser integrada analíticamente	<p>La integral se define como</p> $I = \int_{\Omega} g(x) d\mu(x) + \int_{\Omega} f(x) - g(x) d\mu(x)$ <p>y se usa el estimador</p> $\hat{I} = \int_{\Omega} g(x) d\mu(x) + \frac{1}{N} \sum_{i=1}^N \frac{f(X_i) - g(X_i)}{p(X_i)}$

© Universidad de Las Palmas de Gran Canaria. Biblioteca Digital. 2003

Fig. 2.42 bis.- ALGUNOS MÉTODOS DE REDUCCIÓN DE VARIANZA Y SUS ESTIMADORES

Uso de Monte Carlo en Fiabilidad y Árbol de Fallos

El método de Monte Carlo se suele emplear para analizar la fiabilidad de sistemas grandes o complejos, en los cuales el empleo de los métodos deterministas es muy costoso o imposible. En la mayor parte de los casos es necesario encontrar la combinación adecuada con un método de reducción de varianza afín a la formulación del problema estudiado para optimizar el tiempo- costo de la simulación.

En el análisis de Árboles de Fallos el procedimiento de simulación por Monte Carlo consiste en realizar múltiples ensayos (historias) de los sucesos básicos observando si ocurren o no, en el primer caso se asignan los tiempos de reparación apropiados, se van anotando las veces que ocurre el suceso final no deseado (top) y se calcula su probabilidad comparando los casos favorables de ocurrencia con los casos posibles (ensayos totales). El valor obtenido es una estimación que será tanto mas aproximada al valor real como se incrementa el número de ensayos. Otras informaciones que se pueden obtener de esta aproximación son: la probabilidad de todos los sucesos intermedios del árbol y los grupos de corte que intervienen en la ocurrencia del suceso final.

Formulación general del problema

Si se define un espacio de muestra S_R por medio de:

$$S_R = \{R = (R_1, \dots, R_m): R_n\}$$

siendo S_R es una variable aleatoria uniformemente distribuida en el intervalo (0,1) y n es fija y finita. Es decir, S_R es el espacio de todos los vectores de longitud n, cuyos componentes son números aleatorios

Todos los puntos r de S_R están relacionados con una función de densidad de probabilidad de forma unívoca. Sea Y la variable aleatoria definida implícitamente en S_R , mediante un programa de computadora. Esto significa que dado un vector de números aleatorios $R = r$, el programa de computadora transformara el vector aleatorio en un valor de la variable aleatoria Y. O sea:

dado un $n \Rightarrow \forall$ vector $R = r \Rightarrow \exists$ un valor de $Y \subset S_R$

Para generalizar mas, conviene que esta transformación dependa de un vector de parámetros de entrada (esto es, $Y = Y(r; \gamma)$). Para este caso, el programa de computadora contendrá un modelo lógico (función de estructura) de un árbol de fallos.

Y puede ser vectorial o escalar, pero este estudio se restringe al caso escalar, Y será el suceso final del árbol de fallos.

El objetivo de la simulación es estimar el valor de $Y(E(Y))$ dado por:

$$E(Y) = \int_0^1 \dots \int_0^1 Y(r_i) f(r_i) dr \quad \forall_i \exists f.d.p.(r_i)$$

El procedimiento que se sigue normalmente para estimar $E(Y)$ consiste en generar una muestra de j vectores (compuestos de n números aleatorios cada uno), calcular el valor de Y para cada uno y obtener la media de estas j observaciones como valor estimado de Y

$$E(y) = \frac{\sum_{i=1}^j Y_i}{j}$$

Muy pocas veces se conoce Y de manera explícita, pero si se tiene cierta libertad al especificar su forma y, se reconoce y aprovecha, se puede reducir al varianza y, por tanto, mejorar el rendimiento de Monte Carlo.

Proceso general de simulación del árbol de fallos

El proceso general de simulación del árbol de fallos fue ya introducido tempranamente, Haals (1965), por la literatura especializada. Los trabajos posteriores se han dedicado mas a la optimización en cuanto a tiempo de ordenador y precisión de la estimación que a modificar el proceso de simulación en si. Consta de: Generador de Fallos, Generador de Tiempos de Reparación, Controlador del árbol de fallos y programa de Evaluación de Probabilidades, cuyas características esenciales se describen en los siguientes apartados.

Generador de Fallos

El generador de fallos da como salida un vector de estado que contiene el estado de cada suceso básico independiente de entrada al árbol de fallos. El estado de cada suceso básico se genera aleatoriamente de acuerdo con una probabilidad y una función de distribución característica de cada suceso. El procedimiento de generación es el siguiente: Sea un suceso básico independiente con una probabilidad $P(A)$ y una función de densidad de probabilidad $f_A(t)$. Se resuelve la ecuación

$$X = \int_0^Y f_A(t) dt \qquad \int_0^{\infty} f_A(t) dt = 1$$

donde:

X = número aleatorio de distribución uniforme entre 0 y 1 (dato)

Y = número aleatorio que se desea encontrar (incógnita) para $f_A(t)$

$f_A(t)$ conocida y acotada (dato)

si

$Y \geq P(A) \Rightarrow$ el suceso A ocurre

$Y < P(A) \Rightarrow$ el suceso A no ocurre

Generalmente cuando el suceso (fallo) ocurre se representa con un " 1 " y cuando no ocurre con un " 0 ". Así pues el vector de estado contendrá un símbolo (0 o 1) por cada suceso básico de entrada al árbol de fallos

Generador de tiempos de reparación

Cuando tiene lugar un suceso básico (fallo)(presenta un 1 en el vector de estado), se genera un tiempo de reparación asociado a dicho suceso mediante el generador de tiempos de reparación. Para ello es necesario que cada fallo tenga asociado una función de densidad de probabilidad del tiempo de su reparación. El proceso de generación será análogo al de generación de estados del apartado anterior. Sea:

$f_{RA}(t)$ = función de densidad de probabilidad de los tiempos de reparación del suceso A

se resuelve la ecuación

$$X = \int_0^{Y_R} f_{RA}(t) dt \qquad \int_0^{\infty} f_{RA}(t) dt = 1$$

donde:

X = número aleatorio uniformemente distribuido entre 0 y 1 (dato)

Y_R = tiempo de reparación aleatorio buscado (incógnita)

$f_{RA}(t)$ conocida y acotada (dato)

Para cada suceso intermedio del árbol de fallos deberá generarse aleatoriamente un tiempo de reparación en caso de presentarse

Controlador del Árbol de Fallos

En el controlador del árbol de fallos, los fallos de entrada inician las secuencias de fallos tal como son descritas por el propio árbol. Las secuencias de fallos siguen a través del árbol y aparecen fallos secundarios para los cuales también se generaran tiempos de reparación, de esta manera que la cadena de fallos continua hasta que todas las averías son reparadas o aparece el suceso final no deseado que estudia el árbol

El árbol puede resolverse también sin considerar tiempos de reparación, lo cual conduce, como se ha demostrado (Yellman, 1981), a resultados mas optimistas que los reales, excepto si se trata de calcular la No Disponibilidad del Sistema con probabilidades de los sucesos básicos estimadas en un horizonte temporal homogéneo.

Programa de evaluación de probabilidades

Los resultados que aparecen a la salida del controlador del árbol de fallos son acumulados por el programa d evaluación de probabilidades, el cual, una vez concluida la simulación, estima las probabilidades de los sucesos en base a los resultados estadísticos obtenidos.

Reducción de Varianza en simulación de Árboles de Fallos

En la evaluación de Árboles de fallos se han empleado diversos métodos con distinta efectividad. En el cuadro siguiente se presenta una selección de aportaciones, su descripción y referencias.

REFERENCIA	TIPO DE REDUCCIÓN DE VARIANZA
Van Slyke y Frank, 1972	Muestreo Estratificado
Kumamoto et al., 1977	Variable de Control
Boyd y Bavuso, 1993	Muestreo por Importancia
Manian y Dugan, 1998	"
Easton y Wong, 1980	Dstrucción Secuencial
Kumamoto et al., 1980	Variable Antitética (Daga)
Altschule & Nagel,	Muestreo por Importancia Power Rule
Karp y Luby, 1983	Intervalo Previo
Fishman, 1986	Comparación de cuatro métodos
Kumamoto et al, 1987	Intervalo Previo
Doreste y Galván, 1987	Variable Antitética (Daga Generalizado)
Kumamoto & Henley, 1993	Importancia y Variable Antitética

Fig 2.43.- REDUCCIONES DE VARIANZA EMPLEADAS EN SIMULACIÓN DE ÁRBOLES DE FALLOS.

Muestreo Restringido

Sean ϕ_L y ϕ_U dos funciones binarias que satisfacen las ecuaciones

$$\begin{aligned}\phi_L(b) &\leq \phi(b) \leq \phi_U(b) \\ \phi_L(b) &\neq 0, \phi_U(b) \neq 1, \forall b\end{aligned}$$

donde

$$b = (b_1, \dots, b_K) \equiv \text{vector binario}$$

$$\phi = \text{Función de estructura}$$

$K = n^\circ$ de sucesos básicos del árbol de fallos

Si se quiere evaluar la no disponibilidad Q_S un estimador no sesgado de la misma es \hat{Q}_R (R= restringido):

$$\hat{Q}_R = [Q_U - Q_L] N^{-1} \sum_{v=1}^N \phi(S_v) + Q_L \quad (1)$$

Donde Q_U, Q_L son las no disponibilidades de ϕ_U, ϕ_L respectivamente y las muestras aleatorias S_1, \dots, S_N son generadas usando las nuevas probabilidades asignadas a los sucesos básicos mediante

$$\Pr(y_i = b_i / y_1 = b_1, \dots, y_{i-1} = b_{i-1}) = \frac{Q_{U,i}(b_1, \dots, b_i) - Q_{L,i}(b_1, \dots, b_i)}{Q_{U,i-1}(b_1, \dots, b_{i-1}) - Q_{L,i-1}(b_1, \dots, b_{i-1})}$$

La varianza del estimador (1) es:

$$\text{Var}(\hat{Q}_R) = N^{-1} (Q_U - Q_S)(Q_S - Q_L)$$

y es menor que la obtenida por el muestreo directo.

Muestreo Daga

El muestreo Daga utiliza correlación negativa entre las muestras para obtener estimaciones de $E(\phi(Y))$ con varianza más pequeña, al mismo tiempo es muy eficiente porque con un sólo número aleatorio genera múltiples muestras. El estimador del muestreo Daga es:

$$\hat{Q}_D = N^{-1} \sum_{v=1}^N \phi(Z_v)$$

$Z_1, \dots, Z_N = N$ muestras generadas por muestreo daga

La varianza de este estimador es:

$$\text{Var}(\hat{Q}_D) = \text{Var}(\hat{Q}_C) + \sum_{i \neq j} \text{Cov}(\phi(Z_i), \phi(Z_j)) \quad (2)$$

$$\hat{Q}_C = \text{estimador del Monte Carlo Directo o Crudo} = N^{-1} \sum_{v=1}^N \phi(C_v)$$

Si la correlación entre las muestras es negativa entonces el segundo miembro de (2) es negativo y por consiguiente

$$\text{Var}(\hat{Q}_D) \leq \text{Var}(\hat{Q}_C)$$

Destrucción Secuencial

Easton y Wong (1980) introdujeron el método de Destrucción (ó Construcción) secuencial para obtener tanto reducción de la varianza como ahorro de tiempo de cálculo. Este método no requiere análisis topológico y es aplicable incluso cuando las probabilidades son pequeñas, pueden ser aplicado a una amplia gama de medidas de fiabilidad y no es necesario que los fallos de los componentes sean estadísticamente independientes.

La probabilidad del suceso top se estima utilizando

$$E\{\phi\} = \sum_{i=1}^{n!} \left[\sum_{j=0}^n \phi(\pi_i; j) \Pr(\pi_i; j) \right]$$

$$\Pr\{\pi_i; j\} = \Pr\{S_K\} / [(n-j)! j!]$$

$S_i \equiv n$ -upla representando el estado del sistema

$n \equiv$ número de elementos del sistema

La Destrucción Secuencial se basa en definir un nuevo espacio de muestreo a partir del espacio original usando un estado S_K con probabilidad $\Pr(S_K)$. Si j es el número de elementos con valor "0" en S_K y $(n-j)$ el de elementos de valor "1", se construirá una permutación aleatoria $\{1, \dots, n\} \equiv \pi_i$ uniendo una permutación aleatoria de cada uno de esos subconjuntos de estados y ello corresponderá la primera salida (resultado) siendo j el segundo. El conjunto de posibles salidas se identificará por $\{\pi_i; j\}$. Este muestreo posee una menor varianza que el Monte Carlo directo o Crudo.

Programas de ordenador

Han sido desarrollados muchos programas que permiten el estudio de Árboles de fallos por el método de Monte Carlo. En el siguiente cuadro se nombran los mas conocidos y se menciona su procedimiento de trabajo así como su referencia bibliográfica.

PROGRAMA	AUTOR(ES)
SAFTE	Garrick & Geklar, (1967)
SAMPLE	WASH 1400, (1975)
STADIC	Cairns & Fleming, (1977)
MOCARS	Matthews, (1977)
RALLY	German Risk Study,(1978)
PATREC-MC	Blin et al.,(1978)
STADIC-II	Orvic & Frank, (1981)
PROSA-2	Vaurio, (1981)
RELY 4	Kongsoe, 1972
DIFTREE	Dugan et al. 1997
REDIS	Kongsoe, 1975

Fig. 2.44.- PROGRAMAS QUE USAN MONTE CARLO PARA EVALUAR ÁRBOLES DE FALLOS

2.7.8.- Importancia

La importancia de un componente del sistema o de un grupo de corte es una medida de su contribución a la aparición del suceso top del árbol de fallos (Lee et al., 1997). El análisis de importancia es, por tanto, útil para el diseño, diagnóstico y la optimización de un sistema técnico. También la Inspección, el Mantenimiento y la Detección de fallos puede llevarse a cabo siguiendo el orden de prioridades sugerido por la importancia de los elementos y por supuesto, una forma de evaluar los efectos de la incertidumbre en los datos de los componentes es observar como varían las medidas de importancia al variar los datos.

Medidas de Importancia

Las medidas de importancia se pueden referir tanto a los componentes como a los Grupos de Corte y han de producir entre otras, las siguientes conclusiones:

- Lista de componentes del sistema que requieran mayor investigación y desarrollo ordenada de mayor a menor influencia sobre la fiabilidad del sistema, a coste o esfuerzo mínimo.
- Listas de chequeo para Mantenimiento y Reparación a emplear por los operarios de las brigadas de Mantenimiento preventivo, respuesta al fallo y Mantenimiento Oportunista.

En el cuadro siguiente se puede encontrar el cuadro resumen de las medidas de importancia de los sucesos básicos recopiladas por Henley y Kumamoto (1993):

SÍMBOLO	MEDIDA DE IMPORTANCIA	EXPRESIÓN	SIGNIFICADO
Δg_i	Importancia de un suceso básico según Birnbaum	$g(1_i, Q(t)) - g(0_i, Q(t))$	Probabilidad de que el sistema esté en un estado en el que la ocurrencia del suceso i sea crítica
I_i^{CR}	Importancia Crítica de un suceso básico	$\frac{Q_i(t[g(1_i, Q(t)) - g(0_i, Q(t))])}{g(Q(t))}$	Probabilidad de que el suceso i haya ocurrido y sea crítico para el sistema
I_i^{FV}	Importancia de un suceso básico según Fussell-Vesely	$\frac{g_i(Q(t))}{g(Q(t))}$	Dado un fallo del sistema, probabilidad de que el suceso i esté contribuyendo
I_i^{UF}	Función Escalada de la Importancia de un suceso básico	$\frac{\lambda_i(t)}{\partial(\lambda_i(t))} \times \frac{\partial g(Q(t))}{g(Q(t))}$	Reducción fraccional de la probabilidad del suceso top cuando la tasa de fallos de un componente se reduce fraccionalmente
I_i^{BP}	Importancia de un suceso básico según Barlow-Proschan	$\int_0^t \{g(1_i, Q(t)) - g(0_i, Q(t))\} w_i(t) dt$	Número esperado de fallos causados por el suceso i en $[0, t]$
I_i^{SC}	Importancia de Contribución Secuencial de un suceso básico	$\sum_j \{g(1_{i,j}, Q(t)) - g(1_{i,0}, Q(t))\} Q_j(t) w_j dt$	Número esperado de fallos del sistema en $[0, t]$ causados por un GCM que contiene a un suceso i , habiendo ocurrido este suceso antes de ocurrir el fallo del sistema
I_i^{VF}	Importancia de un Grupo de Corte según Fussell-Vesely	$\frac{Q_i(t)}{Q(t)}$	Dado un fallo del sistema, probabilidad de que el grupo de corte i este contribuyendo
I_i^{BP}	Importancia de un Grupo de Corte según Barlow-Proschan	$\sum_{i \in j} \int_0^t [1 - g(0_i, 1^{j-i}, Q(t))] \times \prod_{\substack{l \neq i \\ l \in j}} Q_l(t) dw_i(t) dt Q(t)$	Número esperado de fallos del sistema causados por el grupo de corte j

© Universidad de Las Palmas de Gran Canaria. Biblioteca Digital. 2003

Fig. 2.45.- MEDIDAS DE IMPORTANCIA DE SUCESOS Y GRUPOS DE CORTE

Importancia cuando se permite la reparación

Todos los métodos de medida de importancia pueden calcularla también en los casos en este permitida la reparación excepto la Función Escalada (Henley & Kumamoto, 1993). En todas las formulaciones, excepto en la de Barlow y Proschan, la no disponibilidad puede ser sustituida por la función de Fiabilidad sin cambio en el sentido probabilístico (Lambert, 1975).

Programas de ordenador para medida de importancia

Entre los programas de ordenador que contemplan las medidas de importancia son de mencionar: IMPORTANCE (Lambert, 1975), PL-MOD (Olmos y Wolf, 1978). Los cuales realizan los cálculos de importancia según las diversas medidas expuestas.

2.7.9.- Los fallos con Causa Común en el análisis cuantitativo

A la mayor parte de las causas de fallo secundario se les puede asignar una tasa de ocurrencia dependiente del tiempo en cada dominio. Con frecuencia esas tasas de ocurrencia son idénticas a las de fallos hardware que se traducen en causas de fallo genéricas. La tasa de fallo para la causa genérica " temperatura " puede corresponder a la tasa de fallo del sistema de refrigeración. De forma diferente que la característica de fiabilidad de un componente, la característica de fiabilidad de un candidato de causa común no se describe completamente por la tasa de fallo dependiente del tiempo de la causa de fallo. Cada ocurrencia de una causa de un fallo secundario no ocasionara el fallo del sistema. Los candidatos de causa común se clasifican por su sensibilidad a cada causa de fallo secundario. Esa clasificación se usa para determinar la fracción de ocurrencias de una causa de fallo secundario que causa actualmente el fallo del sistema. La tasa de fallo del candidato de causa común es la tasa de ocurrencia de la causa de fallo secundario multiplicada por la fracción de fallos del sistema que se obtiene para la causa de fallo secundario. La posición en la clasificación de la sensibilidad del candidato de causa común es la menor sensibilidad de componente cuando todos los componentes de un GCM deben fallar para que el sistema falle.

Una tasa de fallo dependiente del tiempo se puede determinar para el nuevo grupo de corte que resulta de los sucesos con causa común. La reparación del fallo secundario se debe considerar añadida a la reparación de los elementos del candidato de causa común.

Vesely (1977) desarrollo técnicas de estimación estadística para los fallos con causa común. El modelo de Marshall-Olkin (1967) permite obtener técnicas de estimación eficiente con los datos, normalmente dispersos, disponibles para análisis cuantitativo de fallos con causa común. Esta implícita en el uso del modelo de Marshall-Olkin la suposición de que cada causa de fallo tiene una distribución exponencial para la primera vez que ocurra y que todas las causas de fallo están en competencia, así como que la población es homogénea en el sentido de que se compone de elementos sujetos a causas de fallo similares. Se suelen considerar dos casos con el modelo homogéneo: Tasas de fallo constantes y Tasas de fallo Binomiales.

2.8.- APLICACIONES Y NUEVAS TENDENCIAS

2.8.1.- Introducción

El análisis de árbol de fallos ha sido utilizado con profusión en las últimas décadas en muy diversas aplicaciones técnicas, es una enorme tarea intentar detallar tales aplicaciones, por lo que hemos recurrido a realizar una clasificación, que sin llegar a ser exhaustiva, pretende mostrar algunas referencias en áreas de interés actual. En cuanto a las nuevas tendencias, intentan reflejar aquellas áreas de conocimiento en que se están produciendo actualmente aportaciones al estado del arte que pueden trazar los caminos de Investigación y desarrollo para el futuro próximo. No cabe duda de que el análisis de árbol de fallos como parte de la Fiabilidad de Sistemas tomará participación activa en la llamada Ingeniería Concurrente que pretende incorporar esta disciplina a un proceso global que considere los productos técnicos desde su concepción hasta su desecho, incorporando todas las etapas reales del ciclo de vida del producto como elementos para su diseño e implantación.

2.8.2.- Aplicaciones del Análisis de Árbol de Fallos

En el cuadro siguiente se pueden encontrar algunas referencias significativas que muestran diferentes aplicaciones de esta metodología en diversas áreas de la técnica. Como puede apreciarse, el abanico de posibilidades es amplio, por lo que es inadecuado para este trabajo intentar una descripción exhaustiva de todas ellas. Si se puede destacar, sin embargo, el Análisis Probabilístico de Riesgos (PRA) que se realiza para garantizar la seguridad colectiva de personas y bienes, cuyo grado de implantación es muy alto y que será descrito con algo de detalle a continuación.

<u>ANÁLISIS PROBABILÍSTICO DE SEGURIDAD (PRA)</u>	<u>DISEÑO</u>
Roodbol, 1986 (Zona Portuaria) Wash 1400 (Centrales Nucleares) Galyean, Fowler, Close, 1989 Parascos, 1990 Budnitz, 1998 (Seismos) Carlsson et al., 1998 (PWR) Risk Analysis, 1998 (PWR) Zamalani, 1998 (NPP)	Kuzawinski & Smurthwaite, 1989 Hessian et al., 1989 Agarwalla, 1991 Teng & Ho, 1995 Andrews y Pattisson, 1997 Bouissou y Bogarde, 1997 Georgiannis, 1997 Cepin y Mavko, 1998 Levitin et al., 1998 Ren y Dugan, 1998 Shetty et al., 1998 (Sist. Antiincendios) Annand y Somani, 1998 (Sist. Dependientes) Roush y Wang, 1995 (Sist. Dependientes)
<u>AERONÁUTICA</u>	<u>SISTEMAS TOLERANTES A FALLOS</u>
Khodabandehloo & Sayles, 1986 Zielinski, 1989 Redgate, Mckelvey, Jolly, 1994	Bavuso et al., 1987 Belli & Jedrzejowicz, 1989 Sato, Henley, Inoue, 1989 Bechta, Bavuso, Boyd, 1990 Boyd & Tuazon, 1991 Dugan, Doyle, Patterson-Hine, 1994 Dugan & Liu, 1994 Doyle, Dugan, Patterson-Hine, 1995 Gulati y Dugan, 1997 Manian et al., 1998 Hura & Atwood, 1988 Dugan, Bavuso, Boyd, 1992 Patterson-Hine, Dugan, 1992 Roush & Wang, 1995
<u>FERROCARRILES y MINAS</u>	
Min, Zou, Li, 1994 Collins, 1987	
<u>SOFTWARE</u>	
Munera, 1986 Liu, 1998 Smidts y Sova, 1999	
<u>PROCESOS QUÍMICOS</u>	
Mulvihill, 1988 Ulrich & Powers, 1988 Sloane & Sherbine, 1988 Dhillon & Rayapati, 1988	

Fig. 2.46.- APLICACIONES DE LOS ÁRBOLES DE FALLOS

Análisis Probabilístico de Riesgos (PRA)

También denominado Análisis Probabilístico de Seguridad (PSA), constituye sin duda una importante aplicación de los árboles de fallos. Este análisis pretende determinar la probabilidad de pérdida o daño sobre las personas y/o los bienes en una determinada región sometidas a algún riesgo causado por dispositivos tecnológicos. Un ejemplo típico lo constituye el PRA de las regiones que contienen Centrales Nucleares o Plantas Químicas de alto riesgo. La seguridad no se puede garantizar de forma individual sino que se determina de forma colectiva en base a ciertos parámetros de tipo empírico, por ejemplo, un nivel de

riesgo de 10^{-6} es asumido por muchos como aceptable para las actividades industriales. En general este tipo de análisis requiere de la intervención de diferentes estamentos sociales así como técnicos especializados, entidades públicas y gestores, para alcanzar resultados precisos y aceptables por todos. En la figura siguiente se puede apreciar el marco general de interacciones para un Análisis Probabilístico de Riesgos.



Fig. 2.47.- MARCO GENERAL DE ACTUACIONES EN UN PRA

Aunque paulatinamente nuestra sociedad va evolucionando hacia concepciones claras respecto a su seguridad colectiva, todavía hoy en día “ *los PRA se realizan más para satisfacer al público que para disminuir el riesgo* “ (Henley & Kumamoto, 1993). Un análisis de este tipo se compone de diferentes etapas que se pueden sintetizar en:

ETAPA1.- ANÁLISIS INICIAL DE RIESGOS, encaminada a determinar los riesgos y sus causas desencadenantes.

ETAPA2.- IDENTIFICACIÓN DE SECUENCIAS DE ACCIDENTES, ÁRBOLES DE SUCESOS Y ÁRBOLES DE FALLOS, que permiten encadenar las causas a los efectos que se desea analizar, y cuantificar los diferentes escenarios posibles de desarrollo de los acontecimientos.

ETAPA3.- ANÁLISIS DE CONSECUENCIAS, que determinará los riesgos sobre las personas y/o bienes comprendidos dentro de la región en análisis.

En la figura siguiente se puede apreciar un esquema del proceso en su conjunto.

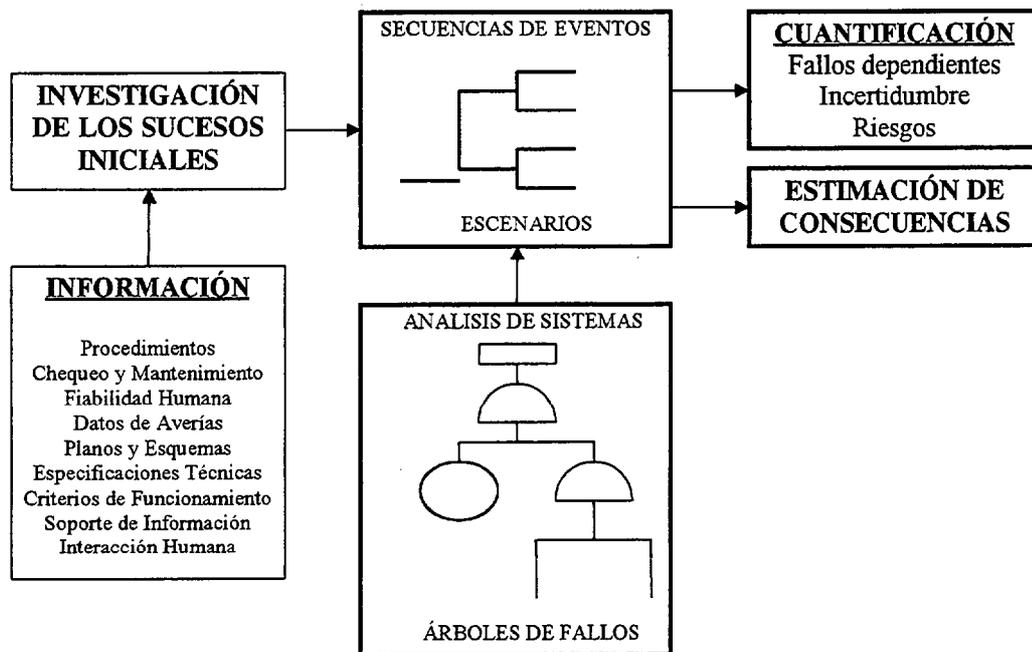


Fig.2.48.- EL PROCESO DEL ANÁLISIS PROBABILÍSTICO DE RIESGOS

2.8.3.- Nuevas tendencias del Análisis de Árbol de Fallos

Como es frecuente, no existe acuerdo general entre los investigadores sobre la totalidad de las líneas de investigación a seguir en el análisis de árbol de fallos. Se han realizado esfuerzos notables en tal dirección (Misra, 1993), lo que ha permitido cierto grado de entendimiento. En lo que sigue emplearemos una clasificación basada en cinco líneas maestras:

- Evaluación Directa de Grandes Árboles de Fallos
- Análisis de Sistemas Dinámicos y/o Tolerantes a Fallos
- Análisis de Incertidumbre
- Modelado de Sistemas Dependientes
- Combinación con otras técnicas del Algebra Discreta

Evaluación Directa de Grandes Árboles de Fallos

La creciente complejidad técnica de los sistemas técnicos unida a la interdisciplinariedad de los estudios a realizar conduce a árboles de fallos cada vez más grandes y/o complejos, la evaluación cuantitativa de estos es muy difícil, cuando no imposible, por métodos tradicionales basados en los Primeros Implicantes (PI's) o en los Grupos de Corte Mínimos (GCM). Por todo ello se han desarrollado en los últimos años diversas técnicas de Evaluación Directa, sin recurrir a los PI's o a los GCM, que abordan el árbol tal cual se obtiene del proceso de construcción. El contenido de esta Tesis Doctoral trata de este tema, por lo que se le ha dedicado un apartado muy amplio en la presente memoria del estado del arte (Apdo. 2.7.3).

Análisis de Sistemas Dinámicos y/o Tolerantes a Fallos

El análisis de fiabilidad de Sistemas Tolerantes a Fallos para aplicaciones críticas es ciertamente complicado debido a diferentes factores. Los sistemas diseñados para alcanzar altos grados de fiabilidad emplean frecuentemente un alto grado de redundancia, gestión dinámica de la redundancia y técnicas complejas de recuperación cuando se producen fallos y/o errores. Sistemas técnicos típicos son Procesadores Paralelos Tolerantes a fallos, Sistemas de Aviónica y Procesadores Hipercúbicos. Estos sistemas pueden ser modelados usando árboles de fallos (Dugan et al., 1992), (Rayapati, V.N., 1989), (Belli & Jezdrzejowicz, 1990), (Ciciani, B., 1991), (Doyle et al., 1995), (Bavuso et al., 1987), (Roush & Wang, 1995), (Boyd & Tuazon, 1991), (Patterson-Hine & Dugan, 1992), (Gulati y Dugan, 1997), (Manian et al., 1998).

Muchos sistemas técnicos actuales incluyen ordenadores activos con software dedicado a tareas de supervisión, control y maniobrabilidad. En misiones delicadas es necesario estudiar la fiabilidad de la combinación de hardware y software, incluyendo análisis cuantitativo. El modelo ha de contemplar fallos transitorios y permanentes del hardware así como fallos independientes y dependientes en el software. Un árbol de fallos, en combinación con modelos de Markov, es capaz de incorporar todos estos tipos de fallos

si se usa una descripción estructural del proceso adecuada. Existen tres aproximaciones estructurales para ello: Bloques Distribuidos (Distributed Recovery Bloks), Programación N-version (NVP) y Programación N-autochequeable (NSCP). (Dugan et al., 1994), (Howell et al., 1990).

Análisis de Incertidumbre

La evaluación de árboles de fallos en condiciones de Incertidumbre es una es una faceta de importancia creciente en la metodología, pretende dar respuesta a una de las limitaciones conocidas de la metodología constituida por la presencia de datos vagos o imprecisos respecto a los sucesos básicos. Cuando todos o parte de los datos son de esta naturaleza es necesario contemplar a través de todo el árbol, puerta a puerta hasta el suceso top, esa incertidumbre. Las soluciones a tal inconveniente provienen de diferentes áreas de conocimiento, así desde el campo de la Estadística provienen la utilización del Método de los Momentos (Rushdi, 1985), (Rushdi & Kafrawi, 1988) y, recientemente, la propuesta de uso de la Desviación Estándar (Page & Perry, 1994). Sin embargo, las aportaciones de mayor aceptación provienen del área de la Inteligencia Artificial, en la cual se usan diversas alternativas para contemplar la Incertidumbre. La teoría de Dempster-Shafer fue propuesta para su utilización en árboles de fallos por Guth (1991) la cual tiene sus fundamentos en la teoría de Probabilidades. De gran importancia ha sido la utilización de la Lógica Difusa (Borrosa), propuesta por Misra y Weber (1988), que se fundamenta en un conjunto propio de axiomas que permiten desarrollar una estructura matemática completa y aplicable a los árboles de fallos, existen grandes aportaciones teóricas al efecto (Misra, 1993), así como múltiples aplicaciones algunas de ellas de avanzado diseño (Beymair & Ebecken, 1995). Una última aportación pendiente de publicación pretende la combinación de la Lógica Borrosa y las Redes Neuronales (Pedrycz, W).

Modelado de Sistemas Dependientes

Los modelos de fiabilidad que contemplan dependencias en sistemas complejos (Annand y Somani, 1998), tales como Sistemas de Control de Tráfico Aéreo y aproximación de Estaciones Espaciales, son incapaces de reflejar fielmente, tanto por tamaño como por complejidad, los diseños altamente integrados de los mismos. Para abordarlos se puede recurrir a sofisticadas técnicas que incluyen árboles de fallos, cadenas de Markov y al desarrollo de representaciones jerárquicas del comportamiento de los sistemas, así como estrategias de solución basadas en Programación Orientada a Objetos (POO), (Patterson-Hine & Dugan, 1992). Una de las limitaciones de los modelos de árboles de fallos es su incapacidad para reflejar las dependencias secuenciales. Existen muchas situaciones en los sistemas técnicos que requieren ser modelados usando cierta dependencia en secuencia obligatoria, la cual no es fácil de reflejar usando las puertas lógicas convencionales. Una forma de ser incorporadas al árbol de fallos es usando soluciones tipo Markov y un conjunto de nuevas puertas (Bechta et al., 1990), (Vermaak y Viranjan, 1998). En la figura 2.4 se pueden encontrar estas puertas de dependencia secuencial.

Combinación con otras técnicas del Algebra Discreta

Tanto las limitaciones referentes a los árboles de fallos como las nuevas aplicaciones, constituyen los nuevos hitos a superar en los próximos años. Los investigadores están procediendo al estudio sistemático de técnicas combinadas, con otras disciplinas del Algebra Discreta, que incorporan al análisis de árbol de fallos las ventajas características de estas otras o viceversa. En el cuadro siguiente se pueden apreciar algunas contribuciones significativas al estudio de estas técnicas combinadas.

REFERENCIA	TÉCNICA COMBINADA
Ulerich y Powers, 1988 Boyd e Iverson, 1988	Digrafo + Árbol de Fallos
Lin y Wang, 1997	Árbol de Fallos y Lógica Borrosa
Dugan ,1988 Vermaak y Viranjan, 1998	Markov + Árbol de Fallos
Hura y Atwood, 1988 Malhortra y Trivedi, 1995	Redes de Petri + Árbol de Fallos
Mulvihill, 1993	Riesgo y Op. + Árboles de Fallos + Árboles de Sucesos
Tan y Cramer, 1993	Bayes + Árbol de Fallos
Malhortra y Trivedi, 1994 Annand y Somani, 1998	Combinación jerárquica de métodos
Doyle et al., 1995	Árbol de Fallos + Cobertura

Fig. 2.49.- COMBINACIÓN DEL ÁRBOL DE FALLOS CON OTRAS TÉCNICAS

CAPÍTULO III
CONTRIBUCIONES A LA EVALUACIÓN DIRECTA DE
ÁRBOLES DE FALLOS

3.1.-INTRODUCCIÓN

Cuando en un sistema técnico moderno se está realizando un **Análisis de Fiabilidad (AF)**, un **Análisis Probabilístico de Riesgos (PRA)** o una **Optimización Multicriterio** que incluya AF o PRA, se suelen presentar árboles de fallos de gran tamaño y/o complejidad, para los cuales no es fácil saber a simple vista si son resolubles por métodos analíticos o no. En la mayoría de los casos se intentará primeramente por un método analítico y, si este falla, se empleará simulación o se optará por simplificar el árbol. La revisión de la bibliografía permite concluir que los métodos analíticos normalmente empleados cuando se estudian árboles de fallos se agrupan en tres grandes líneas:

- Utilizando Grupos de Corte Mínimos(GCM)-MÓDULOS (métodos clásicos)
- Transformando a DIAGRAMAS de DECISIÓN BINARIOS. (nueva tendencia)
- Usando el árbol DIRECTAMENTE. (nueva tendencia)

Si el método analítico comienza por determinar los GCM el proceso puede ser bastante largo y no se sabrá si tiene éxito hasta que concluya, el crecimiento exponencial del número de GCM hace con frecuencia inviable este método. El empleo de MÓDULOS sólo es aplicable si el árbol tiene módulos, lo cual no se sabe a priori. Si se utilizan DIAGRAMAS DE DECISIÓN BINARIOS será necesario decidir el orden de las variables que conduce al Diagrama óptimo, para lo cual no existen criterios fiables matemáticamente demostrados, por lo que existe riesgo de fallar en la conversión, sobre todo en árboles de gran tamaño y complejidad. Los métodos DIRECTOS abordan el árbol sin ninguna determinación previa por lo que son conceptualmente simples, sin embargo no se han desarrollado métodos analíticos directos eficaces para árboles de mas de 100 sucesos básicos. En algunos casos se recurre al truncamiento para poder dar resultados cuando los árboles son demasiado grandes o complejos. Este truncamiento se realiza por tamaño o probabilidad de los GCM ó por criterios heurísticos en los otros métodos.

Los métodos de simulación sólo se pueden emplear con árboles grandes si es posible aplicar un método de reducción de varianza que garantice la convergencia hacia la solución buscada en un tiempo de CPU razonable, además los resultados que se obtienen son aceptables con un cierto nivel de confianza, lo que hace que se apliquen casi exclusivamente en caso de fallo de los métodos analíticos.

El problema de la evaluación cuantitativa de grandes árboles de fallos se vuelve especialmente grave cuando es necesario evaluarlos cientos o miles de veces, lo cual ocurre cuando el proceso está inmerso en uno mucho mayor de Optimización Multicriterio. En este tipo de trabajos se evalúa la Fiabilidad de un sistema técnico pero además han de cumplirse otros condicionantes tanto de tipo técnico como económico y social. Uno de estos modernos procesos de optimización es conocido como Algoritmos Genéticos (GAs) (Goldberg, 1989), el cual está encontrando cada vez más aplicaciones en la optimización de la Fiabilidad de Sistemas (Muñoz, Martorell y Serradell, 1997), (Painton y Campbell, 1995), (Kumar, Pathak, Gupta, 1995), y en particular con la presencia de árboles de fallos (Andrews y Pattison 1997), siendo uno de los actuales inconvenientes la ausencia de subprogramas de evaluación cuantitativa de árboles de fallos adaptados a dichos procesos y eficientes (en tiempo y precisión) para cualquier circunstancia que pueda ocurrir durante la optimización.

En este capítulo se parte de los últimos resultados conocidos y reseñados en los dos capítulos anteriores para el estudio y evaluación de árboles de fallos. Se comienza realizando un análisis de las ventajas e inconvenientes fundamentales de las metodologías determinista o analítica y la aleatoria o de simulación. Tras este análisis se plantea la cuestión consiguiente de la posibilidad de construir un procedimiento que aproveche las ventajas de ambas metodologías, minimice los inconvenientes, y que permita evaluar cuantitativamente grandes árboles de fallos (***Dar respuesta adecuada a dicha cuestión constituye el objetivo central de esta tesis***). Finalmente se decide una estrategia de evaluación combinada a aplicar y en base a ella se especifican las características deseables de los métodos a emplear.

Para el desarrollo de los métodos de evaluación se realiza un replanteamiento teórico del problema de fondo a resolver, que pone de manifiesto los aspectos fundamentales del proceso de resolución más idóneo. A partir de dichos aspectos se desarrollarán modelos matemáticos, que serán estudiados de forma individual y conjunta, especificando las condiciones óptimas de aplicación, su capacidad, versatilidad y los diversos algoritmos que permiten su implementación en ordenadores. El procedimiento anterior concluirá con un nuevo sistema para el análisis de árboles de fallos, que en el capítulo siguiente será validado con casos prácticos. Todos los procesos y rutinas producidos en este trabajo están especialmente diseñados para ser parte de una Optimización Multicriterio basada en Algoritmos Genéticos.

3.2.-ANÁLISIS COMPARATIVO DE LOS MÉTODOS DE EVALUACIÓN EXISTENTES

La lectura de los métodos expuestos en el capítulo anterior nos permite clasificarlos en dos grandes grupos, a los que denominaremos: métodos de Simulación y Analíticos. El siguiente estudio comparativo analiza las características esenciales de ambos, clasificándolas según sus ventajas e inconvenientes.

3.2.1.-Métodos de simulación

ventajas

- Gran Flexibilidad, son aplicables a cualquier tipo de árbol e incorporan fácilmente nuevas puertas lógicas, por lo que se recurre a estos métodos cada vez que se incorporan nuevos elementos a los modelos de los sistemas a evaluar por árboles de fallos.
- Gran Capacidad, permiten la simulación de sistemas grandes y complejos.
- Algoritmos y Programas simples y cortos.
- Permiten evaluar todo tipo de parámetros de Fiabilidad y Mantenimiento
- Libertad para elegir la función de densidad de probabilidad de los sucesos básicos.

inconvenientes

- El resultado que se obtiene es aproximado por lo que existe incertidumbre sobre la validez del mismo, lo cual se refleja en un intervalo de confianza sobre dicho resultado.
- Empleo de mucho tiempo de cálculo cuando las probabilidades son muy pequeñas y/o los árboles muy grandes.
- Convergencia lenta hacia la solución cuando las probabilidades de los sucesos básicos son altas.
- Para aumentar la eficiencia del proceso de simulación es necesario recurrir a métodos de reducir la varianza, estos métodos nos son fáciles de aplicar, en especial cuando se evalúan grandes árboles de fallos.

3.2.2.-Métodos analíticos

BASADOS EN GCM-MÓDULOS

ventajas

- Los conjuntos mínimos facilitan una visión profunda del sistema ya que expresan relaciones explícitas que describen su estructura así como las combinaciones de fallos a las que hay prestar atención prioritaria.
- Pueden dar soluciones exactas para árboles de pequeño tamaño.
- Si se encuentran módulos de tamaño apropiado el árbol se simplifica notablemente

inconvenientes

- Solo utilizan parte de la información disponible
- Dan soluciones aproximadas para árboles medianos o grandes, pero se conocen las cotas de error cometido, se obtiene un intervalo de seguridad exacto.
- El número de GCM puede ser inabordable con lógicas complicadas incluso en árboles pequeños.
- Requieren códigos auxiliares para evaluar los GCM ya que hay que eliminar las repeticiones de un suceso en mas de un GCM.
- La eficiencia de los algoritmos para obtener GCM disminuye con el aumento del tamaño de los árboles y /o la lógica de estos.

BASADOS EN DIAGRAMAS DE DECISIÓN BINARIOS (DDB)

ventajas

- producen soluciones exactas incluso en árboles grandes
- Si el DDB es óptimo se evalúa rápidamente.

inconvenientes

- Para obtener el DDB óptimo es necesario decidir a priori el orden adecuado de los sucesos básicos, pero para ello no existen criterios matemáticamente demostrados lo que obliga a recurrir a heurísticos , con riesgo de error y de que se produzcan crecimientos exponenciales como los que se producen en métodos basados en GCM.
- Requieren dos códigos adicionales : Conversor a DDB y Evaluador de DDBs

- Dificultad para incorporar nuevas puertas lógicas

DIRECTOS

ventajas

- Se abordan los árboles desde el primer instante, sin procesos previos.
- Es posible conocer el error que se comete paso a paso.

inconvenientes

- No se ha demostrado la aplicabilidad para árboles de mas de 100 sucesos básicos
- Los mejores resultados se han obtenido empleando técnicas sofisticadas de difícil generalización.
- La determinación de los parámetros de fiabilidad resulta complicada.

3.2.3.-Conclusiones del análisis comparativo

A la vista de lo anterior, y de la revisión del estado del arte de los capítulos I y II, es posible establecer las siguientes conclusiones:

- La creciente complejidad de los problemas técnicos y el alto nivel de coordinación e integración entre subsistemas grandes, aconsejan la evaluación de parámetros de Fiabilidad obtenidos de forma conjunta, por lo que los árboles de fallos alcanzan gran tamaño. Con frecuencia esos árboles han de ser evaluados muchas veces en los procesos de optimización (Andrews et al., 1997), por lo que es de gran valor disponer de métodos rápidos, fiables y capaces de ofrecer siempre una solución aceptable, sea cual sea el tamaño o la complejidad de la estructura lógica.
- No se ha desarrollado ningún método directo que incorpore todas las ventajas de los métodos analíticos y de simulación, cuya aplicación conjunta minimize los inconvenientes de cada uno de ellos y que permita abordar grandes árboles de fallos.
- No existe ningún código de ordenador que incorpore las ventajas mencionadas, que elija automáticamente los métodos a aplicar y que aproveche el máximo posible de

información obtenida por un método para mejorar los resultados del siguiente y que sea utilizable integrado en un proceso de optimización multicriterio de tipo evolutivo.

A estas conclusiones podemos añadir las opiniones de algunos autores sobre el particular:

- ZIPF(1982) concluye su trabajo de comparación de ambos métodos diciendo que *"para Árboles grandes es recomendable la simulación y para Árboles pequeños los métodos analíticos"*.
- (Hauptmans, 1986) afirma que *"es recomendable, para el trabajo del ingeniero, disponer de un programa con métodos analíticos y otro de simulación directa y reducción de varianza"*.
- (Koen, 1989) afirma que *"sería conveniente poner límites exactos a la probabilidad de suceso top cuando esta se calcula por métodos aproximados con truncamiento"*.
- Un trabajo reciente (Boyd y Bavuso, 1993) comparaba el empleo de simulación frente a métodos analíticos para analizar parámetros de fiabilidad de sistemas técnicos complejos. Entre sus conclusiones se puede encontrar el siguiente párrafo :*"la experiencia ha demostrado que es usualmente preferible emplear métodos analíticos , mientras sea posible, que simuladores para evaluar un modelo de fiabilidad. En particular, cuando la precisión de los resultados es importante, creemos que el uso de simulación generalmente será el último recurso a emplearse después de que se haya encontrado que las técnicas de modelado analítico no son factibles,*

3.3.-OBJETIVOS DEL ESTUDIO

Asumiendo como correctas las conclusiones de Boyd y Bavuso (1993) antes mencionadas, y decididos a encontrar un proceso que consuma la menor cantidad de tiempo posible en evaluar cuantitativamente un árbol de fallos de cualquier complejidad y/o tamaño, parece lógico que debieran cumplirse las siguientes condiciones:

a)Respecto a la secuencia de métodos a emplear

- Se comenzará siempre por un método analítico y, si sus resultados no son satisfactorios, se continuará por uno de simulación.

b)Respecto al método analítico inicial

- Abordar la evaluación cuantitativa desde el primer momento (tendencia actual)
- Aprovechar toda la información disponible
- Ser capaz de resolver árboles de gran tamaño y/o complejidad
- Si no es capaz de resolver un árbol deberá entonces suministrar información absolutamente fiable sobre la probabilidad buscada, por ejemplo cotas seguras de la misma.
- Saber en cada instante el error máximo que está cometiendo, lo que significa que debe ser autoevaluable paso a paso
- Que la información que facilite sea susceptible de ser empleada por un método de simulación siguiente, tanto para iniciar sus cálculos como para mejorar su convergencia y precisión.

c)Respecto al método de simulación siguiente

- Ser capaz de aprovechar la información previa facilitada por el método analítico de forma óptima, pudiendo retomar los cálculos donde quedaron analíticamente.
- Poseer una ó varias formas de muestreo muy eficaces, que estén lo menos influenciadas posibles por el número de sucesos básicos, puertas lógicas y la lógica del árbol de fallos.

La revisión desde este punto de vista de la bibliografía existente permite observar algunos resultados parciales:

- Kumamoto & Henley (1987, 1993) y Karp & Luby (1983) presentan métodos de simulación que reducen su varianza utilizando información previa obtenida analíticamente, el principal inconveniente de estos métodos se deriva del hecho de que usan Grupos de Corte y/o Caminos Mínimos para definir cotas seguras, con lo que se cae de nuevo en el problema del crecimiento exponencial si se quiere obtener cotas suficientemente próximas entre si en árboles grandes.
- Un trabajo de Fishman (1986) compara la eficacia de diversos métodos de reducción de varianza aplicables a priori en la situación que planteamos y concluye que para probabilidades pequeñas el método más eficiente es el Muestreo Daga (Kumamoto et al, 1980), seguido de un método basado en cotas deterministas que usa Grupos de Corte y Caminos Mínimos para determinar unas cotas iniciales seguras. El muestreo Daga vuelve a ser mencionado como muy eficaz en (Doreste y Galván, 1986, 1987), (Henley y Kumamoto, 1993).
- Otro método de muestreo empleado es el Muestreo por Importancia (Boyd y Bavuso, 1993), sin embargo tiene el inconveniente de que es necesario cambiar paulatinamente las funciones de distribución de las que hay que muestrear, lo que también va en contra de nuestras condiciones iniciales, ya que complica la aplicación en árboles muy grandes de forma eficiente.

Dado que los métodos de simulación pueden considerarse como DIRECTOS por abordar la evaluación del árbol sin ningún proceso previo, parece lógico que el método analítico inicial fuese también DIRECTO para facilitar el cambio de uno a otro, ya que se abordaría todo el proceso como un todo cuyas partes realizarán cálculos definidos desde una filosofía común. También deberá existir un mecanismo que permita diferenciar claramente los elementos que se usarán con el método analítico de los que se usarán en el de simulación.

3.4.- ESTUDIO GENERAL DEL PROBLEMA DE LA EVALUACIÓN DE UN ÁRBOL DE FALLOS

El apartado anterior ha puesto de manifiesto que tanto la metodologías analíticas como las de simulación incorporan ventajas e inconvenientes al problema de la evaluación de grandes árboles de fallos. Por tanto, y como se indicó en la introducción de este capítulo, cabe preguntarse si: *¿Existirá algún procedimiento mixto que aprovechando las ventajas esenciales de ambas metodologías minimice los inconvenientes de estas?*. La respuesta a esta cuestión requiere necesariamente del replanteamiento del problema teórico básico a resolver, lo que se realizará a continuación.

Supuestos iniciales:

- 1) El árbol de fallos tiene k sucesos básicos
- 2) El árbol de fallos tiene M puertas lógicas entre los sucesos básicos y el top
- 3) Los sucesos básicos son estadísticamente independientes
- 4) No se conocen a priori, ni grupos de corte ni caminos mínimos
- 5) Las probabilidades de los sucesos básicos serán < 0.1 (sucesos raros)

Notación :

$$x_i = \text{estado del suceso básico } i \left\{ \begin{array}{l} 1 \quad \text{si el suceso ocurre} \\ 0 \quad \text{si el suceso no ocurre} \end{array} \right\}$$

$\mathbf{x} = (x_1, x_2, \dots, x_k)$ vector de estado de sucesos básicos

$\mathbf{v} = (v_1, v_2, \dots, v_k)$ vector binario

$\psi(\mathbf{x}) =$ Función de Variable Indicadora binaria que representa al suceso top del árbol de fallos (Función de Estructura)

$$\psi(\mathbf{x}) = \left(\begin{array}{l} 1 \quad \text{si el suceso top ocurre} \\ 0 \quad \text{si el suceso top no ocurre} \end{array} \right)$$

$P(x_i) =$ probabilidad ad de que ocurra el suceso básico i

3.4.1.-Formulación general del problema

Nos proponemos calcular la *No Disponibilidad* del sistema, que es la probabilidad de que la función de variable indicadora binaria del suceso top del árbol sea igual a uno, siendo conocidas únicamente las probabilidades de cada suceso básico, o sea:

$$Q_s = P(\psi(\mathbf{x})=1) / \exists P(x_i=1) \forall i \in U_\psi \quad (3.1)$$

Siendo U_ψ el conjunto Universal que contiene todos los vectores de estado disjuntos entre si, que constituyen las combinaciones posibles de sucesos básicos a ensayar en el árbol de fallos.

$$U_\psi = \{\mathbf{x} / \mathbf{x} \text{ vector de estado}\} = \{0,1\}^k$$

$$\forall \mathbf{x} \in U_\psi \rightarrow \psi(\mathbf{x}) = \{0,1\}$$

El tamaño de este conjunto universal es 2^k , de gran tamaño para valores de "k" grandes. Todos los elementos se pueden obtener mediante un desarrollo de la tabla de Quine de los k sucesos básicos. Cada fila de la tabla de Quine es un vector de estado posible, y para todos ellos se cumplirá:

$$0 < P(\mathbf{x} = \mathbf{v}) = \prod_{i=1}^k P(x_i = v_i) = \prod_{i=1}^k P_i^{v_i} (1 - P_i)^{(1-v_i)} < 1$$

Por lo que la expresión 3.1 puede escribirse de la forma

$$Q_s = P(\psi(\mathbf{x})=1) = \sum_{\mathbf{v}} \psi(\mathbf{v}) P(\mathbf{x} = \mathbf{v}) = \sum_{\mathbf{v}} \psi(\mathbf{v}) \prod_{i=1}^k P_i^{v_i} (1 - P_i)^{(1-v_i)} = E_{\mathbf{v}}(\psi(\mathbf{v})) \quad (3.2)$$

Esta expresión describe al mismo tiempo dos formas de abordar el problema planteado:

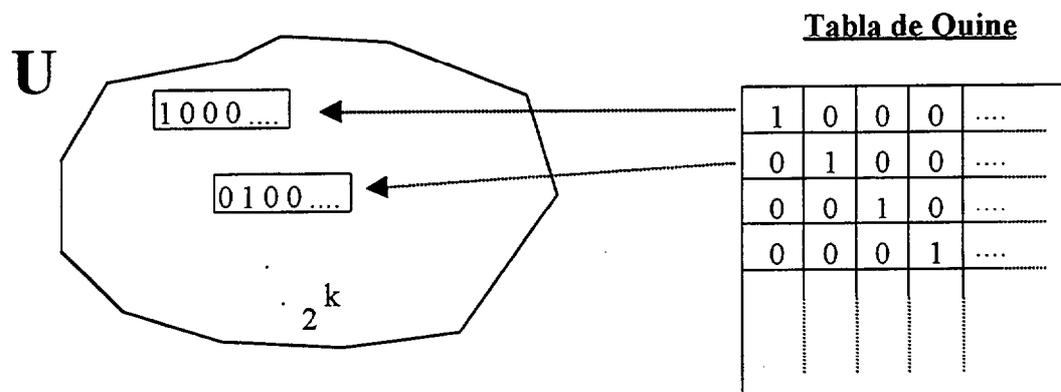
Por un lado, el penúltimo miembro nos dice que la probabilidad buscada se obtendrá sumando los productos de las probabilidades de los sucesos que aparecen en cada vector de estado que produzca un "uno" en la función de estructura, cada vector de estado contribuirá al valor buscado con su producto de

probabilidades, por lo que lo llamaremos Peso. Este proceso es de tipo determinista y lo denominaremos "Método de los Pesos".

Por otro lado el último miembro nos dice que la probabilidad buscada puede determinarse mediante el cálculo de la esperanza matemática de la Función de Estructura, lo cual se realizará por simulación mediante el método de Monte Carlo. En este proceso de simulación se emplearán dos estimadores eficientes: El primero se realizará mediante la aplicación de una técnica de reducción de la varianza que denominaremos Muestreo Daga Generalizado, el segundo mediante un estimador mixto de gran reducción de varianza que denominaremos Método PRD (Pesos-Restringido-Daga), obtenido a partir de un estimador general de Muestreo Restringido.

3.4.2.-Particiones en U

El conjunto Universal U contiene todas las filas de la tabla de Quine, o sea, dos elevado a "k" elementos, siendo k el número de sucesos básicos del árbol de fallos. El proceso de calcular la probabilidad del suceso top del árbol consiste en obtener la suma de los productos de probabilidades de todos los elementos (filas) que producen un uno en el suceso top (Ec. 3.2). A lo largo del desarrollo de los métodos de evaluación se ha ido abordando el problema desde diferentes perspectivas, en este apartado describiremos dichas perspectivas considerando siempre como referencia las filas de la Tabla de Quine, dado que parte de las aportaciones de este trabajo se basan en conclusiones obtenidas a partir de este enfoque.

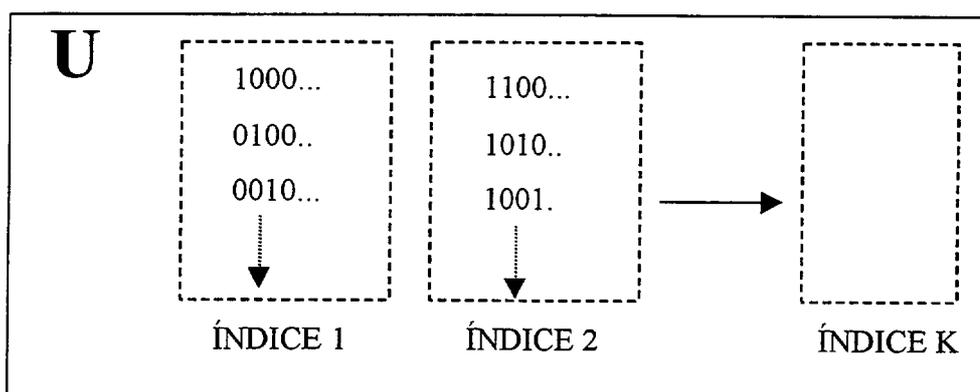


Construir la Tabla de Quine es una forma segura y regular de obtener todos los elementos del conjunto universal U , al mismo tiempo permite observar características de los elementos que se pueden aprovechar para distinguir unos de otros. Dado que los elementos (filas, vectores de estado, o cadenas) están compuestos de ceros y unos, las características diferenciadoras se basarán en estos *valores* y/o en *agrupaciones* de ellos. Los valores posibles son dos (1,0), mientras que las agrupaciones pueden considerarse como elementos que contienen algunas posiciones fijas "Esquemas" (Goldberg,1986). Por ejemplo el esquema "00111#####" representa a todas las filas que comienzan por "00111".

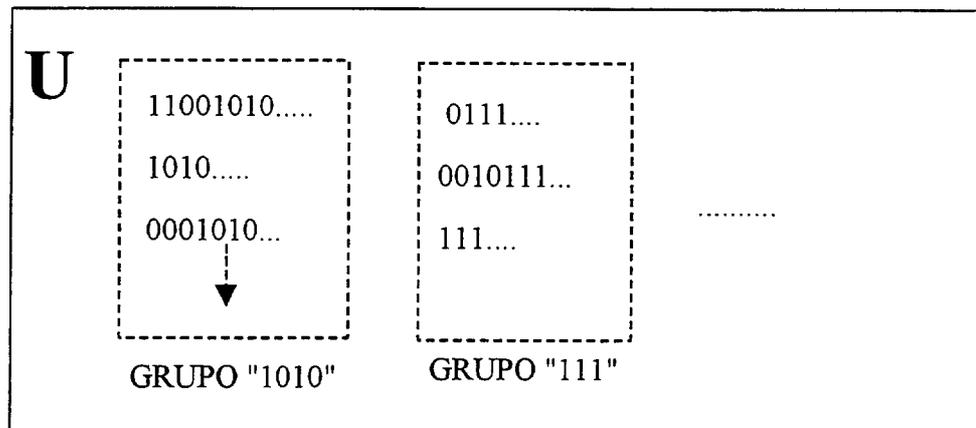
a)PARTICIONES INTERNAS

El conjunto universal U se puede partir en clases de individuos atendiendo a *valores* y/o *esquemas* de diferentes formas, cuando esas particiones se realizan atendiendo sólo a características internas de U , sin considerar correspondencias que provienen de otras consideraciones, se denominarán INTERNAS. Las Particiones Internas se pueden agrupar en dos clases:

1) Atendiendo al número de unos que están presentes en los elementos.- Se considerarán pertenecientes a la misma clase todos aquellos elementos que tengan igual número de unos sea cual sea su posición. Llamaremos ÍNDICE Z a la clase compuesta por elementos con Z unos. El dual con el número de ceros también es posible. Si se consideran todos los miembros de un orden se dirá TOTAL(T) y si consideran solo algunos miembros se dirá PARCIAL(P). Los miembros de estas particiones son fáciles de diferenciar si se añade un contador de unos (o ceros) al proceso de obtención de elementos. dado que las particiones atendiendo al número de unos solo se fijan en los unos (o en los ceros la dual) estas se denominarán SIMPLES(S),



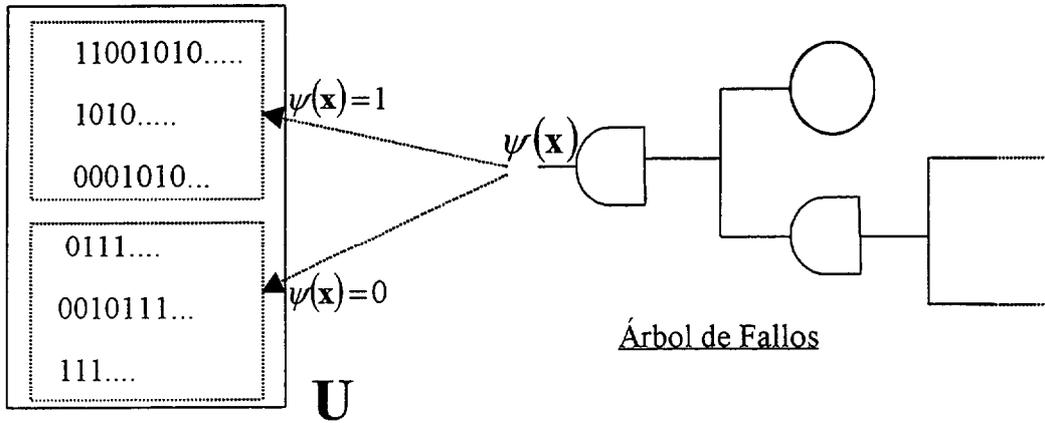
2) Atendiendo a los esquemas.- Se considerarán pertenecientes a la misma clase todos aquellos elementos que contengan a un esquema dado. Denominamos **CARDINAL** de un esquema al número de posiciones fijas que contiene. Dado un esquema se denominará **GRUPO** al subconjunto formado por elementos de U que contienen el esquema. Las particiones atendiendo a los esquemas admiten unos o ceros, por lo que las denominaremos **MÚLTIPLES(M)**.



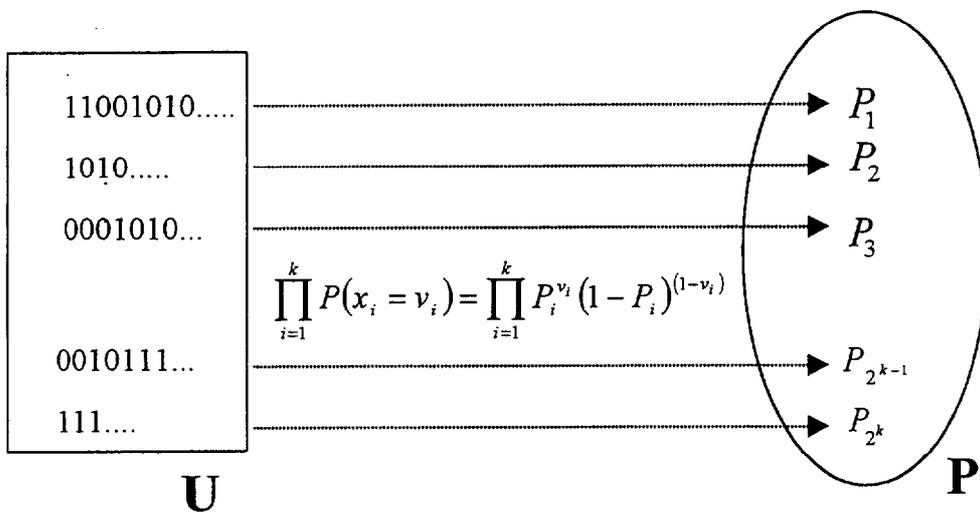
B)PARTICIONES EXTERNAS

Es posible definir clases de en U según valores externos al mismo, en el caso de árboles de fallos, son posibles las siguientes:

1) Según el valor lógico del suceso top de árbol.- Formarán parte de una misma clase aquellos elementos que produzcan un mismo valor lógico en la función de estructura o un subconjunto de ella. Por tanto existirán dos clases: los elementos que producen un "1" y los elementos que producen un "0". A las particiones de este tipo las denominaremos **ESTRUCTURALES(E)**. Si se consideran todos los elementos la partición se dirá **TOTAL(T)**, en caso contrario se dirá **PARCIAL(P)**.



2) Según la probabilidad de los sucesos básicos del Árbol.- El producto de las probabilidades de los sucesos básicos de un elemento (vector de estado o fila de la tabla de Quine) es un valor real que lo representa y que puede emplearse para clasificar, ya sea mediante orden de dichos valores de mayor a menor (o viceversa), ya sea mediante la consideración de intervalos para dichos valores. A las particiones de este tipo las denominaremos LINEALES(L).



3.4.3.-Ordenamientos en U. Teorema de Ordenación Intrínseca

El conjunto U puede ordenarse Internamente por el número de unos (o ceros) que tienen sus elementos siguiendo el orden "natural" para construir la Tabla de Quine (Orden LEXICOGRAFICO) y Externamente según el valor del producto de las probabilidades de los sucesos básicos (PESO, ec. 3.2) de tal forma que los elementos aparezcan de mayor a menor producto (o viceversa). El siguiente Teorema permite definir sobre U una relación de orden parcial que respeta es orden Lexicográfico y que permitirá establecer un proceso de cálculo analítico eficiente basado en los Pesos de las filas de la Tabla.

TEOREMA DE ORDENACIÓN INTRÍNSECA (González, García y Galván, 1995).-

"Supuestas, sin mengua de generalidad, ordenadas la variables booleanas de la tabla de Quine en orden creciente de sus probabilidades, no mayores que 1/2 ;

$$0 \leq P_1 \leq \dots \leq P_k \leq 1/2$$

entonces la condición necesaria y suficiente para que el peso de la N-Upla

$$(x_1, x_2, \dots, x_k) \in \{0,1\}^k$$

sea mayor o igual que el peso de la N-Upla

$$(y_1, y_2, \dots, y_k) \in \{0,1\}^k$$

con independencia de las probabilidades P_1, P_2, \dots, P_k , es que en la matriz

$$M = \begin{Bmatrix} x_1, x_2, \dots, x_k \\ y_1, y_2, \dots, y_k \end{Bmatrix}$$

cada columna $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ esté precedida por una columna $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ diferente.

NOTA: "Diferente" debe entenderse en el sentido de Inyectividad: Dos columnas $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ distintas en M estarán precedidas por dos columnas $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ distintas en M.

Este Teorema puede formularse dualmente para pesos menores o iguales y se debe por entero a la aritmética de la construcción de la Tabla de Quine y no al cálculo de Probabilístico.

3.4.4.-Combinación de Particiones y Ordenamientos

Las particiones y ordenamientos pueden combinarse según se considere un método u otro. Así por ejemplo se pueden considerar subconjuntos de U que sean:

"ESTRUCTURAL SIMPLE ORDEN 1T-2P(20)".- Cuyos elementos serán aquellos que produzcan un "1" en la función de estructura incluidos todos aquellos que tienen un solo "1" y veinte de los que tienen dos "1". Que podría representarse por "E S O-1T-2P(20)"

"ESTRUCTURAL SIMPLE DUAL ORDEN 1T-2P(15) INTRÍNSECA".- Cuyos elementos serán aquellos que produzcan un "0" en la función de estructura, incluidos todos aquellos que tienen un solo "0" y los quince mas influyentes que tengan dos "0". Que se puede representar abreviadamente como "E S D O-1T-2P(15) I"

Según esta nomenclatura los dos principales métodos analíticos para evaluar árboles de fallos, basados en Grupos de Corte Mínimos (GCM) o basados en Módulos, se pueden clasificar como PARTICIONES ESTRUCTURALES parciales siendo la primera de ellas además MÚLTIPLE. Si se considera que cada partición del conjunto universal es una forma de observar a este, forma sesgada o punto de vista, que permite concretar conocimientos sobre (evaluar) la función de estructura asociada, parece claro, al menos cualitativamente, que un método de evaluación incrementará su probabilidad de concluir con eficiencia (precisión y velocidad) la tarea al aumentar el número de particiones y/o ordenamientos que sea capaz de considerar a lo largo del proceso, tanto de forma individual como mediante combinaciones de las mismas. En este sentido se puede afirmar que los métodos reflejados en el análisis bibliográfico realizado en el capítulo II muestran pobreza de recursos ya que emplean como máximo dos particiones para realizar la tarea, quedándose sin alternativas (fallando), si las particiones no son suficientemente buenas. Esta situación se pone de manifiesto en cuanto el tamaño o la complejidad de los árboles crece ligeramente, llegando a invalidar dichos métodos para árboles de fallos grandes. *Así pues se puede concluir que la búsqueda de un método eficiente para evaluar grandes árboles de fallos pasa obligatoriamente por considerar diferentes particiones y/o ordenamientos encadenados y/o embebidos unos en otros, como parte de un único proceso, capaz cambiar de partición o de método de evaluación y aprovechar todo la información obtenida cualitativa y cuantitativamente.*

3.5.-EL MÉTODO DE LOS PESOS

3.5.1.-Descripción del método

Este método se basa en expansiones incompletas de Shannon de la función de estructura del árbol de fallos (Vesely et al., 1981) que determinan los términos de una función disyuntiva normal así como los de una función conjuntiva normal, a la que se incorpora un criterio para la elección de los términos de mayor grado de contribución basado en el teorema de Ordenación Intrínseca citado en el apartado anterior.

Las representaciones de la función de estructura del árbol de fallos por las formas disyuntiva-conjuntiva normales completas que determina la expansión de Shannon (Vesely et al., 1981) son:

$$\psi(\mathbf{x}) = \sum_{j=1}^m \rho_j(\mathbf{x}) = \sum_{j=1}^m \prod_{i \in \rho_j} x_i^{v_i} \quad (3.3) \text{ (Forma Disyuntiva)}$$

$$\psi(\mathbf{x}) = \prod_{j=1}^n K_j(\mathbf{x}) = \prod_{j=1}^n \sum_{i \in K_j} x_i^{v_i} = \prod_{j=1}^m \sum_{i=1}^k x_i^{v_i} \quad (3.4) \text{ (Forma Conjuntiva)}$$

La simplificación algebraica se consigue por ser los implicantes (implicados) dos a dos incompatibles:

$$P\{\psi(\mathbf{x}) = 1\} = P\left\{\sum_{j=1}^m \prod_{i=1}^k x_i^{v_i} = 1\right\} = \sum_{j=1}^m P\left\{\prod_{i=1}^k x_i^{v_i} = 1\right\}$$

Y, supuesto el carácter independiente de las variables booleanas:

$$P\{\psi(\mathbf{x}) = 1\} = \sum_{j=1}^m \prod_{i=1}^k P\{x_i^{v_i} = 1\} = \sum_{j=1}^m \prod_{i=1}^k P_i^{v_i} (1 - P_i)^{(1-v_i)}$$

Si la suma de Shannon es incompleta, la suma de las probabilidades de s implicantes proporciona una cota inferior $C_1(s)$ del resultado:

$$C_1(s) = \sum_{j=1}^s \prod_{i=1}^k P_i^{v_i} (1 - P_i)^{(1-v_i)} \leq \sum_{j=1}^m \prod_{i=1}^k P_i^{v_i} (1 - P_i)^{(1-v_i)} = P\{\psi(\mathbf{x}) = 1\} \quad s \leq m \quad (3.5)$$

Dualmente, la suma de las probabilidades de t implicados proporciona una cota superior $1 - C_0(t)$:

$$\begin{aligned} P\{\psi(\mathbf{x})=0\} &= P\{\bar{\psi}(\mathbf{x})=1\} = P\left\{\prod_{j=1}^n \sum_{i=1}^k x_i^{v_i} = 1\right\} = P\left\{\sum_{j=1}^n \prod_{i=1}^k x_i^{1-v_i} = 1\right\} = \\ &= \sum_{j=1}^n P\left\{\prod_{i=1}^k x_i^{1-v_i} = 1\right\} = \sum_{j=1}^n \prod_{i=1}^k P_i^{1-v_i} (1 - P_i)^{v_i} \geq \sum_{j=1}^t \prod_{i=1}^k P_i^{1-v_i} (1 - P_i)^{v_i} = C_0(t) \quad (t \leq n) \end{aligned} \quad (3.6)$$

con lo que:

$$C_1(s) \leq P\{\psi(\mathbf{x})=1\} = 1 - P\{\psi(\mathbf{x})=0\} \leq 1 - C_0(t) \quad (3.7)$$

De esta forma, las representaciones de ψ elegidas proporcionan un intervalo de valor $[C_1(s), 1 - C_0(t)]$ que contiene la No Disponibilidad Q_s buscada del sistema, que no requiere de los implicantes o implicados minimales de la función de estructura. Este tratamiento establece una Partición Estructural incompleta sobre U , siendo su principal inconveniente el crecimiento exponencial del número de filas (N-Uplas) de la tabla de Quine de la función de estructura:

$$2^k = m + n$$

Con lo que es posible que el problema se haga inabordable para nuestro ordenador antes de que las cotas estén lo suficientemente cerca para ofrecer una precisión aceptable. Este inconveniente puede ser eliminado o retrasado si se escogen para el cálculo de las cotas aquellas filas (vectores de estado) de la tabla de Quine que contribuyen mas al acercamiento de las cotas, lo que aceleraría el proceso de convergencia de la sucesión de intervalos encajados:

$$\lim_{\substack{s \rightarrow m \\ t \rightarrow n}} [C_1(s), 1 - C_0(t)] = [C_1(m), 1 - C_0(n)] = P\{\psi(\mathbf{x})=1\} \quad (3.8)$$

La ecuación 3.2 ya introducía cual es la medida de la contribución de cada fila de la tablas de Quine, a la que denominamos **PESO**:

$$\text{PESO} = \rho(x_1, x_2, x_3, \dots, x_k) = P(\mathbf{x} = \mathbf{v}) = \prod_{i=1}^k P_i^{v_i} (1 - P_i)^{(1-v_i)} \quad (3.9)$$

Se trata entonces de escoger aquellas filas de mayor PESO de forma prioritaria para proceder al cálculo de las cotas. Una forma efectiva de elegir dichas filas es introducir un Ordenamiento en U basado en el **Teorema de Ordenación Intrínseca** (González et al., 1995) y que ya fue presentado en el apartado anterior. Las ventajas de aplicar este teorema se pueden condensar en:

- Las filas con mas peso se obtienen por incremento del ÍNDICE (número de unos de la fila). Partiendo de las de ÍNDICE 0 (ningún 1 en la fila) seguidas de las de ÍNDICE 1 (Un 1 en la fila), etc. Se seguirá el orden LEXICOGRÁFICO para generar las filas ordenadas.
- Establece una Partición Simple de U basada en propiedades internas del mismo (sus N-Uplas o filas), por tanto el orden de las filas por su peso no depende de las probabilidades de los sucesos básicos.
- La lógica del árbol no influye en la aproximación entre las cotas de la sucesión de intervalos encajados, solo permite decidir si el peso de una fila se empleará en la cota superior o en la inferior.

El empleo de este teorema sólo introduce un pequeño inconveniente derivado de que la relación de orden que define en U es Parcial, lo que significa en la práctica que algunas filas de peso menor aparecerán antes que otras de peso superior. Sin embargo en González, et al, (1995) se demuestra que este efecto es despreciable frente a los beneficios que se obtienen. Una consecuencia muy importante de la aplicación de este teorema es que la Partición Simple establecida sobre U tiene como identificador el ÍNDICE de las filas (número de unos presentes en las mismas), lo que permite establecer, de forma simple para un algoritmo, que elementos se han empleado para aproximar las cotas, y por tanto, que filas quedan libres para aplicar otros métodos de cálculo. Esto será particularmente importante en aquellos árboles en los que por tamaño y/o complejidad se deba continuar por simulación el cálculo de la No Disponibilidad.

3.5.2.-Algoritmo del método de los Pesos

El método de los Pesos puede implementarse en un computador siguiendo el siguiente algoritmo:

- 1) Indicar el número k de sucesos básicos del árbol de fallos
Indicar la precisión ε deseada en los cálculos
Introducir las probabilidades $P(k)$ de los sucesos básicos

- 2) Reordenar de mayor a menor las probabilidades de los sucesos básicos

$$P_o(k) \leftarrow P(k)$$

- 3) Generar una fila de la tabla de Quine usando orden lexicográfico aplicado sobre el conjunto de probabilidades reordenadas $P_o(k)$

$$P_o(1,2,\dots,k) \xrightarrow{\text{lexicográfico}} \mathbf{v} = (v_1, v_2, v_3, \dots, v_k)$$

- 4) Comprobar en la función de estructura del árbol de fallos si la fila produce un "1" o un "0".

$$\psi(\mathbf{v}) = \psi(v_1, v_2, v_3, \dots, v_k) = \begin{cases} 0 \\ 1 \end{cases}$$

- 5) calcular el peso de la fila

$$Peso = \rho(x_1, x_2, x_3, \dots, x_k) = P(\mathbf{x} = \mathbf{v}) = \prod_{i=1}^k P_{oi}^{v_i} (1 - P_{oi})^{(1-v_i)}$$

- 6) Si $\psi(\mathbf{v}) = 1$ acumular el peso para la cota inferior $C_1 \leftarrow C_1 + Peso$

Si $\psi(\mathbf{v}) = 0$ acumular el peso para la cota superior $C_0 \leftarrow C_0 + Peso$

- 7) Hallar la diferencia entre las cotas superior e inferior $\delta \leftarrow C_1 - (1 - C_0)$

- 8) Si $\delta > \varepsilon$ volver al paso 3
Si $\delta < \varepsilon$ ir al paso 9

- 9) imprimir/guardar valores de $C_1, 1 - C_0, \delta$

- 10) terminar

3.6.-SIMULACIÓN POR MONTE CARLO CRUDO O DIRECTO

Estimar la No Disponibilidad del sistema según define el miembro derecho de la ec. 3.2 empleando el método de Monte Carlo Crudo o Directo consistirá en obtener N muestras aleatorias (Vectores de estado o filas de la tabla de Quine), y emplear el estimador Binomial no sesgado:

$$\hat{Q}_C = N^{-1} \sum_{v=1}^N \psi(c_v) \quad (3.10)$$

Donde el subíndice C quiere significar Crudo o Directo. Las muestras " c " serán vectores de estado " v " con un valor binario "0" o "1" para cada suceso básico del árbol de fallos. Cada muestra contendrá pues " k " muestras individuales obtenidas cada una de ellas en base a la probabilidad del suceso básico al que represente. La varianza de este estimador viene dada por la expresión:

$$Var(\hat{Q}_C) = N^{-1} Q_s (1 - Q_s) \quad (3.11)$$

Los algoritmos basados en Monte Carlo Crudo son conceptualmente simples por lo que los códigos de ordenador son cortos e implementables en cualquier tipo de CPU. Cuando se trata de evaluar árboles de gran tamaño es necesario realizar un gran número de simulaciones para obtener una respuesta con un nivel de confianza aceptable, lo cual se vuelve impracticable debido a que son necesarios tiempos de ejecución muy largos. La causa fundamental de estos tiempos de CPU inaceptables es el tiempo necesario para generar números aleatorios ya que para generar N muestras hay que obtener $N \times k$ números aleatorios. Para reducir los tiempos de ejecución y aumentar la precisión será necesario acudir a métodos de reducción de la varianza muestral que incrementen la eficiencia del proceso basándose en algún conocimiento que pueda ser establecido a priori sobre el valor de la No Disponibilidad. En el siguiente apartado se desarrolla uno de estos métodos que emplea correlación negativa para reducir la varianza y utiliza muchos menos números aleatorios que el Monte Carlo Crudo.

3.7.- MUESTREO DAGA GENERALIZADO

3.7.1.-Introducción

El método de los pesos, expuesto anteriormente, pierde su eficacia de forma creciente a medida que crece el NSB. Por tanto, si se pretende obtener una mayor exactitud en la estimación de $P(\psi(\mathbf{x})=1)$ será necesario proseguir con un nuevo método todos los cálculos. Tratándose de árboles de un elevado número de variables, parece más apropiado emplear métodos de simulación, coincidiendo nuestro criterio con la experiencia señalada en la bibliografía.

El primer paso es establecer las condiciones de dicha simulación para que sea eficaz y, si es posible, complementaria del método de los pesos. Estas condiciones son:

- Muestrear el árbol de fallos de tal manera que se favorezca la aparición de filas de la tabla de Quine de mayor peso.
- Evitar en lo posible la aparición de filas repetidas, ya que estas no aportan nada nuevo al método de los pesos.
- Rápida convergencia, lo cual es equivalente a emplear un método muy eficaz de reducción de varianza.
- Algoritmo sencillo, rápido y fiable.

La primera de estas condiciones nos impide simular la tabla de Quine en su forma más directa (asignando probabilidad $1/2$ a todos los sucesos básicos), ya que esto iría en contra de la condición de favorecer la aparición de las filas de mayor peso. En efecto, tal simulación asignaría la misma probabilidad a la fila de peso máximo que a la de peso mínimo y, en general, asignaría a todas las filas la probabilidad $\left(\frac{1}{2}\right)^k$

Descartadas las probabilidades $1/2$, sólo queda muestrear con las probabilidades conocidas de cada suceso básico y utilizar, si es posible, un procedimiento para obtener

muestras que reduzcan la varianza al mismo tiempo que favorecen la aparición de las filas de mayor peso.

Una búsqueda en los métodos presentados hasta la actualidad, conduce al denominado MUESTREO DAGA, presentado por Kumamoto, Tanaka, Inoue y Henley en 1980. Sin embargo, dicho método sólo fue estudiado para un árbol pequeño que tenía probabilidades iguales en todos los sucesos básicos. Por tanto, habrán de estudiarse sus características cuando se emplea con cualquier número de sucesos básicos, todos ellos con probabilidades distintas y usualmente muy pequeñas. Este estudio lo denominaremos GENERALIZACIÓN del Muestreo Daga y se realiza a continuación.

3.7.2.-Generalización del muestreo Daga

Estudiando la varianza (ec. 3.11) vemos que:

$$\begin{aligned} \text{Var}[\hat{Q}_C] &= \text{Var}\left[\frac{\sum_{i=1}^N \psi(\mathbf{x}_i)}{N}\right] = \frac{1}{N^2} \text{Var}\left[\sum_{i=1}^N \psi(\mathbf{x}_i)\right] = \\ &= \frac{1}{N^2} \left\{ \sum_{i=1}^N \text{Var}(\psi(\mathbf{x}_i)) + 2 \sum_{\substack{i=1, j=1 \\ i \neq j}}^N \text{Cov}(\psi(\mathbf{x}_i), \psi(\mathbf{x}_j)) \right\} = \frac{1}{N^2} \left\{ \sum_{i=1}^N \text{Var}(\psi(\mathbf{x}_i)) \right\} = \frac{1}{N^2} N \sigma^2 = \frac{\sigma^2}{N} \end{aligned} \quad (3.12)$$

Esta simplificación es cierta sólo si no existe Covarianza entre las muestras, esto es, las muestras no están correladas. Por el contrario si existiese una correlación negativa entre las muestras sería posible obtener una reducción de varianza adicional por tal hecho, la técnica se conoce como reducción de varianza por el método de la variable antitética y se describe como sigue.

Basta tomar como estadístico muestral en lugar del habitual \hat{Q}_C a la media muestral \hat{Q}_D de las combinaciones lineales convexas θ_i :

$$\hat{Q}_D = \frac{\sum_{i=1}^N \theta_i}{N} \quad (3.13)$$

donde

$$\theta_i = \alpha\psi(\mathbf{x}) + (1-\alpha)\psi(\mathbf{x}') \quad (3.14)$$

$$0 < \alpha <$$

Siendo $\psi(\mathbf{x}), \psi(\mathbf{x}')$ negativamente correlados. Uno de los métodos más frecuentemente utilizados para la definición de las parejas $(\mathbf{x}, \mathbf{x}')$ viene dado por:

$$R_i = \int_{-\infty}^{x_i} f(x)dx = P(X \leq x_i) \quad 1 - R_i = \int_{-\infty}^{x_i} f(x)dx = P(X \leq x_i')$$

Siendo R_i un numero aleatorio escogido en el intervalo $(0, 1)$, o de otra forma:

$$\alpha R_i = \int_{-\infty}^{x_i} f(x)dx \quad \alpha + (1-\alpha)R_i = \int_{-\infty}^{x_i} f(x)dx$$

El estadístico \hat{Q}_D así definido sigue siendo insesgado:

$$E[\hat{Q}_D] = E\left[\frac{\sum_{i=1}^N \theta_i}{N}\right] = \frac{1}{N} E\left[\sum_{i=1}^N \theta_i\right] = \frac{1}{N} E\left\{\sum_{i=1}^N [\alpha\psi(\mathbf{x}) + (1-\alpha)\psi(\mathbf{x}')]\right\} =$$

$$= \frac{1}{N} \sum_{i=1}^N \{\alpha E[\psi(\mathbf{x})] + (1-\alpha)E[\psi(\mathbf{x}')]\} = \frac{1}{N} N(\alpha Q_s + (1-\alpha)Q_s) = Q_s$$

En cuanto a su varianza:

$$Var[\hat{Q}_D] = Var\left[\frac{1}{N} \sum_{i=1}^N \theta_i\right] = \frac{1}{N^2} Var\left[\sum_{i=1}^N \theta_i\right] = \frac{1}{N^2} Var\left[\sum_{i=1}^N \alpha\psi(\mathbf{x}) + (1-\alpha)\psi(\mathbf{x}')\right] =$$

$$\begin{aligned}
&= \frac{1}{N^2} \text{Var}[\alpha\psi(x_1) + \alpha\psi(x_2) + \dots + \alpha\psi(x_N) + (1-\alpha)\psi(x'_1) + (1-\alpha)\psi(x'_2) + \dots + (1-\alpha)\psi(x'_N)] = \\
&= \frac{1}{N^2} \left\{ \sum_{i=1}^N \text{Var}(\alpha\psi(\mathbf{x}_i)) + \sum_{i=1}^N \text{Var}((1-\alpha)\psi(\mathbf{x}'_i)) + 2\alpha^2 \sum_{\substack{i=1, j=1 \\ i \neq j}}^N \text{Cov}(\psi(\mathbf{x}_i), \psi(\mathbf{x}_j)) + \right. \\
&\quad \left. 2(1-\alpha)^2 \sum_{\substack{i=1, j=1 \\ i \neq j}}^N \text{Cov}(\psi(\mathbf{x}'_i), \psi(\mathbf{x}'_j)) + 2\alpha(1-\alpha) \sum_{\substack{i=1, j=1 \\ i \neq j}}^N \text{Cov}(\psi(\mathbf{x}_i), \psi(\mathbf{x}'_j)) \right\} = \\
&= \frac{1}{N^2} \left\{ [\alpha^2 + (1-\alpha)^2] N\sigma^2 + 2\alpha(1-\alpha) \sum_{\substack{i=1, j=1 \\ i \neq j}}^N \text{Cov}(\psi(\mathbf{x}_i), \psi(\mathbf{x}'_j)) \right\} = \\
&= \frac{\sigma^2}{N} [\alpha^2 + 1 - 2\alpha + \alpha^2] + \frac{2}{N^2} \alpha(1-\alpha) \sum_{\substack{i=1, j=1 \\ i \neq j}}^N \text{Cov}(\psi(\mathbf{x}_i), \psi(\mathbf{x}'_j)) = \\
&= \frac{\sigma^2}{N} + \frac{\sigma^2}{N} [2\alpha(\alpha - 1)] + \frac{2\alpha(1-\alpha)}{N^2} \sum_{\substack{i=1, j=1 \\ i \neq j}}^N \text{Cov}(\psi(\mathbf{x}_i), \psi(\mathbf{x}'_j)) =
\end{aligned}$$

Pero la varianza del estimador de Monte Carlo Crudo es

$$\text{Var}[\hat{Q}_C] = Q_s(1 - Q_s) = \frac{\sigma^2}{N}$$

por lo que nos queda

$$\text{Var}(\hat{Q}_D) = \text{Var}(\hat{Q}_C) + [2\alpha(\alpha - 1)] \text{Var}(\hat{Q}_C) + \frac{2\alpha(1-\alpha)}{N^2} \sum_{\substack{i=1, j=1 \\ i \neq j}}^N \text{Cov}(\psi(\mathbf{x}_i), \psi(\mathbf{x}'_j))$$

Donde se aprecia que el segundo término del segundo miembro es siempre negativo (para alfa menor que la unidad), y que el tercero será negativo si la Covarianza es negativa, lo cual ocurre si las muestras están correladas negativamente. Por lo que la varianza del Muestreo Daga es menor que la varianza del Monte Carlo Crudo:

$$\text{Var}(\hat{Q}_D) < \text{Var}(\hat{Q}_C) \quad (3.15)$$

3.7.3.-Forma de realizar el muestreo Daga

El muestreo Daga construye todo un conjunto de filas de la tabla de Quine al mismo tiempo ya que genera varias muestras para un mismo suceso básico, negativamente correladas, antes de muestrear al siguiente. Concretamente, para cada suceso básico generará un uno en cada $1/P$ muestras, siendo P la probabilidad de ese suceso básico, siendo el resto de valores cero.

De esta forma la probabilidad de aparición de 1 en cada columna coincide exactamente con la probabilidad del correspondiente suceso básico según el modelo de la ley uniforme. Es obvia la correlación negativa al forzar el valor cero en el resto de las casillas de la columna y concretamente la deseada reducción de varianza según lo probado anteriormente.

En cuanto a la forma de elegir en que muestra va el valor uno, se genera un número aleatorio (RND) entre 0 y 1 según una distribución uniforme, la posición o número de la muestra que tendrá el uno vendrá dada por:

$$Posición_del_uno = POS = INT\left(RND \times \frac{1}{PROB.}\right)$$

En el apartado anterior se ha establecido que el muestreo Daga generalizado procederá construyendo para cada suceso básico "i" un grupo de muestras cuyo número

será $\frac{1}{P_i}$, más concretamente la parte entera de este valor, o sea:

$$NM_i = INT\left[\frac{1}{P_i}\right] \quad i = 1, \dots, k$$

Esta forma de proceder obliga a la existencia de una matriz que tendrá un número de filas igual al valor más grande de NM_i y un número de columnas igual al número de sucesos básicos del árbol de fallos en estudio, o sea:

$$\text{MATRIZ_DAGA} \rightarrow \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \vdots & & & & & & & & & \\ \vdots & & & & & & & & & \\ \vdots & & & & & & & & & \\ \vdots & & & & & & & & & \\ \vdots & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

El tamaño de la matriz crece rápidamente, desbordando con facilidad la capacidad de memoria disponible. Como ejemplo consideremos que un árbol con 33 sucesos básicos ($k = 33$) y cuya probabilidad más pequeña es 0.00005, requerirá de 660.000 posiciones de memoria para la misma.

Esta es la principal dificultad que presenta la implementación del muestreo Daga generalizado en un ordenador, sobre todo si se trata de un PC de sobremesa. La clave para superarla está en la constatación de que la mayor parte de dichas posiciones están ocupadas por ceros, existiendo sólo "algunos" unos distribuidos aleatoriamente dentro de la matriz. Esta característica sugiere que se debe encontrar un algoritmo que permita saber y almacenar sólo las posiciones de los unos de dicha matriz. Es decir, emplear técnicas de almacenamiento, ordenación e indexación de matrices huecas.

Aún así es necesario establecer con exactitud cual es la efectividad máxima que cabe esperar de tal algoritmo, para poder establecer la viabilidad de la implementación en ordenadores de pequeño tamaño. Para ello, consideremos que obtenemos una matriz Daga para " k " sucesos básicos, siendo las probabilidades de los mismos:

$$P_i \quad i = 1, \dots, k$$

la dimensión de esa matriz es:

$$DI = k \times MAX[NM_i] = k \times \left[\frac{1}{MIN(P_i)} \right]$$

el número de unos que aparecerán en cada columna será:

$$n^\circ_unos_columna_i = NU_i = INT \left[\frac{P_i}{MIN(P_i)} \right] \quad i = 1, \dots, k$$

La posición de los unos en cada columna se puede calcular mediante:

$$POS_i^j = INT \left[RND_j \times \left(\frac{1}{P_i} \right) \right] \quad i = 1, \dots, k$$

con lo cual el número de datos a almacenar (NDA) será:

$$\begin{aligned} POS_1^1 \dots POS_1^2 \dots POS_1^{NU_1} &\equiv NU_1_datos \\ POS_2^1 \dots POS_2^2 \dots POS_2^{NU_2} &\equiv NU_2_datos \\ \dots &\dots \\ POS_n^1 \dots POS_n^2 \dots POS_n^{NU_n} &\equiv NU_n_datos \end{aligned}$$

o de otra forma:

$$NDA = NU_1 + NU_2 + \dots + NU_n = \sum_{i=1}^k NU_i \quad (3.16)$$

Por tanto se almacenarán NDA posiciones de unos, lo que será equivalente a una matriz Daga de dimensiones $k \times MAX[NM_i]$. Como ejemplo numérico consideremos que el árbol de fallos tiene 17 sucesos básicos con probabilidad 0.01 para todos ellos. En estas circunstancias, la matriz Daga completa tendrá una dimensión de:

$$k \times MAX[NM_i] = k \times \left[\frac{1}{MIN(P_i)} \right] = 17 \times 100 = 1700_posiciones$$

mientras que según (3.16), sólo necesitaremos:

$$NDA = \sum_1^{17} 1 = 17_posiciones$$

cantidad que es el 1 % de la anterior.

Este resultado se puede generalizar desarrollando la fórmula (3.16):

$$\begin{aligned} NDA = NU_1 + NU_2 + \dots + NU_k &= \frac{P_1}{MIN(P_i)} + \frac{P_2}{MIN(P_i)} + \dots + \frac{P_i}{MIN(P_i)} + \dots + \frac{P_k}{MIN(P_i)} = \\ &= \frac{P_1 + P_2 + \dots + P_i + \dots + P_k}{MIN(P_i)} = \frac{\sum_{i=1}^k P_i}{MIN(P_i)} = \frac{P_{media} \times k}{MIN(P_i)} = \\ &= \frac{P_{media} \times NSB}{MIN(P_i)} = P_{media} \times NSB \times \left(\frac{1}{MIN(P_i)} \right) = P_{media} \times DI \end{aligned}$$

dado que se trata de sucesos raros ($P_i \ll 1$), la P_{media} tendrá un valor muy pequeño y $NDA \ll DI$.

El número de datos que será necesario almacenar se puede determinar como un tanto por ciento de los de la matriz Daga inicial según la fórmula:

$$\frac{NDA}{DI} \times 100 = \frac{P_{media} \times DI}{DI} \times 100 = P_{media} \times 100$$

y la reducción de necesidades de memoria respecto a las iniciales es

$$RM = 100 - P_{media} \times 100 = 100 \times (1 - P_{media}) \%_de_DI$$

En el apartado siguiente se puede apreciar el algoritmo de implementación en ordenador de éste método, al que se ha dado el nombre de DAGA.

3.7.4.-Algoritmo del muestreo Daga Generalizado

- 1) Indicar el número k de sucesos básicos del árbol de fallos
 Indicar la precisión ε deseada en los cálculos
 Introducir las probabilidades $P(k)$ de los sucesos básicos
 Indicar la dimensión máxima de vectores que admite su ordenador ($DMAX$)
- 2) Estudio de la dimensión máxima que alcanzará el vector de posiciones de los unos. Corregir probabilidades mas pequeñas si es necesario hasta que:

$$k \frac{1}{\text{MIN}(P_i)} < DMAX$$

- 3) Indicador de número de sucesos básicos a uno: $i \leftarrow 1$
- 4) Obtener el número de unos del suceso básico i

$$NU_i \leftarrow \text{INT} \left[\frac{P_i}{\text{MIN}(P_i)} \right]$$

- 5) Obtener la posición de los unos del suceso básico i

$$a \leftarrow 0$$

Para $j=1$ hasta NU_i

$$POS_i^j \leftarrow a + \text{RND}_j \left[\frac{1}{P_i} \right]$$

$$a \leftarrow a + \text{INT} \left[\frac{1}{P_i} \right]$$

Siguiente j

- 6) Si $i < k$ incrementar i y volver al paso 4
- 7) Obtener un vector de estado (\mathbf{v}) a partir de las posiciones de los unos
- 8) Comprobar el resultado lógico del árbol de fallos para el vector de estado

$$\text{Si } \psi(\mathbf{v})=1 \text{ entonces } M \leftarrow M + 1$$

- 9) Obtener una estimación de la probabilidad buscada

$$P[\psi(\mathbf{x})=1] = \frac{M}{N}$$

- 10) Si se pueden obtener mas vectores de estado volver al paso 7
- 11) Imprimir las estimaciones obtenidas, si la precisión alcanzada es suficiente
 Si no lo es volver al paso 3
- 12) Terminar

3.8.-MUESTREO RESTRINGIDO

3.8.1.- Descripción del muestreo

Consideremos en el conjunto U_ψ una partición según los valores de la función de estructura $\psi(\mathbf{x})$, o sea:

$$U_\psi = \{\mathbf{v} / \mathbf{v} \text{ vector de estado}\} = \{0,1\}^N = U_0 \cup U_1$$

$$U_0 = \{\mathbf{v} \in U_\psi / \psi(\mathbf{v}) = 0\}$$

$$U_1 = \{\mathbf{v} \in U_\psi / \psi(\mathbf{v}) = 1\}$$

Supongamos que conocemos un subconjunto S de cada uno de los conjuntos

$$S_0 \subset U_0$$

$$S_1 \subset U_1$$

Utilizando estos subconjuntos es posible calcular el valor exacto de una cota inferior y una cota superior de la No Disponibilidad del sistema, para ello emplearemos:

$$Q_L = \sum_{\mathbf{v} \in S_1} \psi(\mathbf{v})P(\mathbf{x} = \mathbf{v}) \leq \sum_{\mathbf{v} \in U_1} \psi(\mathbf{v})P(\mathbf{x} = \mathbf{v}) \rightarrow Q_L \leq Q_S$$

$$Q_U = \sum_{\mathbf{v} \in S_0} \psi(\mathbf{v})P(\mathbf{x} = \mathbf{v}) \leq \sum_{\mathbf{v} \in U_0} \psi(\mathbf{v})P(\mathbf{x} = \mathbf{v}) = 1 - Q_S \rightarrow Q_S \leq Q_U$$

Por tanto se cumplirá en todo caso que :

$$0 \leq Q_L \leq Q_S \leq Q_U \leq 1$$

La solución exacta se encuentra cuando son iguales las cotas superior e inferior según se discutió en el método de los pesos. Supongamos que esta situación no se ha producido, y por lo tanto:

$$Q_U - Q_L > 0$$

Q_L es también la no disponibilidad exacta de un árbol cuya función de estructura será:

$$\psi_L(\mathbf{v}) = \begin{cases} \psi(\mathbf{v}) = 1 & \forall \mathbf{v} \in S_1 \\ 0 & \forall \mathbf{v} \notin S_1 \end{cases} P(\psi_L = 1) = \sum_{\mathbf{v} \in U} \psi_L(\mathbf{v})P(\mathbf{x} = \mathbf{v}) = Q_L \quad (3.17)$$

Q_U es la no disponibilidad de un árbol cuya función de estructura será:

$$\psi_U(\mathbf{v}) = \begin{cases} 1 & \forall \mathbf{v} \notin S_0 \\ \psi(\mathbf{v}) = 0 & \forall \mathbf{v} \in S_0 \end{cases} P(\psi_U = 0) = Q_U \quad (3.18)$$

Y por tanto

$$\begin{aligned} \psi_L(\mathbf{v}) &\leq \psi(\mathbf{v}) \leq \psi_U(\mathbf{v}) & \forall \mathbf{v} \in U \\ \psi_L &\neq 0, \quad \psi_U \neq 1 \end{aligned}$$

Dado que ψ_L contiene parte de la solución buscada podemos emplearla como función de control de la función ψ para reducir la varianza de esta. Así partiendo de la expresión 3.2 podemos escribir:

$$\begin{aligned} Q_S &= \sum_{\mathbf{v} \in U} \psi(\mathbf{v}) P(\mathbf{x} = \mathbf{v}) = \sum_{\mathbf{v} \in U} [\psi(\mathbf{v}) - \psi_L(\mathbf{v}) + \psi_L(\mathbf{v})] P(\mathbf{x} = \mathbf{v}) = \\ &= \sum_{\mathbf{v} \in U} [\psi(\mathbf{v}) - \psi_L(\mathbf{v})] P(\mathbf{x} = \mathbf{v}) + \sum_{\mathbf{v} \in U} \psi_L(\mathbf{v}) P(\mathbf{x} = \mathbf{v}) \end{aligned}$$

Pero el segundo miembro es conocido, según la ec. 3.4 quedará:

$$Q_S = \sum_{\mathbf{v}} [\psi(\mathbf{v}) - \psi_L(\mathbf{v})] P(\mathbf{x} = \mathbf{v}) + Q_L \quad (3.19)$$

Esta ecuación ya representa una mejora respecto al muestreo que se realiza en el Monte Carlo Crudo, ya que calcula parte de la solución de forma determinista, y por lo tanto se tendrá menor varianza en la estimación. La mejora se ha conseguido, sin embargo, aprovechando solo la información procedente de una función conocida que produce una cota inferior de la solución buscada. Para introducir el conocimiento procedente de una función que produce una cota superior del valor buscado, sin perder lo ya conseguido en la expresión 3.5, será necesario añadir una nueva partición al conjunto universal, consideremos los conjuntos:

$$\begin{aligned} X &= \{\mathbf{v} / \psi(\mathbf{v}) - \psi_L(\mathbf{v}) = 1\} = \{\mathbf{v} / \psi(\mathbf{v}) = 1, \psi_L(\mathbf{v}) = 0\} \\ Y &= \{\mathbf{v} / \psi_U(\mathbf{v}) - \psi_L(\mathbf{v}) = 1\} = \{\mathbf{v} / \psi_U(\mathbf{v}) = 1, \psi_L(\mathbf{v}) = 0\} \end{aligned}$$

El conjunto X contiene los contribuidores a “1” no elegidos, mientras que el conjunto Y contiene los contribuidores no elegidos. Por tanto Y es un conjunto más amplio que contiene al X y la ecuación 3.5 puede ser reescrita como:

$$Q_S = \sum_{\mathbf{v} \in X} [\psi(\mathbf{v}) - \psi_L(\mathbf{v})] P(\mathbf{x} = \mathbf{v}) + Q_L = \sum_{\mathbf{v} \in Y} [\psi(\mathbf{v}) - \psi_L(\mathbf{v})] P(\mathbf{x} = \mathbf{v}) \quad (3.20)$$

En el conjunto Y han cambiado los pesos relativos de las filas de la tabla de Quine, ya que no están presentes aquellas filas elegidas para las funciones auxiliares. El nuevo peso relativo de las filas presentes será entonces:

$$P(\mathbf{y} = \mathbf{v}) = \frac{P(\mathbf{x} = \mathbf{v})}{[Q_U - Q_L]}$$

Con lo que la ecuación 3.6 quedará como:

$$Q_S = [Q_U - Q_L] \sum_{\mathbf{v} \in Y} [\psi(\mathbf{v}) - \psi_L(\mathbf{v})] P(\mathbf{y} = \mathbf{v}) + Q_L$$

Pero $\psi_L(\mathbf{v}) = 0 \forall \mathbf{v} \in Y$, por tanto

$$Q_S = [Q_U - Q_L] \sum_{\mathbf{v} \in Y} \psi(\mathbf{v}) P(\mathbf{y} = \mathbf{v}) + Q_L \quad (3.21)$$

Expresión que puede emplearse para plantear el cálculo de la No Disponibilidad mediante simulación, teniendo en cuenta que:

$$\sum_{\mathbf{v} \in Y} \psi(\mathbf{v}) P(\mathbf{y} = \mathbf{v}) = E_y(\psi(\mathbf{y}))$$

La expresión (3.21) quedará como

$$Q_S = [Q_U - Q_L] E_y(\psi(\mathbf{y})) + Q_L \quad (3.22)$$

Que es una expresión formal del Método de Monte Carlo con Muestreo restringido. El valor esperado de la No Disponibilidad puede obtenerse mediante el estimador no sesgado:

$$\hat{Q}_R = [Q_U - Q_L] N^{-1} \sum_{j=1}^k \psi(m_j) + Q_L \quad (3.23)$$

El sufijo R quiere significar "Restringido". Una forma eficiente de generar las muestras m_1, m_2, \dots, m_k Se presentará en el apartado siguiente. El estimador (3.23) debe cumplir las propiedades de ser Insesgado y de menor varianza que el de Monte Carlo Crudo.

Si el estimador es insesgado deberá cumplir que:

$$E[\hat{Q}_R] = Q_S = P(\psi(\mathbf{x}) = 1)$$

Para demostrarlo desarrollaremos el primer miembro como sigue:

$$\begin{aligned} E[\hat{Q}_R] &= E\left\{ [Q_U - Q_L] N^{-1} \sum_{j=1}^N \psi(m_j) + Q_L \right\} = (Q_U - Q_L) E\left[\frac{\sum_{j=1}^N \psi(m_j)}{N} \right] + E[Q_L] = \\ &= (Q_U - Q_L) \frac{M}{N} + Q_L = (Q_U - Q_L) \frac{Q_S - Q_L}{Q_U - Q_L} + Q_L = Q_S \end{aligned}$$

Donde se ha tenido en cuenta que en el conjunto Y se cumple que $\frac{M}{N} \cong \frac{Q_S - Q_L}{Q_U - Q_L}$

Por otra parte, la varianza de este estimador es menor que la del Monte Carlo Crudo.

Teniendo en cuenta que $Var(Z) = E(Z^2) - (E(Z))^2$ obtenemos

$$Var(\hat{Q}_R) = N^{-1} Var[(Q_U - Q_L)\psi(m)] = N^{-1} E[(Q_U - Q_L)^2 \psi^2(m)] - N^{-1} \{E[(Q_U - Q_L)\psi(m)]\}^2$$

$$\begin{aligned} Var(\hat{Q}_R) &= Var\left[(Q_U - Q_L) N^{-1} \sum_{j=1}^N \psi(m_j) + Q_L \right] = Var\left[N^{-1} \sum_{j=1}^N (Q_U - Q_L) \psi(m_j) \right] = \\ N^{-2} Var\left[\sum_{j=1}^N (Q_U - Q_L) \psi(m_j) \right] &= N^{-2} \sum_{j=1}^N Var[(Q_U - Q_L) \psi(m_j)] = N^{-2} N Var[(Q_U - Q_L) \psi(m)] = \end{aligned}$$

$$= N^{-1} Var[(Q_U - Q_L) \psi(m)] \quad (3.24)$$

Pero de 3.8 se deduce la equivalencia siguiente:

$$E[(Q_U - Q_L)\psi(m)] = Q_S - Q_L \quad (3.25)$$

Multiplicando a ambos lados de la ecuación por $(Q_U - Q_L)$ y teniendo en cuenta que se cumple

$$\psi(m) = \begin{cases} 0 \\ 1 \end{cases} = \begin{cases} 0^2 \\ 1^2 \end{cases} = \psi^2(m)$$

Se obtiene

$$E[(Q_U - Q_L)^2 \psi^2(m)] = (Q_U - Q_L)(Q_S - Q_L) \quad (3.26)$$

Sustituyendo ahora (3.25) y (3.26) en (3.24) se obtiene

$$\text{Var}(\hat{Q}_R) = N^{-1}(Q_U - Q_L)(Q_S - Q_L) - N^{-1}(Q_S - Q_L)^2 = N^{-1}(Q_S - Q_L)[(Q_U - Q_L) - (Q_S - Q_L)]$$

Que simplificando nos conduce a la expresión

$$\boxed{\text{Var}(\hat{Q}_R) = N^{-1}(Q_U - Q_S)(Q_S - Q_L)} \quad (3.27)$$

comparándola con la varianza del estimador de Monte Carlo crudo, deducida en el apartado 3.6, se comprueba que es menor la de este estimador Restringido

$$N^{-1}(Q_U - Q_S)(Q_S - Q_L) < N^{-1}Q_S(1 - Q_S) \Leftrightarrow \text{Var}(\hat{Q}_R) < \text{Var}(\hat{Q}_C) \quad (3.28)$$

3.8.2.-Forma de Obtener las muestras para el estimador Restringido

La vía clásica de uso de este estimador Restringido ha sido la de definir dos funciones auxiliares, una a partir de algunos GCM y otra a partir de algunos Caminos Mínimos, para calcular las cotas y utilizar Muestro por Importancia asignando probabilidades en función de la contribución relativa a las cotas (véase páginas 127 y 128 de esta memoria). Este método incurre en dos graves inconvenientes (Henley & Kumamoto, 1993) que permiten descartar su aplicación a grandes árboles: Utilizar GCM y Asignar probabilidades para la obtención de sucesivas muestras. Por lo que es necesario replantearse su uso de manera más eficiente ya que, sin embargo, este estimador contiene en sí mismo una unión de método determinista con método de simulación, que pudiera cumplir con los criterios de búsqueda definidos en los objetivos de este estudio.

Una forma simple de obtener las muestras es muestrear sobre el conjunto universal U utilizando las probabilidades de los sucesos originales conocidas, se aceptarán como válidas sólo aquellas que estén dentro del subconjunto Y , o lo que es lo mismo, aquellas que no estén en la partición elegida. Los inconvenientes de este proceder son dos: La dificultad para diferenciar el subconjunto Y , y la baja probabilidad que los elementos de Y tendrán dentro del conjunto universal si la partición es grande.

La primera dificultad es salvable si se adopta una partición Simple de U , o sea, la partición se hará atendiendo al número de unos presentes (orden) en las muestras, empezando por las de menor orden y más peso (INTRÍNSECO).

La segunda dificultad será un inconveniente grave en aquellos casos en que las cotas no se aproximen mucho y los árboles sean grandes. En estos casos la reducción de varianza lograda no será significativa respecto del estimador de Monte Carlo Crudo.

Esta forma general de proceder se concreta en al algoritmo para el muestreo restringido que se presenta a continuación, en el cual se ponen de manifiesto las particiones y ordenamientos empleados así como el mecanismo de control del muestreo basado en el índice de los vectores de estado.

3.8.3.-Algoritmo del muestreo Restringido

- 1) Indicar el número k de sucesos básicos del árbol de fallos
 Indicar la precisión ε deseada en los cálculos
 Indicar el ÍNDICE al que se desea llegar analíticamente (ÍNDICE)
 Indicar el número de sucesos básicos a escoger en cada ÍNDICE (NSCUT)
 Introducir las probabilidades $P(k)$ de los sucesos básicos

- 2) Reordenar de mayor a menor las probabilidades de los sucesos básicos

$$P_o(k) \leftarrow P(k)$$

- 3) Generar una fila de la tabla de Quine usando orden lexicográfico aplicado sobre el conjunto de probabilidades reordenadas $P_o(k)$

$$P_o(1,2,\dots,k) \xrightarrow{\text{lexicográfico}} \mathbf{v} = (v_1, v_2, v_3, \dots, v_k)$$

- 4) Comprobar en la función de estructura del árbol de fallos si la fila produce un "1" o un "0".

$$\psi(\mathbf{v}) = \psi(v_1, v_2, v_3, \dots, v_k) = \begin{cases} 0 \\ 1 \end{cases}$$

- 5) calcular el peso de la fila

$$Peso = \rho(x_1, x_2, x_3, \dots, x_k) = P(\mathbf{x} = \mathbf{v}) = \prod_{i=1}^k P_{oi}^{v_i} (1 - P_{oi})^{(1-v_i)}$$

- 6) Si $\psi(\mathbf{v})=1$ acumular el peso para la cota inferior $C_1 \leftarrow C_1 + Peso$

Si $\psi(\mathbf{v})=0$ acumular el peso para la cota superior $C_0 \leftarrow C_0 + Peso$

- 7) Hallar la diferencia entre las cotas superior e inferior $\delta \leftarrow C_1 - (1 - C_0)$

- 8) Si ($\delta > \varepsilon$ y no se han alcanzado los valores ÍNDICE y NSCUT) volver al paso 3

Si $\delta < \varepsilon$ ir al paso 16

Si $\delta > \varepsilon$ pero se han alcanzado los valores de ÍNDICE y NSCUT, ir al paso 9

- 9) Generar un vector de estado (\mathbf{v}) por Monte Carlo Crudo

- 10) Comprobar que los valores ÍNDICE y/o Nscut de la muestra son mayores que los ÍNDICE y NSCUT en que ha terminado la fase analítica.

- 11) Comprobar el valor de la función de estructura $\psi(\mathbf{v})$

12) Acumular valores

$$M \leftarrow M + \psi(\mathbf{v})$$

$$N \leftarrow N + 1$$

13) Obtener el estimado de la No Disponibilidad mediante

$$Q \leftarrow C_1 + [(1 - C_0) - C_1] \frac{M}{N}$$

14) Obtener estadísticos de la No Disponibilidad (Varianza)

15) Si se ha alcanzado precisión suficiente ir a 16, en caso contrario volver al paso 9.

16) imprimir/guardar valores de $C_1, 1 - C_0, \delta, \frac{M}{N}$

17) terminar

3.9.-MÉTODO PESOS-RESTRINGIDO-DAGA (PRD)

3.9.1.-Descripción del método

El muestreo restringido visto en el apartado anterior utiliza dos cotas seguras de la probabilidad del suceso final (top) del árbol de fallos para reducir la varianza muestral y Monte Carlo Crudo para muestrear los vectores de estado necesarios para aproximar dicha probabilidad. Analizando el proceso en su conjunto se observa que sobre las muestras solo existe una restricción (pertenecer a Y) que ya esta contemplada si se adopta la partición Simple como base para aplicar este muestreo. El único esfuerzo de simulación que queda por hacer entonces es generar muchas muestras sobre U y elegir aquellas que están en Y, antes de comprobar el valor de la función de estructura del árbol de fallos. La mayor parte del tiempo computacional de este esfuerzo de simulación será entonces el de generar números aleatorios para las muestras aleatorias, por lo que parece idóneo emplear Muestreo Daga Generalizado para obtener dichas muestras, dado que sus principales características son (como se vio en el apdo. 3.7.2); el empleo de pocos números aleatorios para obtener muchas muestras y obtener reducción de varianza por establecer correlación negativa entre las muestras sin alterar la naturaleza de las mismas. El estimador a emplear sería entonces:

$$\hat{Q}_{PRD} = [Q_U - Q_L] N^{-1} \sum_{j=1}^N \theta_j + Q_L \quad (3.29)$$

Donde θ_j son muestras obtenidas por Muestreo Daga Generalizado aplicado sobre las filas de la tabla de Quine (vectores de estado) que no fueron considerados en la partición empleada para obtener las cotas Q_U, Q_L . Este estimador es insesgado porque lo son los del muestreo restringido y el del muestreo daga, en cuanto a su varianza vemos que:

$$\begin{aligned} \hat{Q}_R &= [Q_U - Q_L] N^{-1} \sum_{i=1}^N \psi(m_j) + Q_L = [Q_U - Q_L] \hat{Q}_C + Q_L \\ \hat{Q}_{PRD} &= [Q_U - Q_L] N^{-1} \sum_{j=1}^N \theta_j + Q_L = [Q_U - Q_L] \hat{Q}_D + Q_L \end{aligned}$$

Pero ya se ha demostrado que

$$Var[\hat{Q}_D] < Var[\hat{Q}_C] \quad y \quad Var[\hat{Q}_R] < Var[\hat{Q}_C]$$

Por tanto

$$\begin{aligned} Var[\hat{Q}_{PRD}] &< Var[\hat{Q}_R] < Var[\hat{Q}_C \\ Var[\hat{Q}_{PRD}] &< Var[\hat{Q}_D] < Var[\hat{Q}_C \end{aligned} \quad (3.30)$$

3.9.2.-Características del método PRD

El método PRD consiste en la aplicación consecutiva de un método analítico (Pesos) y una simulación con doble reducción de varianza (Restringida y Daga). Cumple con todas las características deseables descritas en el apdo. 3.3 ya que:

- Se comienza por un método analítico y si sus resultados no son satisfactorios se continúa automáticamente por simulación.
- Aborda la evaluación cuantitativa desde el primer instante, aprovechando toda la información disponible de forma directa (árbol de fallos y probabilidades)
- Suministra en cada instante del cálculo cotas seguras de la probabilidad buscada, y conoce el error máximo cometido.
- Toda la información que produce el método analítico inicial se puede emplear, en caso de ser necesario, para continuar por simulación. Esta información es de dos tipos: Cotas seguras de la probabilidad buscada y Elementos de la Tabla de Quine que se han utilizado para calcularlas.
- El método de simulación utiliza las cotas seguras para restringir la muestra (y su varianza) al intervalo entre ellas y utiliza un segundo método (Daga) para reducir la varianza de las muestras que se obtendrán sobre los elementos de la Tabla de Quine no utilizados para el cálculo de las cotas, por lo que tendrá una doble reducción de varianza. El muestreo Restringido es especialmente indicado cuando se conocen cotas seguras y el Daga cuando se muestrea sobre sucesos de probabilidad muy pequeña.
- El método de simulación continúa el cálculo de la probabilidad buscada a partir de donde quedó analíticamente.

- El método PRD en su conjunto es muy poco sensible al número de sucesos básicos y a la lógica del árbol de fallos que se evalúa.
- El mecanismo para controlar el paso del método analítico al de simulación se basa en establecer una Partición Estructural Simple sobre el conjunto de vectores de estado posibles (Tabla de Quine), y utiliza como máximo dos variables:
 - Indicador del ÍNDICE máximo alcanzado en el método de los Pesos
 - Indicador del número del suceso menos probable alcanzado en el ÍNDICE Máximo

Será necesario establecer por experimentación la eficiencia final del método.

3.9.3.-Algoritmo del método PRD

- 1) Indicar el número k de sucesos básicos del árbol de fallos
 Indicar la precisión ε deseada en los cálculos
 Indicar el ÍNDICE al que se desea llegar analíticamente (ÍNDICE)
 Indicar el número de sucesos básicos a escoger en cada ÍNDICE (NSCUT)
 Introducir las probabilidades $P(k)$ de los sucesos básicos

- 2) Reordenar de mayor a menor las probabilidades de los sucesos básicos

$$P_o(k) \leftarrow P(k)$$

- 3) Generar una fila de la tabla de Quine usando orden lexicográfico aplicado sobre el conjunto de probabilidades reordenadas $P_o(k)$

$$P_o(1,2,\dots,k) \xrightarrow{\text{lexicográfico}} \mathbf{v} = (v_1, v_2, v_3, \dots, v_k)$$

- 4) Comprobar en la función de estructura del árbol de fallos si la fila produce un "1" o un "0".

$$\psi(\mathbf{v}) = \psi(v_1, v_2, v_3, \dots, v_k) = \begin{cases} 0 \\ 1 \end{cases}$$

- 5) calcular el peso de la fila

$$Peso = \rho(x_1, x_2, x_3, \dots, x_k) = P(\mathbf{x} = \mathbf{v}) = \prod_{i=1}^k P_{oi}^{v_i} (1 - P_{oi})^{(1-v_i)}$$

- 6) Si $\psi(\mathbf{v}) = 1$ acumular el peso para la cota inferior $C_1 \leftarrow C_1 + Peso$

Si $\psi(\mathbf{v}) = 0$ acumular el peso para la cota superior $C_0 \leftarrow C_0 + Peso$

- 7) Hallar la diferencia entre las cotas superior e inferior $\delta \leftarrow C_1 - (1 - C_0)$

- 8) Si ($\delta > \varepsilon$ y no se han alcanzado los valores ÍNDICE y NSCUT) volver al paso 3

Si $\delta < \varepsilon$ ir al paso 16

Si $\delta > \varepsilon$ pero se han alcanzado los valores de ÍNDICE y NSCUT, ir al paso 9

- 9) Generar un vector de estado (\mathbf{v}) por muestreo Daga, aplicar el algoritmo descrito en el apartado 3.7.4.

- 10) Comprobar que los valores ÍNDICE y/o Nscut de la muestra son mayores que los ÍNDICE y NSCUT en que ha terminado la fase analítica.

- 11) Comprobar el valor de la función de estructura $\psi(\mathbf{v})$

12) Acumular valores

$$M \leftarrow M + \psi(\mathbf{v})$$
$$N \leftarrow N + 1$$

13) Obtener el estimado de la No Disponibilidad mediante

$$Q \leftarrow C_1 + [(1 - C_0) - C_1] \frac{M}{N}$$

14) Obtener estadísticos de la No Disponibilidad (Varianza)

15) Si se ha alcanzado precisión suficiente ir a 16, en caso contrario volver al paso 9.

16) imprimir/guardar valores de $C_1, 1 - C_0, \delta, \frac{M}{N}$

17) terminar

3.10.-UN NUEVO SISTEMA PARA ANALIZAR ÁRBOLES DE FALLOS

Todos los algoritmos anteriormente desarrollados han sido programados con objeto de comprobar en la práctica sus características evaluando cuantitativamente árboles de fallos de diferentes tamaños y/o complejidad lógica.

Por otra parte es conveniente poder comparar resultados con los obtenidos mediante algunos otros métodos de evaluación cuantitativa. Para que dicha comparación indique realmente la eficiencia de los métodos desarrollados en esta tesis no es aconsejable utilizar códigos desarrollados en plataformas y lenguajes diferentes, por ello se ha optado por desarrollar adicionalmente varios códigos que implementan otros métodos ya conocidos. Todo ello conduce a un conjunto de códigos capaz por si solo de evaluar cualitativa y cuantitativamente árboles de fallos de gran tamaño y/o complejidad, totalmente integrados unos con otros y especialmente diseñados para formar parte de una optimización multicriterio vía Algoritmos Genéticos o Estrategias Evolutivas. Los Códigos que forman parte de nuevo sistema son:

MCRUDO1.- Evaluación cuantitativa mediante simulación por Monte Carlo Crudo o directo.

MINCUT2.- Evaluación cuantitativa mediante el método de los Pesos calculando cotas seguras de la probabilidad buscada. Este código calcula adicionalmente los Grupos de Corte Mínimos con o sin truncamiento tanto por índices como por probabilidad, a decidir por el usuario.

MONDAG5.- Evaluación cuantitativa mediante Monte Carlo con reducción de varianza por Muestreo Daga Generalizado.

MPRD.- Evaluación cuantitativa multicombinación. Admite utilizar muestreo Pesos-Restringido-Daga partiendo de cualquier índice a decidir por el usuario. Emplea las cotas seguras dadas por el método de los pesos para lanzar el muestreo restringido y aplica muestreo Daga al subconjunto restante de combinaciones de sucesos básicos.

SOLOCUT.- Determinación de los Grupos de Corte Mínimos por el método de prueba de inclusión. El usuario decide hasta que índice desea aplicarlo en función de sus características de CPU disponibles. Puede utilizar truncamiento hasta cierto índice a decidir por el usuario.

PESCUT.- Evalúa cuantitativamente el árbol de fallos utilizando los Grupos de Corte Mínimos. Puede utilizar truncamiento hasta cierto índice o por probabilidad.

CAPÍTULO IV

APLICACIONES, CÁLCULOS Y DISCUSIÓN

4.1.- INTRODUCCIÓN

La evaluación cuantitativa de árboles de fallos sigue siendo hoy en día una tarea ingente cuando los mismos son de tamaño mediano o grande, o su complejidad notoria (Rauzy y Dutuit, 1997), por ello no es fácil encontrar en la bibliografía especializada ejemplos resueltos, siendo lo más común la referencia a árboles pequeños. Entre los autores incluyen referencias a grandes árboles en sus experiencias prácticas publicadas podemos citar a: Stecher (1986), Coudert y Madre (1993, 1994), Rai (1994), Heger et al. (1995), Aralia Group, (1995), Hennings y Kuznetsov (1995), Rauzy y Dutuit (1997). Algo muy común en las citadas referencias es que los grandes árboles de fallos que emplean proceden de compañías privadas, tales como Electricite de France (EDF) o Dassault Aviation, las cuales son muy reservadas en cuanto a información como la que refleja un árbol de fallos, por ello es meramente imposible obtenerlos salvo que exista algún convenio de colaboración específico en vigor, además, y de todas maneras, no superan la decena los grandes árboles citados, lo cual es una muestra muy pequeña para poder dar resultados que permitan extrapolar las características de un método de evaluación cuantitativa con un nivel de confianza aceptable. Por ello, para poder realizar ensayos de evaluación cuantitativa sobre un número suficiente de grandes árboles, se ha optado en esta tesis por una doble vía: Para árboles pequeños y medianos se han utilizado ejemplos extraídos de las referencias bibliográficas mientras que para grandes árboles se ha diseñado e implementado un software capaz de generar árboles de cualquier tamaño y/o complejidad al que se ha denominado GENARB. Este programa genera en segundos un árbol del tamaño que se desee (máximo 7.000 puertas lógicas y 7.000 sucesos básicos) con cualquier complejidad a elección del usuario, comprobando que el mismo es conexo y completo, y por lo tanto evaluable. Los árboles pequeños y medianos extraídos de la bibliografía permiten validar los códigos desarrollados y/o mostrar aplicaciones concretas para la optimización de sistemas técnicos, mientras que los grandes árboles permitirán explorar los límites de aplicación de los métodos propuestos en esta tesis.

4.2.- VALIDACIÓN DE ALGORITMOS Y APLICACIONES EN INGENIERÍA

Los árboles de fallos y aplicaciones que se muestran a continuación se han extraído de la bibliografía especializada. Con ellos se pretende validar los códigos desarrollados mediante la comparación con los resultados obtenidos por diversos autores y mostrar como es posible aplicar dichos códigos para optimizar el diseño de sistemas técnicos de forma eficiente. Se ha seguido para ello un esquema que comienza con una presentación en la que se justifica el motivo y la utilidad del mismo, seguida de una descripción técnica somera que incluye la especificación del fallo objeto de estudio, a continuación se presentan los resultados obtenidos al aplicar los métodos desarrollados en este trabajo.

4.2.1.- ÁRBOL PROPUESTO POR KUMAMOTO ET AL.

Presentación

Este árbol fue presentado por Kumamoto, Tanaka, Inoue y Henley (1980), contiene 17 sucesos básicos y trece puertas lógicas, es estadísticamente coherente (s-coherente), con los sucesos básicos estadísticamente independientes y se ha evaluado frecuentemente considerando todas las probabilidades de los sucesos de igual valor (0.01). Con el paso de los años se ha convertido en uno de los árboles más usados en la bibliografía para comparar resultados tanto de análisis cualitativo como cuantitativo, por lo que su inclusión en esta memoria parece adecuada como punto de partida. La aplicación de los trabajos de ésta tesis a dicho árbol de fallos permite:

- Validar el método de los Pesos
- Validar y comprobar la convergencia y rapidez del algoritmo del muestreo Daga generalizado.
- Validar y comprobar el método PRD
- Verificar la precisión de todos los métodos en condiciones bastante extremas, dado que el hecho de tener todas las probabilidades iguales y altas en los sucesos básicos es uno de los factores más perjudiciales para los algoritmos desarrollados.

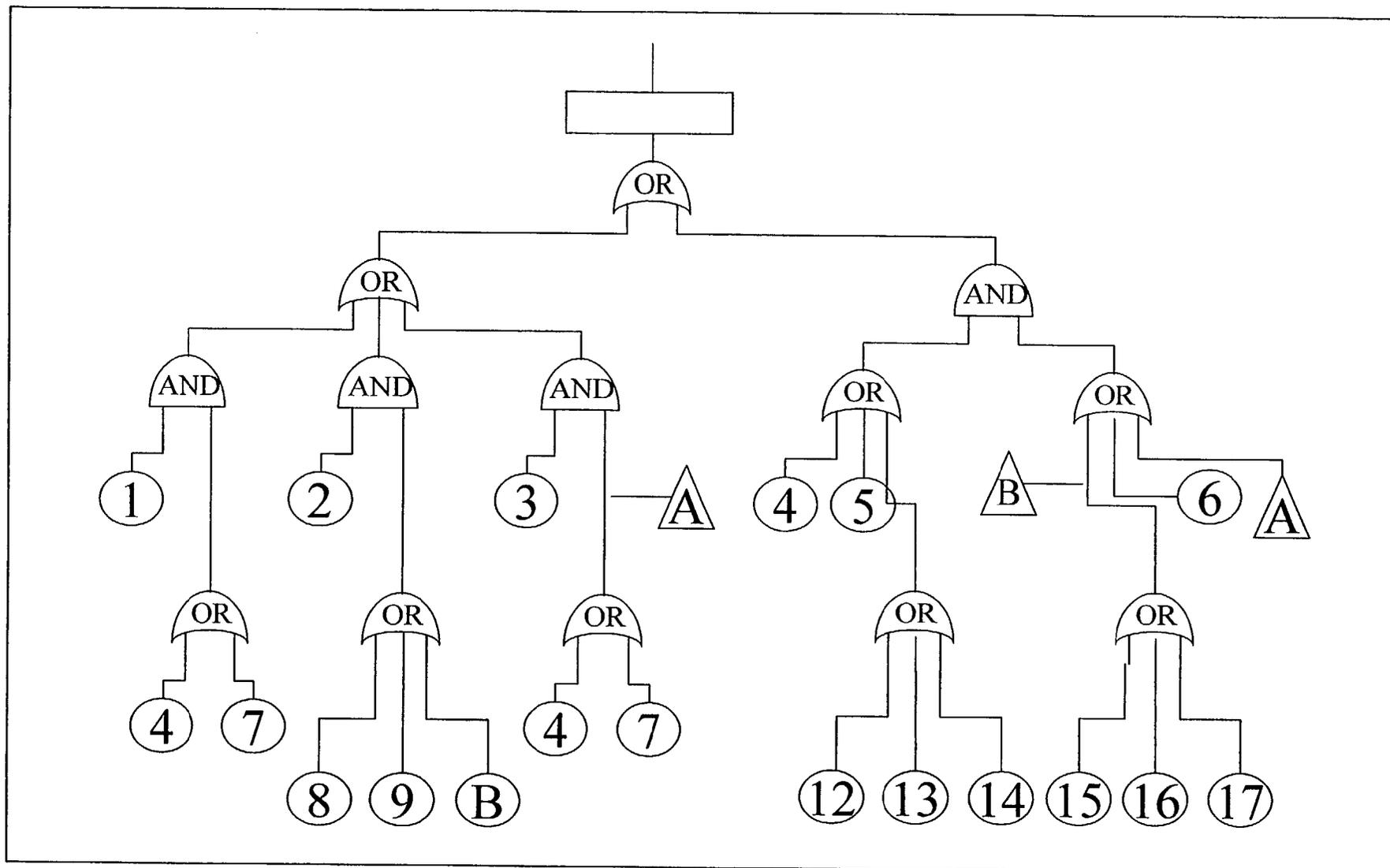


Fig. 4.1.- ÁRBOL DE FALLOS PROPUESTO POR KUMAMOTO ET AL. (1980)

RESULTADOS OBTENIDOS CON PROBABILIDADES 0.01

El método de los Pesos evalúa completamente el árbol, sus resultados son:

Índice	GCM	GCM sel.	Cota Inferior	Cota Superior
1	0	0	0.0000000000000000E+000	1.230898552994431E-002
2	39	39	3.354227583101026E-003	3.966419489922490E-003
3	0	0	3.707807128897997E-003	3.729251883046625E-003
4	0	0	3.724787160692858E-003	3.725346914496441E-003
5	0	0	3.725294970585884E-003	3.725306229406145E-003
6	0	0	3.725305776420896E-003	3.725305954554003E-003
7	0	0	3.725305951150606E-003	3.725305953402924E-003

La solución se calcula instantáneamente con diez cifras significativas. En el gráfico siguiente se muestran los resultados obtenidos para este árbol con los métodos propuestos más por Monte Carlo. Para el método Restringido y el PRD se ha adoptado como índice de partida el 2, tomando como cotas seguras las del índice 1 de la respuesta del método de los Pesos anterior.

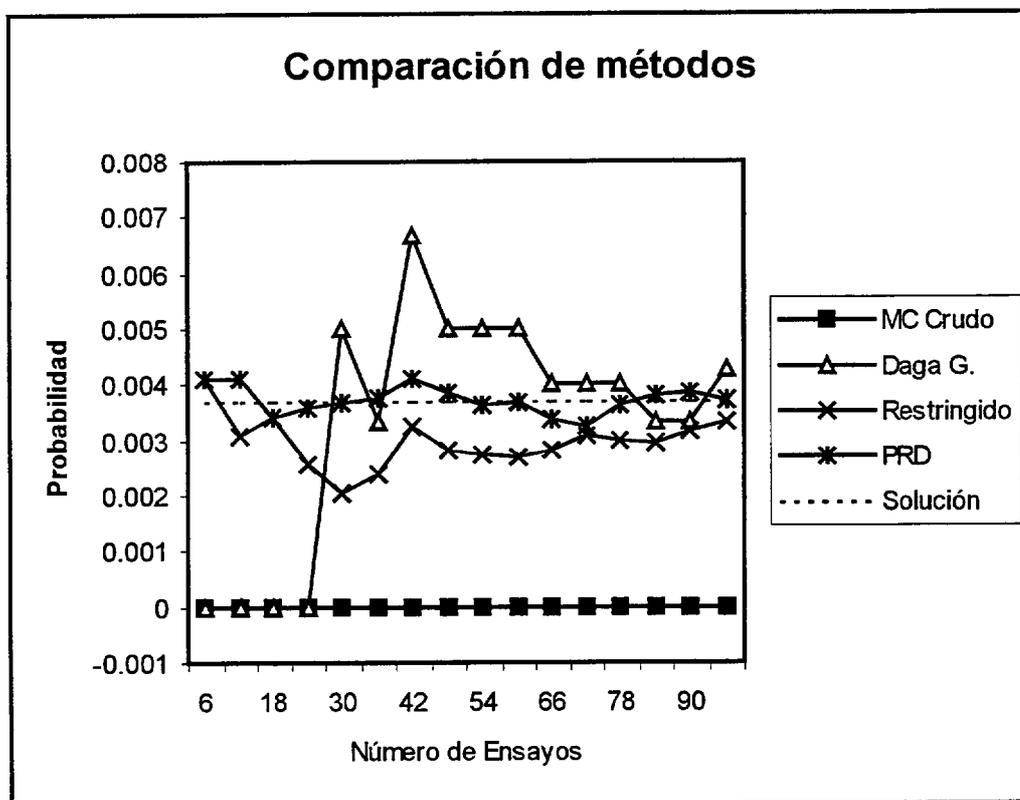


Fig. 4.2.- COMPARACIÓN DE ESTIMADORES

RESUMEN Y COMENTARIOS

La comparación muestra claramente la mejor convergencia del estimador PRD, seguido del Restringido y del Daga Generalizado. Con solo 96 muestras los tres estimadores se sitúan en un rango muy cercano a la solución exacta mientras que la versión del muestreo Daga presentada por los autores en el artículo original necesitaba del orden de 3000 muestras para lograr situarse en dicho rango. El muestreo Crudo no logra encontrar ninguna muestra significativa con este número de ensayos. Los tres estimadores llegan a la solución $3.72E-03$ en menos de un segundo de CPU. Por otra parte es de señalar que el árbol no es simple de resolver por métodos analíticos debido a la combinación de puertas lógicas que contiene, hasta tal punto que solo existe 39 grupos de corte mínimos de orden dos y ninguno más hasta orden 7, con los que solo se consigue el resultado aproximado $3.35E-03$ para la probabilidad del suceso top. En cuanto a las varianzas sus valores fueron:

PRD	$6.36326E-08$
Restringido		$2.18380E-07$
Daga G.		$4.59320E-06$

Donde se aprecia que la varianza del PRD es un 71% menor que la del Restringido y un 98% menor que la de Daga Generalizado. Los autores mencionados no indicaron la reducción de varianza obtenida con su algoritmo por lo que no es posible comparar resultados.

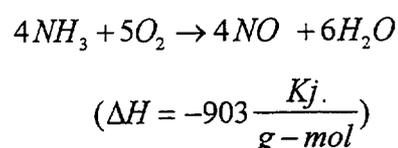
4.2.2.- SISTEMA DE MEZCLA DE AMONIACO Y AIRE DE UNA PLANTA PARA LA FABRICACIÓN DE ÁCIDO NÍTRICO

Presentación

La fiabilidad de las plantas de fabricación de Acido Nítrico ha sido estudiada en la bibliografía (Lapp y Powers, 1977), (Hauptmans, Yllera y Sastre, 1982), (Hauptmans, 1986). En este ejemplo evaluaremos un árbol de fallos del sistema de mezcla de Amoniacio y Aire partiendo de los datos de esta ultima cita bibliográfica. Como parámetro de optimización se utilizará la No Disponibilidad del sistema.

Descripción

Uno de los subsistemas de las plantas de fabricación de ácido nítrico es el correspondiente a la combustión del amoniaco con el oxigeno del aire según la reacción:



bajo una presión de 4,2 bar y a una temperatura de 880 °C. En el diagrama de la página siguiente se pueden apreciar los detalles de la configuración. La proporción volumétrica del amoniaco y aire en el proceso es aproximadamente del 0,5%, el vapor de amoniaco entra en el sobrecalentador S1 bajo una presión de 4,5 bar y a una temperatura de 15 °C y sale a una temperatura próxima a 60 °C. Después de pasar por el filtro F1 el vapor se mezcla en el Venturi E1 con aire que se encuentra a una presión de 4,3 bar y a una temperatura de 190 °C. La mezcla entra en los reactores R1 y R2 a través del filtro F2.

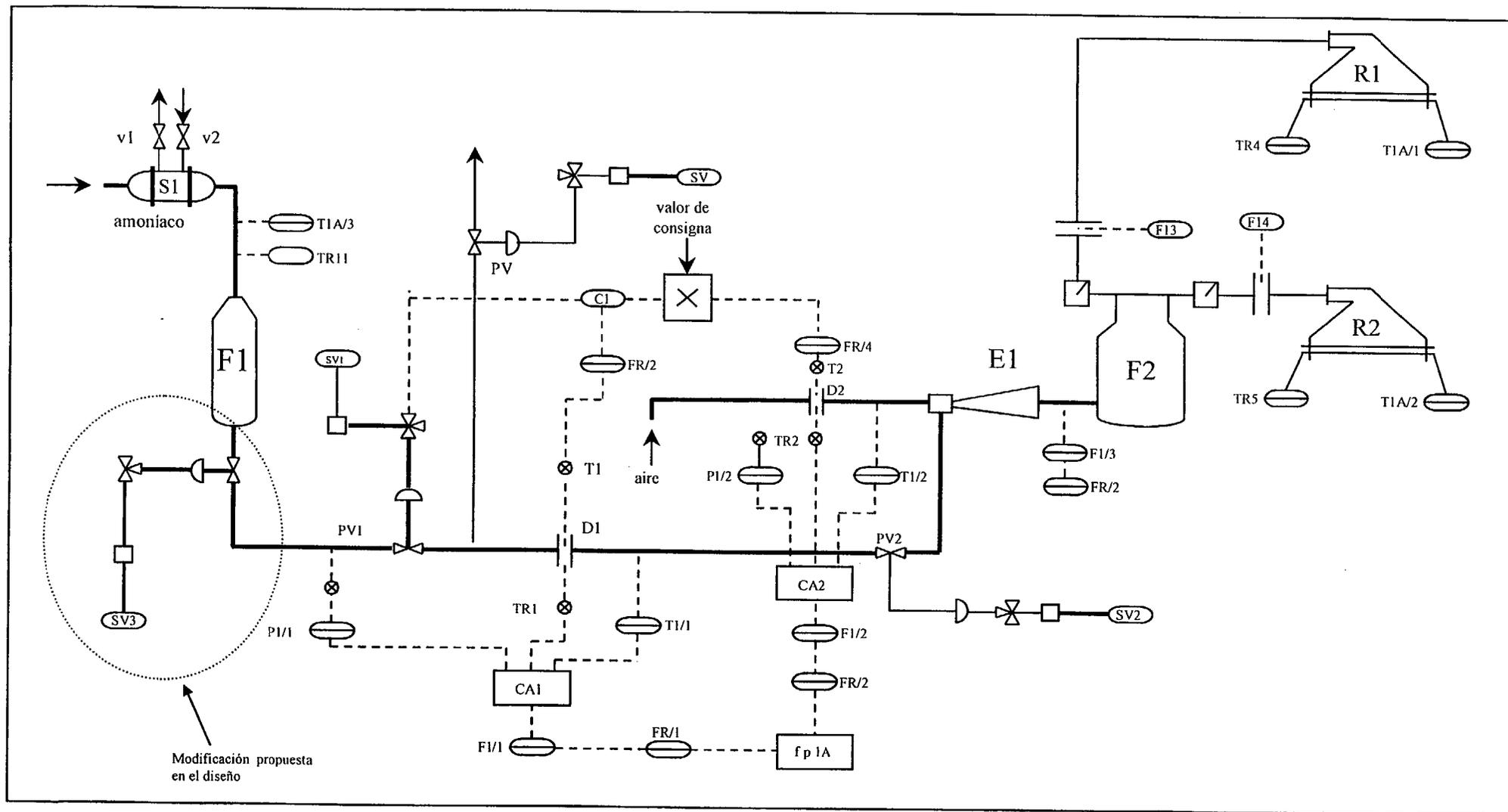


Fig. 4.3.- DIAGRAMA DE FLUJO DEL SISTEMA DE MEZCLA Y COMBUSTIÓN DE AMONIACO Y AIRE

Un análisis del potencial de peligro muestra que una proporción de amoníaco-aire demasiado alta podría ocasionar la explosión de los reactores, ya que:

- El amoníaco tiene un límite de explosión en aire en % de volumen a presión atmosférica del 16-25
- El sistema neumático de control mantiene la proporción de amoníaco-aire entre 9% y el 12 % en volumen para que el rendimiento sea elevado.

Este peligro se trata de controlar con los siguientes dispositivos de seguridad:

1.- Control Neumático

El primer nivel de seguridad se puede considerar que es aportado por el mecanismo de control neumático. Este sistema mide las diferencias de presión a través de los orificios D1 y D2. Estas diferencias se convierten en los transformadores T1 y T2 en señales que representan los caudales de amoníaco y aire respectivamente. En conformidad con un valor de consigna prefijado, el relé MR transforma la señal de T2 en una señal de consigna para el flujo de amoníaco. Dicha señal se compara en el controlador C1 con el valor verdadero procedente de T1. En el caso de una discrepancia entre ambas señales, una señal neumática originada por el controlador corrige la posición de la válvula PV1.

2.- Sistema Eléctrico de alarma

La proporción de amoníaco-aire se comprueba también mediante un sistema eléctrico de alarma que para la planta si la misma esta fuera del intervalo 8%-13,5% en volumen de amoníaco.

Este sistema comparte los orificios D1 y D2 con el sistema de control neumático y convierte las diferencias de presión medidas en corrientes eléctricas mediante los transductores TR1 y TR2. Puesto que los orificios están diseñados para una presión y temperatura determinadas, desviaciones de sus valores nominales se deben tener en cuenta para corregir las medidas de los caudales del amoníaco y del aire. por esta razón se miden las temperaturas y presiones del amoníaco(TI1, PI1) y del aire (TI2,PI2) y se

introducen dichos valores en las calculadoras CA1 y CA2 respectivamente. Estas calculadoras llevan a cabo la corrección. Los caudales corregidos se comparan con los permisibles en la unidad indicadora de flujo y alarma FpIA, que activa la parada de la planta en caso de exceder los valores permisibles.

3.- Medidores T1A y T2A

El tercer nivel de seguridad lo constituyen los medidores de temperatura TIA1 y TIA2 que paran la planta si las temperaturas de reacción medidas por uno de ellos o ambos superan los 910 GGG. Al recibir la señal de parada, las válvulas de solenoide SV1 y SV2 se abren para que el aire de instrumentos pueda escapar. Por tanto las válvulas PV1 y PV2 se cierran costando el paso del amoniaco al Venturi.

Con todos estos conocimientos se ha realizado el árbol de fallos que se muestra en la figura siguiente, para el que se han realizado las siguientes hipótesis:

- A.- Si las válvulas PV1 y PV2 fallan en posición abierta, el sistema de parada no es capaz de cerrarlas
- B.- Si el vapor de amoniaco arrastra gotitas de amoniaco liquido, ni el sistema de control ni el de seguridad pueden descubrir el consiguiente exceso de amoniaco en la mezcla y una explosión resulta posible.
- C.- Un aumento del diámetro o una obstrucción de los orificios D1 y D2 no se nota.
- D.- El fallo de los instrumentos de parada se incluye en el árbol aunque prácticamente todas las desviaciones mayores de su funcionamiento normal activan el sistema de parada de la planta.
- E.- La descarga del amoniaco a través de la válvula PV no es suficiente para cortar el suministro de los reactores, si no se cierra la válvula PV2.
- F.- Si la válvula PV no se abre, no se rompe ninguna válvula de tal manera que se produce una proporción demasiado alta de amoniaco en la mezcla.

Todas las hipótesis, con excepción de la f que representa un suceso altamente probable, se deben considerar como pesimistas. La suposición (B) se basa en hechos ya ocurridos y (E) esta confirmada por la experiencia de explotación de la planta.

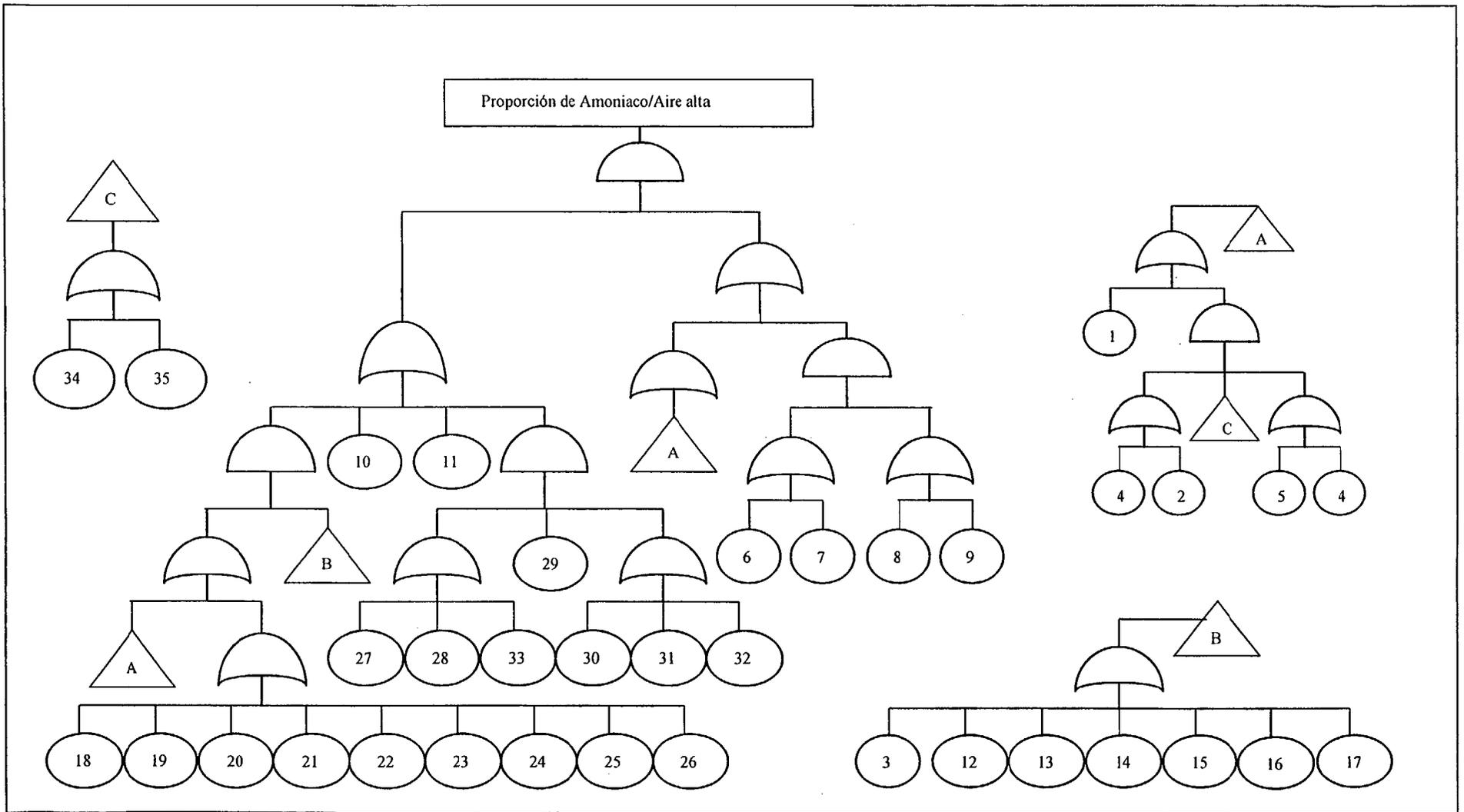


Fig 4.4.- ÁRBOL DE FALLOS PARA EL SUCESO "PROPORCIÓN DE AMONIACO/AIRE ALTA"

Ecuaciones del árbol de fallos

El árbol de fallos anterior permite escribir el siguiente conjunto de funciones booleanas que representan su estructura:

$$C(1) = S(4) \text{ OR } S(5)$$

$$C(2) = S(3) \text{ OR } S(2)$$

$$C(3) = C(2) \text{ AND } C(1)$$

$$C(4) = S(1) \text{ OR } C(3)$$

$$C(5) = S(3) \text{ OR } S(12) \text{ OR } S(13) \text{ OR } S(14) \text{ OR } S(15) \text{ OR } S(16) \text{ OR } S(17)$$

$$C(6) = S(18) \text{ OR } S(19) \text{ OR } S(20) \text{ OR } S(21) \text{ OR } S(22) \text{ OR } S(23) \text{ OR } S(24) \\ \text{OR } S(25) \text{ OR } S(26)$$

$$C(7) = C(6) \text{ OR } C(4)$$

$$C(8) = S(8) \text{ OR } S(9)$$

$$C(9) = S(6) \text{ OR } S(7)$$

$$C(10) = C(9) \text{ AND } C(8)$$

$$C(11) = C(4) \text{ OR } C(10)$$

$$C(12) = S(30) \text{ OR } S(31) \text{ OR } S(32)$$

$$C(13) = S(27) \text{ OR } S(28) \text{ OR } S(33)$$

$$C(14) = C(13) \text{ AND } S(29) \text{ AND } C(12)$$

$$C(15) = C(5) \text{ AND } C(7)$$

$$C(16) = S(10) \text{ OR } S(11) \text{ OR } C(15) \text{ OR } C(14)$$

$$C(17) = C(16) \text{ AND } C(11)$$

En las ecuaciones anteriores se ha empleado la letra S para designar a los sucesos básicos y la letra C para los sucesos compuestos ó puertas lógicas intermedias del árbol.

SUCESO BÁSICO		TASA DE FALLOS (x 10 ⁻⁶)	PROBABILIDAD DE FALLO A LA DEMANDA (x 10 ⁻²)
Sistema de Parada falla	X1		0.01
SV1 falla	X2	36.75	2.611
PV1 no cierra	X3	43.05	3.052
SV2 falla	X4	36.75	2.611
PV2 no cierra	X5	43.05	3.052
TIA-1 activación de parada falla	X6	13.32	0.954
TIA-1 falla el instrumento	X7	33.10	2.355
TIA-2 activación de parada falla	X8	13.32	0.954
TIA-2 falla el instrumento	X9	33.10	2.355
D1 orificio ampliado	X10	0.15	0.01079
D2 orificio obstruido	X11	0.05	0.000359
T1 señal baja	X12	1.25	0.08959
Fuga desde la tubería neumática medida del flujo de amoníaco	X13	1.15	0.08276
Relé MR falla	X14	6.20	0.4454
Controlador C1 falla	X15	33.10	2.355
T2 señal alta	X16	1.25	0.08959
Valor de Consigna	X17	14.84	1.06279
TI-1 valor alto	X18	46.81	3.3141
TI-2 valor bajo	X19	46.81	3.3141
TR-1 señal baja	X20	13.30	0.9530
TR-2 señal baja	X21	13.30	0.9530
PI-1 valor bajo	X22	110.66	7.6583
PI-2 valor bajo	X23	110.66	7.6583
Activación de parada falla	X24	13.32	0.954
CA1 falla	X25	13.30	0.9530
CA2 falla	X26	13.30	0.9530
V1 falla por cierre	X27	4.69	0.3371
V2 falla por cierre	X28	4.69	0.3371
Filtros F1 y F2 fallan	X29	0.25	0.01799
TIA-3 falla	X30	33.10	2.355
Alarma para la parada falla	X31	13.32	0.954
Operador no para la planta	X32		0.01
Cierre erroneo de V1 y V2	X33	0.60	0.04319
PV3 falla abierta	X34	43.05	3.052
SV3 falla	X35	36.75	2.611

Fig. 4.5- SUCESOS BÁSICOS Y PROBABILIDADES CONSIDERADAS

SOLUCIÓN Y OPTIMIZACIÓN DEL SISTEMA

Este ejemplo va a ser solucionado desde el punto de vista de un Ingeniero de Planta, cuya preocupación fundamental es anular los fallos conjuntos de cualquier tipo (Excepto los inherentes a fallo del sistema de parada en si) que alcancen o superen cierto umbral de probabilidad típico de este tipo de plantas. En este caso se consideran significativos los que aporten mas de un cinco por mil a la probabilidad del suceso peligroso " proporción de Amoniaco/Aire alta ".

RESULTADOS OBTENIDOS

El método de los Pesos evalúa completamente el árbol dando los siguientes valores

Indice	GCM	GCM sel.	Cota Inferior	Cota Superior
1	0	0	0.0000000000000000E+000	9.535846260648173E-002
2	11	2	1.076567941954836E-003	1.525489667084057E-002
3	24	2	1.624147687737810E-003	3.116421724175145E-003
4	261	0	1.763428034802701E-003	1.880740903788980E-003
5	54	0	1.784167198888709E-003	1.791302726011468E-003
6	0	0	1.786181959870516E-003	1.786526034960767E-003
7	0	0	1.786319732016614E-003	1.786333123235728E-003

El tiempo de CPU hasta asegurar cuatro cifras en la solución es menor a 1 segundo, siendo el total hasta el orden 7 de 4 segundos. El estudio de los Grupos de Corte Mínimos encuentra 350 de los cuales selecciona 4 por importancia, dos de ellos de orden dos y otros dos de orden 3. En las tablas que siguen se puede apreciar en detalle la configuración de todos los Grupos de Corte mínimo existentes de índice dos y tres así como su nivel de importancia sobre el suceso final no deseado " Proporción de Amoniaco/Aire alta".

GRUPOS DE CORTE MÍNIMO DE SEGUNDO ORDEN

Total de grupos encontrados = 11 (Seleccionados por importancia = 2)

Nº	G.C.M.		P E S O	IMPORTANCIA	SELECCIONADO POR IMPORTANCIA
1	1	11	0,0000000002094214	0,0000001168966079	
2	3	1	0,0000018364141852	0,0010250654304400	
3	3	4	0,0004922936204821	0,2747926414012909	SI
4	3	5	0,0005780599894933	0,3226664364337921	SI
5	10	1	0,0000000062949645	0,0000035137770737	
6	12	1	0,0000000523086818	0,0000291981068585	
7	13	1	0,0000000483175633	0,0000269703105005	
8	14	1	0,0000002609839953	0,0001456782629248	
9	15	1	0,0000014069086092	0,0007853202405386	
10	16	1	0,0000000523086818	0,0000291981068585	
11	17	1	0,0000006266322998	0,0003497789730318	

GRUPOS DE CORTE MÍNIMO DE TERCER ORDEN

Total de grupos encontrados = 24 (Seleccionados por importancia = 2)

Nº	G.C.M.			P E S O	IMP.	SELECCIONADO POR IMP.
1	5	2	15	1,1873126E-005	0,006984	SI
2	5	2	17	5,288250471E-006	0,003111	
3	5	2	14	2,202485609E-006	0,001296	
4	5	2	16	4,414413459E-007	0,00026	
5	5	2	12	4,414413459E-007	0,00026	
6	5	2	13	4,077596682E-007	0,00024	
7	5	2	10	5,312421327E-008	3,12E-05	
8	5	2	11	1,767340438E-009	1,04E-06	
9	2	4	15	1,0111519E-005	0,005948	SI
10	2	4	17	4,503636486E-006	0,002649	
11	2	4	14	1,875704376E-006	0,001103	
12	2	4	16	3,759450054E-007	0,000221	
13	2	4	12	3,759450054E-007	0,000221	
14	2	4	13	3,472606544E-007	0,000204	
15	2	4	10	4,524221129E-008	2,66E-05	
16	2	4	11	1,505121386E-009	8,85E-07	
17	7	9	10	3,661269240E-008	2,15E-05	
18	7	9	11	1,218033884E-009	7,16E-07	
19	7	8	10	1,462184574E-008	8,6E-06	
20	7	8	11	4,864406959E-010	2,86E-07	
21	9	6	10	1,462184574E-008	8,6E-06	
22	9	6	11	4,864406959E-010	2,86E-07	
23	6	8	10	5,839460553E-009	3,43E-06	
24	6	8	11	1,942676256E-010	1,14E-07	

Como se puede comprobar en los cuadros anteriores sólo existen cuatro grupos de corte cuya importancia sea superior al cinco por mil de la probabilidad de ocurrencia del suceso " Proporción de Amoníaco/Aire alta ", siendo los grupos de corte de segundo orden los de mayor valor, que son:

G.C.M.	DESCRIPCIÓN
3 4	PV1 no cierra y SV2 falla
3 5	PV1 no cierra y PV2 no cierra

Esto pone de manifiesto que todos los equipos con relevancia para la seguridad (Sistema de Control, Sistema eléctrico de alarma y Parada por exceso de temperatura) requieren el funcionamiento correcto de al menos una de las dos válvulas PV1 y PV2, que son componentes que los citados sistemas tienen en común. Por tanto, la redundancia del equipo que activa la parada se pierde al nivel de ejecución de la misma.

A la luz de las conclusiones anteriores la acción correctora más simple consiste en instalar una válvula adicional PV3 para cortar el flujo de Amoníaco. Este cambio se ha añadido en el diagrama del sistema y está representado en el árbol de fallos por la puerta de transferencia número 2, siendo la nueva ecuación del árbol:

- C(1) = S(4) OR S(5)
- C(2) = S(3) OR S(2)
- C(18) = S(34) OR S(35)
- C(3) = C(2) AND C(1) AND C(18)
- C(4) = S(1) OR C(3)
- C(5) = S(3) OR S(12) OR S(13) OR S(14) OR S(15) OR S(16) OR S(17)
- C(6) = S(18) OR S(19) OR S(20) OR S(21) OR S(22) OR S(23) OR S(24)
OR S(25) OR S(26)
- C(7) = C(6) OR C(4)
- C(8) = S(8) OR S(9)
- C(9) = S(6) OR S(7)
- C(10) = C(9) AND C(8)
- C(11) = C(4) OR C(10)
- C(12) = S(30) OR S(31) OR S(32)
- C(13) = S(27) OR S(28) OR S(33)
- C(14) = C(13) AND S(29) AND C(12)
- C(15) = C(5) AND C(7)
- C(16) = S(10) OR S(11) OR C(15) OR C(14)
- C(17) = C(16) AND C(11)

RESULTADOS PARA EL SISTEMA CORREGIDO

A continuación se resumen los resultados obtenidos por el método de los Pesos para la propuesta nuevo diseño del sistema

<u>Índice</u>	<u>GCM</u>	<u>GCM sel.</u>	<u>Cota Inferior</u>	<u>Cota Superior</u>
1	0	0	0.000000000000000E+000	0.113708079760779
2	9	0	4.058085642061593E-006	1.897017593483630E-002
3	12	4	6.540610149912115E-005	2.330672785201493E-003
4	293	0	1.098529753315345E-004	3.140434478142717E-004
5	36	0	1.216055291871698E-004	1.359882767766329E-004
6	36	0	1.233402438282762E-004	1.241513046130738E-004
7	0	0	1.235082337957143E-004	1.235455061184831E-004

Aunque ningún grupo de corte tiene ya la importancia mínima exigida es útil saber las combinaciones de fallos de elementos que contribuyen mas al suceso peligroso

" Proporción de Amoníaco/Aire alta ". Siendo estas:

Nº	G.C.M.	PESO	IMPORTANCIA
1	3 4 5	1,718174E-5	14,25 %
2	3 5 35	1,463250E-5	13,58 %
3	3 4 34	1,463250E-5	13,58 %
4	3 4 35	1,246148E-5	10,33 %

COMPARACION DE MÉTODOS

Aunque el método de los Pesos ha mostrado su capacidad para resolver completamente los árboles de esta magnitud, a continuación se presenta una comparación con los resultados obtenidos por los demás métodos, suponiendo que fuese necesaria su intervención. En el gráfico se observa la evolución de los estimadores del muestreo Daga Generalizado y PRD frente al Restringido en la que se ha adoptado escala logarítmica en el eje de ordenadas para mostrar el tipo de convergencia.

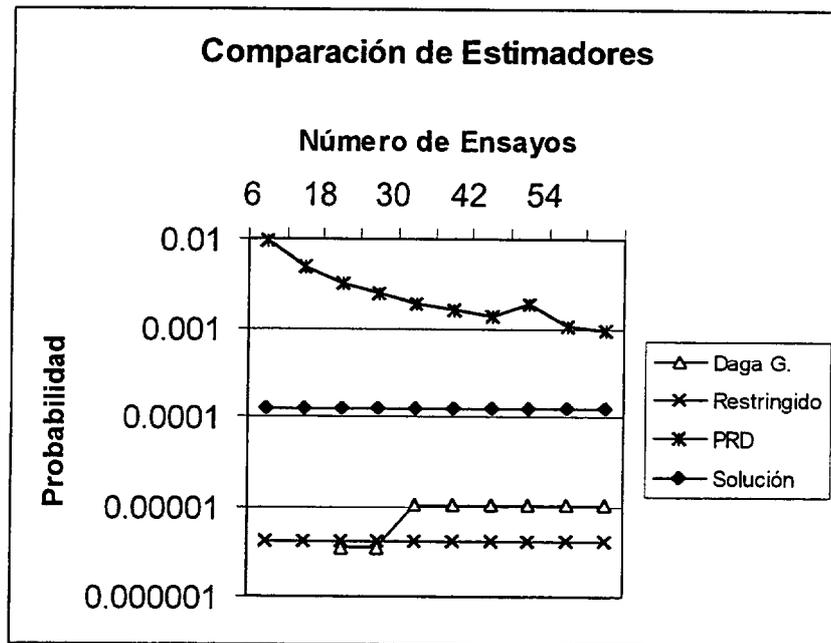


Fig. 4.6.- COMPARACIÓN DE ESTIMADORES

Se puede observar como el estimador PRD muestra una tendencia exponencial hacia la solución exacta, mientras que el Daga Generalizado converge a saltos y el Restringido aún no ha encontrado ninguna muestra válida siendo su valor el de la cota inferior determinista suministrada por el método de los Pesos hasta índice 2 que, por otra parte, es el mismo dato que se ha suministrado al PRD. El Estimador de Monte Carlo Crudo no se ha representado porque no ha encontrado ninguna muestra ya que necesita unas tres mil para comenzar a converger. Los tres estimadores obtiene la solución con más de dos cifras significativas sin alcanzar 1 segundo de tiempo en CPU.

RESUMEN Y COMENTARIOS

Se ha analizado utilizando los métodos desarrollados en esta tesis, el subsistema de combustión de Amoníaco y Aire de una planta de fabricación de Acido Nítrico. Utilizando el árbol de fallos para el suceso altamente peligroso definido como " Proporción de Amoníaco/Aire alta ", que podría causar una explosión en la planta, se ha localizado un punto débil en la instalación, que es común a todos los equipos con relevancia para la seguridad, cuantificando su importancia, así como todas las combinaciones de fallos que tienen una importancia o aportación a la probabilidad del

suceso peligroso superior al 5 por mil. Se ha definido e introducido la acción correctora mas oportuna, construyendo y evaluando el nuevo árbol de fallos resultante, en el que se comprueba que:

- La probabilidad del suceso altamente peligroso ha disminuido en un 93 %, pasando de un valor $1,7 \text{ E-}3$ antes de la acción correctora a $1,2\text{E-}4$ después de la misma.
- Excluyendo el fallo del sistema que realiza físicamente la parada, será necesario que fallen tres componentes para que se produzca la proporción de Amoníaco/Aire alta. Existen cuatro combinaciones de elementos posibles para ello que tengan importancia relevante y deberán ser, por tanto, contempladas en los planes de mantenimiento preventivo y programado con especial atención.

Se ha comprobado que el método de los Pesos es capaz de evaluar por si solo el árbol estudiado con cuatro cifras significativas en menos de 1 segundo de CPU. La observación del tipo de convergencia que presentan los estimadores PRD, Daga Generalizado y Restringido muestra convergencia exponencial solo en el primero, mientras que el segundo comienza a encontrar algún caso evaluable y el tercero ninguno.

4.2.3.- SISTEMA DE INYECCIÓN SPRAY EN LA CONTENCIÓN DE UNA CENTRAL NUCLEAR (CSIS)

Presentación

Este ejemplo presenta el árbol de fallos de un sistema de seguridad de una central nuclear, el Sistema de Inyección por aspersión (Spray) de agua borada al interior de la contención en el caso de producirse un accidente grande del tipo Perdida de Refrigerante (LOCA, Loss of Coolant Accident). Es el típico sistema para el cual el estudio de la No Disponibilidad o fallo a la demanda mediante árbol de fallos es particularmente útil. Ha sido tomado del WASH-1400 (1975)

Descripción

El sistema se compone de dos subsistemas idénticos, cada uno de ellos es capaz de inyectar 3200 GPM de agua a la atmósfera interior de la contención a través de aspersores de 360 °. Cada aspersor tiene 368 inyectores equidistantes y a 120 pies de altura. Ambos subsistemas succionan del RWST (Refueling Water Storage Tank) de tal modo que la solución de hidróxido de sodio enviada al mismo es extraída con preferencia por el CSIS. El sistema deberá funcionar durante la primera media hora después del LOCA. En la figura siguiente se presentan varias representaciones del sistema. Las posiciones de las válvulas de la figura son para el funcionamiento normal de la planta. Para que funcionen los subsistemas del CSIS será necesario que las válvulas (V5 o V6) y (V7 o V8) sean abiertas y que las bombas P1 y P2 arranquen, lo que ocurrirá por medio de una señal enviada al CSIS por el CLCS (Consequence Limiting Control System). El sistema esta diseñado con las siguientes bases:

- 1.- Cada subsistema Spray S1 o S2 será capaz de suministrar toda la inyección en spray suficiente para la atmósfera de la contención.
- 2.- Se requerirá su intervención solo hasta que se agota el suministro de agua en el RWST.

El árbol de fallos se ha construido tomando como suceso final no deseado a: Flujo de Fluido insuficiente a través de los inyectores del CSIS en la contención. En la figura siguiente se puede apreciar el árbol obtenido y en la figura los datos para su evaluación.

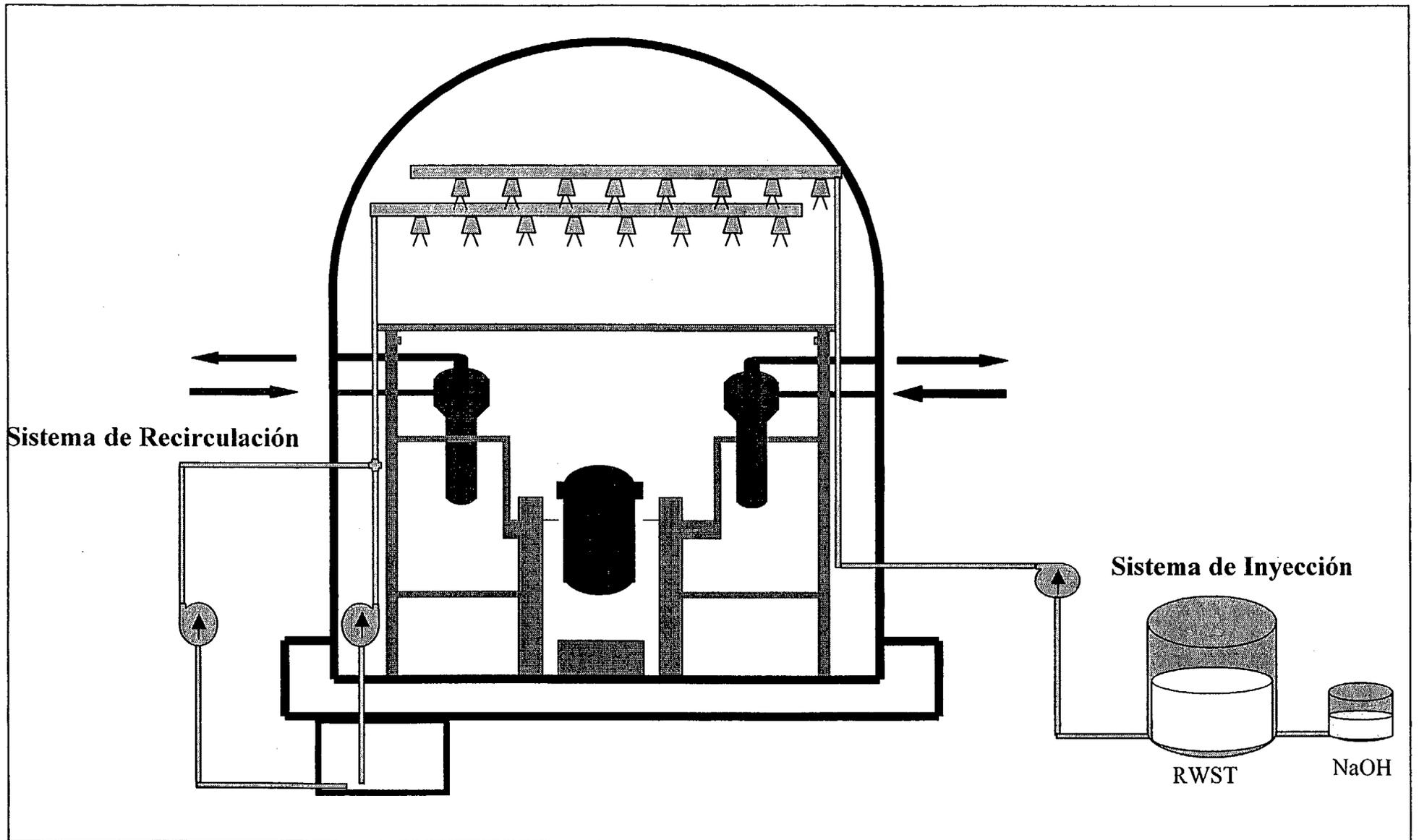


Fig. 4.7.- ESQUEMA DEL SISTEMA DE INYECCIÓN SPRAY EN LA CONTENCIÓN (CSIS)

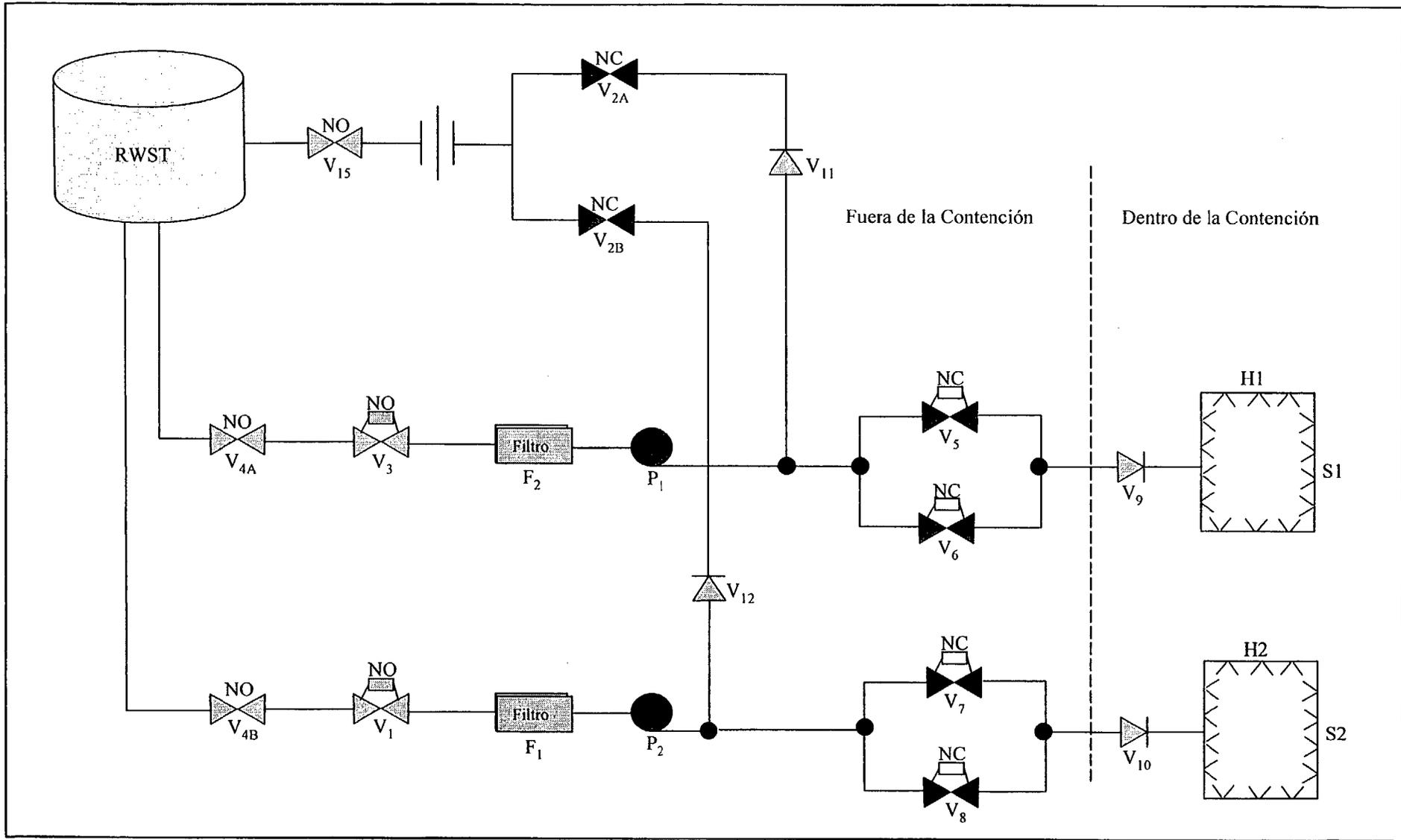


Fig. 4.8.- DIAGRAMA DE FLUJO SIMPLIFICADO DEL CSIS

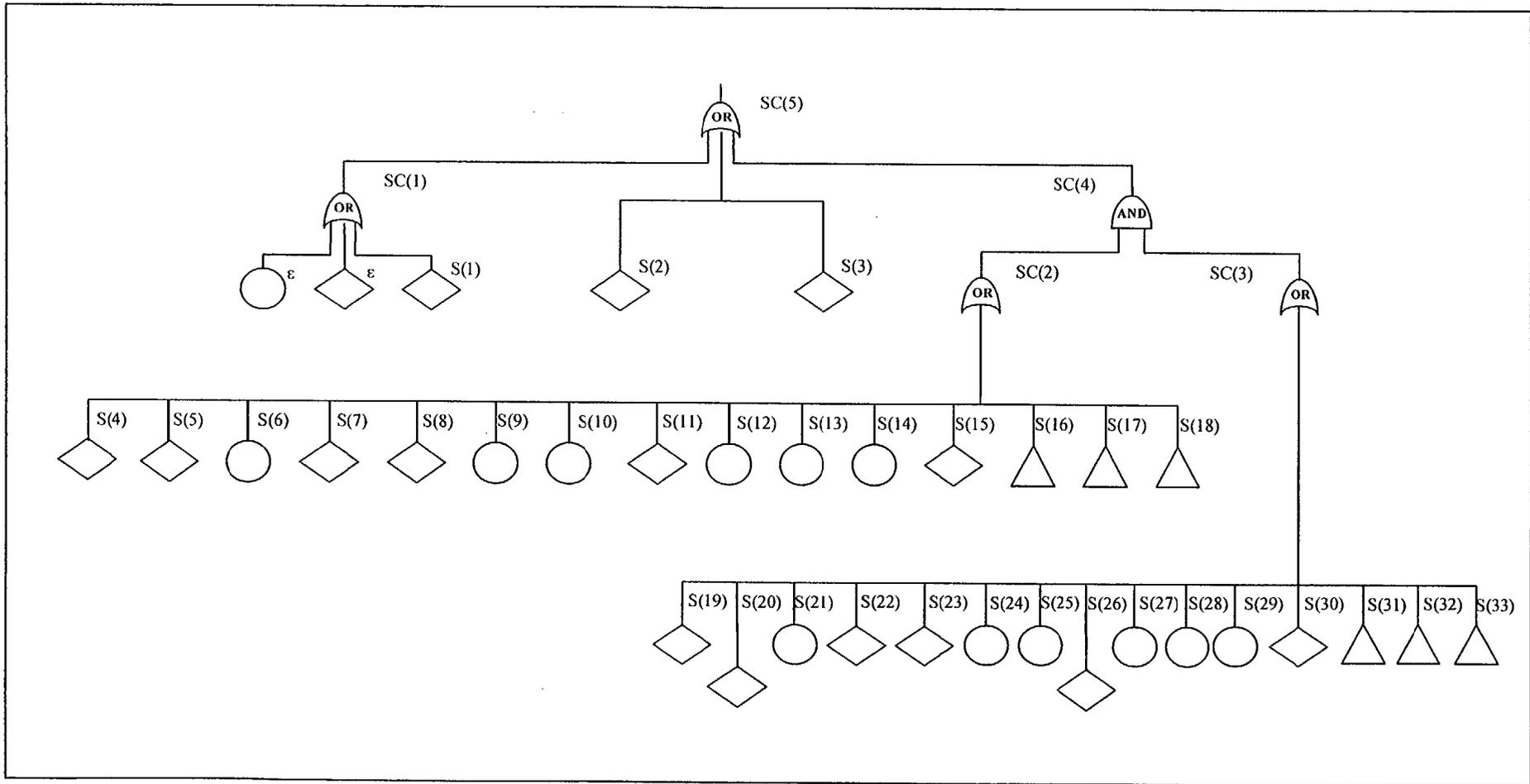


Fig. 4.9.- ÁRBOL DE FALLOS SIMPLIFICADO DEL CSIS

DATOS DE PARTIDA

El árbol de fallos se ha evaluado con las probabilidades siguientes:

SUCESO	PROB.	SUCESO	PROB.	SUCESO	PROB.
1	4,4E-7	12	1,0E-3	23	1,1E-4
2	1,5E-4	13	1,3E-5	24	3,0E-4
3	1,9E-3	14	1,0E-4	25	1,0E-10
4	1,0E-3	15	1,3E-4	26	1,0E-3
5	1,0E-10	16	4,6E-3	27	1,0E-3
6	1,0E-4	17	4,1E-5	28	1,3E-5
7	1,0E-2	18	1,1E-6	29	1,0E-4
8	1,1E-4	19	1,1E-3	30	1,3E-4
9	3,0E-4	20	1,0E-10	31	4,6E-3
10	1,0E-10	21	1,0E-4	32	4,1E-5
11	1,0E-3	22	1,0E-2	33	1,1E-6

RESULTADOS

La evaluación del árbol usando los diferentes métodos proporciona los siguientes resultados:

Índice	GCM Sel	Cota Inferior	Cota Superior
1	3	2 1.975378741518481E-003	2.596113096424157E-003
2	225	0 2.380674363867774E-003	2.386242781334591E-003
3	0	0 2.385694263930880E-003	2.385725305572017E-003

Instantáneamente el método de los Pesos encuentra la solución además de estudiar la existencia de Grupos de Corte Mínimo hasta índice 3 y seleccionando dos de índice 1 por ser su valor superior a 1.0E-4. El estudio completo hasta el índice 7 se realiza en 10 segundos de CPU y no existen más GCM hasta éste orden. Los GCM de orden uno seleccionados son:

GCM nº	SB	Peso del GCM seleccionado
1	3	1.830681104965133E-003
2	2	1.442744946635968E-004

El muestreo Daga Generalizado alcanza en 2 segundos de CPU el valor 2.385669E-003 realizando un total de 14.285.715 muestras daga, calculando además las probabilidades de las puertas lógicas, que son:

<u>PL n°</u>	<u>PROBABILIDAD</u>
1	0.000000420000
2	0.018304578960
3	0.018395718187
4	0.000334669981
5	0.002385669854
6	0.005887909792
7	0.005888049491

El Monte Carlo Crudo tarda 2 minutos 30 segundos en realizar los 14.285.715 muestras y alcanza el valor final de 2.3387857E-03. El método PRD se ha lanzado suponiendo que el método de los Pesos solo alcanza el índice 1, por lo que la cota superior será: 2.596113096424157E-003, y la inferior: 1.975378741518481E-003. En estas condiciones se alcanza el valor 2.37957621E-03 en 1 segundo y encontrando 8233 muestras significativas. El muestreo Restringido obtiene después de dos minutos 30 segundos el valor 2.3549760E-03.

COMENTARIOS

Todos los método propuestos en este trabajo muestran un buen comportamiento tanto en tiempo de CPU como en precisión. La conclusión más importante es que el método determinista de los Pesos es capaz de evaluar por si solo un árbol de fallos de esta magnitud, lo cual coincide con lo concluido en el ejercicio práctico anterior. El muestreo Gaga Generalizado obtiene una precisión mayor que el Restringido y al igual que el método PRD, teniendo éste último la ventaja de realizar los cálculos en la mitad de tiempo para alcanzar estas precisiones.

4.2.4.- SISTEMA DE SUMINISTRO ELÉCTRICO DESDE UNA GRAN CENTRAL

Presentación

En este ejemplo presentamos un sistema de suministro eléctrico de una gran central de tipo nuclear, fue utilizado por Garribba et al.(1977) para presentar un método de obtención de Grupos de Corte Mínimos a partir de un árbol de fallos, ilustra todos los aspectos que se han presentado en los restantes ejemplos de forma conjunta y además presenta un árbol de considerables proporciones.

Descripción

Una central nuclear proporciona energía a una red de suministro eléctrico mediante cuatro líneas, dos de ellas a una tensión de 380 KV. (L3 y L4) y las otras dos a 150 KV. (L1 y L2), la tensión en generación es de 6 KV, por lo que ha de ser elevada a las tensiones de transporte mediante transformadores (T1,T2,TO12,TO22). Es de destacar la multiplicidad típica de estos sistemas, en los que cada punto con carga puede ser suministrado al menos por dos caminos diferentes y cada protección es al menos doble, lo que conduce a árboles de fallos con ramas redundantes y en general a probabilidades de fallo muy pequeñas, así como a la posible presencia de Grupos de Corte con probabilidad mensurable pero de índice muy alto.

El esquema completo se puede apreciar en la figura siguiente, el fallo que se desea estudiar es la ausencia de energía eléctrica en el BUS 12B que trabaja a 380 V y proporciona energía a circuitos internos de seguridad de vital importancia (seguridad y control), también se puede apreciar el árbol de fallos propuesto por los autores. En esta tipo de instalaciones ha de prestarse especial atención a la existencia de Causas Comunes de fallos de varios sistema a la vez, tales como Fuego, Explosiones, Caídas y Roturas de dispositivos y Líneas, las cuales han de ser consideradas para la correcta evaluación del árbol de fallos resultante. Los autores no publicaron las probabilidades de los sucesos básicos, por lo que ha sido evaluado en las condiciones mas desfavorables para los métodos desarrollados.

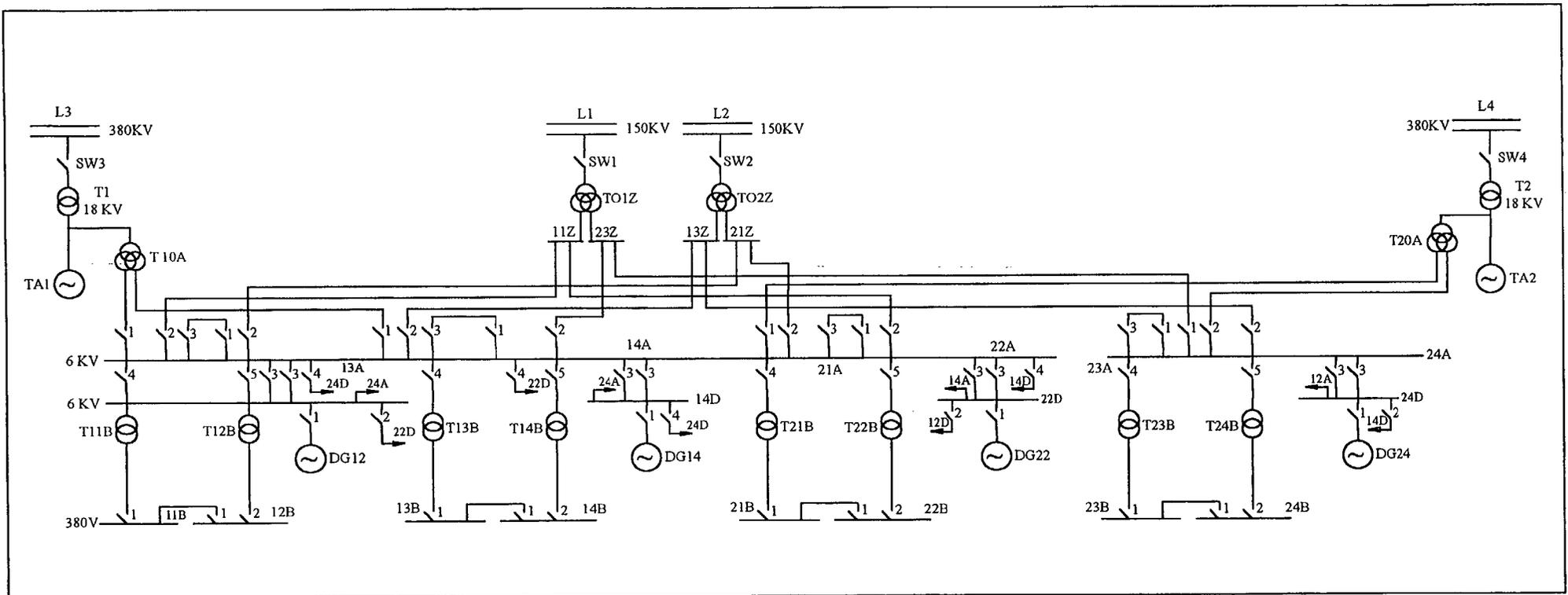


Fig. 4.10.- SISTEMA DE SUMINISTRO ELÉCTRICO DE UNA GRAN CENTRAL

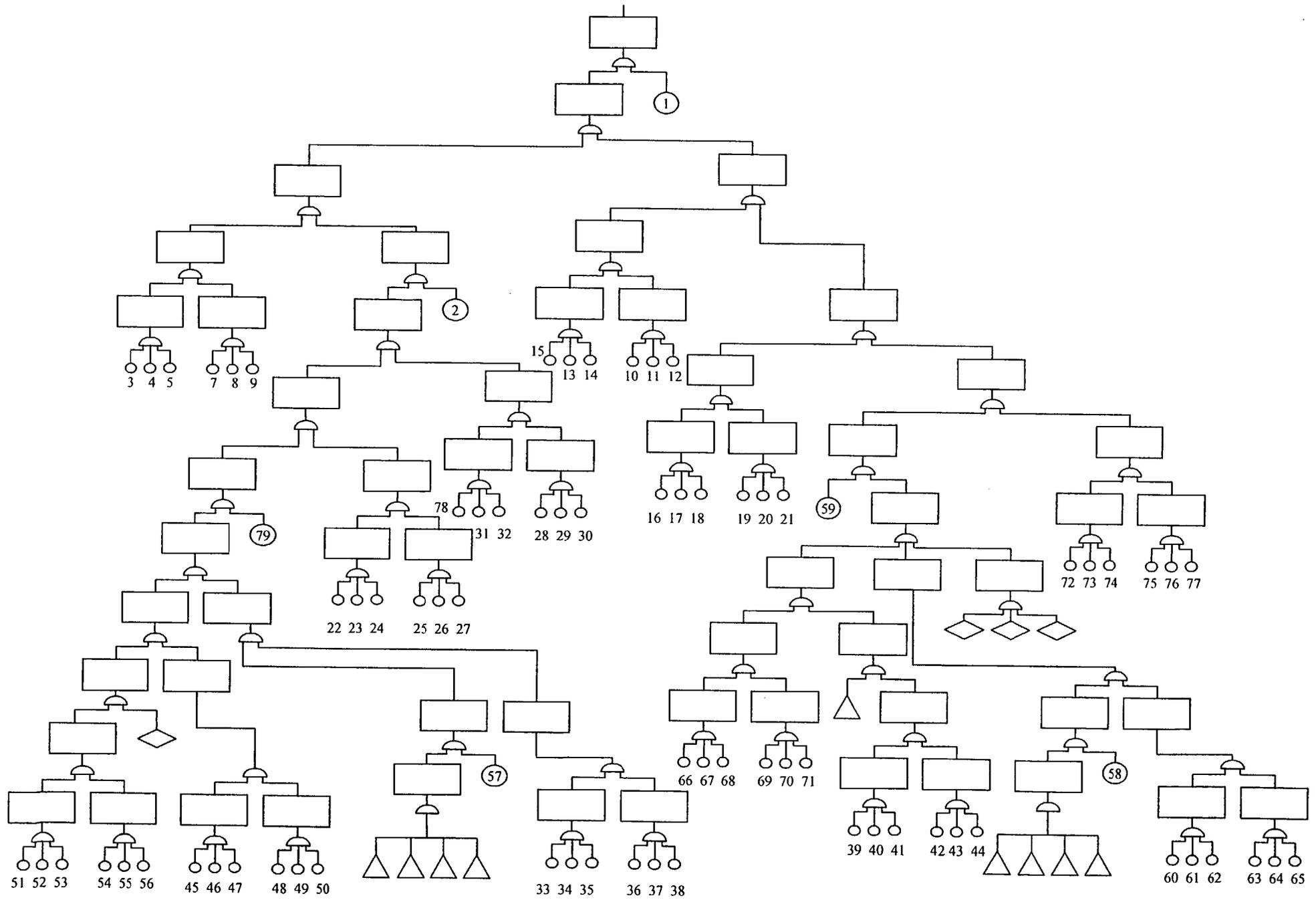


Fig. 4.11.- ÁRBOL DE FALLOS DEL SISTEMA ELÉCTRICO DE UNA GRAN CENTRAL

Para el estudio de la evaluación cuantitativa de éste árbol hemos usado unas condiciones de test de uso frecuente entre los investigadores del área (Rauzy y Dutuit, 1997), consistentes en realizar todas las pruebas para cuatro conjuntos de probabilidades, siendo en cada conjunto todas las probabilidades iguales, de valor 0.1, 0.01, 0.001 y 0.0001 respectivamente. Cualquier otro conjunto de probabilidades quedará entre las aquí consideradas.

RESULTADOS OBTENIDOS CON LAS PROBABILIDADES a 0.1

<u>Índice GCM</u>	<u>Cota Inferior</u>	<u>Cota Superior</u>	<u>tiempo CPU</u>
1	1	8.464149782874062E-006	0.999170513321278 -
2	380	4.495403995793158E-004	0.995845042917696 -
3	0	3.578767923012230E-003	0.986698118385651 1 seg.
4	0	1.523693254851336E-002	0.968688915544109 4 seg.
5	0	4.455571147855014E-002	0.941310058853558 45 min.

Fig. Resultados obtenidos con el método de los Pesos.

Ensayos	Crudo	Daga G.	PRD
6	0.6666667	0.4000000	0.6640465
12	0.8333333	0.4000000	0.6640465
18	0.8888889	0.6500000	0.7193463
24	0.9166667	0.5666667	0.7055213
30	0.9333333	0.7333333	0.7304062
36	0.9444444	0.6750000	0.7469962
42	0.9523840	0.6600000	0.7825460

Fig. 4.12.- DETALLE DE LOS PRIMEROS ENSAYOS CON CADA TIPO DE MUESTREO

El muestreo Daga Generalizado alcanza el valor 0.7858999 instantáneamente y el método PRD el valor 0.78259247 también sin tiempo mensurable. La solución exacta es 7.783E-1.

Se observa que el hecho de tener probabilidades altas perjudica al método de los pesos pero beneficia a los Monte Carlo ya que se encuentran muchas muestras rápidamente.

RESULTADOS OBTENIDOS CON LAS PROBABILIDADES a 0,01

Indice GCM		Cota Inferior	Cota Superior
1	1	4.088201744225494E-003	0.231418072085605
2	380	2.345554132040505E-002	8.539906837276834E-002
3	0	3.594664319089721E-002	4.888680926744415E-002
4	0	4.017723483768563E-002	4.235151098683110E-002
5	0	4.114445226351330E-002	4.144828981605886E-002

Fig. 4.13.- RESULTADOS OBTENIDOS CON EL MÉTODO DE LOS PESOS.
(Mismos tiempos de CPU que en los casos anteriores.)

Crudo		PRD Índice >2 c.i. = 2.3455542E-02 c.s. = 8.5399069E-02	
Nº casos	Probabilidad	Nº casos	Probabilidad
1000000	4.0369999E-02	100	4.3081213E-02
2000000	4.0580001E-02	200	4.1547958E-02
3000000	4.0453333E-02	300	4.1036874E-02
4000000	4.0364999E-02	400	4.1394632E-02
5000000	4.0456001E-02	500	4.0995985E-02
6000000	4.0461667E-02	600	4.0730223E-02
7000000	4.0454287E-02	700	4.1153692E-02
8000000	4.0386248E-02	800	4.0857993E-02
9000000	4.0432222E-02	900	4.1105017E-02
10000000	4.0449001E-02	1000	4.1118644E-02
8 segundos de CPU		< 1 segundo de CPU	

Fig. 4.14.- COMPARACIÓN PRD-DIRECTO

El PRD a partir de los valores finales de la tabla se mantiene muy estable entorno a esta solución. Por lo que es mas beneficioso en tiempo resolver por el método de las cotas hasta el orden dos (instantáneo) y continuar por PRD (instantáneo).

Comparación entre métodos para probabilidades 0.01

Núm. de Casos	Crudo	Daga G.	Restringido índice >2 c.i. = 2.3455542E-02 c.s. = 8.5399069E-02	PRD Índice >2 c.i. = 2.3455542E-02 c.s. = 8.5399069E-02
5	0.0000000E+00	0.0000000E+00	3.5844248E-02	3.3779465E-02
10	0.0000000E+00	0.0000000E+00	2.9649895E-02	3.3779465E-02
15	0.0000000E+00	0.0000000E+00	2.7585110E-02	3.0338155E-02
20	0.0000000E+00	0.0000000E+00	2.9649895E-02	4.1522402E-02
25	0.0000000E+00	0.0000000E+00	3.0888764E-02	3.9973818E-02
30	0.0000000E+00	9.9999999E-03	3.1714678E-02	3.8941424E-02
35	2.8571429E-02	9.9999999E-03	3.2304619E-02	3.9678849E-02
40	2.5000000E-02	9.9999999E-03	3.4295660E-02	4.1522402E-02
45	2.222223E-02	9.9999999E-03	3.5844248E-02	4.1809179E-02
50	3.9999999E-02	9.9999999E-03	3.4605376E-02	4.1006207E-02
55	3.6363635E-02	2.0000000E-02	3.4717999E-02	4.1287769E-02
60	5.0000001E-02	9.9999999E-03	3.5844248E-02	3.3779465E-02

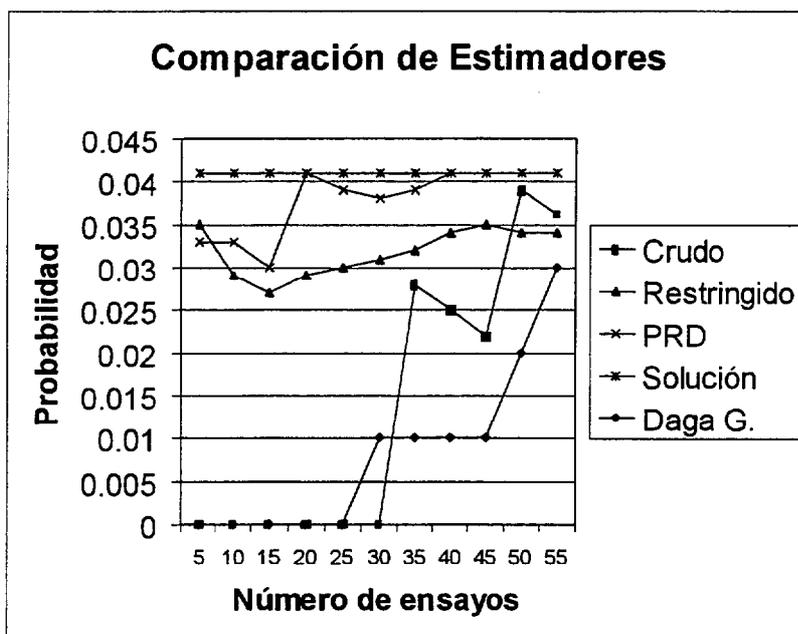


Fig. 4.15.- COMPARACIÓN DE ESTIMADORES

En la que el error medio cometido por el método PRD es un 88,61% menor que el cometido por el muestreo Crudo, un 66,33% menor que el cometido por el muestreo Restringido y un 90 % menor que el cometido por el Daga Generalizado. Este caso, como el anterior, muestra los problemas de "despegue" del estimador Daga generalizado cuando las probabilidades son todas iguales. Constituye para este estimador la peor circunstancia de muestreo efectivo. De todas formas y en todos los casos, transcurridos unos ensayos más, estimador se recupera rápidamente y supera ampliamente la

exactitud del Restringido y el Crudo. En este caso alcanza el valor 4.106E-02 en 2 segundos de CPU.

RESULTADOS OBTENIDOS CON LAS PROBABILIDADES a 0.001 y 0.0001

	<u>Índice GCM</u>	<u>Cota Inferior</u>	<u>Cota Superior</u>
1	1	9.148048366926450E-004	4.692337678406600E-003
2	380	1.344277778042858E-003	1.454349787897269E-003
3	0	1.371727395470354E-003	1.374112866306398E-003
4	0	1.372648706599858E-003	1.372689581656261E-003
5	0	1.372669581711011E-003	1.372670732052605E-003

Fig. 4.16.- RESULTADOS OBTENIDOS CON PROBABILIDADES A 0.001

	<u>Índice GCM</u>	<u>Cota Inferior</u>	<u>Cota Superior</u>
1	1	9.911390466797493E-005	1.389297094673836E-004
2	380	1.037628116876058E-004	1.038795278001103E-004
3	0	1.037924982744503E-004	1.037927537581584E-004
4	0	1.037925978306603E-004	1.037925815519092E-004
5	0	1.037925981386618E-004	1.037925815519092E-004

Fig. 4.17.- RESULTADOS OBTENIDOS CON PROBABILIDADES A 0.0001

El método de los Pesos llega instantáneamente a la solución en ambos casos. El Daga Generalizado llega instantáneamente a 0.000101 con 100 matrices daga. No se emplean los muestreos Restringido y PRD dado que ya se ha llegado a la solución.

RESUMEN Y COMENTARIOS

Se ha estudiado un árbol de fallos de un sistema eléctrico de una gran Central el cual es representativo de los que pueden presentarse en centrales tanto Térmicas como Nucleares. Por su número de sucesos básicos puede considerarse que esta en el límite de los de mediano tamaño o en el principio de los de gran tamaño. Se han utilizado cuatro conjuntos de probabilidades todas iguales para los sucesos básicos, desde 0.1 hasta 0.0001. El conjunto de probabilidades 0.1 ha resultado el más desfavorable para el método de los pesos, en cuanto a precisión se refiere, pero el más favorable para los muestreos de Monte Carlo. Aún así el árbol ha podido ser evaluado eficientemente con el método PRD en todos los casos (con precisión suficiente y en un tiempo inferior a 2 segundos de CPU).

4.2.5.- ESTUDIO DE APLICACIONES A GRANDES ÁRBOLES DE FALLOS

Una vez comprobados y validados los métodos y programas desarrollados con las aplicaciones anteriores queda por estudiar la capacidad de los mismos para analizar grandes árboles de fallos, con diferentes magnitudes y complejidad lógica. Para ello es conveniente disponer de un número elevado de árboles de estas características lo cual es bastante complicado dado el carácter confidencial que suelen dar a sus trabajos las compañías e investigadores dedicados a ello. Con objeto de salvar este inconveniente se ha desarrollado una aplicación auxiliar capaz de escribir las ecuaciones booleanas de árboles de cualquier magnitud y/o complejidad según se le especifique en ciertos parámetros de entrada. Con ella hemos obtenido un conjunto de árboles suficientes para averiguar el comportamiento de las aplicaciones principales. En las tablas siguientes se presenta una muestra de los resultados obtenidos y seguidamente se suman los resultados obtenidos sobre más de 100 árboles estudiados en diferentes condiciones.

En primer lugar se ha realizado un estudio, típico entre las publicaciones del área, en el cual se evalúan los árboles con conjuntos de probabilidades todas iguales que recorren desde la probabilidad 0.1 hasta la 0.0001, con ello se explora el comportamiento de la evaluación en condiciones extremas, dado que las probabilidades todas iguales y altas lo son para árboles de fallos.

Seguidamente se realiza el estudio con conjuntos de probabilidades, que emulan a aquellas que pueden aparecer en la mayor parte de las evaluaciones reales, obtenidos mediante variaciones aleatorias sobre uno inicial al que hemos denominado Probabilidades Base (expandido de una muestra obtenida del WASH 1400). Con esto se pueden explorar los límites de aplicación razonable de los métodos para procesos de optimización, para los cuales hemos considerado deseable que los tiempos de ejecución sean del orden de segundos. Como límite de tiempos hemos explorado hasta 2 min. 30 segundos efectivos de CPU.

Finalmente se realiza un estudio sobre todos y cada uno de los factores que pueden influir en la evaluación, reflejando este capítulo los resultados del mismo.

Árbol	NSB	PL	Pr = 0.1	Pr = 0.01	Pr = 0.001	Pr = 0.0001
Ar012	160	287	S = 0.9214 DG = 0.9233(<1 seg.) Cs = 0.9999 0.9204(<1seg.) Ci = 2.139E-6	S = 0.2064 DG = 0.210 (<1 seg.) Cs = 0.3326 0.2075(<1seg.) Ci = 0.1164	S = 0.0227 DG = 0.0227 (<1 seg.) Cs = 0.02254 0.023(<1seg.) Ci = 0.02313	S = 0.0022976 DG = 0.0023(<1 seg.) Cs = 0.002298 0.00229(<1seg.) Ci = 0.00297
Ar015	218	123	S = 0.9951 DG = 0.997 (<1 seg.) Cs = 0.9999 0.9961(<1seg.) Ci = 6.29E-08	S = 0.2323 DG = 0.2305(<1 seg.) Cs = 0.4434 0.2339(<1seg.) Ci = 0.07133	S = 0.0171 DG = 0.0160(<1 seg.) Cs = 0.01808 0.01704(<1seg.) Ci = 0.0166	S = 0.0016119 DG= 0.00159(<1 seg.) Cs = 0.00161302 0.00161(<1seg.) Ci = 0.00161134
Ar037	306	337	S = 0.9294 DG = 0.921(<1 seg.) Cs = 0.9999 0.890 (<1seg.) Ci = 6.767E-13	S = 0.1772 DG = 0.1778(<1 seg.) Cs = 0.4387 0.1627(<1seg.) Ci = 0.07246	S = 0.017 DG = 0.0170 (<1 seg.) Cs = 0.01736 0.001708(<1seg.) Ci = 0.01708	S = 0.001701 DG= 0.00170(<1 seg.) Cs = 0.001705 0.001701 (<1seg.) Ci = 0.001701
Ar052	400	162	S = 1.000 DG = 1.000 (<1 seg.) Cs = 1.000 1.000 (<1seg.) Ci = 3.73E-16	S = 0.8791 DG = 0.875 (<1 seg.) Cs = 0.9036 0.8747 (<1seg.) Ci = 0.1403	S = 0.1639 DG = 0.1639 (<1 seg.) Cs = 0.1650 0.1642(<1seg.) Ci = 0.1572	S = 0.0173 DG= 0.01730(<1 seg.) Cs = 0.01739 0.01738(<1seg.) Ci = 0.01738
Ar112	500	500	S = 0.888 DG = 0.895 (<1 seg.) Cs = 1.000 0.890 (<1seg.) Ci = 1.47E-21	S = 0.170 DG = 0.173 (<1 seg.) Cs = 0.8837 0.174(<1seg.) Ci = 0.00716	S = 0.0178 DG = 0.0179 (<1 seg.) Cs = 0.0306 0.01776(<1seg.) Ci = 0.0163	S = 0.001799 DG= 0.00179(<1 seg.) Cs = 0.001817 0.001797 (<1seg.) Ci = 0.00179

© Universidad de Las Palmas de Gran Canaria. Biblioteca Digital. 2003

Fig. 4.18.- MUESTRA DE RESULTADOS OBTENIDOS PARA ALGUNOS
ÁRBOLES CON PROBABILIDADES IGUALES

Árbol	NSB	PL	Solución	Daga Gen.	Restringido	PRD
Ar012	160	287	4.908E-03	4.923E-03 1 min. 30 seg.	Falla al partir de Indice>2	4.9095E-03 solo Pesos (I=3) 30 seg.
Ar015	218	123	4.4E-03	4.419E-03 1 min. 45 seg.	Falla al partir de Indice>2	0.004401974 solo Pesos (I=2) 15 seg.
Ar037	306	337	0.091	0.09437 2 min.	Falla al partir de Indice>2	0.90711 1 min. 15 seg.
Ar052	400	162	0.31899	0.326322316 1 min.	Falla al partir de Indice>2	0.3170347 40 seg. Indice 2
Ar112	500	500	2.49E-03	2.493E-03 2 min.	2.29E-03 1 min. Indice 2	2.4710E-03 1 min. 45 seg. Indice 2

Fig. 4.19.- MUESTRA DE RESULTADOS OBTENIDOS PARA DIFERENTES ÁRBOLES Y PROBABILIDADES

A continuación se resumen los resultados y conclusiones obtenidos analizando en primer lugar cada método por separado y finalmente su aplicación conjunta en el método PRD

El Método de Los Pesos

El método de los Pesos ha mostrado ser capaz de evaluar cuantitativamente por si solo árboles de fallos de hasta 230 sucesos básicos, con cualquier número de puertas lógicas. El número de sucesos básicos del árbol condiciona el índice máximo (número máximo de unos por vector de estado) explorable por el método, así, para arboles de 100 sucesos básicos e índices mayores que 7 se cae rápidamente en el problema del crecimiento exponencial del número de soluciones, reduciéndose este índice hasta el valor 2 en los árboles de 500 sucesos básicos. El punto más negativo del método es la existencia de probabilidades altas (>0.01) entre las de los sucesos básicos, lo que hace que las cotas encontradas estén muy lejanas entre si.

Daga Generalizado

El muestreo Daga es aplicable individualmente ya que su reducción de varianza se basa en establecer correlación negativa entre las muestras. La experimentación práctica ha demostrado que este método es capaz de valorar cuantitativamente árboles de fallos de hasta 700 sucesos básicos con cualquier número de puertas lógicas. Su factor más negativo es la presencia de valores extremos (grandes o pequeños) de la probabilidad de los sucesos básicos. La presencia conjunta de probabilidades mayores que 0.01 y/o menores que 10^{-7} para los sucesos básicos de un mismo árbol de fallos conducen a pérdidas de eficiencia, por lo que el algoritmo ha de truncar por las probabilidades más bajas para permitir la coherencia (correlación negativa) entre todas las muestras, de todas formas más allá de los 700 sucesos básicos se pierde eficiencia, ya sea porque los tiempos de CPU superan ampliamente los 30 minutos o bien porque la precisión de la respuesta disminuye debido al truncamiento. En los árboles de menos sucesos básicos se observa un incremento en los tiempos de ejecución debido a que el algoritmo admite probabilidades muy bajas (truncamiento escaso), lo que implica que obtiene un alto número de muestras que han de ser estudiadas. En los ensayos se ha observado que el tamaño de árbol que optimiza el tiempo de ejecución está entre 300 y 400 sucesos básicos. La reducción del error medio respecto al cometido por el Monte Carlo Crudo fue de un 53% con una desviación típica del 8%, mientras que siempre se obtuvieron al menos dos cifras significativas exactas de la solución buscada.

Muestreo Restringido

El muestreo Restringido sólo consigue reducción de varianza respecto al Monte Carlo Crudo cuando se le suministran cotas seguras lo suficientemente cercanas, lo que significa en la práctica que va precedido del método de los pesos y que la reducción de varianza está fuertemente condicionada por este. En la práctica ha mostrado un comportamiento muy pobre al obtener cotas por el Método de los Pesos, de hecho cuando las cotas obtenidas por este están muy próximas, situación deseable, en la mayoría de los casos es necesario un número de ensayos tan alto que dispara los tiempos de CPU que permiten obtener precisión aceptable. Por tanto ambos métodos (Pesos y Restringido), en lugar de cooperar, compiten entre si, el tiempo que se gana

aproximando lo mas posible las cotas por el método de los Pesos, se pierde intentando aproximar la solución con el muestreo Restringido.

Método PRD

Este método ha mostrado un buen comportamiento en cuanto a precisión y tiempos de CPU hasta árboles de tamaño sobre 500 sucesos básicos. En primer lugar el método de los Pesos, como ya se ha establecido, es capaz de evaluar por si la mayoría de los árboles de hasta 230 sucesos básicos (excepto aquellos que poseen probabilidades muy altas en promedio), siendo el índice máximo alcanzado en los mas grandes el 2. En segundo lugar el estimador Restringido con el Daga Generalizado insertado ha sido capaz de terminar la evaluación con error aceptable y en tiempos que nunca has sobrepasado los 2 minutos 30 segundos de CPU. El método muestra un balance muy positivo entre el uso del cálculo intensivo sobre los primeros índices (mediante los Pesos) y el uso del muestreo selectivo sobre los restantes (muestreo Restringido-Daga). O dicho de otra forma; en primer lugar se evalúan todos los contribuidores a ambas cotas mientras el tiempo de CPU es admisible y en segundo lugar se evalúan solo los contribuidores más importantes del resto pero con alta correlación negativa. Un factor que optimiza la aplicación conjunta es el hecho de que el muestreo Daga aumenta su velocidad por el hecho de no tener que considerar todas aquellas muestras que ya fueron consideradas por el método de Los Pesos, las cuales suelen ser la mayoría, por lo que los tiempos conjuntos disminuyen pero aumenta la precisión. De todas formas este equilibrio se rompe a partir del entorno de los 550 sucesos básicos, por lo que consideramos como un límite razonable de aplicación del método para procesos de optimización el de 500 sucesos básicos.

Un valor añadido del método PRD es el de que en todos los casos el método de los Pesos determina los Grupos de Corte Mínimos hasta el índice que realiza los cálculos, además los clasifica, almacena y selecciona por su importancia (valor) aquellos que estén por encima de un umbral determinado elegido por el usuario. A pesar del ligero esfuerzo computacional que ello significa, se ha preferido mantener dicha opción dado que esta información siempre puede ser útil a la hora de optimizar eficientemente determinados sistemas técnicos.

Estudio factorial del Método PRD

Los factores de los que depende el método PRD son: Número de puertas lógicas, Complejidad de la estructura del árbol, Número de sucesos básicos y Probabilidades de los mismos. En base a los resultados obtenidos en la experimentación práctica de los algoritmos que implementan el método PRD se puede concluir que:

- El método siempre producirá una salida o probabilidad del suceso top del árbol de fallos que se evalúe sean cuales sean los valores de los cuatro factores descritos. Esta probabilidad obtenida será mas o menos precisa en función de la combinación de factores que se de en cada caso. En los siguientes párrafos se muestran las conclusiones obtenidas tras la experimentación práctica.
- El número de puertas lógicas y/o la complejidad de la estructura no son determinantes para la precisión alcanzada por el método. El único límite aparente para el número de puertas lógicas es el de líneas de código compilable que admita la CPU disponible, lo cual permite en cualquier caso práctico superar las diez mil puertas lógicas. Esta cifra es suficiente para los árboles que se espera puedan aparecer en las instalaciones más grandes o sistemas de alto nivel de integración.
- El número de sucesos básicos del árbol condiciona el índice máximo (número máximo de unos por vector de estado) explorable por el método de los Pesos, sin embargo, se han podido evaluar árboles de hasta 500 sucesos básicos con el PRD dadas las características ya mencionadas anteriormente.
- El conjunto de probabilidades de los sucesos básicos ha sido en la práctica el factor más influyente sobre el método, hasta tal punto que es el único factor realmente limitativo de la precisión alcanzable. El análisis detallado muestra las siguientes características:
 - Si las probabilidades son todas muy altas el método de los Pesos pierde precisión ya que aproxima muy poco las cotas superior e inferior de la probabilidad del suceso top, pero los estimadores Restringido-Daga ganan

precisión ya que encuentran muchas muestras (significativas y negativamente correladas) con poco esfuerzo computacional. El resultado global es que la probabilidad buscada es aproximada eficientemente.

- Si las probabilidades son todas muy bajas el método de los Pesos gana eficiencia y es capaz de aproximar en gran medida las cotas, en la mayor parte de los casos no es necesaria la intervención subsiguiente de los estimadores Restringido-Daga.
- Cuando las probabilidades de los sucesos básicos comprenden tanto valores altos como valores bajos, caso esperable en la práctica, la combinación PRD ha mostrado un comportamiento muy aceptable en cuanto a precisión/tiempo de CPU. Las reducciones de error cometido al estimar la probabilidad del suceso top con el método respecto del Monte Carlo Crudo han sido en valor medio de un 88% con una desviación típica del 6%, obtenidas para mas de 100 muestras diferentes partiendo del conjunto de probabilidades base. Las reducciones con respecto al Muestreo Daga y Muestreo restringido fueron de media 33% y 56% con desviaciones típicas del 4% y 7% respectivamente, para las mismas condiciones de test. Los tiempos de CPU no superaron los 2 minutos 30 segundos, siendo su valor medio de 45 segundos, para llegar a las soluciones con al menos dos cifras significativas exactas.

CAPÍTULO V

CONCLUSIONES Y LÍNEAS FUTURAS

5.1.-PRÓLOGO

La necesidad de optimizar desde diversos puntos de vista (multicriterio) los cada vez mas complicados sistemas técnicos, en los cuales la Fiabilidad y/o la Disponibilidad es un factor importante (Andrews y Pattison, 1997)(Yansong y Dugan, 1998)(Levitin et al., 1998)(Cancela y Khadiri, 1998)(Pierre y Elgibaoui, 1997)(Bouissou y Bogarde, 1997), ha reforzado la necesidad de disponer de métodos de evaluación cuantitativa de árboles de fallos capaces proceder con seguridad ante cualquiera de ellos sea cual sea su tamaño y/o su complejidad. Los métodos desarrollados hasta la actualidad, algunos de los cuales son muy potentes, como es el caso del uso de Metaproductos (Bryant, 1986)(Coudert y Madre, 1994), no son de aplicación universal, lo que significa en la práctica que los árboles a evaluar han de cumplir determinados requisitos o que es necesario conocer a priori cierta característica que asegura el éxito de la evaluación (Bouissou, 1996)(Andrews y Bartlett, 1998). Por otro lado la aparición de nuevas puertas lógicas derivadas de la necesidad de modelar complicadas interrelaciones entre equipos y/o subsistemas, típicas en sistemas dependientes (Schneeweiss, 1989)(Anand y Somani, 1998), en sistemas tolerantes a fallos (Dugan, Bavuso y Boyd, 1992)(Dugan, Venkatamaran y Gulati, 1997) o la moderna y prometedora disciplina de desarrollo de software tolerante a fallos (Belli y Jedrzejowicz, 1989)(Smidts y Sova, 1999), han restado aplicabilidad a las fórmulas deterministas de evaluación que hasta hace poco años eran consideradas indispensables (Vesely y Narum, 1970)(Blin, Carline et al., 1978) y devuelto cierto protagonismo a métodos de evaluación basados en simulación tipo Monte Carlo, los cuales vuelven a ser considerados como parte indispensable de los códigos actuales (Dugan, Coppit y Sullivan, 1998)(Saraiva y Sousa, 1998), así como han propiciado el desarrollo de técnicas combinadas con otros tipos de estructuras como pueden ser las cadenas de Markov (Dugan, 1988) o las Redes de Petri (Malhortra y Trivedi, 1995). La evaluación cuantitativa de árboles de fallos se vuelve especialmente crítica cuando esta es parte de una Análisis Probabilístico de Riesgos "PRA" (Henley y Kumamoto, 1993), también llamado Análisis Probabilístico de Seguridad (PSA), técnicas con las que se evalúa el impacto, en forma de riesgo para la vida y/o los bienes humanos, de determinadas actividades o fenómenos. Es conocido que estas técnicas se emplean para analizar riesgos derivados del funcionamiento de Centrales Nucleares (Zamalani, 1998), Industrias Químicas, Almacenamiento de Combustibles o sus Residuos, Terremotos (Budnitz, 1998), etc. En muchas de estas

actividades esta legislada la necesidad de realización de estos análisis, los cuales suelen incluir interacciones de tipo humano o medioambiental de difícil modelado lógico. La combinación de los factores mencionados suele conducir a árboles de fallos de tamaño y complejidad crecientes cuya construcción requiere un enorme esfuerzo de muchos expertos y el empleo de sofisticados sistemas de ayuda (Geymayr y Ebecken, 1995)(Liggesmeyer y Rothfelder, 1998), para los cuales sería deseable asegurar, a priori, que son evaluables cuantitativamente ó disponer de un método general de evaluación seguro y eficiente en toda circunstancia. En esta memoria se han presentado las conclusiones de la búsqueda de un método de esas características.

5.2.- CONCLUSIONES GENERALES

Las conclusiones basadas en la revisión del estado del arte y opiniones de diversos investigadores en los últimos años (epígrafe 3.2.3), permiten establecer un criterio de búsqueda de un método general de evaluación cuantitativa de árboles de fallos de gran tamaño y/o complejidad, así como algunas de las características deseables del mismo. Estas características, según se ha establecido en el capítulo III, se pueden concretar en utilizar un método combinado Determinista-Simulación, donde el método determinista sea el primero en abordar los cálculos y permita evaluar grandes árboles, en caso de no poder determinar exactamente la probabilidad buscada suministrará cotas seguras de la misma. El método de simulación será el segundo en aplicarse para, en este caso, estimar la probabilidad mediante el uso de estimadores que incorporen uno o varios métodos de reducción de varianza y aprovechen la información calculada por el método determinista inicial. Todo el proceso deberá estar dotado de un sistema de control capaz de decidir cuando termina el método determinista inicial y empieza el método de simulación subsiguiente, capaz de actuar de forma que no interrumpa el proceso de evaluación, y que optimice el conjunto del proceso.

Partiendo del anterior criterio se han estudiado las causas de pérdida de eficiencia de los métodos de evaluación conocidos, lo que ha permitido establecer los siguientes criterios:

- Ambos métodos deben ser Directos, o sea, abordar la evaluación cuantitativa desde el primer instante evitando la necesidad de determinar Grupos de

Corte Mínimos o transformaciones a estructuras equivalentes como los Diagramas de Decisión Binarios.

- El método Determinista inicial ha de replantearse desde los conocimientos básicos sobre funciones Booleanas, dado que utilizar las líneas de razonamiento que han conducido a los métodos clásicos muy probablemente conducirá a resultados con parecidos defectos. Fundamentalmente se tratará de evitar los problemas derivados del crecimiento exponencial del número de combinaciones lógicas. Para todo ello parece aconsejable retroceder hasta la Tabla de Quine de las funciones Booleanas y comenzar en ella la deducción.
- La búsqueda de un método eficiente para evaluar grandes árboles de fallos pasa obligatoriamente por considerar diferentes particiones y/o ordenamientos, del conjunto universal de soluciones posibles, encadenados y/o embebidos unos en otros como parte de un único proceso.

La conclusión fundamental de los trabajos realizados en esta tesis es que es posible definir un método de evaluación de árboles de fallos, al que hemos llamado **método PRD**, que responda a las características deseadas especificadas anteriormente cuyas propiedades fundamentales son:

1. El método determinista inicial será el que hemos denominado **Método de los Pesos** (epígrafe 3.5.1) el cual es capaz de determinar la probabilidad del suceso final o top del árbol de fallos mediante la aproximación, hasta coincidir, de dos cotas seguras (inferior y superior) de la misma, calculadas acumulando el peso de cada fila de la tabla de Quine correspondiente al espacio de vectores de estado que son soluciones del árbol de fallos que se estudie, utilizando Ordenación Intrínseca (epígrafe 3.4.3), mediante:

$$C_1(s) \leq P\{\psi(\mathbf{x})=1\} = 1 - P\{\psi(\mathbf{x})=0\} \leq 1 - C_0(t)$$

$$C_1(s) = \sum_{j=1}^s \prod_{i=1}^k P_i^{v_i} (1 - P_i)^{(1-v_i)} \leq \sum_{j=1}^m \prod_{i=1}^k P_i^{v_i} (1 - P_i)^{(1-v_i)} = P\{\psi(\mathbf{x})=1\} \quad s \leq m$$

$$\begin{aligned} P\{\psi(\mathbf{x})=0\} &= P\{\overline{\psi}(\mathbf{x})=1\} = P\left\{\prod_{j=1}^n \sum_{i=1}^k x_i^{v_i} = 1\right\} = P\left\{\sum_{j=1}^n \prod_{i=1}^k x_i^{1-v_i} = 1\right\} = \\ &= \sum_{j=1}^n P\left\{\prod_{i=1}^k x_i^{1-v_i} = 1\right\} = \sum_{j=1}^n \prod_{i=1}^k P_i^{1-v_i} (1 - P_i)^{v_i} \geq \sum_{j=1}^t \prod_{i=1}^k P_i^{1-v_i} (1 - P_i)^{v_i} = C_0(t) \quad (t \leq n) \end{aligned}$$

Este método fallará cuando estas cotas no converjan en un mismo valor en un tiempo de CPU razonable, en este caso el resultado del mismo serán ambas cotas, las cuales son seguras, o sea que la probabilidad buscada estará contenida entre ellas.

2. El método de simulación subsiguiente será el que denominamos **Muestreo RD**, el cual emplea el estimador de Muestreo Restringido (epígrafe 3.8) para aprovechar las cotas seguras provenientes del método de los Pesos y el estimador del Muestreo Daga Generalizado (epígrafe 3.7) insertado en el anterior para establecer una correlación negativa entre las muestras, provocando la convergencia rápida de los estimadores hacia la probabilidad buscada. Por ello los estimadores utilizados serán:

$$Q_{PRD} = [Q_U - Q_L] \hat{Q}_D - Q_L$$

$$\hat{Q}_D = N^{-1} \sum_{i=1}^N \theta_i$$

3. El mecanismo de control de cambio de método se basa en el **índice** o número de unos presentes en los vectores de estado pertenecientes a una partición estructural simple (epígrafe 3.4.4) del conjunto universal de soluciones del árbol.

El método de los Pesos ha mostrado ser capaz de evaluar cuantitativamente por sí solo árboles de fallos hasta 230 sucesos básicos, con cualquier número de puertas lógicas. El número de sucesos básicos del árbol condiciona el índice máximo (número máximo de unos por vector de estado) explorable por el método de los Pesos, así, para árboles de 100 sucesos básicos e índices mayores que 7 se cae rápidamente en el problema del crecimiento exponencial del número de soluciones, reduciéndose este índice hasta el valor 2 en los árboles de 500 sucesos básicos. El punto más negativo del método es la existencia de probabilidades altas (>0.01) entre las de los sucesos básicos, lo que hace que las cotas encontradas estén muy lejanas entre sí.

El muestreo Restringido, aplicado tras el método de los Pesos pero sin aplicar muestreo Daga, sólo consigue reducción de varianza respecto al Monte Carlo Crudo cuando se le suministran cotas seguras cercanas entre sí, lo que significa en la práctica que va precedido del método de los pesos y que la reducción de varianza está fuertemente condicionada por este. En la práctica ha mostrado un comportamiento muy pobre al obtener cotas por el Método de los Pesos, de hecho cuando las cotas obtenidas por este están muy próximas, situación deseable, en la mayoría de los casos es necesario un número de ensayos tan alto que dispara los tiempos de CPU que permiten obtener precisión aceptable. Por tanto ambos métodos (Pesos y Restringido), en lugar de cooperar, compiten entre sí, el tiempo que se gana aproximando lo más posible las cotas por el método de los Pesos, se pierde intentando aproximar la solución con el muestreo Restringido.

El muestreo Daga es aplicable individualmente ya que su reducción de varianza se basa en establecer correlación negativa entre las muestras. La experimentación práctica ha demostrado que este método es capaz de valorar cuantitativamente árboles de fallos de hasta 700 sucesos básicos con cualquier número de puertas lógicas. Su factor más negativo es la presencia de valores extremos (grandes o pequeños) de la probabilidad de los sucesos básicos. La experimentación práctica ha mostrado que la presencia conjunta de probabilidades mayores que 0.01 y/o menores que 10^{-7} para los sucesos básicos de un mismo árbol de fallos conducen a pérdidas de eficiencia, por lo que el algoritmo ha de truncar por las probabilidades más bajas para permitir la coherencia (correlación negativa) entre todas las muestras. De todas formas más allá de los 700 sucesos básicos se pierde eficiencia, ya sea porque los tiempos de CPU superan ampliamente los 30 minutos o bien porque la precisión de la respuesta disminuye debido al truncamiento. En los árboles de menos sucesos básicos se observa un incremento en los tiempos de ejecución debido a que el algoritmo admite probabilidades muy bajas (truncamiento escaso), lo que implica que obtiene un alto número de muestras que han de ser estudiadas. En los ensayos se ha observado que el tamaño de árbol que optimiza el tiempo de ejecución está entre 300 y 400 sucesos básicos. La reducción del error medio respecto al cometido por el Monte Carlo Crudo fue de un 53% con una desviación típica del 8%, mientras que siempre se obtuvieron al menos dos cifras significativas exactas de la solución buscada.

El método PRD ha mostrado un buen comportamiento en cuanto a precisión y tiempos de CPU hasta árboles de tamaño sobre 500 sucesos básicos. En primer lugar el método de los Pesos, como ya se ha establecido, es capaz de evaluar por sí la mayoría de los árboles de hasta 230 sucesos básicos (excepto aquellos que poseen probabilidades muy altas en promedio), siendo el índice máximo alcanzado en los más grandes el 2. En segundo lugar el estimador Restringido con el Daga Generalizado insertado ha sido capaz de terminar la evaluación con error aceptable y en tiempos que nunca ha sobrepasado los 2 minutos 30 segundos de CPU. El método muestra un balance muy positivo entre el uso del cálculo intensivo sobre los primeros índices (mediante los Pesos) y el uso del muestreo selectivo sobre los restantes (muestreo Restringido-Daga). O dicho de otra forma; en primer lugar se evalúan todos los contribuidores a ambas cotas mientras el tiempo de CPU es admisible y en segundo lugar se evalúan solo los contribuidores más importantes del resto pero con alta correlación negativa. Un factor que optimiza la aplicación conjunta es el hecho de que el muestreo Daga aumenta su velocidad por el hecho de no tener que considerar todas aquellas muestras que ya fueron consideradas por el método de Los Pesos, las cuales suelen ser la mayoría, por lo que los tiempos conjuntos disminuyen pero aumenta la precisión. De todas formas este equilibrio se rompe a partir del entorno de los 550 sucesos básicos, por lo que consideramos como un límite razonable de aplicación del método para procesos de optimización el de 500 sucesos básicos.

Un valor añadido del método PRD es el de que en todos los casos el método de los Pesos determina los Grupos de Corte Mínimos hasta el índice que realiza los cálculos, además los clasifica, almacena y selecciona por su importancia (valor) aquellos que estén por encima de un umbral determinado elegido por el usuario. A pesar del ligero esfuerzo computacional que ello significa, se ha preferido mantener dicha opción dado que esta información siempre puede ser útil a la hora de optimizar eficientemente determinados sistemas técnicos.

5.3.- CONCLUSIONES DEL ANÁLISIS FACTORIAL DEL MÉTODO PRD

Los factores de los que depende el método PRD son: Número de puertas lógicas, Complejidad de la estructura del árbol, Número de sucesos básicos y Probabilidades de los mismos. En base a los resultados obtenidos en la experimentación práctica de los algoritmos que implementan el método PRD se puede concluir que:

- El método siempre producirá una salida o probabilidad del suceso top del árbol de fallos que se evalúe sean cuales sean los valores de los cuatro factores descritos. Esta probabilidad obtenida será mas o menos precisa en función de la combinación de factores que se de en cada caso. En los siguientes párrafos se muestran las conclusiones obtenidas tras la experimentación práctica.
- El número de puertas lógicas y/o la complejidad de la estructura no son determinantes para la precisión alcanzada por el método. El único límite aparente para el número de puertas lógicas es el de líneas de código compilable que admita la CPU disponible, lo cual permite en cualquier caso práctico superar las diez mil puertas lógicas. Esta cifra es suficiente para los árboles que se espera puedan aparecer en las instalaciones más grandes o sistemas de alto nivel de integración.
- El número de sucesos básicos del árbol condiciona el índice máximo (número máximo de unos por vector de estado) explorable por el método de los Pesos, sin embargo, se han podido evaluar árboles de hasta 500 sucesos básicos con el PRD dadas las características ya mencionadas anteriormente.
- El conjunto de probabilidades de los sucesos básicos ha sido en la práctica el factor más influyente sobre el método, hasta tal punto que es el único factor realmente limitativo de la precisión alcanzable. El análisis detallado muestra las siguientes características:
 - Si las probabilidades son todas muy altas el método de los Pesos pierde precisión ya que aproxima muy poco las cotas superior e inferior de la probabilidad del suceso top, pero los estimadores Restringido-Daga ganan precisión ya que

encuentran muchas muestras (significativas y negativamente correladas) con poco esfuerzo computacional. El resultado global es que la probabilidad buscada es aproximada eficientemente.

- Si las probabilidades son todas muy bajas el método de los Pesos gana eficiencia y es capaz de aproximar en gran medida las cotas, en la mayor parte de los casos no es necesaria la intervención subsiguiente de los estimadores Restringido-Daga.

- Cuando las probabilidades de los sucesos básicos comprenden tanto valores altos como valores bajos, caso esperable en la práctica, la combinación PRD ha mostrado un comportamiento muy aceptable en cuanto a precisión/tiempo de CPU. Las reducciones de error cometido al estimar la probabilidad del suceso top con el método respecto del Monte Carlo Crudo han sido en valor medio de un 88% con una desviación típica del 6%, obtenidas para mas de 100 muestras diferentes partiendo del conjunto de probabilidades base. Las reducciones con respecto al Muestreo Daga y Muestreo restringido fueron de media 33% y 56% con desviaciones típicas del 4% y 7% respectivamente, para las mismas condiciones de test. Los tiempos de CPU no superaron los 2 minutos 30 segundos, siendo su valor medio de 45 segundos, para llegar a las soluciones con al menos dos cifras significativas exactas.

5.4.- CONSECUENCIAS

El hecho de que sea el conjunto de probabilidades de los sucesos básicos el único realmente influyente unido a las propiedades del método de los Pesos, permite una conclusión práctica novedosa y útil en cuanto a la evaluación cuantitativa de árboles de fallos con este método, y por tanto con el PRD. Se trata de que **es posible saber a priori (antes de construir el árbol de fallos) con que precisión mínima va a responder el método, o lo que es lo mismo cual será el error máximo cometido**. En efecto dado que para evaluar un árbol de fallos es necesario conocer las probabilidades de los sucesos básicos, y que estas se suelen conocer a priori, por estar recogidas en bases de datos de fallos que contienen el historial de todos los componentes de los sistemas técnicos, es posible calcular el peso acumulado de todas las combinaciones de sucesos básicos (filas de la tabla de Quine) hasta un índice determinado. Una vez calculado este peso acumulado el error máximo cometido será el valor absoluto de su diferencia con el valor uno, o sea:

$$\text{error máximo cometido} = 1 - \sum_{i=1}^k \text{Peso}_i$$

La conclusión del párrafo anterior permite establecer un patrón de trabajo capaz de asegurar el éxito de la evaluación cuantitativa de árboles de fallo de cualquier tamaño o complejidad basado solamente en el estudio de las probabilidades de los sucesos básicos presentes en la base de datos de fallos del sistema técnico. En efecto **bastará con agrupar o descomponer elementos del sistema de tal forma que las probabilidades de los sucesos básicos resultantes conduzcan a valores del error, dado por la fórmula anterior, menores que la tolerancia admisible**. Ello significa esencialmente que **cuando nos disponemos a obtener el árbol de fallos de un sistema técnico complicado (la tarea más ardua y costosa del Análisis de Arboles de Fallos) ya estamos seguros de que el árbol es evaluable mediante el método PRD sin importar ni su tamaño ni su complejidad**.

La conclusión anterior es muy útil en cualquier circunstancia pero muy especialmente cuando se realiza un proceso de optimización multicriterio en el cual es necesario ensayar miles de alternativas de diseño de un sistema técnico, lo cual implica

miles de árboles de fallos con complejidad y tamaño diferente, más aún si al mismo tiempo que se cambia el diseño se ensayan diferentes dispositivos (con Disponibilidades diferentes) para cada función a realizar. Esto es **al mismo tiempo que se cambia la disposición física de los componentes se ensayan componentes alternativos para realizar una misma tarea**. En estas circunstancias bastará con emplear dispositivos cuya Disponibilidad esté dentro del rango que asegura el error máximo cometido, dado por la fórmula anterior, para asegurar el éxito de la evaluación cuantitativa por el método PRD de cualquier árbol de fallos que se obtenga durante el proceso de optimización.

5.5.- LÍNEAS FUTURAS

Dos vías son posibles para continuar los trabajos realizados en esta tesis.

La primera es explorar la viabilidad del método PRD para:

- Incluir todo tipo de puertas lógicas en el árbol de fallos
- Evaluación de árboles en sistemas tolerantes a fallos, incluso aquellos que llevan ordenadores cuyos fallos aparecen en el árbol.
- Analizar la aplicabilidad para el estudio de la Fiabilidad del Software, para lo cual el árbol de fallos es una herramienta de gran proyección en los próximos años.
- Estudiar la Fiabilidad de sistemas técnicos.

La segunda sería aplicar el método PRD y el muestreo Daga generalizado al estudio y optimización de sistemas técnicos, en particular el software desarrollado es aplicable directamente para analizar la probabilidad de Fallo a la Demanda (No Disponibilidad) de sistemas de seguridad, cuya principal característica es la de que solo deben funcionar cuando se les demande para ello (raras veces), permaneciendo el resto del tiempo en espera. Tales como:

- Sistemas de detección y extinción de incendios en edificios (viviendas, hoteles, teatros, estadios, etc.), instalaciones industriales, plantas de alto riesgo (químicas,

nucleares, bacteriológicas,...), sistemas de almacenamiento (combustibles en puertos, aeropuertos, residuos, vertederos, etc.).

- Sistemas de control de explosiones y fugas en instalaciones industriales
- Sistemas de respuesta al fallo y adopción de contramedidas en centrales (eléctricas, químicas, etc.).
- Sistemas Eléctricos de protección
- Sistemas y circuitos de emergencia
- Sistemas de frenado de emergencia (vehículos, ascensores, etc.)
- Sistemas de lanzamiento y aterrizaje (aeronáutica)
- Sistemas de bombeo para circulación de la sangre (medicina)
- Sistemas de Electrochoque (medicina)
- Sistemas de iluminación de emergencia
- Sistemas de protección antichoques (estructuras absorbentes, airbag, etc.)



BIBLIOGRAFÍA

- Abraham, J. [1979]. "An Improved algorithm for network reliability", *IEEE Transactions on Reliability* **R-28**(1): 58-61.
- Agarwalla, A.S. [1991]. "A Technique for proper design and impact analysis of event sequencing for safety and availability", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 280-284
- Alesso, H.P. & Benson, H.J. [1980]. "Fault Tree and reliability Relationships for analyzing non_coherent two state system". *Nuclear Engineering and Design* **56**:309-320.
- Allen, D.J. & Madhava-Rao, M.S. [1980]. "New Algorithms for the Synthesis and Analysis of Fault Trees", *Industrial Engineering Chemical Fundamentals* **19**: 79-85.
- Anand, A. & Somani, A. [1998]. "Hierarchical Analysis of Fault Trees with Dependencies using decomposition", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 69-75.
- Andrews, J.D. & Pattison, R.L. [1997]. "Optimal Safety System Performance", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 76-83.
- Andrews, J.D. & Bartlett, L.M. [1998]. "Efficient Basic Event Orderings for Binary Decision Diagrams", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 61-68.
- Apostolakis, G. & Moieni, P. [1986]. "On the Correlation of Failure rates", *Data Collection and use in Reliability and Availability Assessment, Proceedings of the 5th EuReData Conference on Reliability*, pp. 56-61, Heidelberg.
- ARALIA Group [1995]. "Computation of Prime Implicants of a Fault-tree Within ARALIA", *Proceedings of the ESREL Conference*, pp. 77-81, Bournemouth, England.
- Barlow, R.E. & Proschan, F. [1975]. *Statistical Theory of Reliability and Life Testing*, Mc. Ardle Press Incorporated, New York.
- Bavuso, S. & Hampton, J. [1997]. "Aerospace Applications of Weibull and Monte Carlo Simulation with Importance Sampling", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 208-210.
- Bechta, J., Bavuso, S. & Boyd, M.A. [1990]. "Fault Trees and sequence dependencies", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 286-293.
- Bell Telephone Labs. [1961]. "Launch control safety study", *Bell Telephone Laboratories* **VII** (1), U.S.A.

- Belli, F. & Jdrzejowicz, P. [1989]. "Fault-tolerant programs and their reliability", *IEEE Transactions on Reliability* **39**(2): 184-192.
- Blin, P. et al. [1978]. "PATREC, a computer code for fault tree calculations", En Apostolakis, G.(ed.), *Synthesis and Analysis Methods for Safety & Reliability*, pp. 33-43.
- Billington, R. & Jonnavithula, H. [1997]. "Composite system adequacy assessment using sequential Monte Carlo simulation with variance reduction techniques", *IEEE Proceedings Generation, Transmission and Distribution* **144**(1): 1-6, January.
- Bossche, A. [1985]. "Computer Aided Fault Tree Synthesis", *Microelectronics and Reliability* **25**(4): 773-788.
- Bouissou, M. [1996]. "An Ordering Heuristic for Building Binary Decision Diagrams from Fault-Trees", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 208-214.
- Bouissou, M. & Bogarde, E. [1997]. "Unavailability Evaluation and Allocation at the Design stage for Electric power plants: Methods and Tools", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 91-99.
- Bourne, A.J. et al. [1981]. *Defences against Common-Mode Failures in Redundancy systems, a guide for Management Designers and Operators*, UKAEA, SRD R 196.
- Bowles John B., Bonnell Ronald D. [1996], "Failure Mode, Effects and Criticality Analysis [What It Is and How To Use It". En IEEE. , *Tutorial Notes, Annual Reliability and Maintainability Symposium*, pp. 1-34.
- Boyd, M. & Iverson, D. [1993]. "Digraphs and Fault Trees: A tale of two combinatorial modeling methods", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 220-226.
- Boyd, M. & Bavuso, S.J. [1993]. "Simulation modeling for long duration spacecraft control systems", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 106-113.
- Boyd, M.A. & Tuazon, J.O. [1991]. "Fault Tree models for Fault Tolerant Hypercube Multiprocesors", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 120-123.
- Braun, I. [1991]. *Approximate fault tree analysis by means of an incomplete Shannon decomposition of the redundancy structure function*, FernUniversitat. Informatik-Bericht 115 [9/1991] D58-Hagen.
- Bryant, R.E. [1986]. "Graph-Based Algorithms for Boolean Function Manipulation", *IEEE Transactions on Computers* **C-35**(8).

- Bryant, R.E. [1992]. "Symbolic boolean manipulation with ordered binary-decision diagrams", *ACM Computing Surveys* **24**(3): 293-318.
- Budnitz, R.J. [1998]. "Current status of methodologies for seismic probabilistic safety analysis", *Reliability Engineering and System Safety* **62**(1-2): 71-88. Elsevier Science Limited.
- Camarda, P., Corsi, F. & Trentadue, R. [1978]. "An Efficient simple algorithm for fault tree automatic synthesis from the Reliability graph", *IEEE Transactions on Reliability* **R-27**(3): 215-221.
- Camarinopoulos, L. & Yllera, J. [1985]. "An improved top-down algorithm combined with modularization as highly efficient method for fault tree analysis", *Reliability Engineering and System Safety* **11**: 93-108. Elsevier Science Limited.
- Cancela, H. & El Khadiri, M. [1998]. "Series-Parallel Reductions in Monte carlo Network-Reliability Evaluation", *IEEE Transactions on Reliability* **47**(2):120-123. June.
- Carlsson, L., Erikson, H. & Werner, W. [1998]. "Generic probabilistic safety analysis: application to the comparison of the emergency feedwater system designs of three PWRs and to the modernisation and backfitting of older Swedish NPPs", *Reliability Engineering and System Safety* **63**(2): 141-154. Elsevier Science Limited.
- Cepin, M. & Mavko, B. [1998]. "Fault tree developed by an object-based method improves requirements specification for safety-related systems", *Reliability Engineering and System Safety* **63**(2): 111-125. Elsevier Science Limited.
- Chatterjee, P. [1975]. "Modularization of fault trees: a method to reduce the cost of analysis", *SIAM Reliability and Fault Tree Analysis*, pp.101-106.
- Chen, L.W. & Modarres, M. [1990]. *Autonomous decision making expert system for fault administration*, Gensym Corporation Published G2 Success Stories.
- Chu, T.L. & Apostolakis, G. [1980]. "Methods for probabilistic analysis of noncoherent fault trees", *IEEE Transactions on Reliability* **R-29**: 5.
- Ciciani, B. [1991]. "Fault-Tolerance considerations for redundant binary tree dynamic random access memory [RAM] chips", *IEEE Transactions on Reliability* **41**(1): 139-148.
- Cochran, W.G. [1963]. *Sampling Techniques*, second ed., John Wiley & Sons, New York.
- Coudert, O. & Madre, J.C. [1994]. "Metaprime: An interactive Fault-Tree analyzer", *IEEE Transactions on Reliability* **43**(1): 121-127.

- Coudert, O. & Madre, J.C. [1993]. "Fault Tree analysis: 10^{20} Prime Implicants and beyond", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 240-245.
- Curtiss, J.H., et al. [1951]. "Monte Carlo Method", *National Bureau of Standards Applied Mathematics NBSAM Series 12*, U.S.A..
- Dagum, P., Karp, R. Luby, M. & Ross, S.[1995]. "An optimal algorithm for Monte Carlo estimation", *IEEE Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pp. 142-149.
- Dhillon, B.S. & Rayapati, S.N. [1988]. "Chemical-system reliability; literature review". *IEEE Transactions on Reliability* **R-28**: 199-208.
- Dhurjati, P.S. [1992]. *Preprints of the IFAC Symposium on On-line Fault Detection and Supervision in the Chemical Process Industries*, University of Delaware, Newark.
- Doreste, L.D. y Galván, B.J.[1986], "Generalización del Muestreo Daga para evaluar la No Disponibilidad de Sistemas", *XII Reunión de la Sociedad Nuclear Española*, Salamanca. España.
- Doreste, L.D. y Galván, B.J.[1987], "Nuevas aplicaciones del Muestreo daga Generalizado", *XIII Reunión de la Sociedad Nuclear Española*, La Toja. España.
- Doyle, S.A., Dugan, J.B. and Patterson-Hine, F.A. [1995]. "A combinatorial approach to modeling Imperfect Coverage ", *IEEE Transactions on Reliability* **44**(1):87-94.
- Dugan, J.B. & Lyu, M.R. [1993]. "System reliability analysis of an N-version programming application", *Proceedings of The International Symposium on Software reliability engineering*, November.
- Dugan, J.B [1988]. "Fault Trees and Imperfect Coverage", *IEEE Transactions on Reliability* **38**(2): 177-185.
- Dugan, J.B., Bavuso, S.J. & Boyd, M.A. [1992]. "Dynamic Fault-Tree models for fault-tolerant computer systems", *IEEE Transactions on Reliability* **41**(3): 363-377.
- Dugan, J.B., Venkataraman B. & Gulati R. [1997]. "A software package for the analysis of dynamics fault tree models", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 64-70.
- Easton, M.C. & Wong, C.K. [1980]. "Sequential Destruction Method for Monte Carlo Evaluation of System Reliability", *IEEE Transactions on Reliability*, **R-29**(1):27-32.
- Efron, B. & Stein, C. [1981]. "The jackknife estimate of variance", *The Annals of Statistics* **9**: 586-596.
- Esary, J.D. & Proschan, F. [1963]. "Coherent structures of non_ identical components". *Technometrics* **5**(2).

- Feo, T. [1986]. "PAFT F77, Program for the analysis of Fault Trees", *IEEE Transactions on Reliability* **35**(1): 48-50.
- Fleming, K.N. & Kalinowski, A.M. [1983]. "An extension of the Beta Factor Method to Systems with High Levels of Redundancy", *Pickard, Lowe and Garrick, Inc.*
- Fleming, K.N., Mosleh, A. & Deremer, R.K. [1986]. "A Systematic Procedure for the incorporation of Common Cause Events into Risk and Reliability Models", *Nuclear Engineering and Design* **93**: 245-273.
- Fragola, J.R. [1993]. "Designing for Success: Reliability Technology in the Concurrent Engineering Era", *IEEE Proceedings Annual Reliability and Maintainability Symposium*.
- Fussell, J.B. [1973a]. *Fault tree analysis concepts and techniques*, NATO advanced Study Institute on Generic Techniques of System Reliability Assesment.
- Fussell, J.B. [1973b]. "A formal methodology for fault tree construction", *Nuclear Engineering and Design* **52**: 337-360.
- Galyean, W.J., Fowler, R.D., Close J.A. & Donley M.E. [1989]. "Case Study: Reliability of the INEL-SITE power system", *IEEE Transactions on Reliability* **38**(3): 279-284.
- Georgiannis, V.C., Caragiannis, I.E. & Tsoukarellas, M.A. [1997]. "A general framework for applying safety analysis to safety critical real-time applications using fault trees", *Proceedings Ninth Euromicro workshop on real-time systems*, pp. 168-175.
- Geymayr, J. & Ebecken, N. [1995]. "Fault-Tree analysis: A Knowledge-Engineering approach", *IEEE Transactions on Reliability* **44**(1): 37-45.
- Gray, M.G., Booth, T.E., Kwan T.J.T. & Snell, C.M. [1998]. "A multicomponent variance reduction scheme for Monte Carlo semiconductor simulators", *IEEE Transactions on Electron Devices* **45**(4): 918-924. April.
- González, L., García, D., y Galván, B.J. [1995], "Sobre el análisis computacional de funciones booleanas estocásticas de muchas variables", *Libro de actas de la 1ª Reunión de Algebra Computacional*, Santander.
- González, L., García, D., y Galván, B.J. [1995], "Un método determinista para la evaluación de funciones booleanas estocásticas: Análisis comparativo", *Libro de actas del IV Congreso de Matemática aplicada*, CEDYA, Vic.
- Gulati, R. & Dugan, J.B. [1997]. "A Modular Approach for Analyzing Static and Dynamics Fault Trees", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 57-63.

- Gulati, R. [1996]. *A modular approach to static and dynamic fault tree analysis*, Master's thesis. University of Virginia.
- Guth, M.A.S. [1991]. "A Probabilistic Foundation for Vagueness & Imprecision in Fault-Tree Analysis", *IEEE Transactions on Reliability* **40**(5): 563-571.
- Haasl, D.F. [1965]. "Advanced concepts on Fault Tree analysis", *The Boeing Company System Safety Symposium*, U.S.A.
- Hammersley, J.M. & Handscomb, D.C. [1964]. *Monte Carlo Methods*, Chapman and Hall, New York.
- Hauptmanns, U. [1986]. *Análisis de Arboles de Fallos*, Editorial Bellaterra.
- Heger, A., Sharif, B., Jayaram K., Stack D.W. & Talbott D.V. [1995]. "Calculating Exact Top-Event Probabilities Using Sigma-Pi PATREC", *IEEE Transactions on Reliability* **44**(4): 640-644.
- Heger, A.S., Koen B.V. & Wegmann H.F. [1993]. "Parallel Processing of Fault Trees on a Locally Distributed Multiple-Processor Network", *IEEE Transactions on Reliability* 436-441.
- Henley, E. & Kumamoto, H. [1993]. *Probabilistic Risk Assessment*, IEEE Press. New York.
- Hennings, W. & Kuznetsov, N. [1995]. "FAMOCUTN & CUTQN: Programs for Fast Analysis of Large Fault Trees with Replicated & Negated Events", *IEEE Transactions on Reliability* **44**(3): 368-376.
- Hessian, R.T. Jr., Salter, B.B. & Goodwin, E.F. [1989]. "Fault-tree analysis for system design, development, modification and verification", *IEEE Transactions on Reliability* **39**(1): 87-91.
- Hocevar, D.E., Lightner M.R. & Trick, J. [1983]. "A study of variance reduction techniques for estimating circuits yields", *IEEE Transactions on Computer-Aided Design CAD-2*(3): 180-192. July.
- Howell, S.V., Bavuso S.J. & Haley P.J. [1990]. "A graphical language for reliability model generation", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 471-475.
- Huang, P. & Li, Z. [1992]. "The Measure of Effect in an Expert System for Automatic Generation of Fault Trees", *IEEE Transactions on Reliability* **41**(1): 57-62.
- Hura, G.S. & Atwood, J.W. [1988]. "The use of Petri Nets to analyze coherent Fault Trees", *IEEE Transactions on Reliability* **37**(5): 469-474.
- Iverson, D.L. [1992]. "Automatic Translation of diagraph to Fault-Tree models", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 354-362.

- Kalos, M.H. & Whitlock, P.A. [1986]. *Monte Carlo Methods, Volume I: Basics*, John Wiley & Sons, New York.
- Keramat, M. & Kielbasa, R. [1997]. "Efficient Average Quality Index Estimation of Integrated Circuits by Modified Latin Hypercube Sampling Monte Carlo [MLHSMC]", *IEEE International Symposium on Circuits and Systems*, pp. 1648-1651. Hong Kong.
- Keramat, M. & Kielbasa, R. [1997]. "A study of stratified sampling in variance reduction techniques for parametric yield estimation", *Proceedings of the International conference on Circuits and Systems, ISCAS 97* 3: 1652-1655.
- Keramat, M. & Kielbasa, R. [1998]. "Study of optimal importance sampling in Monte Carlo estimation of average quality index", *Proceedings of the International conference on Circuits and Systems, ISCAS 98* 6:106-109.
- Khodabandehloo, K. & Sayles, R.S. [1986]. "Production Reliability in flexible manufacturing systems", *Quality and Reliability Engineering* 2(3): 171-182.
- Kim I.S., Modarres M. [1987], "Application of a goal tree-success tree model as the knowledge-base of operator advisory system". , *Nuclear Eng. and Design*, 104: 67-81.
- Kirk, D.B. & Arvo, J. [1991]. Umbiased sampling techniques for image synthesis, *Computer Graphics (SIGGRAPH 91 Porceedings)*, 25:153-156.
- Kneale, J.B. [1997]. "Application of expert system techniques to fault tree diagnostic data", *IEEE Autotestcon Proceedings, AUTOTESTCON 97*, pp. 100-104.
- Kócza, G. & Bossche, A. [1997]. "Automatic Fault Tree Synthesis and Real-Time Tree Trimming, Based on Computer Models", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 71-75.
- Kodha, T., Henley, E. & Inoue, K. [1988], "Finding modules in Fault Trees", *IEEE Transactions on Reliability* 38(2):165-176.
- Koen, B.V. & Carnino, A. [1972]. "Calculation of systems reliability by pattern-recognition", *Computational methods and techniques*, pp. 141-142.
- Kongsoe, E. [1972]. *RELY-4 : A Monte Carlo computer program for systems reliability analysis*, Danish Atomic Energy Commission, RISO_M_1500, June.
- Kongsoe, E. [1975]. "REDIS a computer program for system reliability analysis by direct simulation", *International Symposium on Reliability of Nuclear Power Plants*, Insbruck, Austria, April 4-18.
- Kumamoto, H., Tanaka, T., Inoue, K. & Henley, E. [1980]. "Dagger Sampling Monte Carlo for system Unavailability Evaluation", *IEEE Transactions on Reliability* R-29(2):122-125.

- Kumamoto, H., Tanaka, T. & Inoue, K. [1987]. "A New Monte Carlo Method for Evaluating System-Failure Probability", *IEEE Transactions on Reliability* **R-36**(1):63-69.
- Kumamoto, H. & Henley, E.J. [1978]. "Top Dawn algorithm for obtaining prime implicants sets of non-coherent fault trees", *IEEE Transactions on Reliability* **R-27**: 242-249.
- Kumar, A., Pathak, R. & Gupta, Y. [1995]. "Genetic-Algorithm-based reliability optimization for computer network expansion, *IEEE Transactions on Reliability* **44**(1):63-68.
- Kuzawinski, K.M. & Smurthwaite, R. [1989]. "Automated Fault Tree Analysis via AI/ES", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 331-335.
- Kvam, P.H. [1998]. "A Parametric Mixture-Model for Common-Cause failure data", *IEEE Transactions on Reliability* **47**(1): 30-34. March.
- Lafortune, E.P. & Willems, Y.D. [1995]. "A 5D tree to reduce the variance of Monte Carlo ray tracing", *Rendering Techniques '95*, pp. 11-20.
- Lambert, H.E. [1975]. "Measures of importance of events and cut sets in fault trees", En Barlow, R.E, et al.(ed.), *SIAM Reliability and Fault Tree Analysis*, pp. 77-100.
- Lambert, H.E. [1975]. *Fault trees for decision making in system analysis*, Lawrence Livermore Laboratory, University of California.
- Lapp S.A., Powers G.J. [1977], "The synthesis of fault trees". En Fussell, J.B., Burdick, G.R., SIAM. Risk Assessment, Nuclear Systems Reliability Eng. and, 778-799.
- Lapp, S.A. & Powers, G.J. [1979]. "Update of Lapp_Powers fault tree synthesis algorithm", *IEEE Transactions on Reliability* **R-28**(1): 12-14.
- Lapp, S.A. & Powers, G.J. [1977]. "Computer-Aided Synthesis of Fault trees", *IEEE Transactions on Reliability* **R-26**: 2-13.
- Lassila, P.E. & Virtamo, J.T. [1998], "Variance reduction in Monte Carlo simulation of product form systems", *Electronic letters* **34**(12): 1204-1205.
- Lee, W.S., Grosh, D.L., Tillman, F.A. & Lie, C.H. [1985]. "Fault Tree Analysis, Methods, and Applications-- A Review", *IEEE Transactions on Reliability* **R-34**(3): 194-203.
- Lee, W. [1982]. *A Study of Fault Tree Analysis for System Safety and Reliability*, Phd dissertation, Kansas State University.

- Lee, S.L., Lie, C.H. & Hong, J.S. [1997]. "A Computation Method for Evaluating Importance-Measures of Gates in a Fault Tree", *IEEE Transactions on Reliability* **46**(3):360-365, September.
- Lee, J.H., Oka, Y. & Kosshizuka, S. [1999]. "Safety system consideration of supercritical-water cooled fast reactor with simplified PSA", *Reliability Engineering and System Safety* **64**(3): 327-338.
- Levitin, G., Lisnianski, A., Ben-Haim, H. & Elmakis, D.[1998]. "Redundancy Optimization for Series-Parallel multi-State Systems", *IEEE Transactions on Reliability* **47**(2):165-172, June.
- Liggesmeyer, P. & Rothfelder, M. [1998]. "Improving System Reliability with Automatic fault tree generation", *IEEE Transactions on Reliability* **3**(2): 90-99.
- Lin, C. & Wang, M.J. [1997]. "Hybrid fault tree analysis using fuzzy sets", *Reliability Engineering and System Safety* **58**(3): 205-213 .
- Luo, T. & Trivedi, K.S. [1998]. "An improved algorithm for Coherent-System reliability", *IEEE Transactions on Reliability* **47**(1): 73-78. March.
- Lyu, M.R. [1998]. "An integrated approach to achieving high software reliability", *IEEE Aerospace Conference* **4**: 123-136.
- Malhortra, M. & Trivedi, K.S. [1995]. "Dependability Modeling using Petri Nets", *IEEE Transactions on Reliability* **44**(3): 428-440.
- Malhortra, M. & Trivedi, K.S. [1994]. "Power - Hierarchy of Dependability model types", *IEEE Transactions on Reliability* **43**(2): 493-502.
- Manian, R., Dugan, J.B., Coppit, D. & Sullivan, K.J. [1998]. "Combining various solution techniques for Dynamic Fault Tree Analysis of Computer systems", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 21-28.
- Marshall, A.W. & Olkin, I. [1967]. "A multivariate exponential distribution", *JASA* **62**:30-44, U.S.A.
- McGrath, E.J. & Irving, D.C. [1975], *Techniques for efficient Monte Carlo Simulation*, Radiation Shielding information Center, U.S.A.
- McGraw, R.J. [1989]. *Petri Net and Fault Tree Analysis: Combining two techniques for software safety analysis on an embedded military application*, Master's Thesis NTIS, Naval Postgraduate School. AD-A225 563.
- Metropolis, N. & Ulam, S. [1949]. "The monte-carlo method", *Journal of American Statistical Association* **44**: 335-341.
- Meyer, M.A.(ed.) [1956]. *Symposium on Monte Carlo Methods*, John Wiley & Sons, New York.

- Min Y., Zhou, Y. & Li, Z. [1994]. "A Fail-Safe Microprocesor-based system for interlocking on railways", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 414-420.
- Misra, K.B. [1993]. *Reliability Analysis and Prediction [A Methodology Oriented Treatment]*, Elsevier Science Publishers BV, Amsterdam.
- Misra, K.B. & Weber, G.G. [1988]. "A New Method for Fuzzy Fault Tree Analysis", *Microelectronics and Reliability* **29**(2): 195-216.
- Mitchell, D.P. [1996]. "Consequences of stratified sampling in computer graphics", *SIGGRAPH 96 Proceedings*, pp. 277-280, Addison-Wesley.
- Muñoz, A., Martorell, S. Y Serradell, V. [1997] "Genetic algorithms in optimizing surveillance and maintenance of components", *Reliability Engineering and System Safety* **57**:107-120.
- Mulvihill, R.J. [1993]. "Design-Safety Enhancement through the use of Hazard & Risk Analysis", *IEEE Transactions on Reliability* **37**(2):149-158.
- NUREG [1975], *Reactor Safety Study-An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants, WASH-1400*, U.S. Nuclear Regulatory Commission NUREG-75/014, Washington, D.C.
- Niederreiter, H. [1992]. *Random Number generation and Quasi-Monte Carlo Methods*, Society for Industrial and Applied Mathematics, Philadelphia.
- Olmos, J. & Wolf, L. [1977]. *A modular approach to fault tree and reliability analysis*, Nuclear Engineering Dept., M.I.T. Technical Report 209, Cambridge, Massachussets.
- Olmos, J. & Wolf, L. [1978]. "A modular representation and analysis of fault trees". *Nuclear Engineering and Design* **48**, U.S.A.
- Owen, A.B. [1992]. "Orthogonal arrays for computer experiments, integration and visualization", *Statistica Sinica* **2**: 439-452.
- Owen, A.B. [1994]. "Lattice sampling revisited: Monte Carlo variance of means over randomized orthogonal arrays", *The Annals of Statistics* **22**: 930-945.
- Owen, A.B. [1997]. "Monte Carlo variance of scrambled net quadrature", *SIAM Journal on Numerical Analysis* **34**: 1884-1910.
- Page, L.B. & Perry, J.E. [1987]. "An Algorithm for Fault-tree probabilities using the factoring theorem", *IEEE Microelectronics and Reliability* **28**(2): 273-286.
- Page, L.B. & Perry, J.E. [1986]. "An algorithm for exact fault-tree probabilities without cut sets", *IEEE Transactions on Reliability* **35**(5): 544-558.

- Page, L.B. & Perry, J.E. [1994]. "Standard deviation as an alternative to fuzziness in fault tree models", *IEEE Transactions on Reliability* **43**(3): 402-407.
- Page, L.B. & Perry, J.E. [1991]. "Direct Evaluation Algorithms for Fault-Tree Probabilities", *Computers chem. Engng.* **15**(3):157-169.
- Painton, L & Campbell, J. [1995]. "Genetic algorithms in optimization of system reliability", *IEEE Transactions on Reliability* **44**(2):172-178.
- Papazoglou, I.A. [1998]. "Mathematical foundations of event trees", *Reliability Engineering and System Safety* **61**(3): 169-183.
- Parascos, E.T. [1990]. "Tools of the Trade-utility application", *INTER-RAM* **17**: 67-73.
- Patterson-Hine, F.A. & Dugan J.B. [1992]. "Modular techniques for dynamic Fault-Tree Analysis" *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 363-369.
- Patterson-Hine, F.A. & Koen, B.V. [1987]. "Comment on : An algorithm for exact fault-tree probabilities without cut sets", *IEEE Transactions on Reliability* **36**(5):640-641.
- Patterson-Hine, F.A. & Koen, B.V. [1985]. "Implementation of PATREC nuclear reliability program in LISP", *IEEE Transactions on Reliability* 640-641.
- Patterson-Hine, F.A. & Koen, B.V. [1989]. "Direct Evaluation of Fault Trees Using Object-Oriented Programming Techniques", *IEEE Transactions on Reliability* **38**(2):186-192.
- Patton, R.P. & Clark, R. [1989]. *Fault Diagnosys in Dynamic systems: Theory and Applications*, Prentice Hall.
- Pierre, S. & Elgibaoui, A.[1997]. "A Tabu-Search Approach for Designing Computer-Network Topologies with Unreliable Components", *IEEE Transactions on Reliability* **46**(3): 350-359, September.
- Powers, G.M. & Tompkins, F.C. [1973]. "Computer_aided synthesis of fault trees for complex processing systems", *NATO Advanced Study Institute on Generic Techniques of System Reliability Assessment*, pp. 307-314.
- Powers, G.J., Tompkins, F.C. & Lapp, S.A. [1975]. "A Safety Simulation Language for Chemical proceses: A Procedure for Fault Tree synthesis" En Barlow and Fussell (ed.) *SIAM Reliability and Fault Tree analisis*, U.S.A..
- Probabilistic safety assessment [1995]. Special issue in *Kerntechnik*, **60**/2-3. May.
- Purgathofer, W. [1986]. "A statistical method for adaptive stochastic sampling", *Eurographics '86*, pp. 145-152. North-Holland.

- Radke, G.E. & Evanoff, J. [1994]. "A Fast Recursive Algorithm to Compute the Probability of M-out-of-N Events", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 114-117.
- Rai, S. [1994]. "A Direct Approach to Obtain Tighter Bounds for Large Fault Trees with Repeated Events", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 475-480.
- Rauzy, A. [1993]. "New algorithms for fault tree analysis", *Reliability Engineering and Systems Safety* **40**: 203-211.
- Rauzy, A. & Dutuit, Y. [1997]. "Exact and truncated computations of prime implicants of coherent and non-coherent fault trees within Aralia", *Reliability Engineering and System Safety* **58**: 127-144. Elsevier science limited.
- Rayapati, V.N. [1989]. "Modular Fault Tolerant VLSI parallel processor architectures with dynamic redundancy", *Microelectronics and Reliability* **30**(2): 213-236.
- Redgate, M.L., Mckelvey, M.H. & Jolly, C.L. [1994]. "Implementation of an Integrated Safety program: The MD-90 antiskid system", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 52-58.
- Ren, Y. & Dugan, J.B. [1998]. "Design of Reliable Systems using Static & Dynamic Fault Trees", *IEEE Transactions on Reliability* **47**(3): 234-244, September.
- Risk Analysis [1998]. Special Issue of *Reliability Engineering and System Safety* **59**(1).
- Rosenthal, A. [1980]. "Decomposition Methods for Fault Tree Analysis", *IEEE Transactions on Reliability* **R-29**(3).
- Roush, M.L. & Wang, X. [1995]. "Time-dependent logic for goal-oriented dynamic-system analysis", *IEEE Proceedings Annual Reliability and Maintainability Symposium*.
- Rubinstein, R.Y. [1981]. *Simulation and Monte Carlo method*, John Wiley & Sons, New York.
- Rushdi, A.M. [1985]. "Uncertainty Analysis of Fault Tree Outputs", *IEEE Transactions on Reliability* **34**(5): 458-462.
- Rushdi, A.M. & Kafrawy, K.F. [1988]. "Uncertainty propagation in Fault-Tree analysis using an exact method of moments", *IEEE Microelectronics and Reliability* **28**(6): 945-965.
- Salem, S.H. & Apostolakis, G.E. [1978]. *The CAT methodology for Fault Tree construction*, NATO, U.S.A.
- Salem, S.H., Wu, J.S. & Apostolakis G.E. [1979]. "Decision Table development and application to the construction of fault trees", *Nuclear Technology* **42**: 51-64, U.S.A.

- Sato, Y., Henley, E. J. & Inoue, K. [1989]. "Action-chain model for design of hazard-control systems for robots", *IEEE Transactions on Reliability* **38**(2):151-157.
- Saraiva, J.T. & Sousa, A.V. [1998]. "New advances in integrating fuzzy data in Monte Carlo simulation to evaluate reliability indices of composite power systems", *Proceedings of the mediterranean Electrotechnical Conference, MELECON 98 2*, pp. 1084-1088.
- Schneeweiss, W.G. [1989]. *Boolean Functions with Engineering Applications and Computer Programs*, Springer-Verlag.
- Schneeweiss, W.G. [1989]. "Fault Tree synthesis via backtracking", *Proceedings of Reliability '89*, vol. II, pp. 4C11-4C19, Brighton, U.K.
- Schneeweiss, W.G. [1992]. "Approximate Fault-Tree analysis without cut sets", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 370-375.
- Schneeweiss, W.G. [1992]. "Reliability theory for large linear systems with helping neighbors", *IEEE Transactions on Reliability* **41**(3): 343-351.
- Schneeweiss, W.G. [1996], *A simple tighter bound for the Probability of a union of events; with Dependability applications*, Informatik Berichte 194-2/1996. FernUniversitat. Hagen.
- Semanderes, S.N. [1971]. "ELRAFT a computer program for the efficient logic reduction analysis of fault trees", *IEEE Transactions on Nuclear Science* **18**(1): 481-487, U.S.A.
- Shetty, N.K., Soares, C.G., Thoft-Christensen, H. & Jensen, F.M. [1998]. "Fire safety assessment and optimal design of passive fire protection for offshore structures", *Reliability Engineering and System Safety* **61**(1-2): 139-149.
- Sinnamon, R.M. & Andrews, J.D. [1997]. "New approaches to evaluating fault trees", *Reliability Engineering and System Safety* **58**: 89-96.
- Sloane, B.D. & Sherbina, C.A. [1988]. "Incinerator for PCB and municipal solid wastes; likelihood of off-normal releases", *Proceedings Annual Reliability and Maintainability Symposium*, pp. 192-198.
- Slyke, R. Van & Frank, H. [1972]. "Network Reliability Analysis", *IEEE Transactions on Reliability* 279-290.
- Smidts, C. & Sova, D. [1999]. "An architectural model for software reliability quantification: sources of data", *Reliability Engineering and System Safety* **64**(2):135-136.
- Smith, P.J., Shafi, M. & Hongsheng, G. [1997], "Quick simulation: a review of importance sampling techniques in communications systems", *IEEE Journal on Selected Areas in Communications* **15**(4): 597-613. May.

- Sobol, I.M. [1994]. *A primer for the Monte Carlo Method*, CRC Press, Boca Raton, Florida. Traducido de la cuarta edición Rusa (1985).
- Spanier, J. & Gelbard, E.M. [1969]. *Monte Carlo Principles and Neutron Transport Problems*, Addison-Wesley, Reading, Massachusetts.
- Stamatis, D.H. [1995]. *Failure Mode and Effect Analysis FMEA from Theory to Execution*, ASQC Quality Press, Milwaukee, Wisconsin.
- Stecher, K. [1986]. "Evaluation of Large Fault trees with repeated events using an efficient bottom-up algorithm", *IEEE Transactions on Reliability* **35**(1): 51-58.
- Stein, M. [1987]. "Large sample properties of simulations using latin hypercube sampling", *Technometrics* **29**: 143-151.
- Tan, J. & Kramer, M. [1993]. "On-line Hazard aversion: A Bayes approach ", *IEEE Transactions on Reliability* **42**(2): 317-326.
- Tang, B. [1993]. "Orthogonal array-based latin hypercubes", *Journal of american Statistical Association* **88**:1392-1397.
- Teixeira, K., Bhaya, E., Filho, B. & Xerez, L. [1993], "A Knowledge-based system for fault detection and diagnosis in oil production plants in offshore platforms". *International Federation on Automatic Control, III World Congress* **5**, pp. 279-282, Sydney, Australia.
- Teng, S. & Ho, S. [1995]. "Reliability analysis for the decision of fan inflator", *IEEE Quality and Reliability Engineering* **11**(3): 203-214.
- Tezuka, S. [1995]. *Uniform Random Numbers: Theory and Practice*, Kluwer Academic Publishers, Boston.
- Ulerich, N. & Powers, G [1988]. "On-line hazard aversion and diagnosis in chemical processes: The Digraph + Fault Tree method", *IEEE Transactions on Reliability* **37**(2):171-177.
- Ulerich, N.H. [1988]. "Online hazard aversion and fault diagnosis in chemical processes: combined digraph/fault-tree method", *IEEE Transactions on Reliability* 171-177.
- Van Slyke, W.J. & Griffing, D.E [1975]. *ALLCUTS, a fast comprehensive fault tree analysis code*, ARH_ST_112, July. Atlantic Richfield Hanford Company, U.S.A.
- Vaurio, J. [1998]. "An implicit Method for incorporating Common-Cause Failures in System Analysis", *IEEE Transactions on Reliability* **47**(2): 173-180. June.
- Veach, E. & Guibas, L.J. [1995]. "Optimally combining sampling techniques for Monte Carlo rendering", *SIGGRAPH 95 Proceedings*, Addison-Wesley, pp. 419-428.

- Veach, E. [1997]. *Robust Monte Carlo Methods for light transport simulation*. Phd Dissertation, Stanford University, December.
- Vermaak, J. & Niranjana, M. [1998]. "Markov chain Monte carlo methods for speech enhancement", *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing 2*: 134-140.
- Vesely, W.E. [1975]. "Reliability quantification techniques used in the Rasmussen study", *SIAM Reliability and Fault Tree Analysis*, pp. 775-803.
- Vesely, W.E. [1977]. "Time dependent unavailability analysis of nuclear safety system", *IEEE Transactions on Reliability R-26*: 257-260.
- Vesely, W.E. [1969]. *Analysis of fault trees by Kinetic tree theory*, Idaho Nuclear Corporation, U.S.A.
- Vesely, W.E. [1977]. "Estimating common cause failure probabilities in reliability and risk analysis : Marshall_Olkin specialization", *SIAM Nuclear Systems Reliability Engineering and Risk assesment*, pp. 314-341, U.S.A.
- Vesely, W.E. & Narum, R. [1970]. *PREP and KITT computer code for the automatic evaluation of Fault Tree*, Idaho Nuclear Corporation, U.S.A.
- Wagner, D.P., Cate, C.L. & Fussell, J.B. [1977], "Common cause failure analysis methodology for complexsystems", *SIAM Nuclear Systems Reliability Engineering and Risk assesment*, pp. 289-313, U.S.A.
- Wheeler, D.B., Hsuan, J.S., Duersch, R.R. & Glenn, M.R. [1977]. "Fault tree analysis using bit manipulation", *IEEE Transactions on Reliability R-26(2)*: 95-99.
- Worrell, R.B. [1974]. *Set Equation Transformation system [SETS]*, Sandia Laboratories, U.S.A.
- Yellman, T. [1981]. "Event secuence vs the fault tree", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 446-451.
- Zamalani, J. [1998]. "Probabilistic-Risk-Assessment Applications in the Nuclear-Power Industry", *IEEE Transactions on Reliability 47(3)*: 361-364. September.
- Zielinski, P.A. [1989]. "Probabilistic Analysis of Aircraft structure", *IEEE Proceedings Annual Reliability and Maintainability Symposium*, pp. 140-145.

