Métodos Monte Carlo en la estimación de la confiabilidad de redes de comunicaciones – Estado del Arte –

Héctor Cancela *María E. Urquhart*
cancela@fing.edu.uy urquhart@fing.edu.uy
Investigación Operativa *
Instituto de Computación, PeDeCiBa Informática
Facultad Ingeniería, Montevideo, Uruguay

Diciembre 22, 1994

Resumen — Una red de comunicaciones es un sistema compuesto por un conjunto de centros que reciben y envían información, y un conjunto de conexiones que transportan esa información. La capacidad de una red de comunicaciones para resistir a posibles fallos de algunos de sus componentes se evalúa a través de distintas métricas de confiabilidad.
La confiabilidad de la red es una medida definida como la probabilidad de comunicación exitosa entre diferentes pares o conjunto de centros, dadas las probabilidades de buen funcionamiento de los componentes de la red y la topología de las conexiones. Consideraremos el cálculo de la medida $R_{\mathcal{K}}$ (confiabilidad entre el conjunto K de terminales) en el caso de una red con nodos perfectos y líneas falibles. La evaluación exacta de esta medida es, en general, muy costosa computacionalmente dado que pertenece a la familia de problemas NP duros. Desde el punto de vista práctico esto quiere decir que el tiempo necesario para su evaluación es una función exponencial en el tamaño (cantidad de elementos) de la red.
Un enfoque alternativo para evaluar modelos de cierto porte (mas de 100 elementos) es el de realizar estimaciones utilizando métodos de simulación Monte Carlo. En el caso de redes altamente fiables, las técnicas Monte Carlo "crudas" (o ingenuas) resultan tambien prohibitivamente consumidoras de recursos computacionales lo que obliga a la utilización de técnicas de reducción de varianza que dan resultados en tiempos más razonables de cálculo.
En este informe pasamos revista a las diferentes técnicas que han sido propuestas en este contexto, discutiendo brevemente sus ventajas y desventajas.
Palabras clave — Confiabilidad de redes, métodos Monte Carlo, reducción de varianza.

Abstract — In the evaluation of the capacity of a communication network architecture to resist to the possible failures of some of its components, several reliability metrics are currently used. In this paper we consider the *$\mathcal{K}$-terminal reliability* measure $R_{\mathcal{K}}$. The exact evaluation of this parameter is in general a very costly task since it is, as an algorithmic problem, classed in the NP–hard family. An alternative approach to the exact evaluation is to perform estimations using Monte Carlo simulation methods. In the case of highly reliable networks, the crude Monte Carlo approach is also prohibitively expensive and variance reduction techniques must be used. We review here different Monte Carlo schemes specifically designed to this context.
Key words — Network reliability, Monte Carlo methods, variance reduction.

*Este trabajo es patrocinado por el CONICYT y el ECOS francés

1 Introducción

Una red de comunicaciones es un sistema compuesto por un conjunto de centros, puntos, o terminales que reciben y envían información, y un conjunto de enlaces o conexiones que transportan esa información. Una red se puede modelar mediante un grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ conexo y sin bucles, cuyo conjunto $\mathcal{V}$ corresponde a los centros de la red y cuyo conjunto $\mathcal{E}$ de aristas corresponde a las lineas de conexión.

Nosotros consideraremos una red $\mathcal{G}$ donde los nodos son perfectos y las líneas fallan de forma aleatoria, independientemente entre si y con probabilidades conocidas.

La comunicación exitosa entre todos los nodos de un subconjunto $\mathcal{K}$ fijo del conjunto de nodos es un evento aleatorio. La probabilidad $R_{\mathcal{K}}$ de este evento se denomina *$\mathcal{K}$-confiabilidad* o *confiabilidad entre terminales*. El problema de la evaluación de esta medida ha recibido considerable atención de la comunidad académica ([7] y [23] contienen una abundante bibliografía). Una de las razones de este interés es que, en el caso genérico, el problema del cálculo exacto de las medidas de confiabilidad pertenece a la clase de problemas NP-completos [2], [3], [25]. Esto significa, en grandes líneas, que es sumamente improbable que exista un algoritmo general para calcular $R_{\mathcal{K}}$ en forma exacta y cuyo tiempo de ejecución crezca polinomialmente con el tamaño del problema [16]. Hacemos notar que calcular $R_{\mathcal{K}}$ es equivalente a calcular el parámetro de anti-confiabilidad entre terminales $Q_{\mathcal{K}} = 1 - R_{\mathcal{K}}$.

El tiempo de cálculo necesario para evaluar la confiabilidad en un modelo general de tamaño mediano (con algunas docenas de líneas) es prohibitivo, inclusive en una estación de trabajo. Por una ilustración numérica ver [8].

Desde el punto de vista práctico, esto significa que se pueden evaluar en forma exacta, redes con decenas de elementos pero no con un centenar de ellos, tamaño que facilmente alcanza cualquier sistema de comunicaciones.

Una opción más eficaz para evaluar modelos de mayor tamaño es la estimación estadística de esta medida mediante técnicas de tipo Monte Carlo .

La simulación por Monte Carlo "crudo" consiste en ejecutar de forma independiente N veces el siguiente experimento: a) para cada línea de una red dada realizar una prueba Bernoulli, b) deducir una sub-red mediante la eliminanción de aquellas aristas que "no funcionan"; c) la estimación de $R_{\mathcal{K}}$ es entonces la frecuencia de sub-redes en las que los nodos del conjunto $\mathcal{K}$ están conectados.

El principal problema de la simulación estándar (cruda) es el crecimiento no acotado del error relativo cuando $R_{\mathcal{K}}$ se acerca a 1, problema común en todas las redes altamente confiables [13]. Esto hace necesario la realización de una gran cantidad de experimentos para disminuir ese error (N muy grande) lo cual torna el método ineficaz para modelos de tamaño mayor. Entonces se plantea la alternativa de utilizar técnicas de reducción de varianza cuyo propósito general es el de ofrecer un estimador más preciso que aquél que brinda la simulación estándar, con el mismo número de experimentos.

En este trabajo presentamos los más relevantes esquemas Monte Carlo para la estimación del parámetro de confiabilidad de redes $R_{\mathcal{K}}$.

Existen otros métodos tanto determinísticos como probabilísticos de evaluación de la confiabilidad, que están tratados en los trabajos recientes [28], [4], [8].

Este informe está organizado de la forma siguiente. Luego de resumir las notaciones y definiciones principales en la Sección 2, el método Monte Carlo crudo se presenta en la Sección 3; las técnicas de reducción de varianza siguen en la Sección 4. La Sección 5 está dedicada a la presentación de algunos resultados numéricos.

2 Notaciones y definiciones globales

En esta sección se resumen las principales notaciones y definiciones utilizadas a lo largo del presente trabajo.

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: la topología de la red analizada (grafo subyacente);
  - $\mathcal{V} = \{1, \ldots, n\}$: el conjunto de nodos de la red;

- $\mathcal{E} = \{l_1, \ldots, l_m\}$: el conjunto de aristas o líneas de la red;
- m: la cantidad de aristas de $\mathcal{G}$;
- n : la cantidad de nodos de $\mathcal{G}$;

- $\mathcal{K} \subseteq \mathcal{V}$: el conjunto de terminales, es decir el subconjunto de nodos que deben poder comunicarse entre si para considerar a la red como operativa;
- x_l: la variable aleatoria binaria "estado de la línea l en $\mathcal{G}$", definido por

$$x_l = \begin{cases} 1 & \text{si la línea } l \text{ funciona,} \\ 0 & \text{si la línea } l \text{ no funciona;} \end{cases}$$

- p_l: la probabilidad que la línea l funcione, es decir, $p_l = \mathcal{P}(x_l = 1)$;
- q_l: la probabilidad que la línea l no funcione, es decir, $q_l = \mathcal{P}(x_l = 0) = 1 - p_l$;
- $X = (x_1, \ldots, x_m)$: el vector (aleatorio) de estados de la red;
- $\tilde{X} = (\tilde{x}_1, \ldots, \tilde{x}_m)$: es el vector (aleatorio) correspondiente a una observación del estado de la red;
- Para una línea cualquiera $l = \{u, v\} \in \mathcal{E}$,
  - $\mathcal{G} - l$: es el grafo $(\mathcal{V}, \mathcal{E} - \{l\})$;
  - $\mathcal{G} * l$: es el grafo deducido de $\mathcal{G}$ mediante la contracción de la arista l. Esto consiste en eliminar la línea l y unir sus extremos u y v que pasan a ser un único nodo.

Cuando una arista l está en funcionamiento, es posible realizar una contracción de la misma, y cuando una arista l está fallada, es posible eliminarla. Los parámetros de confiabilidad de la red se mantienen incambiados.

- $\mathbf{1}_E$: la función indicatriz del evento E;
- $\overline{E}$: el evento complementario del evento E;
- $\mathcal{P}(E)$: la probabilidad de ocurrencia del evento E;
- $\mathcal{P}\left(X = \tilde{X}\right)$: la probabilidad que el vector de estados del grafo sea $\tilde{X}$.
- Φ: la función de estructura asociada con la medida de $\mathcal{K}$-confiabilidad.
  Es la función de $\{0,1\}^m$ en $\{0,1\}$ que toma valor 1 o 0, según si la red $\mathcal{G} - \{\text{toda arista fallada}\}$ es $\mathcal{K}$-conexa o no respectivamente.
- $\Phi(X)$ la variable aleatoria binaria "estado de la red $\mathcal{G}$",

$$\Phi(X) = \begin{cases} 1 & \text{si el grafo } \mathcal{G} - \{\text{toda arista fallada}\} \text{ es } \mathcal{K}\text{-conexo,} \\ 0 & \text{de otro modo;} \end{cases}$$

- Un $\mathcal{K}$-corte es un conjunto de componentes ($\mathcal{K}$ líneas) cuya falla asegura la falla del sistema;
- $R_{\mathcal{K}} = \mathcal{P}(\Phi(X) = 1) = \mathrm{E}\{\Phi(X)\}$ el parámetro de $\mathcal{K}$ confiabilidad .
- $Q_{\mathcal{K}} = \mathcal{P}(\Phi(X) = 0) = \mathrm{E}\{1 - \Phi(X)\} = 1 - R_{\mathcal{K}}$ el parámetro de anti-confiabilidad.

3 Técnica Monte Carlo "cruda"

El estimador sin sesgo del parámetro de confiabilidad $R_{\mathcal{K}}$ correspondiente a la técnica Monte Carlo cruda es una media de muestreo $\widehat{R}$. Más precisamente,

$$\widehat{R} = \frac{1}{N} \sum_{i=1}^{N} \Phi(X^{(i)})$$

donde $X^{(1)}, \ldots, X^{(N)}$ son muestras aleatorias de X. La varianza de este estimador es

$$\mathrm{Var}\left(\widehat{R}\right) = \frac{\mathrm{Var}\left(\Phi(X)\right)}{N} = \frac{Q_{\mathcal{K}} R_{\mathcal{K}}}{N}$$

aproximada por el estimador sin sesgo

$$\widehat{V} = \frac{\widehat{R}(1-\widehat{R})}{(N-1)} = \frac{1}{N(N-1)} \sum_{i=1}^{N} \left(\Phi(X^{(i)}) - \widehat{R}\right)^2.$$

El algoritmo de simulación consiste en repetir independientemente N veces el experimento siguiente. Una muestra de cada variable x_l es considerada en orden para formar una muestra del vector de estado X. Luego se ejecuta un procedimiento de búsqueda en profundidad (Depth First Search) para verificar si los nodos de $\mathcal{K}$ están conectados en la subred resultante de $\mathcal{G}$. La estimación de $R_{\mathcal{K}}$ es la frecuencia de subredes en las cuales los nodos de $\mathcal{K}$ están conectados. El algoritmo puede ser expresado como sigue:

Inicialización : $\widehat{R} = 0$.
Para cada experimento $n = 1, \ldots, N$
    Para cada línea $l = 1, \ldots, m$
        Sortear una muestra $\mathcal{U}$ $Uniforme(0, 1)$;
        Si ($\mathcal{U} \in [0, p_l[$)
            $x_l = 1$
        si no $x_l = 0$.
    finPara
    Evaluar $\Phi(X)$
    $S = S + \Phi(X)$
finPara
Estimar $R_{\mathcal{K}}$: $\widehat{R} = S/N$.
Estimar $\mathrm{Var}\,(R_{\mathcal{K}})$: $\widehat{V} = \widehat{R}(1-\widehat{R})/(N-1)$.

Por cada muestra se realiza el sorteo del estado de todas las líneas en $\mathcal{E}$ y la evaluación de la función de estructura, esta última a traves de una búsqueda en profundidad (DFS) La complejidad por cada muestra es de $O(max(|\mathcal{E}|, |\mathcal{V}|)) = O(|\mathcal{E}|)$.[30].

Es posible mejorar este algoritmo mediante un esquema de muestreo dinámico. Para esto, se aprovecha que la evaluación de la función de estructura puede realizarse con una simple búsqueda en profundidad (DFS), y entonces no es necesario generar las variables aleatorias correspondientes a todas las aristas, sino solamente a aquellas que van siendo alcanzadas durante la búsqueda. El orden de complejidad por muestra continúa siendo $O(|\mathcal{E}|)$, pero la constante asociada es mucho menor. Resultados experimentales citados en [10] muestran que esta simple estrategia permite, para un grafo de más de 7000 aristas, alcanzar reducciones de 5 a 30 veces el tiempo de ejecución frente a la version original.

El problema del método Monte Carlo crudo es el gran número de muestras necesario para obtener estimadores precisos cuando la red es altamente confiable [20]. Para mitigar parcialmente esta situación una posibilidad es el desarrollo de versiones paralelas del procedimiento. Las referencias [10], [9] discuten la implementación en un hipercubo iPSC2 de 64 procesadores del algoritmo Monte Carlo crudo. La aceleración (speedup) obtenida es prácticamente lineal en el número de procesadores utilizados, lo que muestra el interés en la utilización de esta técnica.

Las dos grandes alternativas para atacar el problema del gran número de muestras para la obtención de estimadores precisos, son: a) disminuir la complejidad computacional de la obtención de cada muestra, o b) utilizar muestras cuya varianza sea menor que la utilizada en el método crudo.

Este segundo tipo de técnicas reciben el nombre de métodos de reducción de varianza, y serán el objeto de la próxima sección.

4 Métodos de reducción de varianza

Diversas técnicas genéricas de muestreo permiten obtener reducción de varianza en los métodos Monte Carlo: uso de variables antitéticas, variables de control, muestreo estratificado, y muestreo según importancia.

Estas técnicas son de distinta aplicabilidad para la estimación de la confiabilidad, y han inspirado varios, aunque no todos, de los métodos que veremos a continuación. En todos los casos, la estructura de base es similar a la de Monte Carlo crudo:

Inicialización $\mathcal{I}$: $S_1 = 0$; $S_2 = 0$;
Para $n = 1, \ldots, N$
    Ejecutar $M(\mathcal{G}, \mathcal{K})$; (devuelve R)
    $S_1 = S_1 + R$; $S_2 = S_2 + R^2$;
finPara
Estimar $R_\mathcal{K}$: $\widehat{R} = S_1/N$.
Estimar $\text{Var}\,(R_\mathcal{K})$: $\widehat{V} = \frac{S_2 - S_1^2/N}{N(N-1)}$.

Es decir, primero se ejecuta un procedimiento de inicialización $\mathcal{I}$; luego se ejecuta N veces un procedimiento $\mathcal{M}()$ y se acumulan sus resultados; por último se estiman $R_\mathcal{K}$ y $\text{Var}\,(R_\mathcal{K})$.
Cada método tendrá que incorporar a este esquema sus propios procedimientos de inicialización $\mathcal{I}$ (que puede ser vacía) y de obtención de muestras $\mathcal{M}()$, ésta última debe generar un valor aleatorio correspondiente a una variable de media $R_\mathcal{K}$, lo que permitirá afirmar que el estimador no posee sesgo. Si además su varianza es menor que la correpondiente al método Monte Carlo crudo, diremos que es más preciso que éste.

4.1 Muestreo basado en cotas

Esta familia de métodos es un híbrido del muestreo según importancia clásico con esquemas de variables de control. Su primer aplicación a problemas de confiabilidad de redes fue presentada por Van Slyke y Frank [31], y posteriormente por Kumamoto [21] y Fishman [15]. Puede ser aplicada a cualquier problema de evaluación de la confiabilidad en el que se conozcan para la función Φ de estructura del sistema dos funciones Φ^L y Φ^U que la acotan inferior y superiormente en forma respectiva. Estas funciones deben cumplir las propiedades siguientes:

- $\Phi^L(X) \leq \Phi(X) \leq \Phi^U(X)$ para todo vector de estado X.
- Para $k = 0, \ldots, |\mathcal{E}|$ y para cualquier asignación de valores $\tilde{X}^{(k)} = (\tilde{x}_1, \ldots, \tilde{x}_k)$ para los primeros k componentes de X, los valores

$$R_k^L(\tilde{X}^{(k)}) \equiv \mathcal{P}\left(\Phi^L(X) = 1 | x_1 = \tilde{x}_1, \ldots, x_k = \tilde{x}_k\right)$$

y

$$R_k^U(\tilde{X}^{(k)}) \equiv \mathcal{P}\left(\Phi^U(X) = 1 | x_1 = \tilde{x}_1, \ldots, x_k = \tilde{x}_k\right)$$

pueden ser calculados en tiempo polinomial.

Los valores R_0^L y $1 - R_0^U$ son las probabilidades de un par de eventos conocidos en los cuales la función de estructura Φ tiene valores conocidos (respectivamente 1 y 0), es decir, $R_0^L = \mathcal{P}\left(\Phi^L(X) = 1\right)$ y $R_0^U = \mathcal{P}\left(\Phi^U(X) = 1\right)$.
Para el muestreo basado en cotas, se define el espacio sobrante

$$W = \{X : \Phi^L(X) = 0, \Phi^U(X) = 1\}$$

de donde se extraerán las muestras en proporción a su probabilidad en el espacio original. A partir del estimador obtenido allí y la información anterior se reconstruye un estimador de la confiabilidad del sistema. La reducción de varianza es directamente proporcional a la fracción de la probabilidad total que queda incluída en el espacio W. A continuación damos una descripción más detallada de la rutina de muestreo $\mathcal{M}$ para el Muestreo basado en cotas:

Entradas: $\mathcal{G}$, $\mathcal{K}$, Φ^L y Φ^U
Salida: un valor de R

Rutina $\mathcal{M}()$ **:**
  Tomar una muestra $\tilde{X} = (\tilde{x}_1, \ldots, \tilde{x}_{|\mathcal{E}|})$
  Calcular $R = R_0^L + \Phi(\tilde{X})(R_0^U - R_0^L)$
  Devolver R
fin$\mathcal{M}()$

La muestra $\tilde{X} = (\tilde{x}_1, \ldots, \tilde{x}_{|\mathcal{E}|})$ se toma mediante el sorteo sucesivo, para $l = 1, \ldots, |\mathcal{E}|$, del estado $\tilde{x}_l$ de la arista l según la probabilidad de funcionamiento

$$\begin{aligned} \tilde{p}_l &= \mathcal{P}\left(x_l = 1 | x_1 = \tilde{x}_1, \ldots, x_{l-1} = \tilde{x}_{l-1} \text{ y } \Phi^U(\tilde{X}) = 1, \Phi^L(\tilde{X}) = 0\right) \\ &= \left[\frac{R_l^U(\tilde{X}^{(l-1)}, 1) - R_l^L(\tilde{X}^{(l-1)}, 1)}{R_{l-1}^U(\tilde{X}^{(l-1)}) - R_{l-1}^L(\tilde{X}^{(l-1)})}\right] p_l \end{aligned}$$

La varianza por muestra, correspondiente a los métodos de esta familia es:

$$\begin{aligned} \mathrm{Var}\,() &= R_{\mathcal{K}}(R_0^U - R_{\mathcal{K}}) - R_0^L(R_0^U - R_{\mathcal{K}}) \\ &= R_{\mathcal{K}}(1 - R_{\mathcal{K}}) - (1 - R_0^U)R_{\mathcal{K}} - R_0^L(R_0^U - R_{\mathcal{K}}) \end{aligned}$$

siendo entonces menor que aquella correspondiente al Monte Carlo crudo ($R_{\mathcal{K}}(1 - R_{\mathcal{K}})$); la diferencia depende de lo ajustadas que sean las cotas Φ^L y Φ^U. La performance en cuanto a su tiempo de ejecución depende de la complejidad computacional del cálculo de esas mismas funciones.

En [12] se propone una variante de esta familia de métodos para el caso particular en el que todas las líneas de la red poseen la misma confiabilidad elemental. El método se basa en calcular eficientemente una cota inferior a la confiabilidad de la red basada en la evaluación del subconjunto de los estados con número de aristas falladas menor que el cardinal del más pequeño corte mínimo de la red, y no utiliza ninguna cota superior. La reducción de varianza obtenida con este método es del orden de $1/\lfloor 1 - R_0^L \rfloor$, para un costo computacional por muestra similar al del método Monte Carlo crudo. El autor clasifica este método dentro de los esquemas basados en variables antitéticas.

4.2 Muestreo “dagger” y derivados

El muestreo dagger fue propuesto por Kumamoto et al. [22] y puede ser visto como una extensión de la técnica de variables antitéticas. La idea base es la de generar bloques de L muestras, dentro de los cuales las variables aleatorias son elegidas de manera de introducir correlación negativa entre las muestras individuales. El tamaño L de cada bloque es elegido de forma tal que para cada arista l la secuencia de L replicaciones puede ser partida en exactamente N_l sub-bloques de tamaño L/N_l, siendo $N_l = \lfloor 1/q_l \rfloor$. Para cada uno de esos subbloques se elige aleatoriamente (utilizando una variable uniforme) una única posición en la que la arista l va a fallar (esto genera un patron de fallos para las L replicaciones en el que la frecuencia de fallas de cada arista está exactamente en proporción a la tasa media de fallo de la misma). Luego se recorre todas las replicaciones del bloque, controlando en cuales de ellas el grafo es $\mathcal{K}$ conexo y en cuales no lo es, para de esta forma obtener una estimación del valor de $R_{\mathcal{K}}$.

En este método, una única invocación a $\mathcal{M}$ equivale a L muestras en Monte Carlo crudo, lo que debe ser tenido en cuenta para la comparación de las complejidades.

Entradas: grafo $\mathcal{G}$, conjunto $\mathcal{K}$
Salida: un valor de R

Inicializacion $\mathcal{I}$:
Calcular el vector de enteros $(N_l : l \in \mathcal{E})$: $N_l = \lfloor 1/q_l \rfloor$.
Elegir el numero de muestras: $L = mcm\{N_l : l \in \mathcal{E}\}$.

Rutina $\mathcal{M}$**:**
Para cada arista l
Para cada subbloque de tamano L/N_l
Elegir aleatoriamente una replicacion dentro del sub-bloque:
Sortear U uniforme, tomar $\tilde{k} = \lceil U/q_l \rceil$
La arista l falla en esa replicacion:
$x_l^k = 1$, $\forall k \neq \tilde{k}$,
$x_l^{\tilde{k}} = 0$, si $\tilde{k} \leq L/N_l$.
finPara
finPara
Inicializar $T = 0$.
Para i de 1 a L
Contar las replicaciones correspondientes a un estado operacional de la red:
$T = T + \Phi(X^i)$.
finPara
Devolver $R = T/L$.
fin$\mathcal{M}$

El análisis de complejidad de este algoritmo realizado en [14] indica que el tiempo de ejecución por cada muestra es $O(|\mathcal{E}|)$, como en el caso de Monte Carlo crudo. Sin embargo, dado que el número de variables aleatorias que es preciso generar durante la ejecución es mucho menor para el método “dagger”, existe una ganancia en tiempo de ejecución considerable, que resulta la principal ventaja de este método respecto al crudo.

La reducción de varianza obtenida por el algoritmo “dagger” está basada en la correlación negativa inducida entre muestras que pertenecen al mismo sub-bloque para alguna arista del grafo, y es relativamente pequeña, tendiendo a 0 cuando la confiabilidad elemental de las aristas se acerca a 1. En [11] El Khadiri y Rubino hacen notar que el método “dagger” presenta algunos problemas, generados por el hecho de que el valor L puede ser muy grande. Dado que es necesario generar y almacenar los valores de L estados del grafo, los requerimientos de memoria pueden hacer infactible la aplicacion del método (en ese trabajo se presentan ejemplos simples de grafos que no pudieron ser evaluados debido a este problema). Esos autores muestran que en realidad el tamaño L del bloque puede ser elegido de forma completamente arbitraria, proponiendo una generalización del método "dagger" bajo el nombre de algoritmo antitético genérico. Este nuevo método obtiene mejores resultados tanto por la posibilidad de ajustar el valor de L como por los mecanismos que incorpora para las etapas de muestreo. Estos mecanismos son muy similares a aquellos empleados en la simulación a eventos discretos, llevándose una lista con los próximos fallos correspondientes a cada arista, y buscando la primer replicación con un fallo. Esto permite no sólo ganar en tiempo de ejecución, sino también reducir la complejidad en memoria al mínimo, y hacerla independiente del tamaño de bloque L.

Para el caso en el que las confiabilidades de todas las aristas son iguales a p, el algoritmo antitético genérico tiene una complejidad por muestra de $O(\alpha|\mathcal{E}|)$, donde

$$\alpha = max(1/L, 1-p).$$

Esto marca una mejora frente al método crudo que es creciente con el tamaño L elegido para los bloques, pero acotada por el inverso de las confiabilidades elementales (en el peor caso, $p = 0.5$, tenemos que la ganancia máxima es de un factor de 2, correspondiendo a una elección de $L = 2$).

4.3 Construcción/Destrucción Secuenciales

Los métodos de Construcción/Destrucción Secuenciales [24] se basan en la consideración de una ordenamiento de las aristas del grafo. Todas las aristas se consideran en un primer momento como falladas, y luego son "reparadas" en forma sucesiva, una a una según el orden elegido, hasta que el sistema pasa a su estado operativo. El estimador de la confiabilidad se puede interpretar como una función de cuanto lleva llegar al estado operativo del sistema.

Estos métodos pueden clasificarse como un híbrido del muestreo estratificado y muestreo según importancia.

El espacio de muestreo para el método de construcción secuencial consiste en parejas $(\tilde{X}, \tilde{\pi})$ donde $\tilde{X}$ es un vector de estado del sistema y $\tilde{\pi} = (\tilde{\pi}_1, \ldots, \tilde{\pi_{|\mathcal{E}|}})$ es una permutación de los índices de aristas de $\mathcal{E}$, y existe un índice k que verifica

$$\tilde{x}_{\tilde{\pi}_1} = \ldots = \tilde{x}_{\tilde{\pi}_k} = 1, \quad \tilde{x}_{\tilde{\pi}_{k+1}} = \ldots = \tilde{x}_{\tilde{\pi}_{|\mathcal{E}|}} = 0.$$

Si el vector $\tilde{X}$ es elegido de acuerdo a las probabilidades de estado del sistema y la permutación $\tilde{\pi}$ es elegida de forma independiente y uniforme sobre el conjunto de permutaciones compatibles, entonces la probabilidad de ocurrencia de una pareja dada $(\tilde{X}, \tilde{\pi})$ es

$$\rho(\tilde{X}, \tilde{\pi}) = \frac{P(X = \tilde{X})}{k!(|\mathcal{E}| - k)!} = \frac{1}{|\mathcal{E}|!} C_k^{|\mathcal{E}|} P(X = \tilde{X})$$

donde k es el número de aristas que funcionan en $\tilde{X}$. El método de construcción secuencial muestrea una permutación $\tilde{\pi}$, y considera en forma simultánea el conjunto $\mathcal{P}_{\tilde{\pi}}$ de todos los posibles pares $(\tilde{X}, \tilde{\pi})$, tales que $\tilde{X}$ es consistente con $\tilde{\pi}$ según el criterio anterior. El valor de confiabilidad R muestreado es entonces la probabilidad condicional de operación del sistema respecto a $\mathcal{P}_{\tilde{\pi}}$, es decir el cociente de la suma de probabilidades de los pares $(\tilde{X}, \tilde{\pi}) \in \mathcal{P}_{\tilde{\pi}}$ tales que $\Phi(\tilde{X}) = 1$ dividido por la probabilidad de $\mathcal{P}_{\tilde{\pi}}$. A continuación damos una descripción más detallada de la rutina de muestreo $\mathcal{M}$ para el Procedimiento de Construcción Secuencial, (en este caso no existe una inicialización particular):

Entradas: grafo $\mathcal{G}$, conjunto $\mathcal{K}$
Salida: un valor de R

Rutina $\mathcal{M}()$ **:**
Sortear $\tilde{\pi} = (\tilde{\pi}_1, \ldots, \tilde{\pi}_{|\mathcal{E}|})$
Para $k = 0, \ldots, |\mathcal{E}|$ (Definir $\tilde{X}^{(k)}$)
$\tilde{x}^{(k)}_{\tilde{\pi}_1} = \ldots = \tilde{x}^{(k)}_{\tilde{\pi}_k} = 1,\ \tilde{x}^{(k)}_{\tilde{\pi}_{k+1}} = \ldots = \tilde{x}^{(k)}_{\tilde{\pi}_{|\mathcal{E}|}} = 0$
finPara
Determinar primer $r \in 0, \ldots, |\mathcal{E}|$
tal que $\Phi(\tilde{X}^{(r)}) = 1$
Calcular
$R = \frac{\sum_{k=0}^{|\mathcal{E}|} \Phi(\tilde{X}^{(k)})\rho(\tilde{X}^{(k)},\tilde{\pi})}{\sum_{k=0}^{|\mathcal{E}|} \rho(\tilde{X}^{(k)},\tilde{\pi})} = \frac{\sum_{k=r}^{|\mathcal{E}|} C_k^{|\mathcal{E}|} P(X=\tilde{X}^{(k)})}{\sum_{k=0}^{|\mathcal{E}|} C_k^{|\mathcal{E}|} P(X=\tilde{X}^{(k)})}$
Devolcer R.
finM()

Es posible mostrar que el estimador R obtenido en cada muestra con este método tiene varianza más pequeña que aquella correspondiente a una muestra del algoritmo crudo. El mayor esfuerzo computacional se realiza al determinar el primer índice r, que depende fundamentalmente de la facilidad de cálculo de $\Phi(\tilde{X}^{(k)})$, es decir de cómo determinar cuando el sistema llega al estado operativo a medida que se reparan las líneas una a una. En el peor caso, y usando un DFS para el cálculo de Φ, la complejidad sería de $O(|\mathcal{E}| \max(|\mathcal{V}|, |\mathcal{E}|))$. Sin embargo, en [14] Fishman muestra que para el caso $\mathcal{K} = \{s, t\}$ es posible determinar el valor de r con un costo similar al de un único cálculo de Φ, por lo que cada muestra de este método tiene un costo parecido al de una muestra en el método crudo.

Es posible construir en forma análoga un método de destrucción secuencial, cuya aplicación puede ser ventajosa en sistemas de muy baja confiabilidad, dado que en ese caso el valor de r se determinará al cabo de menos iteraciones que si se aplicara el método de construcción.

Otros métodos basados en modelos de evolución de grafos han sido publicados en [13]. Se trata de los métodos llamados de procesos de *destrucción y creación, y de creación y unión*, que refinan las ideas anteriores identificando aristas irrelevantes y particiones regulares del grafo subyacente. En este caso, la complejidad por muestra es de $O(|\mathcal{E}|^2)$, pero la varianza es menor que aquella correspondiente a la construcción secuencial.

4.4 Método de cubrimiento

El método de cubrimiento ha sido propuesto por Karp y Luby [19]. Es una variante híbrida de los métodos de muestreo según importancia y muestreo estratificado, y utiliza la lista de los cortes mínimos del sistema para mejorar el muestreo crudo. La idea de base es embeber el conjunto F de eventos de fallo del sistema en un espacio universal ponderado $(\mathcal{U}, w)$, donde w es una función (no negativa) de ponderación en $\mathcal{U}$, y que satisface los siguientes criterios:

- $w(F) = \mathcal{P}(F) = Q_{\mathcal{K}}$.
- $w(\mathcal{U})$ se puede calcular de forma eficiente (en tiempo polinomial). Además, es posible tomar muestras de $\mathcal{U}$ con probabilidad proporcional a sus pesos en forma eficiente.
- Se puede decidir en forma eficiente si un elemento de $\mathcal{U}$ pertenece a F.
- $w(\mathcal{U})/w(F)$ está acotado superiormente por un valor M para todas las instancias en la clase de problemas considerados.
- $w(C)$ es el peso (ponderación) total de los elementos de $\mathcal{U}$ con segunda componente igual a C.

Si se toma una muestra de $\mathcal{U}$, y si se obtiene el estimador $\hat{Q}$ multiplicando la proporción de dicha muestra contenida en F por $w(\mathcal{U})$, entonces $\hat{Q}$ es un estimador sin sesgo de $Q_{\mathcal{K}}$.

Sea $\mathcal{C}$ el conjunto de los $\mathcal{K}$-cortes mínimos del grafo $\mathcal{G}$. Se define el espacio universal ponderado $\mathcal{U}$ como compuesto por los pares (X, C) donde X es un vector de estado, $C \in \mathcal{C}$ es un corte, y $x_l = 0$ para todas las aristas l pertenecientes a C. De esta manera cada estado X de fallo del sistema aparecerá en los elementos de $\mathcal{U}$ tantas veces como el número de cortes mínimos fallados en X; para embeber F en $\mathcal{U}$ es necesario asignar a cada X un único corte $C \in \mathcal{C}$. Para esto se elige un nodo $s \in K$, se busca el conjunto N de nodos alcanzables desde s a través de caminos formados por aristas en funcionamiento, y se elige como $C \equiv C(X)$ al conjunto de aristas que van de N a $\mathcal{V} - N$. Los elementos de F aparecen entonces en $\mathcal{U}$ como parejas (X, C) tales que $C = C(X)$, y se puede decidir si un elemento de $\mathcal{U}$ pertenece a F en tiempo lineal verificando la condición $C = C(X)$. La función de ponderación w está dada por $w(\tilde{X}, C) = Pr(X = \tilde{X})$.

A continuación se da la descripción en seudocódigo de las rutinas de inicialización y muestreo para el Método de cubrimiento:

Entradas: $\mathcal{G}$, $\mathcal{K}$, lista de cortes minimos $\mathcal{C}$
Salida: un valor de R

Inicialización $\mathcal{I}$**:**
  Para cada $C \in \mathcal{C}$
    Calcular $w(C) = \prod_{l \in C} q_l$
  finPara
  Calcular $w(\mathcal{U}) = \sum_{C \in \mathcal{C}} w(C)$
fin$\mathcal{I}$
Rutina $\mathcal{M}()$**:**
    Sortear un $(\tilde{X}, C)$ de $\mathcal{U}$ de acuerdo a w:
    Sortear un corte C con probabilidad $w(C)/w(\mathcal{U})$
    Construir $\tilde{X}$:
      $\forall l \in C \; \tilde{x}_l = 0$
      $\forall l \notin C \; \tilde{x}_l = 1$ con probabilidad p_l,
        $\tilde{x}_l = 0$ sino.
    Si $C = C(\tilde{X})$
      $R = 1 - w(\mathcal{U})$
    Sino
      $R = 1$
    finSi
    Devolver el valor R.
fin$\mathcal{M}()$

Este método puede obtener buenos resultados a nivel de la reducción de la varianza, pero con el gran inconveniente de exigir el cálculo y almacenamiento previo de la lista de todos los cortes mínimos del grafo considerado. Dado que el tamaño de esta lista crece en forma exponencial con el tamaño del grafo, tenemos que muy rápidamente los requerimientos de espacio pueden hacer imposible su aplicación (por ejemplo, para el grafo de 20 nodos y 30 arcos presentado en [14], el espacio requerido en memoria RAM es de 2 Megabytes para almacenar la lista citada). Por lo tanto, la aplicabilidad de este método se encuentra seriamente limitada (es interesante notar que existen métodos que calculan la confiabilidad en forma exacta en tiempo polinomial en el número de cortes mínimos del grafo [26]).

4.5 Método del riesgo total

Las variables de riesgo aleatorio, y en particular la variable de riesgo total, han sido empleadas en diversidad de contextos para el estudio por simulación de modelos aleatorios. En [18] y [27] se propone su utilización para la estimación de la confiabilidad $R_{\mathcal{K}}$.

Sea C_1 un $\mathcal{K}$-corte mínimo (un conjunto mínimo de aristas que al ser sacadas desconectan los nodos de $\mathcal{K}$ en el grafo). El primer riesgo, h_1, es la probabilidad de que todos los componentes en C_1 no funcionen (lo que implica que el grafo no es $\mathcal{K}$-conexo), es decir

$$h_1 = \prod_{i \in C_1} q_i.$$

El método del riesgo total consiste en simular el estado de todas las aristas de C_1. Si todas las aristas están fuera de servicio, el procedimiento finaliza. Si por lo menos una arista funciona, se fijan los estados de las aristas simuladas, y se busca un nuevo corte mínimo C_2 en el grafo modificado. A partir del nuevo corte se calcula el segundo riesgo, h_2, dado por

$$h_2 = \prod_{i \in C_2} q_i.$$

A continuación se simula el estado de las aristas de C_2 y se repite el proceso anterior, generando nuevos grafos hasta que todos los componentes de un corte mínimo esten fallados o hasta llegar a un grafo trivial, sin cortes mínimos (todas las aristas han sido fijadas). Si se consideraron r riesgos, el riesgo total está dado por

$$H(\mathcal{G}) = \sum_{i=1}^{r} h_i,$$

y es un estimador sin sesgo de $Q_{\mathcal{K}}$. La implementación sugerida en [18] utiliza la lista de todos los cortes mínimos del sistema considerado, y la actualiza a medida que se fijan los estados de las aristas simuladas. El corte se elige en cada paso para que resulte en el máximo riesgo.

Entradas: $\mathcal{G}$, conjunto $\mathcal{K}$, lista de $\mathcal{K}$-cortes de $\mathcal{G}$
Salida: una muestra de R

Inicialización $\mathcal{I}$:
$H = 0$.

Rutina $\mathcal{M}$:
  Elegir un $\mathcal{K}$-corte minimo C.
  Simular el estado de las aristas de C
  Repetir hasta que todas las aristas del corte elegido estan falladas
    Actualizar la lista de cortes del grafo $\mathcal{G}$.
    Calcular el riesgo: $h = \prod_{i \in C} q_i$.
    Acumular en H: $H = H + h$.
    Elegir un $\mathcal{K}$-corte minimo C.
    Simular el estado de las aristas de C.
  finRepetir
  Devolver $R = 1 - H$.
fin$\mathcal{M}$

La reducción de varianza que es posible obtener con este método depende fuertemente de la forma de elegir el $\mathcal{K}$-corte mínimo C. La heurística que parece más adecuada es la de elegir a cada paso el corte de mayor riesgo, pero esto conlleva un costo computacional que puede ser bastante elevado, ya que debe realizarse una búsqueda en la lista de cortes (que es de tamaño exponencial en el tamaño del grafo) o mediante un algoritmo de flujo máximo en cada iteración del flujo interno. El análisis de la complejidad por muestra depende fuertemente de ese mismo paso, así como del paso 2.5. De todas formas, la complejidad será en el mejor de los casos de $O(|\mathcal{E}|)$ por muestra, dado que es posible tener que simular el estado de todas las aristas del grafo.

4.6 Método de reducción recursiva de varianza

El método de reducción recursiva de varianza presentado en [6] utiliza una partición del espacio de estado y ciertas técnicas desarrolladas en los métodos exactos (condicionamiento respecto a una partición del espacio de estado, contración y eliminación de aristas, etc. [1], [5], [29]) para obtener una muestra del estimador de $Q_{\mathcal{K}}$.

La siguiente ecuación da una formulación recursiva de la anti-confiabilidad de un grafo $\mathcal{G}$ a partir de la anti-confiabilidad de grafos más pequeños, basándose en un particionamiento del espacio de probabilidad subyacente:

$$Q_{\mathcal{K}}(\mathcal{G}) = Q_{C(\mathcal{G},\mathcal{K})} + \sum_{i=1}^{|C(\mathcal{G},\mathcal{K})|} \mathcal{P}(B_i)\, Q_{\mathcal{K}_i}(\mathcal{G}_i)$$

donde

- $C(\mathcal{G},\mathcal{K}) = \{l_1, l_2, \ldots, l_{|C(\mathcal{G},\mathcal{K})|}\}$ es un $\mathcal{K}$-corte de $\mathcal{G}$;
- $Q_{C(\mathcal{G},\mathcal{K})}$ es la probabilidad que todos los componentes de $C(\mathcal{G},\mathcal{K})$ fallen;
- B_i es el evento que todos los componentes en el conjunto $\{l_1, l_2, \ldots, l_{i-1}\}$ fallan y el componente l_i funciona;
- $G_i = (G - l_1 - \ldots - l_{i-1}) * l_i$ y $\mathcal{K}_i$ es su conjunto de terminales.

En base a la ecuación precedente se plantea el operador recursivo $F(,)$ que proporciona un estimador sin sesgo de la anti-confiabilidad:

$$F(\mathcal{G},\mathcal{K}) = Q_{C(\mathcal{G},\mathcal{K})} + (1 - Q_{C(\mathcal{G},\mathcal{K})}) \sum_{i=1}^{|C(\mathcal{G},\mathcal{K})|} \mathbf{1}_{(\mathcal{U}(\mathcal{G},\mathcal{K})\in J_i)} F(\mathcal{G}_i, \mathcal{K}_i)$$

con

- $\mathcal{U}(\mathcal{G},\mathcal{K})$ una variable aleatoria independiente de distribución uniforme $[0,1]$;
- $(J_i)_{1\le i\le |C(\mathcal{G},\mathcal{K})|}$ es una secuencia de intervalos disjuntos cuya unión es $[0,1]$, donde el largo de cada intervalo J_i es $\mathcal{P}(B_i)/(1 - Q_{C(\mathcal{G},\mathcal{K})})$.

Como sólo una de las indicatrices tendrá valor 1, esto equivale a en cada paso ir acumulando las probabilidades de fallo de un corte y continuar el trabajo en un nuevo grafo $\mathcal{G}_i$ donde las aristas de $\mathcal{G}$ en funcionamiento son contraídas, y las falladas son eliminadas.

El procedimiento $\mathcal{M}()$ corresponde a este método, y debe ser utilizado desde el programa principal con los parámetros $\mathcal{G}$ y $\mathcal{K}$, devolviendo una muestra seudo-aleatoria. En el primer paso del procedimiento se controlan las condiciones de finalización. Si $\mathcal{K}$ es un conjunto unitario, entonces el grafo es trivialmente $\mathcal{K}$-conexo, por lo que el valor de la confiabilidad es 1 (y el de la anti-confiabilidad 0). En el otro extremo, si la topología de base de $\mathcal{G}$ no es $\mathcal{K}$-conexa, tenemos anti-confiabilidad 1. En el segundo paso se elige el corte C compuesto por todas las líneas adyacentes a un nodo arbitrario de $\mathcal{K}$, y a continuación se calcula la probabilidad de que todos los componentes de C hayan fallado. En el cuarto paso se utiliza una variable aleatoria uniforme para determinar uno de los intervalos J_i de la partición dada en la fórmula anterior.

A continuación se actualiza el grafo y el conjunto de terminales de acuerdo al intervalo seleccionado, y por último se realiza la llamada recursiva, utilizándose el valor devuelto en el cálculo de la muestra.

Entrada: grafo $\mathcal{G}$, conjunto $\mathcal{K}$
Salida: una muestra de la variable aleatoria $F(\mathcal{G}, \mathcal{K})$

Rutina $\mathcal{M}(\mathcal{G}, \mathcal{K})$
Controlar condición de parada de la recursión:
Si $|\mathcal{K}| = 1$ devolver(0) ($\mathcal{G}$ es siempre $\mathcal{K}$-conexo).
Si $\mathcal{G}$ no es $\mathcal{K}$-conexo devolver(1).
Buscar un $\mathcal{K}$-corte
$C = \{l_1, \ldots, l_{|C|}\}$
Calcular $Q_C = \prod_{i=1}^{|C|} q_{l_i}$.
Muestrear $\mathcal{U}$ $Uniforme(0,1)$ y elegir J_i tal que $\mathcal{U} \in J_i$
Construir $\mathcal{G}_i = (\mathcal{G} - l_1 - l_2 - \ldots - l_{i-1}) * l_i$.
Determinar $\mathcal{K}_i$ correspondiente a $\mathcal{G}_i$.
Devolver $(Q_C + (1 - Q_C) \times \mathcal{M}(\mathcal{G}_i, \mathcal{K}_i))$.
fin$\mathcal{M}$

Un cálculo sencillo permite evaluar la complejidad total de la obtención de una muestra con este algoritmo, que es de $O(|\mathcal{E}|^2)$, frente a $O(|\mathcal{E}|)$ en el método crudo. El análisis de la reducción de varianza no está resuelto completamente, dado que depende de la elección del corte en cada paso. Una cota inferior de la reducción esperable en un único paso recursivo (que no tiene en cuenta el efecto acumulado) está dada por $Q_C(1 - R_{\mathcal{K}})$, lo que muestra el interés en elegir en el paso 3 un corte con una probabilidad de falla elevada.

5 Comparación Numérica

Esta sección presenta una comparación numérica de distintos métodos presentados, basada en resultados publicados en la literatura y complementada por los autores del presente trabajo para algunos métodos de los que se dispone del código.

Para comparar dos esquemas de simulación es importante tener en cuenta tanto su precisión (varianza o intervalo de confianza) y el esfuerzo computacional requerido para obtener dicha precisión. Fijado un tamaño de muestra, notaremos T_M y V_M respectivamente el tiempo de ejecución y la varianza de un método Monte Carlo. Para tener en cuenta los aspectos nombrados anteriormente utilizaremos la eficiencia relativa (a veces llamada aceleración) del método M respecto a Monte Carlo crudo (MCC). Este parámetro denotado como W_{MCC}/W_M [17] es el el producto de dos términos, el cociente de varianzas V_{MCC}/V_M y el cociente de tiempos T_{MCC}/T_M, para un tamaño fijo de muestra.

Los resultados están organizados en dos grupos. Primero utilizamos la red de 20 nodos y 30 aristas que se ve en la Figura 1 con conjunto de terminales $\mathcal{K} = \{s, t\}$ (los nodos s y t son aquellos etiquetados en dicha figura), tomando todas las aristas la misma confiabilidad elemental $q_l = 0.50$, 0.10, 0.05 y 0.02. Esta topología fue propuesta en el estudio [14] para comparar el comportamiento de varios métodos de reducción de varianza. En la Tabla 1 aparecen los valores estimados de la anti-fiabilidad Q_{st}.

q_l	.50	.10	.05	.02
$\widehat{Q}_{st}$	0.7099	0.002880	0.0002946	0.00001702

Tabla 1: Resultados de simulación para R_{st} del dodecaedro (estimaciones obtenidas por el algoritmo de reducción recursiva de varianza con tamaño de muestra 10^6)

Valores del cociente V_{MCC}/V_M							
	Reducción Recursiva de Varianza (1)	Riesgo Total (1)	Muestreo Dagger (2)	Construcción Secuencial (2)	Cotas (2)	Cubrimiento (2)	Proceso de Creación y Unión (3)
$q_l = .50$	18.9	0.74	1.23	2.70	2.35	0.21	2.8
$q_l = .10$	302	44.1	1.01	5.25	50.22	6981.31	37.1
$q_l = .05$	2510	324	1.00	11.43	594.24	437230.32	246
$q_l = .02$	81000	2510	—	—	—	—	3472
Valores del cociente T_{MCC}/T_M							
	Reducción Recursiva de Varianza (1)	Riesgo Total (1)	Muestreo Dagger (2)	Construcción Secuencial (2)	Cotas (2)	Cubrimiento∗ (2)	Proceso de Creación y Unión (3)
$q_l = .50$	0.35	0.16	1.27	0.25	0.24	0.22	0.24
$q_l = .10$	0.28	0.15	1.79	0.27	0.24	0.01	0.24
$q_l = .05$	0.27	0.15	1.91	0.24	0.23	0.01	0.23
$q_l = .02$	0.27	0.15	—	—	—	—	0.14
Valores del cociente W_{MCC}/W_M							
	Reducción Recursiva de Varianza (1)	Riesgo Total (1)	Muestreo Dagger (2)	Construcción Secuencial (2)	Cotas (2)	Cubrimiento∗ (2)	Proceso de Creación y Unión (3)
$q_l = .50$	6.56	0.115	1.56	0.68	0.56	0.05	0.67
$q_l = .10$	85.5	6.63	1.81	1.40	12.3	70.3	8.8
$q_l = .05$	681	50.10	1.91	2.71	136	3714	55.7
$q_l = .02$	22000	386.0	—	—	—	—	495

(1) Fuente [6]. (2) Fuente [14]. (3) Fuente [13].
— El caso $q_l = .02$ no fue considerado en la fuente [14].
∗ Estas aceleraciones no tienen en cuenta el costo de la generación previa de la lista de cortes.

Tabla 2: Resultados de simulación para los seis métodos considerados, en función de la probabilidad de falla de aristas, para la red en la Figura 1.

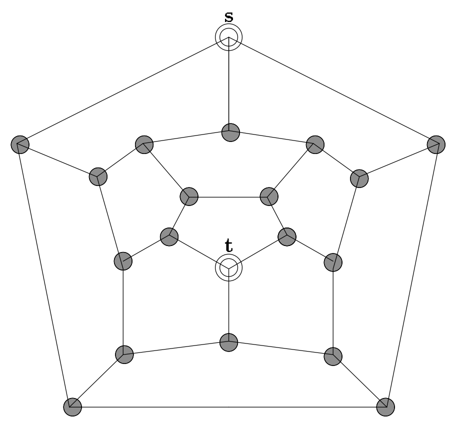

Figura 1: Topología de la red Dodecaedro

La Tabla 2 presenta la aceleración, así como los cocientes de varianzas y de tiempos de cálculo relativos al método Monte Carlo directo, correspondientes a siete métodos de la literatura (muestreo dagger, construcción secuencial, cotas, cubrimiento, proceso de creación/unión, reducción recursiva de varianza, riesgo total). Los resultados correspondientes a los primeros cuatro métodos provienen de [14], y aquellos del proceso de creación/unión de [13]. Los restantes algoritmos fueron programados y ejecutados en una estación Sun SPARCstation 1. Para cada valor de q_l, los resultados de simulación se basaron en $N = 10^6$ muestras. Estos resultados, aún proveniendo de distintas fuentes, son aproximadamente comparables dado que están expresados como cocientes independientes del número de muestras, y tomando como patrón el método Monte Carlo crudo.

Para valores de la probabilidad elemental de falla q_l altos (cercanos a 0.5) los dos únicos métodos que poseen una performance superior al Monte Carlo crudo son el muestreo dagger y el método de reducción recursiva de varianza. Esta situación varía al disminuir q_l, teniendo el método de cubrimiento los mejores resultados, seguido por los métodos de reducción recursiva de varianza, cotas, y creación y unión respectivamente. Sin embargo, como se discute en la referencia [14], el método de cubrimiento necesita el cálculo previo de la lista de todos los $\mathcal{K}$-cortes de la red. Dado que el tamaño de esta lista crece en forma exponencial con el número de aristas de la red, tanto el tiempo de cálculo previo a la simulación como los requerimientos de almacenamiento son limitaciones prácticas muy fuertes para el uso de este método. Según informa [14], el almacenamiento compacto de la lista de cortes para el grafo de la Figura 1 ocupa más de 2 Megabytes. Estas limitaciones no se aplican a los otros tres métodos nombrados, por lo que serían los de mayor aplicabilidad en la práctica.

El segundo grupo de resultados corresponde a grafos en los cuales el conjunto de terminales $\mathcal{K}$ coincide con el conjunto de nodos $\mathcal{V}$, es decir que todos los nodos deben poder comunicar entre si. La literatura posee menos resultados numéricos para este caso. Hemos tomado la familia de grafos completos K_n, con $n \in \{10, 15, 20, 25, 30\}$, y fijando el valor de la confiabilidad elemental $p_l = 0.45$, para todas las aristas l. La Tabla 3 muestra los valores de $Q_{\mathcal{K}}$ estimados por el método de reducción recursiva de varianza, con tamaño de muestra 10^6.

En la Tabla 4 aparecen la aceleración, así como los cocientes de varianzas y de tiempos de cálculo relativos al método Monte Carlo directo, correspondientes a los métodos proceso de creación/unión, reducción recursiva de varianza y riesgo total. Los resultados correspondientes al proceso de creación/unión son de [13], siendo los restantes algoritmos programados y ejecutados en una estación Sun SPARCstation 1. En todos los casos, el tamaño de muestra se tomó igual a 10^4. Estos resultados indican una muy buena performance del método basado en procesos de creación/unión, solamente superada por la reducción recursiva de

Red	K_{10}	K_{15}	K_{20}	K_{25}	K_{30}
$\widehat{Q}_V$	4.58e-2	3.51e-3	2.47e-4	1.47e-5	6.09e-7

Tabla 3: Resultados de simulación de $R_{\mathcal{V}}$ para grafos completos (estimaciones obtenidas por el algoritmo de reducción recursiva de varianza con tamaño de muestra 10^6).

Valores del cociente V_{MCC}/V_M			
	Reducción Recursiva de Varianza	Riesgo Total	Proceso de Creación y Unión (3)
K_{10}	2.42	4.39	70
K_{15}	1.84	6.06	887
K_{20}	1.53	136	15160
K_{25}	4.38e14	3040	2.7e5
K_{30}	2.21e19	30900	5.08e7
Valores del cociente T_{MCC}/T_M			
	Reducción Recursiva de Varianza	Riesgo Total	Proceso de Creación y Unión (3)
K_{10}	.739	.617	.3
K_{15}	.662	.526	.191
K_{20}	.541	.439	.216
K_{25}	.543	.447	.175
K_{30}	.480	.405	.144
Valores del cociente W_{MCC}/W_M			
	Reducción Recursiva de Varianza	Riesgo Total	Proceso de Creación y Unión (3)
K_{10}	1.79	2.71	21
K_{15}	1.22	3.19	169
K_{20}	.828	59.7	3280
K_{25}	2.37e14	1359	47220
K_{30}	1.06e19	12500	7.3e6
(3) Fuente [13].			

Tabla 4: Resultados de simulación para cinco grafos completos, considerando probabilidad de falla elemental de las aristas $q = 0.55$.

varianza para los grafos de mayor número de nodos. Sin embargo, es preciso hacer notar que la estimación de las varianzas con un tamaño de muestra reducido puede dar lugar a resultados poco confiables. Sería por lo tanto deseable contar con valores numéricos para muestras más grandes, como es el caso para el ejemplo anterior.

6 Conclusiones

En este informe hemos presentado un estado del arte de los métodos de reducción de varianza para el cálculo Monte Carlo de los parámetros clásicos de confiabilidad de una red de comunicaciones con nodos perfectos. Se ha discutido las distintas familias de métodos, presentando en cada caso el seudocódigo correspondiente, así como un análisis de complejidad computacional. También se realizó una comparación numérica de resultados basada en algunos problemas de la literatura. Esta comparación es parcial debido a que no existe una serie de problemas de prueba utilizada por todos los autores. Una perspectiva interesante es entonces el desarrollo de una biblioteca con esas características, que haga más sencillo la evaluación ("benchmarking") de nuevas técnicas.

Referencias

[1] J.A. Abraham. An improved algorithm for network reliability. *IEEE Trans. on Reliab.*, R-28(1):58–61, April 1979.

[2] M.O. Ball. The complexity of network reliability computations. *Networks*, 10:153–165, 1980.

[3] M.O. Ball. Computational complexity of network reliability analysis: An overview. *IEEE Trans. Reliab.*, R-35(3), 1986.

[4] M.O. Ball, C.J. Colbourn, and J.S. Provan. Network reliability. *Research Report*, June 1992.

[5] B.E. Barlow and F. Proschan. *Statistical Theory of Reliability and Life Testing.* Holt, Rinehart & Winston, New York, 1975.

[6] H. Cancela and M. El Khadiri. A recursive variance reduction method for estimating communication network reliability. Research Report 860, IRISA, Rennes, France, Campus de Beaulieu 35042 France, September 1994.

[7] C.J. Colbourn. *The Combinatorics of Network Reliability.* Oxford University Press, New York, 1987.

[8] M. El Khadiri. *Evaluation directe et par simulation d'indices de fiabilité de réseaux de communication : algorithmes séquentiels et dédiés à des architectures à mémoire distribuée.* Phd. thesis, Rennes I, Campus de Beaulieu, 35042 Rennes, France, December 1992.

[9] M. El Khadiri, R. Marie, and G. Rubino. Estimation of network reliability on a parallel machine by means of a Monte Carlo technique. Technical Report 545, IRISA, Campus de Beaulieu, 35042 Rennes, France, 1990.

[10] M. El Khadiri, R. Marie, and G. Rubino. Parallel estimation of 2-terminal network reliability by a crude Monte Carlo technique. In Mehmet Baray and Bülent Özgüç, editors, *ISCIS VI, Sixth International Symposium on Computer and Information Sciences*, pages 559–570, Elsevier, Antalya, Turquie, October 1991.

[11] M. El Khadiri and G. Rubino. A Monte-Carlo method based on antithetic variates for network reliability computations. Technical Report 626, IRISA, Campus de Beaulieu, 35042 Rennes, France, 1992.

[12] M. El Khadiri and G. Rubino. Reliability evaluation of communication networks. In *SAFECOMP'92, International Conference on Safety, Security and Reliability of Computers*, Zurich, Suisse, October 1992.

[13] T. Elperin, I. Gertsbakh, and M. Lomonosov. Estimation of network reliability using graph evolution models. *IEEE Trans. Reliab.*, 40(5), 1991.

[14] G.S. Fishman. A comparison of four Monte-Carlo methods for estimating the probability of s-t conectedness. *IEEE Trans. Reliab.*, R-35(2), 1986.

[15] G.S. Fishman. A Monte Carlo sampling plan for estimating network reliability. *Operational Research*, 34, 1986.

[16] M.R. Garey and D.S. Johnson. *Computers and Intractability, a guide to the theory of NP-completeness.* W.H.Freeman and Company, 1979.

[17] J.M. Hammersley and D.C. Handscomb. *Monte Carlo Methods.* Halsted Press, Wiley & Sons. Inc., New York, 1979.

[18] C.H. Jun and S.M. Ross. System reliability by simulation: Random hazards versus importance sampling. *Probability in the Engineering and Informational Sciences*, 6, 1992.

[19] R. Karp and M.G. Luby. A new Monte Carlo method for estimating the failure probability of an n-component system. *Computer Science Division*, University of California(Berkeley), 1983.

[20] R.M. Karp and M. Luby. Monte-Carlo algorithms for the planar multiterminal network reliability problem. *Journal of Complexity*, 1:45–64, 1985.

[21] H. Kumamoto, K. Tanaka, and K. Inoue. Efficient evaluation of system reliability by Monte Carlo method. *IEEE Trans. on Reliab.*, R-26(5), Dec. 1977.

[22] H. Kumamoto, K. Tanaka, K. Inoue, and E.J. Henley. Dagger-sampling Monte Carlo for system unavailability evaluation. *IEEE Trans. Reliab.*, R-29(2), June 1980.

[23] M.O. Locks and A. Satyarayana editors. Network reliability – the state of the art. *IEEE Trans. Reliab.*, R-35(3), 1986.

[24] C.K. Wong M.C. Easton. Sequential destruction method for Monte Carlo evaluation of system reliability. *IEEE Trans. Reliab.*, R-29(1), April 1980.

[25] J.S. Provan. The complexity of reliability computations in planar and acyclic graphs. *SIAM J. Comput.*, 15(3), Aug. 1986.

[26] J.S. Provan and M.O. Ball. Computing network reliability in time polynomial in the number of cuts. *Operational Research*, 32, 1984.

[27] S.M. Ross. Variance reduction in simulation via random hazard. *Probability in the Engineering and Informational Sciences*, 4, 1990.

[28] G. Rubino. Tutorial: Efficient evaluation of network reliability. In *7th. Conference on Modeling Tools and Techniques*, Vienna, Austria, 1994.

[29] A. Satyarayana and M.K. Chang. Network reliability and the factoring theorem. *Networks*, 13, 1983.

[30] R.E. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, June 1972.

[31] R.M. Van Slyke and H. Frank. Network reliability analysis: part i. *Networks*, 1, 1972.