

# Propagación Historica Software RMES

Centro de Desarrollo de Gestión Empresarial.  
1 Oriente 1097 - Viña del Mar, Chile.  
Fono:(56) (32)688987 - Fax:(56) (32)2684079  
empresa@mes.cl

Marzo, 2011

## Resumen

El presente informe tiene como objetivo documentar el algoritmo historico implementado en RMES para el procesamiento sistémico de fallas explicando los todos procesos involucrados de forma que puedan ser entendidos y evaluados por los expertos en mantención. No utiliza language técnico informático pero si necesita de conocimientos básicos en el uso y conceptos asociados al software RMES.

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Conceptos Propagación Histórica</b>	<b>4</b>
2.1. Falla y Tipos de Falla . . . . .	4
2.2. Configuraciones de Subsistema . . . . .	4
2.3. Subsistemas en Fraccionamiento . . . . .	5
2.3.1. Fraccionamiento Simple . . . . .	6
2.3.2. Fraccionamiento con Capacidad Ociosa . . . . .	6
2.3.3. Fraccionamiento con Redundancia . . . . .	6
2.4. Restricciones y Criterios . . . . .	7
<b>3. Algoritmo Propagación Histórica</b>	<b>9</b>
3.1. Recorrido Post-Orden . . . . .	9
3.2. Falla Unitaria y Falla Compuesta . . . . .	10
3.3. Procesamiento de Equipos . . . . .	10
3.4. Procesamiento de Subsistemas . . . . .	11
3.4.1. Subsistemas Redundantes . . . . .	11
3.4.2. Subsistemas en Línea . . . . .	12
3.4.3. Subsistemas en Fraccionamiento . . . . .	13
3.4.4. Subsistemas Equipo/Planta . . . . .	14
3.5. Resultado Final . . . . .	14
<b>4. Conclusiones</b>	<b>16</b>

## 1. Introducción

El software RMES utiliza un esquema RBD (Reliability Block Diagram) para representar plantas, flotas, etc, existentes en la industria con el fin de obtener indicadores de confiabilidad durante un periodo de tiempo.

Esta representación sólo es posible si todos los componentes constituyentes de estas plantas o flotas pueden ser abstraídos utilizando el concepto de **subsistema**. En RMES, un **subsistema** es un componente *recursivo* que puede contener a otros subsistemas y que puede fallar. Dado que un subsistema puede fallar en distintos intervalos de tiempo, una de sus principales propiedades es que contiene un conjunto de **detenciones**. Una **detención** define el intervalo de tiempo en el cual el subsistema estuvo detenido, en cuanto afectó al subsistema y la razón de la falla, entre otras propiedades.

Con estas definiciones, es posible realizar un estudio de como afectan las detenciones de los subSubsistemas hijos sobre los subSubsistemas padres, entendiéndose esta idea como una forma *recursiva* que comienza desde los subSubsistemas **atómicos** (aquellos que no contiene más subsistemas, ya sea porque son componentes finales o porque la información existente llega a ese nivel) hacia los subsistemas padres llegando hasta el **subsistema raíz**, o sea, la planta o flota completa.

**Gracias el esquema RBD y el usos de los conceptos de sub-Subsistema y detención es posible realizar estudios históricos más precisos a todo nivel del subsistema raíz que un estudio probabilístico basado en fórmulas estadísticas.**

En la Teoría de la Ingeniería en Mantenimiento existen muchas **configuraciones** en las que se puede encasillar a un subsistema. Estas configuraciones rigen el comportamiento de cómo deben ser tratadas las detenciones que llegan desde los subsistemas hijos hacia los subsistemas padres.

A continuación se detallarán los algoritmos que hacen posible la propagación histórica en el software RMES.

## 2. Conceptos Propagación Histórica

### 2.1. Falla y Tipos de Falla

En RMES, el elemento básico de datos es la **Detencion**, informáticamente conocido como **Tupla**, que contiene toda la información necesaria para realizar los estudios históricos.

La falla es una reducción de una detención, la cual sólo conserva la información justa y precisa que permita la propagación hacia niveles superiores del árbol de subsistemas. Las propiedades de una falla son:

- Fecha Original de la falla.
- Inicio y Fin de la falla.
- Tipo de Falla.
- Impacto.

Los **Tipos de Falla** son categorizaciones de las fallas. Con esto es posible obtener distintos indicadores los cuales consideran uno u otro tipo de falla. Los Tipos de Falla son:

- MCM: Mantenición (No Programada) Correctiva Mecánica.
- MCE: Mantenición (No Programada) Correctiva Eléctrica.
- MCI: Mantenición (No Programada) Correctiva Instrumental.
- MP: Mantenición Programada.
- DO: Detencion Operativa (Programada).
- DONP: Detencion Operativa No Programada.
- OD: Retraso Operativo (Programado).
- ODNP: Retraso Operativo No Programado.

### 2.2. Configuraciones de Subsistema

Se define como propagación histórica el proceso de estudiar los efectos de las detenciones que se encuentran el subsistemas hijos sobre los subsistemas padres durante un intervalo de tiempo. El efecto de propagar fallas depende de las configuraciones con que los subsistemas cuenten y de los criterios asociados.

En RMES se definen un subSubsistema atómico conocido como **Equipo** y 5 tipos de configuraciones para subsistemas que pueden contener otros subsistemas. Estos se diferencian en como afectan las fallas de sus subsistemas hijos sobre el subsistema padre. Estas configuraciones son:

1. **Línea**: La falla de un subsistema hijo provoca la detención de todo el sistema.
2. **Paralelo**: Este subsistema falla sólo si todos los subsistemas hijos lo hacen.
3. **StandBy**: Configuración especial que contiene sólo 2 subsistemas, ni más, ni menos. Si uno falla, el otro funciona inmediatamente, por ende, el subsistema falla cuando ambos subsistemas lo hacen.
4. **Redundancia Parcial**: Esta configuración define una nomenclatura de tipo (n:m), donde  $n$  es la cantidad total de subsistemas hijos y  $m$  es la cantidad mínima de subsistemas hijos necesarios en funcionamiento para que el subsistema padre no falle. Entonces, si fallan más de  $(n - m)$  subsistemas hijos fallan, el subsistema falla. Ej: (4:3),

se necesitan mínimo 3 subsistemas en funcionamiento, por ende, si más de  $(4-3) = 1$  subsistemas hijos fallan, el subsistema padre falla.

5. **Fraccionamiento:** Configuración que permite que los subsistemas hijos funcionen a una capacidad fraccionada de su total, provocando fallas fraccionadas sobre el padre. El subsistema padre fallará completamente si todos los subsistemas hijos fallan del todo.

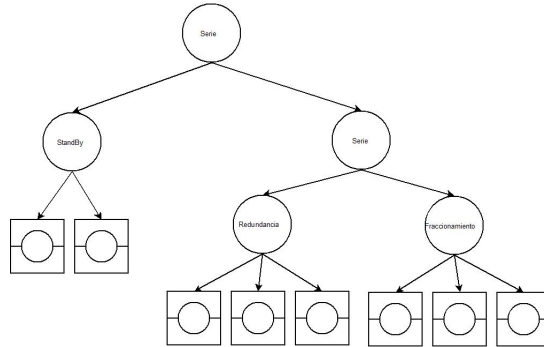


Figura 1: Vista como árbol esquema RBD

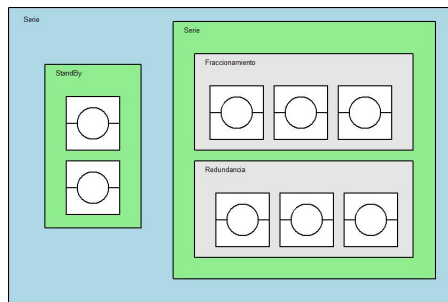


Figura 2: Vista sistémica esquema RBD

### 2.3. Subsistemas en Fraccionamiento

Es necesario realizar un apartado para este tipo de configuración dado que posee ciertas particularidades que lo hacen diferente del resto de los subsistemas y que afectan profundamente al algoritmo de propagación.

Los subsistemas en fraccionamiento poseen hijos "impactados", es decir, sus hijos poseen un **impacto**, que es la tasa de influencia del subSubsistema hijo sobre el subsistema padre en fraccionamiento. El impacto es un valor entre 0 y 1 y corrige el impacto de las fallas que el subsistema hijo contenga (por ejemplo, si un falla tiene impacto de 100 %, y está presente en un subsistema con impacto 30 %, el impacto de la falla se corrige a 30 %, que es simplemente la multiplicación de los 2 impactos).

**El impacto sólo tiene sentido para esta configuración. Para el resto de las configuraciones, el impacto no se utiliza, pero para efectos matemáticos y de uniformidad, todo subsistema que no tenga un padre directo en fraccionamiento posee un impacto 1 (100 %).**

### 2.3.1. Fraccionamiento Simple

La sumatoria de los impactos de los hijos suma exactamente 1, o en término porcentual y más utilizado, 100 %.

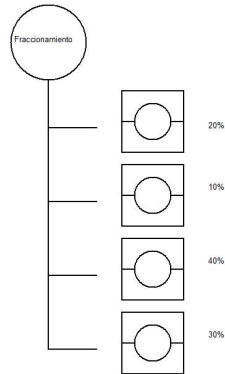


Figura 3: Fraccionamiento Simple

### 2.3.2. Fraccionamiento con Capacidad Ociosa

La sumatoria de los impactos de los hijos puede superar el 100 %, pero si falla 1 subsistema, la sumatoria de impacto de los subsistemas que aún siguen en funcionamiento no debe superar el 100 %. Esto se puede chequear fácilmente restando a la sumatoria de impactos total el menor de los impactos.

Se debe tener completamente en claro que **ningún subsistema, sin importar el tipo de configuración, puede funcionar a más de un 100 %**. Lo que sucede es que existe una sobrecapacidad que se consume a medida que vayan surgiendo fallas. Cuando esta capacidad se consume completamente, el sistema en fraccionamiento comienza a presentar fallas. Este razonamiento se puede aplicar a Sistemas Redundantes, sólo que cuando la sobrecapacidad se consume, el sistema falla completamente, y no de forma fraccionada.

La fórmula matemática que rige la configuración de impactos para los hijos de un Subsistema Fraccionado es:

$$\sum_j I_j - I_i \leq 1 \quad \forall i = \{1, \dots, n\}$$

donde  $\sum_j I_j$  es la sumatoria total de impactos,  $I_i$  es el impacto del subsistema a chequear y  $n$  es la cantidad de subsistemas hijos. Todos los subsistemas hijos deben cumplir esta regla.

### 2.3.3. Fraccionamiento con Redundancia

El fraccionamiento con redundancia es un fraccionamiento con capacidad ociosa pero que no se rige por la regla anteriormente descrita. La sumatoria de impactos puede superar el 100 % y posee una nomenclatura especial  $(n, m)$  en donde  $n$  es la cantidad de subsistemas hijos y  $m$  es la cantidad mínima de subsistemas hijos en funcionamiento para que el subsistema completo funcione a una capacidad del 100 %. Además, **todos los subsistemas hijos poseen el mismo impacto, el cual es  $1/m$** .

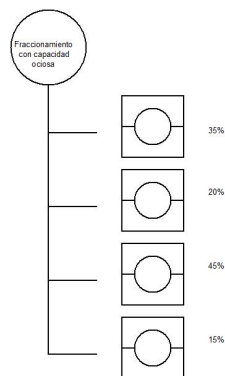


Figura 4: Fraccionamiento con Capacidad Ociosa

Una visión más simple pero incompleta de este tipo de fraccionamiento es que éste se comporta como un subsistema en Redundancia Parcial hasta que el mínimo de equipos en funcionamiento no se cumple. Luego, el subsistema funciona como un fraccionamiento simple. Este enfoque es incompleto dado que no se puede aplicar si los hijos poseen sistemas fraccionados, dado que los impactos generalmente son menores, por ende, es posible que la sumatoria de impactos de los  $m$  subsistemas hijos no sea suficiente para impactar una detención sobre el subsistema padre.

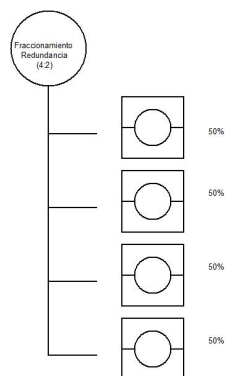


Figura 5: Fraccionamiento con Redundancia (4:2)

## 2.4. Restricciones y Criterios

Últimamente, debido a ciertos problemas en los criterios de la propagación, en como tratar la fallas bajo ciertas situaciones y con el fin de uniformar y pulir ambigüedades, el algoritmo de propagación ahora cuenta con ciertas restricciones que solucionan varios problemas de las versiones antiguas y permite mayor rapidez y eficiencia en los cálculos.

1. La duración mínima de una detención/falla o de un intervalo de tiempo es de 1[s].
2. La duración máxima de una detención/falla o de un intervalo de tiempo es de exactamente **68 [años], 15 [días], 0[meses], 3 [ho-**

- ras], 14 [minutos], 7 [segundos]. Es decir, no se podrán ingresar fallas con una duración superior a esta ni tampoco realizar un estudio que abarcan un intervalo de tiempo superior.
3. Se definen Subsistemas Redundantes como aquellos que poseen "sobrecapacidad entera", es decir, la falla sobre el subsistema ocurre sólo si se han acumulado cierta cantidad de detenciones. Además, las fallas que llegan al subsistema padre no son fraccionadas, siempre impactan en un 100%. Estos subsistemas son: Redundancia, Paralelo y StandBy.
  4. Se definen Subsistemas Fraccionados como aquellos subsistemas que pueden contener detenciones fraccionadas, es decir, una detención en algún hijo puede no impactar en un 100%, sino que en una fracción menor. Estos sistemas funcionan a una capacidad menor al 100% mientras alguno de sus subsistemas hijos continúe funcionando, y sólo se detendrá completamente si todos los subsistemas hijos fallan. Dentro de esta categoría están Fraccionamiento y un subsistema especial conocido como Equipo/Planta.
  5. Un Equipo/Planta es un Equipo que puede contener detenciones fraccionadas y su función principal es poder resumir la información de todas las detenciones de un subsistema mayor en esta configuración especial almacenando las fallas que se hayan obtenido de algún estudio anterior.
  6. De lo anterior, Subsistemas Redundantes no puede contener Subsistemas Fraccionados a ningún nivel. Esto, debido a que las fallas fraccionadas no pueden ser procesadas por los Subsistemas Redundantes.
  7. La configuración en Serie, o un Sistema en Serie, es una configuración especial que acepta todo tipo de subsistemas y puede contener fallas fraccionadas. Por definición, es una configuración ambigua que trabaja tanto como un Sistema Redundante como Fraccionado.
  8. Un Subsistema Redundante puede contener Subsistemas en Serie siempre y cuando este último no contenga Subsistemas Fraccionados a ningún nivel.
  9. Subsistemas en Serie y Subsistemas Fraccionados pueden contener cualquier tipo de Subsistema.
  10. El criterio de ordenamiento de las fallas se realiza según la fecha original de la detención desde donde la falla fue creada. Si 2 fallas tienen exactamente la misma fecha de inicio, el orden se realiza según el tipo de falla el siguiente orden jerárquico: MCM, MCE, MCI, MP, DO, DONP, OD, ODNP.

### 3. Algoritmo Propagación Histórica

#### 3.1. Recorrido Post-Orden

El Recorrido Post-Orden es un concepto informático que se refiere a la forma de recorrer un árbol jerárquico. En el problema de mantenimiento, este árbol jerárquico es la representación RBD de una planta o flota (figura 1). El Recorrido Post-Orden define 2 acciones:

- Visita: que es simplemente llegar al nodo.
- Procesamiento: realizar alguna tarea sobre el nodo.

El **Recorrido Post-Orden primero procesa los hijos de un nodo y luego al nodo mismo**, como se muestra en la figura 6. Esta definición es recursiva, de modo que si los hijos de un nodo a su vez poseen hijos, primero se procesaran los hijos de los hijos del nodo, luego los hijos del nodo y finalmente al nodo mismo.

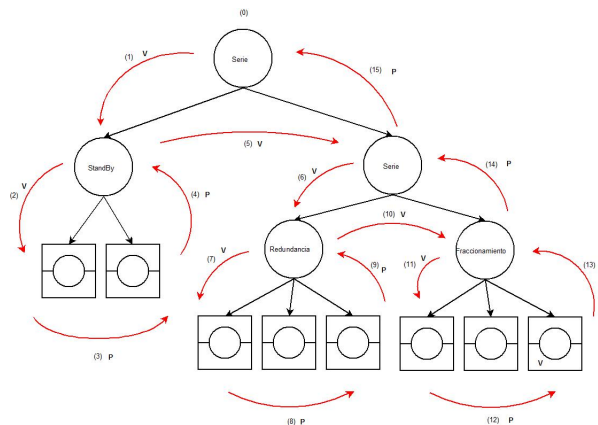


Figura 6: Recorrido Post-Orden. V: Visita, P: Procesamiento

Observando la figura 6:

- En el punto (0) comienza el recorrido. En ese punto, el nodo contiene 2 hijos, por ende primero visita a los hijos.
- En el punto (1) se visita al primer hijo de la raíz. Dado que este contiene 2 hijos más, se visitan éstos primero.
- En el punto (2) se visita al primer hijo del Subsistema StandBy.
- Sólo en el punto (3) se realiza el procesamiento de los hijos del subsistema StandBy.
- En el punto (4), y sólo después de haber procesado todos los hijos del StandBy, se procesa el subsistema.

Los siguientes puntos siguen la misma estructura explicada anteriormente. En resumen, las flechas que apuntan hacia abajo implican visita, y las que apunta hacia arriba, implican procesamiento.

Lo importante de comprender este concepto es que **en todo momento, al procesar un nodo padre, los hijos (y todo su descendencia) ya han sido procesados y tiene toda la información lista para ser usada por el padre.**

A continuación, se explicarán los procesamientos realizados a los nodos según sean sus categorías.

### 3.2. Falla Unitaria y Falla Compuesta

Estos conceptos son la base del algoritmo de propagación. Como ya se explicó anteriormente, la falla (y en este contexto, falla unitaria) es simplemente una copia de una detención, pero más simple y con la información justa y precisa. Una falla compuesta es simplemente un conjunto de fallas (unitarias o compuestas) que comparten el mismo intervalo de tiempo. La sumatoria de impactos del conjunto de fallas de una falla compuesta es menor o igual a 100%. Además, la fecha original de una falla compuesta será la fecha original más anterior de entre todas las fallas que posea.

Como criterio del algoritmo de propagación, el proceso finalizado de propagación dejará en todo Sistema Redundante, incluyendo al Equipo simple, fallas unitarias. Los subsistemas en Serie, Fraccionados o los Equipo/Planta podrán contener fallas compuestas y que sólo contendrán fallas unitarias. Esto será explicado en las próximas secciones.

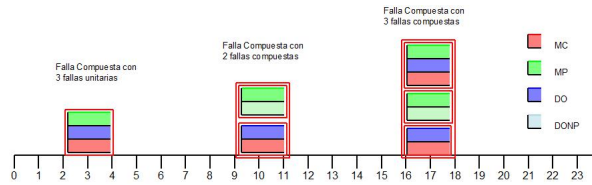


Figura 7: Ejemplos de fallas compuestas.

### 3.3. Procesamiento de Equipos

El Algoritmo de Propagación Histórica a nivel atómico o de equipo es bastante simple y necesita de las siguientes condiciones:

1. **Las detenciones en los equipos viene ordenadas:** condición no necesaria pero útil para agilizar los cálculos.
2. **Las detenciones en los equipos no se encuentran traslapadas:** a lo más, las detenciones pueden venir una inmediatamente de la otra, pero no "una sobre otra" (en otras palabras, de todo el conjunto de detenciones que el equipo contenga, no existe intersección entre 2 detenciones, cualquiera sean).

El algoritmo a este nivel simplemente transforma las detenciones a fallas copiando las propiedades que se necesitan de las primeras sobre las últimas.

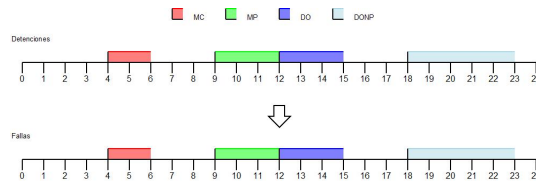


Figura 8: Procesamiento de fallas a nivel de equipo

### 3.4. Procesamiento de Subsistemas

A nivel de Subsistema, cualquiera sea su configuración, es necesario realizar 3 subprocesos básicos:

1. Procesar las detenciones que contenga el subsistema: proceso similar al realizado sobre un equipo, se transforman las detenciones a fallas del subsistema. Las detenciones siempre tendrán impacto igual a 1, exceptuando el caso de detenciones cargadas sobre un Equipo/Planta, en donde las detenciones (que en realidad representa fallas) pueden venir impactadas.
2. Procesar las fallas que tengan los subsistemas hijos: el proceso también es simple. Las fallas de los hijos ya vienen listas para ser utilizadas, por ende, solo se copian directamente hacia el padre, modificando los impactos de las fallas según el impacto del hijo que provenga (simplemente multiplicando ambos impactos), lo cual sólo se aplica a subsistemas en Fraccionamiento, como se explicó en las secciones anteriores.
3. Procesar en conjunto las fallas del padre y de los hijos: este proceso es algo más elaborado, por lo que se explicará a continuación.

El acto de unir las fallas de padre e hijos, informáticamente, se conoce como **Spliteo**. Este subproceso técnicamente une el conjunto completo de fallas padre/hijos en una sólo línea temporal. Luego, se realizan cortes por inicio y fin de todas las fallas, generando más fallas de duración menor. **Todas las fallas que se generaron a partir de otra comparten la misma fecha original**, concepto importante dado que se utiliza como criterio de ordenamiento y jerarquización. Observando la figura 9, la primera falla MCM del subsistema S, se dividió en 2 fallas de duración distinta, con intervalos distintos pero con la misma **fecha original**.

Se reitera que las fallas de los hijos en este punto se encuentran procesadas, en particular, los impactos ya están corregidos. Luego de este subproceso, se continua con el procesamiento particular según sea el tipo de configuración del subsistema. El estudio se realiza intervalo por intervalo respecto a los generados por el subproceso anterior. Por ejemplo, observando la figura 9, se obtiene los siguientes intervalos:

- De 9:00 hrs a 11:00 hrs.
- Intervalos de 1 hora desde las 11:00 hrs a las 19:00 hrs.

Por cada intervalo se elegen las fallas que finalmente quedaran sobre el subSubsistema. **Si alguna de estas fallas pertenece directamente al subsistema, esta será la única que prevalecerá.**

#### 3.4.1. Subsistemas Redundantes

El algoritmo para procesar subSubsistemas en Redundancia Parcial, en Paralelo o en Standby es el mismo. Cada uno de estos subsistemas define un valor informático conocido como "Fail Break Count" (FBC), que se refiere a la cantidad de fallas que el subsistema soporta hasta de dejar de funcionar, es decir, si la cantidad de fallas en un intervalo de tiempo es igual o supera este valor, el subsistema se detiene.

Sea un subsistema Redundante de tamaño  $n$ , si se encuentra en paralelo soporta  $n - 1$  fallas, por ende, su valor FBC sera de  $n$ . El valor para un subsistema en StandBy es 2. Para un subsistema en Redundancia Parcial

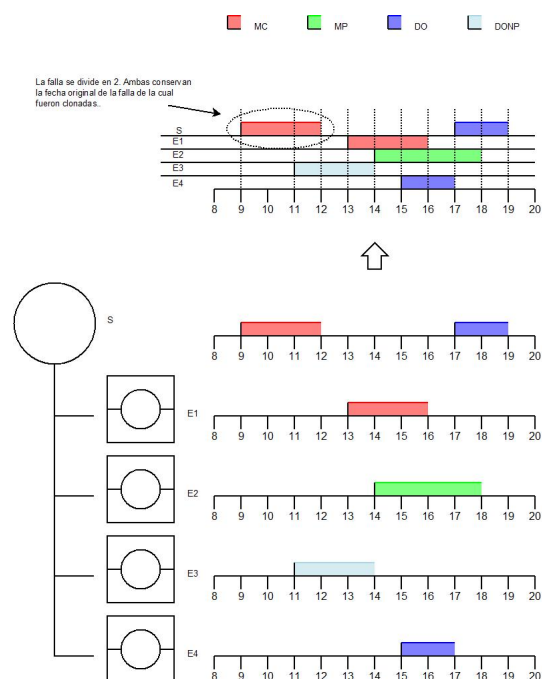


Figura 9: Subproceso de Spliteo. Se unen las fallas provenientes de los hijos y del subSubsistemas. El subproceso guillotina cada falla según las fechas de inicio y fin todas las fallas creando varios intervalos, dejando en cada uno un conjunto de fallas que se procesan según la configuración del subsistema principal.

$(n, m)$ , el valor FBC es  $(n - m + 1)$ . Por ejemplo, para un  $(4:3)$ , el FBC es 2.

Adecuemos el ejemplo de la figura 9 a una Redundancia Parcial  $(4 : 3)$ . El resultado sería el que se presenta en la figura 10. Al subsistema sólo sube una falla unitaria por cada intervalo, y es aquella que hace detener al subsistema. Las fallas asociadas directamente al subsistema prevalecen siempre.

En nuestro caso, el valor FBC es 2. La fecha original de la falla es importante para subsistemas Redundantes. Por ejemplo, si se observa el intervalo entre las 13:00 hrs y 14:00 hrs, existe una falla MCM perteneciente al equipo E1 y una falla DONP perteneciente al equipo E3. Dado que esta última falla es la que inicia primero, la culpable de que en ese intervalo de tiempo el subsistema en Redundancia Parcial falle es la falla MCM del equipo E1, por lo tanto, es la falla que sube al subsistema en Redundancia Parcial.

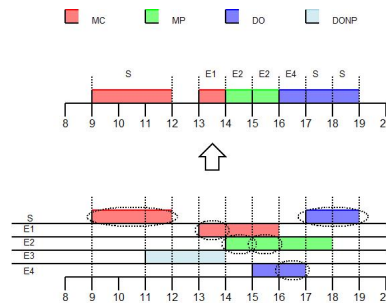


Figura 10: Subproceso para Subsistemas Redundantes

### 3.4.2. Subsistemas en Línea

Los subsistemas en línea tienen la particularidad de que fallan si es que alguno de sus subsistemas hijos lo hace. Dado que este subsistema soporta fallas fraccionadas, el criterio de general de jerarquización es el siguiente: del conjunto de fallas existentes en un intervalo, se elige la de mayor impacto, en caso de que hubiesen varias, se elige la que haya comenzado antes (menor fecha original).

La figura 11 modifica el ejemplo general de forma que las fallas en cada equipo tengan un impacto asociado cambiando los equipos por fraccionamientos, para soporte de fallas fraccionadas y, para no generar confusión, los impactos no están asociados a los subsistemas fraccionados, sino que a las fallas que subieron en ese momento particular (las fallas son las que tienen el impacto, no los fraccionamientos. Recordar que no tiene sentido de que los hijos de un subsistema en serie tenga impacto).

### 3.4.3. Subsistemas en Fraccionamiento

A pesar de que existen 3 tipos de modelos distintos en RMES para Fraccionamiento, el algoritmo histórico puede ser resumido en uno solo. En primer lugar, se busca la sobrecapacidad (OC) del subsistema que es en cuánto más el subsistema soporta su no caída. Esto se obtiene de la sumatoria de impactos menos 100%:

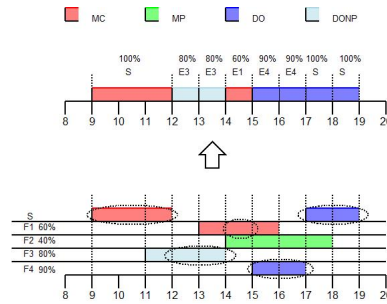


Figura 11: Subproceso para Subsistemas en Línea

$$OC = \sum_i I_i - 1$$

donde  $I_i$  es el impacto de los subsistemas hijos del subsistema en Fraccionamiento.

La sobrecapacidad de un fraccionamiento simple es 0, la de uno con capacidad ociosa no supera al impacto mínimo del conjunto de subsistemas hijos y la de uno con redundancia  $(n, m)$  será  $\frac{n}{m} - 1$ .

Luego, si la sumatoria de los impactos de las fallas que se encuentren en cierto intervalo de tiempo supera la sobrecapacidad, todos estas fallas suben al subsistema corrigiendo sus impactos. Supongamos que tenemos  $k$  fallas que suben al subsistema, la corrección para ese intervalo se realiza de forma ponderada según los impactos individuales:

1. Se obtiene la sumatoria original de impactos de las fallas en el intervalo  $IF_t = \sum IF_i$ , donde  $IF$  es el impacto de la falla  $i$ -ésima.
2. Se obtiene la sumatoria corregida de impactos  $IF'_t = \sum IF_i - OC$
3. Se obtiene la tasa  $r$  de ponderación original para cada falla  $r = IF_i / IF_t \quad \forall i = 1, \dots, k$
4. Se asigna el nuevo impacto a las fallas  $IF_i = r * IF'_t$

La figura 12 muestra esta corrección gráficamente.

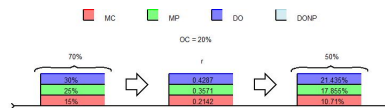


Figura 12: Corrección de Sobrecapacidad

La figura 13 muestra el proceso para un fraccionamiento simple.

La figura 14 muestra el proceso para un fraccionamiento con capacidad ociosa.

En el caso de que al subsistema en fraccionamiento hayan subido varias fallas del mismo tipo, el algoritmo las une en una gran falla del mismo tipo sumando sus impactos. La figura 15 grafica el proceso de unión de fallas.

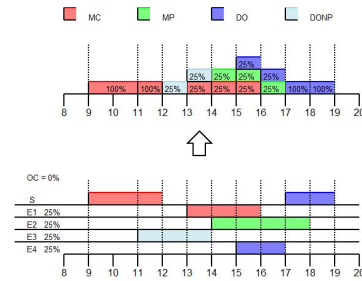


Figura 13: Fraccionamiento Simple

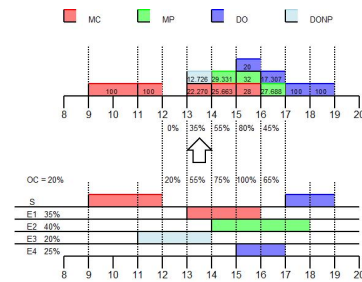


Figura 14: Fraccionamiento con Capacidad Ociosa

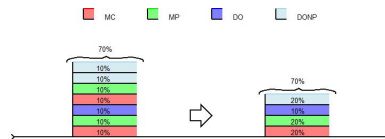


Figura 15: Unión de fallas

#### 3.4.4. Subsistemas Equipo/Planta

En este tipo de subsistemas no existen criterios posteriores, simplemente, del proceso de Spliteo, si en un intervalo la cantidad de fallas es mayor a 1, se crea una falla compuesta que contenga todas las fallas unitarias del intervalo, de lo contrario, la unica falla que el intervalo contiene se deja como tal, la cual sólo puede ser unitaria.

### 3.5. Resultado Final

Como ya es de esperarse en vista a los ejemplos de las secciones anteriores, el resultado final del algoritmo de propagación es un conjunto de fallas unitarias y/o compuestas, donde estas últimas sólo contendrán fallas unitarias, conjunto que no tendrá un tamaño superior a 8. Esto debido a que los subsistemas en fraccionamientos son los únicos capaces de crear fallas compuestas, y a su vez, unen fallas del mismo tipo en una sólo, dejando a lo mas 8 posibles fallas (la cantidad de tipos de falla) presentes en una falla compuesta.

Esta regla puede ser rota por subsistemas Equipo/Planta, dado que la carga de fallas se realiza de forma externa. Esto no afecta al proceso de conteo de RMES (posterior a la propagación y que no se analizará en este documento).

## 4. Conclusiones

El documento presentó de la forma más clara posible los algoritmos y criterios asociados al algoritmo de propagación histórica implementado últimamente en el software RMES.

Todas las modificaciones realizadas al algoritmo antiguo de propagación fueron con el fin de soportar cantidades masivas de detenciones, situación que ocurre en flotas (conjuntos de fraccionamientos). Los cambios hechos agilizan el cálculo y utilizan menos recursos computacionales, pero generan pérdida de información que en el futuro puede utilizarse y actualmente no se hace, como lo es el origen preciso de las fallas desde los equipos hacia los subsistemas. Actualmente, sólo se almacena el origen de la falla y el subsistema desde cual fue clonada la última vez, información que se pierde en sistemas con Fraccionamiento debido a la unión de fallas según tipo, en donde se crean fallas unitarias nuevas.

Es labor de los expertos verificar la correctitud de los criterios implementados y su modificación en caso de encontrar detalles.