



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Confiabilidad Diámetro Acotada para el Modelo Hostil

Construcción de Estrategias de Selección de Cortes Minimales para el método de la
Reducción Recursiva de la Varianza

Ing. Leticia Grassi Mari

Programa de Posgrado en Ingeniería Eléctrica
Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
Noviembre de 2021



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Confiabilidad Diámetro Acotada para el Modelo Hostil

Construcción de Estrategias de Selección de Cortes Minimales para el método de la
Reducción Recursiva de la Varianza

Ing. Leticia Grassi Mari

Tesis de Maestría presentada al Programa de
Posgrado en Ingeniería Eléctrica, Facultad de
Ingeniería de la Universidad de la República, como
parte de los requisitos necesarios para la obtención
del título de Magíster en Ingeniería Eléctrica.

Directores:

Dr. Ing. Prof. Franco Robledo

Dr. Ing. Prof. Pablo Romero

Director académico:

Mag. Ing. Prof. Agregado Gabriel Gómez Sena

Montevideo – Uruguay

Noviembre de 2021

Grassi Mari, Ing. Leticia

Confiabilidad Diámetro Acotada para el Modelo Hostil
/ Ing. Leticia Grassi Mari. - Montevideo: Universidad de
la República, Facultad de Ingeniería, 2021.

XII, 152 p. 29, 7cm.

Directores:

Franco Robledo

Pablo Romero

Director académico:

Gabriel Gómez Sena

Tesis de Maestría – Universidad de la República,
Programa en Ingeniería Eléctrica, 2021.

Referencias bibliográficas: p. 111 – 114.

1. Sistemas Binarios Estocásticos Monótonos,
2. Confiabilidad Diámetro Acotado, 3. Modelo Hostil,
4. Método Reducción Recursiva de la Varianza, 5. Método Monte Carlo. I. Robledo, Franco, Romero, Pablo, .
- II. Universidad de la República, Programa de Posgrado en Ingeniería Eléctrica. III. Título.

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE TESIS

Dr.Ing Gonzalo Mateos

Dr.Ing Matthieu Jonckheere

Dr.Ing Federico La Rocca

Montevideo – Uruguay
Noviembre de 2021

A todas las niñas cuyas voces
fueron calladas, y alas cortadas.

Agradecimientos

Quisiera agradecer a los que acompañaron el camino, y a los que se quedaron a medias. A los que confiaron en dar ánimo en vez de críticas. Agradezco que aprendí a confiar en mi proceso, mis tiempos, y no en las expectativas que otros generan en uno.

Agradezco a mis tutores Pablo y Franco que guiaron este trabajo con constante empeño, aportes y dedicación. Quiero agradecer a mi padre por confiar en mí y en mis decisiones, por ser luz en el cielo acompañando este proceso desde mi corazón. A mi madre por inculcar que mi futuro lo construyo yo por y para mí, a mis hermanas por sus fieles miradas y abrazos. Agradecer con los que recordaré el tiempo invertido, las lágrimas, las risas y la alegría, gracias por acompañar mi proceso Milton, Caro, Seba, Ani, Laris, Stefi, Pato, Mati, Juan, Pancho y Gri-gri.

(Epígrafe:) *"Dicen que antes de
entrar en el mar, EL RIO
tiembla de miedo.
mira para atrás, para todo el día
recorrido,
para las cumbres y las montañas,
para el largo y sinuoso camino
que atravesó entre selvas y
pueblos,
y vé hacia adelante un océano
tan extenso, que entrar en él es
nada más que desaparecer para
siempre.
Pero no existe otra manera.
El río no puede volver.
Nadie puede volver.
Volver es imposible en la
existencia.
El río precisa arriesgarse y
entrar al océano.
Solamente al entrar en él, el
miedo desaparecerá,
porque apenas en ese momento,
sabrás que no se trata de
desaparecer en él, sino volverse
océano".*

Khalil Gilbran

RESUMEN

Los Sistemas Binarios Estocásticos (SBS por sus siglas en inglés) son modelos matemáticos que permiten generalizar la noción clásica de confiabilidad de redes de comunicación. Los SBS consisten en un conjunto finito de componentes sujetos a fallas aleatorias, y una función lógica que describe la operación o falla del sistema, para cada estado posible de sus componentes. Instancias particulares de estos problemas pueden ser modelados bajo funciones monótonas, clasificándolos como Sistemas Binarios Estocásticos Monótonos (SMBS), los cuales heredan el orden natural del producto cartesiano.

En el marco de Internet e inspirado en aplicaciones sensibles a la latencia, el problema de Confiabilidad Diámetro Acotado analiza la probabilidad de que dos o más nodos que desean comunicarse estén conectados por caminos acotados, ante fallas en los elementos de la red. Este problema ha sido estudiado principalmente en estructuras de redes donde fallan únicamente aristas o únicamente fallan nodos, y no tanto en fallas conjuntas (o "fallas simultáneas" de nodos y aristas). El problema de Confiabilidad Diámetro Acotada para el modelo hostil es una instancia de los sistemas binarios estocásticos monótonos, y la complejidad de su evaluación se clasifica como NP -difícil.

Las contribuciones de este documento son las siguientes. En primera instancia se implementa el método de Reducción Recursiva de la Varianza (RVR) para el cálculo de Confiabilidad Diámetro Acotada del modelo hostil. Se realiza un estudio de sensibilidad del método RVR ante distintas estrategias de selección de cortes minimales. Finalmente, se brinda un estudio comparativo entre el método Monte Carlo Crudo y RVR, observando el desempeño de los métodos elegidos.

Palabras claves:

Sistemas Binarios Estocásticos Monótonos, Confiabilidad Diámetro Acotado, Modelo Hostil, Método Reducción Recursiva de la Varianza, Método Monte Carlo.

ABSTRACT

Stochastic Binary Systems (SBS) are mathematical models that allow extend the classical notion of reliability in communication networks. An SBS consists of a set of finite components subject to random failures, and a logical function that describes the operation or failure of the system, for each possible configuration. Particular instances of these problems are modeled under monotone functions, known as Stochastic Monotone Binary System (SMBS), which inherit the natural order of the cartesian product.

Within the framework of the Internet and inspired by latency-sensitive applications, the Diameter Constrained Reliability is the probability that two or more target nodes are connected by bounded paths, in the resulting network after potential failures on its components. The scientific literature considers structural links or node failures. Here, we consider joint failures instead, what is known as the Hostile Network model.

The Diameter Constrained Reliability problem for the hostile model is an instance of Stochastic Monotone Binary System, and its reliability evaluation belongs to the \mathcal{NP} -Hard class.

The contributions of this thesis can be summarized as follows. First, the Recursive Variance Reduction (RVR) method is implemented to estimate the Diameter Constrained Reliability in this novel hostile environment with joint node/link failures. Then, a sensibility analysis of RVR method is performed, following different cut-selection strategies. Finally, a performance analysis between Monte Carlo Crudo and RVR methods is provided.

Keywords:

Diameter Constrained Reliability, Stochastic Monotone Binary System, Hostile Model, Recursive Variance Reduction, Crude Monte Carlo.

Tabla de contenidos

1	Introducción	1
1.1	Organización del documento	5
2	Fundamentos teóricos	6
2.1	Teoría de grafos	6
2.2	Sistemas Binarios Estocásticos	9
2.3	Sistemas Binarios Estocásticos Monótonos	11
2.4	Medida de Confiabilidad	13
2.4.1	Confiabilidad Diámetro Acotada	15
2.5	Complejidad Computacional	20
2.5.1	Complejidad computacional en confiabilidad de redes	21
2.5.2	Complejidad computacional en un SMBS	23
2.6	Cálculo de Confiabilidad	26
2.6.1	Métodos de Enumeración Completa	26
2.6.1.1	Método de Factorización	26
2.6.1.2	Principio de Inclusión y Exclusión	27
2.6.2	Métodos aproximados	28
2.6.2.1	Métodos Monte Carlo	28
2.6.2.2	Reducción Recursiva de la Varianza	32
2.6.2.3	RVR en el Modelo Arista	32
2.6.2.4	RVR en SMBS	38
3	Metodología	41
3.1	Implementación de RVR para DCR en el modelo hostil	42
3.2	Estrategias de separación	43
3.2.1	Corte Lineal	44
3.2.2	Corte Trivial	49
3.2.3	Corte Estructura	53

3.2.4	Corte Optimizado	58
4	Resultados Experimentales	62
4.1	Análisis de eficiencia	62
4.2	Presentación de datos	64
4.2.1	Infraestructura de cómputo	64
4.2.2	Casos de estudio	65
4.2.2.1	Grafo Puente	67
4.2.2.2	Grafo Completo K_4	77
4.2.2.3	Grafo Grilla	86
4.2.2.4	Grafo Arpanet	95
4.2.2.5	Grafo Dodecaedro	99
5	Consideraciones finales	105
5.1	Trabajo futuro	109
	Referencias bibliográficas	111
	Anexos	115
Anexo 1	Conceptos Preliminares	116
1.1	Fundamentos básicos	116
1.1.1	Concepto de generalización	116
1.1.2	Teoría del orden	116
1.1.2.1	Conjuntos parcialmente ordenados	117
1.1.2.2	Par ordenado	117
1.1.2.3	Producto Cartesiano	118
1.1.3	Teoría de grafos	118
Anexo 2	Algoritmos auxiliares	122
2.1	Algoritmo de búsqueda en profundidad	123
2.2	Algoritmo de búsqueda en anchura	125
2.3	Algoritmo Dijkstra	127
2.4	Método Ford-Fulkerson	129
Anexo 3	Resultados auxiliares	133
3.1	Resultados grafo puente	133
3.2	Resultados grafo completo	137
3.3	Resultados grafo grilla $n = 3$	141
3.4	Resultados grafo grilla $n = 5$	143

3.5	Resultados grafo arpanet	145
3.6	Resultados grafo dodecaedro	147
Anexo 4	Algoritmo RVR en acción	150

Capítulo 1

Introducción

Las redes de telecomunicaciones se encuentran cada vez más presentes en nuestra vida diaria. Su crecimiento masivo, la diversificación de los servicios y funcionalidades que ofrecen y el número creciente de usuarios, han impulsado distintas líneas de investigación dedicadas al análisis y diseño de redes. Un sin fin de métricas que cuantifican la calidad del servicio brindado pueden ser definidas. Algunas de éstas son simplemente métricas binarias (por ejemplo, la red puede estar en correcta operación o en falla). Otras pueden tener una naturaleza no binaria (por ejemplo, rendimiento de la red). Es posible incluso considerar métricas complejas, que consideran distintas entidades competitivas, asignando diferentes pesos a distintas clases de actores, teniendo en cuenta la capacidad del medio, la pérdida de paquetes, los retrasos, etc. Por ejemplo el RTT (round trip-time), probabilidad de bloqueo, largo medio de ráfaga de pérdidas, retrasos, variabilidad en los retrasos (jitter) entre otras.

Durante las primeras décadas de desarrollo de redes de telecomunicaciones, se hizo hincapié en métricas que expresaban la capacidad de una red para seguir funcionando [36]. Cuando se desplegaron las primeras redes de paquetes conmutados utilizando líneas telefónicas, el principal objetivo era mantener en funcionamiento las líneas (y la topología de red subyacente). Dentro de este contexto, una simple diferenciación entre un estado de operación y otro de falla fue lo que motivó a la investigación en el campo *Análisis de Confiabilidad*. Las mejoras tecnológicas y reducidos costos de los equipos de red, permitieron acceso a la redundancia, conduciendo al diseño y despliegue de redes mucho más confiables en términos de conectividad. Al mismo tiempo, un mayor tráfico de datos y aplicaciones más complejas condujeron a un cambio de enfoque,

pasando de conectividad a métricas de rendimiento más complejas [29]. Luego, a fines de los años 90, llegó el momento del despliegue masivo de la fibra óptica, líneas altamente confiables. La extraordinaria capacidad de éstas hizo posible que redes de una única línea entreguen los niveles de rendimiento requeridos, siempre que ésta esté operacional. Cobrando relevancia nuevamente, los modelos de confiabilidad. En la actualidad, el despliegue masivo de redes inalámbricas, cuyos enlaces punto a punto son menos confiables debido a razones como factores ambientales y preocupaciones por el ahorro de energía, hizo que las propiedades de conectividad vuelvan a ganar relevancia. Actualmente, un requerimiento altamente desafiante para el desarrollo masivo de redes celulares 5G es lograr redes ultra-confiables de baja latencia, cumpliendo con el lema URLLC del inglés *ultra-reliable low-latency communication* [21]. Podemos decir que el énfasis histórico por la conectividad condujo a un concepto más integral, abarcando tanto preocupaciones de confiabilidad como rendimiento [36], [22].

Asimismo, una red de comunicación evidencia vulnerabilidad frente a sus elementos, desastres naturales y/o causas humanas. Esto conlleva a grandes esfuerzos desde el punto de vista de la ingeniería, que batalla frente a fallas simples o catastróficas de una red. Con los objetivos de asegurar la comunicación necesaria que respalde a primeros auxilios, en [19] desarrollan estrategias de restauración progresiva luego de una catástrofe. Reforzando la afirmación de que las redes de telecomunicaciones son críticas para la sociedad.

Las redes de comunicación, bajo la clásica *Teoría de Grafos* se modelan como grafos probabilísticos. Los sitios y enlaces son representados como nodos y aristas, y se etiqueta cada componente sus respectiva probabilidad de falla. Los modelos probabilísticos de fallas aleatorias en aristas son altamente estudiados dentro de la literatura científica [16]. En segundo lugar pero no menos importante están los modelos de fallas en nodos. No obstante, en redes reales, las fallas aleatorias pueden ocurrir tanto en nodos como aristas. Inspirados en redes reales en esta tesis se trabaja el modelo probabilístico hostil. En el mismo, un conjunto de nodos distinguidos llamados terminales son perfectos, mientras que las aristas y los nodos no terminales fallan con una probabilidad conocida e independiente.

A través de la métrica de confiabilidad clásica (CLR, por sus siglas en inglés), el concepto de *comunicación exitosa* supone la posibilidad de contactar dos o más puntos en la red. Asimismo, existen situaciones donde se requiere que los lapsos de tiempo entre la recepción y la transmisión en la comunicación o en el largo de los caminos no superen cierto umbral. En el año 2001, Petingi y Rodríguez [32] definen una nueva métrica denominada Confiabilidad Diámetro Acotada (DCR), ampliando el concepto de comunicación exitosa como la probabilidad de conectar dos o más puntos de la red por caminos de largo menor o igual a una cota d dada. La medida de confiabilidad diámetro acotada generaliza el concepto de confiabilidad clásica, incluyendo un número mayor de problemas en donde existen restricciones de tiempo y/o distancia en la comunicación.

Adicionalmente, los problemas de confiabilidad en redes pueden ser representados matemáticamente como Sistemas Binarios Estocásticos (o SBS), o representados por ciertos sistemas que poseen un buen comportamiento, que denominamos Sistemas Binarios Estocásticos Monótonos (o SMBS). En los SMBS la función lógica o estructura es monótona creciente, bajo el orden natural de palabras binarias. En esta tesis se aprovecha el hecho de que la monotonía es una propiedad invariante bajo restricciones, y las propiedades de construcción óptima de configuraciones operativas minimales en todo SMBS, para aplicar el método de la Reducción Recursiva de la Varianza (o RVR), a efectos de la estimación de confiabilidad [3]. Conceptualmente el problema de confiabilidad diámetro acotada para el modelo hostil, es un concepto sencillo de comprender, sin embargo el cálculo de su confiabilidad es complejo desde un punto de vista computacional, y pertenece a la familia de problemas \mathcal{NP} -difíciles

A efecto de la estimación de confiabilidad, el método de Reducción Recursiva de la Varianza ha sido condecorado como mejor paper de IEEE, superando al clásico método Monte Carlo Crudo (o MCC) en la medida de mérito que es producto entre tiempo de cómputo y error [8]. Adicionalmente, la eficiencia del método depende de la construcción óptima de conjuntos de corte utilizado en cada llamada recursiva, y su eficiencia disminuye al evaluar redes de gran porte.

Las principales contribuciones de esta tesis se pueden resumir en los tres puntos siguientes:

- Adaptación del método de RVR para la estimación de la confiabilidad diámetro acotada en el modelo hostil.
- Análisis de sensibilidad del método RVR en base a distintas estrategias de construcción de conjuntos separadores de grafos.
- Estudio comparativo entre los métodos Monte Carlo Crudo y RVR, a efectos de analizar el rendimiento de estos en el problema planteado.

Hasta lo mejor de nuestro conocimiento, existen trabajos en la literatura relativos al modelo hostil [25], pero el citado estudio no incluye restricciones en el diámetro.

1.1. Organización del documento

En el Capítulo 1 se introdujo la motivación y contexto de la tesis.

El Capítulo 2 presenta conceptos fundamentales para la comprensión del problema. Se realiza una breve introducción a la teoría de grafos, modelos probabilísticos para redes de telecomunicaciones, y se presentan conceptos esenciales relativos a la confiabilidad como los fundamentos teóricos de los Sistemas Binarios Estocásticos y Sistemas Binarios Estocásticos Monótonos. La Confiabilidad Diámetro Acotada es formalmente definida, examinado su complejidad computacional y se muestra la intratabilidad de su cálculo, hecho que promueve el uso de métodos aproximados. Finalmente, se presentan los métodos aproximados de Monte Carlo Crudo y Reducción de la Varianza Recursiva [8], [3] utilizados para la estimación de la confiabilidad.

El Capítulo 3 presenta el enfoque del trabajo, objetivo e integración con el marco teórico planteado. Con la finalidad de evaluar la eficiencia del método RVR, se analizan distintas estrategias de construcción de conjuntos separadores de grafos. En la Sección 3.2.1 se plantean cortes a partir de la estructura abstracta de un SMBS. En la Sección 3.2.2 se adapta para el sistema bajo estudio una estrategia ya expuesta en trabajos fundacionales del área [8]. En la Sección 3.2.3 se desarrolla una estrategia de corte que juega con la singularidad del camino acotado en nuestro sistema. Y en la Sección 3.2.4 se expone el problema de obtener cortes de máxima probabilidad de falla, mediante un problema de optimización.

El Capítulo 4 desarrolla el marco práctico de la investigación regida por los objetivos planteados. Se expone un análisis cuantitativo del método RVR, en base al cálculo de la confiabilidad diámetro acotada del modelo hostil, y distintas estrategias de separación de grafos planteadas en el Capítulo 3. Se contrastan estos resultados contra los obtenidos con el método Monte Carlo Crudo en función de estimaciones obtenidas para casos de prueba. Se detallan resultados empíricos que analizan la sensibilidad de la versión de RVR implementada, en función de las estrategias de cortes establecidas.

Finalmente, el Capítulo 5 sintetiza las conclusiones generales y principales contribuciones que surgen como producto de esta tesis.

Capítulo 2

Fundamentos teóricos

En este capítulo nos centraremos en establecer la base matemática detrás de las métricas de confiabilidad utilizadas en esta tesis. Comenzamos presentando definiciones necesarias y la notación de Teoría de Grafos. Continuamos con los conceptos de Sistemas Binarios Estocásticos y Sistemas Binarios Estocásticos Monótonos. Se introduce brevemente la complejidad computacional del cálculo de la confiabilidad, lo que conlleva a estudiar distintos métodos exactos y aproximados para la obtención de la medida. Centrándonos en el método de Reducción Recursiva de Varianza.

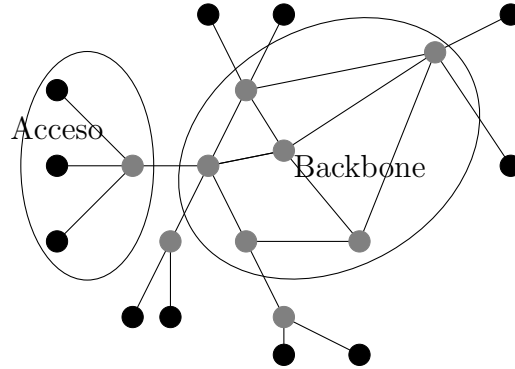
2.1. Teoría de grafos

La topología de una red de comunicación se puede modelar matemáticamente a través de un grafo no dirigido $G = (V, E)$, donde el conjunto de nodos V representa sitios (llamado vértices o nodos) y el conjunto de aristas o enlaces E representa la conexión entre ellos. En el ejemplo de la [Figura 2.1](#), la topología de la red está representada por el grafo, donde los distintos sitios de acceso y backbone son remplazados por nodos, y la interconexión entre ellos por aristas.

Definición 2.1.1. ¹ Un grafo es un par $G = (V, E)$ donde V son llamados nodos y $E \subseteq \{\{x, y\} : x \in V \wedge y \in V\}$ son llamados aristas. Notamos, la cardinalidad de los conjuntos como $|V| = n$ y $|E| = m$, y un enlace como uv o $\{u, v\}$ para sus correspondientes nodos u y v .

¹Ver definiciones en [\[33\]](#)

Figura 2.1: Red de comunicación modelada por un grafo



Definición 2.1.2. ¹ Un *camino* es una secuencia de enlaces e_1, \dots, e_l , tal que, $e_i = v_i v_{i+1}$ para $i = 0 \dots l - 1$. Un camino es *simple* si no repite vértices, es decir, $v_i \neq v_j$ para todo $i \neq j$. Se dice entonces que el *camino simple* conecta los nodos v_0 y v_l con un longitud igual a l .

En el interés de evaluar ciertas propiedades de conectividad del grafo probabilístico (definido en 2.1.8), se introducen conceptos relacionados a la conectividad, y d -conectividad entre otros.

Definición 2.1.3. ¹ Un grafo se dice *conexo*, si y solo si existe un *camino simple* para cada par de nodos $s, t \in V$.

Se distinguen ciertos nodos $K \subseteq V$, denominados *terminales*. En general, vamos a estar interesados en propiedades de conectividad que aplica únicamente para este conjunto nodos K y se considera $\bar{V} = V/K$ como nodos opcionales, que llamaremos nodos no terminales.

Definición 2.1.4. ¹ Un grafo se dice K -conexo si y solo si existe un *camino simple* para cada par de nodos $s, t \in K$.

También, es de nuestro interés familiarizarnos con la longitud del camino que conecta nodos terminales, por lo que se introducen definiciones relacionadas.

Definición 2.1.5. ¹ La *distancia* entre dos nodos s y t es la longitud del *camino simple* más corto. Dado un entero no negativo d , los nodos son d -conexo si y solo si la distancia entre ellos no excede d .

Definición 2.1.6. ¹ El *diámetro* de un grafo, es la *distancia* máxima entre cualquier par de nodos del grafo. Dado $K \subseteq V$, se define K -*diámetro* de un grafo como la *distancia* máxima entre cualquier par de nodos $s, t \in K$.

Definición 2.1.7. [4] Un grafo se dice d - K -conexo si y solo si su K -diámetro no supera d .

Adicionalmente, asignando una probabilidad de operación (o falla) a cada elemento del grafo, se incurre en ciertas propiedades como la definición del grafo probabilístico, y modelos probabilísticos.

Definición 2.1.8. [31] Un grafo probabilístico es aquel donde los nodos y los enlaces son etiquetados con una probabilidad de operación. Eso es, $G = (V, E, \vec{p})$, donde $p_i \in [0, 1]$, $\forall i \in V \cup E$.

En una red de comunicación existen o se diferencian modelos probabilísticos dependiendo de los componentes a los que se les asigna la aleatoriedad de funcionamiento, los casos más generales son:

- Modelo de Aristas: Nodos perfectos, aristas fallan. Obteniendo el grafo aleatorio $G = (V, E, \vec{p})$, donde $p_i \in [0, 1]$, $\forall i \in E$.
- Modelo de Nodos: Nodos fallan, aristas perfectas. Obteniendo el grafo aleatorio $G = (V, E, \vec{p})$, donde $p_i \in [0, 1]$, $\forall i \in V$.
- Modelo Nodos-Aristas: Tanto nodos y aristas en falla. Obteniendo el grafo aleatorio $G = (V, E, \vec{p})$, donde $p_i \in [0, 1]$, $\forall i \in V \cup E$
- Modelo Hostil: Aristas y nodos opcionales fallan, terminales perfectos. Modelo tratado en esta tesis.

Modelo hostil

Partiendo del modelo Nodo-Arista, donde aristas y nodos se etiquetan con una probabilidad de operación, el modelo hostil asume como hipótesis de trabajo: “*que los nodos terminales son perfectos, y la fallas se dan sobre los nodos no terminales y aristas*”.

Consideremos un grafo simple no dirigido $G = (V, E)$, y el conjunto distinguido de terminales $K \subseteq V$. Tanto los enlaces como los nodos no terminales se etiquetan con una probabilidad de operación, y $\forall u \in K$ se cumple $p_u = 1$. Consecuentemente el vector de probabilidades definido $\vec{p} = \{\{p_e\}, \{p_v\}\}$ cumple, $p_e = \{p_{e_1}, p_{e_2}, \dots, p_{e_{|m|}}\}$ es el vector de probabilidad elemental de los $e_i \in E$ y, $p_v = \{p_{v_1}, p_{v_2}, \dots, p_{v_{|n-k|}}, p_{u_1}, \dots, p_{u_k}\}$ es el vector de probabilidad elemental de los $v_j \in V$. En este contexto, notamos que para el modelo hostil el grafo aleatorio $G = (V, E, \vec{p})$, donde $p_i \in [0, 1]$, $\forall i \in \bar{V} \cup E$, con $\bar{V} = V/K$.

Las definiciones previas nos permiten introducir brevemente el problema de computar la medida de confiabilidad que tratamos en esta tesis. Asumiendo que, la red de comunicación se modela por un grafo probabilístico Modelo Hostil, contamos con un conjunto de terminales K y una cota de distancia d dada. Computar la confiabilidad diámetro acotada consiste en calcular la probabilidad de que el grafo aleatorio resultante sea d - K -conexo.

El modelo hostil ha sido tratado con anterioridad. En el documento [25], estudian la confiabilidad clásica del modelo y comparan tres métodos de cálculo aproximados, Monte Carlo Crudo (MCC), Reducción de Varianza Recursiva (RVR) y Muestreo por Importancia (MI). En [17] el autor utiliza el método Artificial Neural Network (ANN) para resolver el problema de optimización para el modelo hostil. También el autor Jacques Carlier desarrolla un método de cálculo exacto de confiabilidad en modelos clásico y hostil para grafos de tamaño reducido, utilizando la descomposición de Rosenthal [28].

2.2. Sistemas Binarios Estocásticos

En esta sección se expone el problema bajo estudio por sistemas de múltiples componentes.

Un *Sistema Binario Estocástico* o SBS por sus siglas en Inglés (Stochastic Binary Systems), representa un sistema de fallas aleatorias de sus componentes. Cada elemento en el conjunto de componentes, denotado S , está representado por dos estados (operativo o falla), o por la representación binaria $\{0, 1\}$. Cada componente trabaja de forma independiente y estocásticamente con cierta probabilidad de operación, denominada probabilidad elemental. Se toman notación y definiciones de los siguientes documentos [34, 3, 25].

Definición 2.2.1. ² Un *Sistema Binario Estocástico* es la terna $\mathcal{S} = (S, p, \phi)$:

- $S = \{s_1, \dots, s_m\}$ conjunto elemental de *componentes*.
- $\vec{p} = (p_1, \dots, p_m)$ vector de probabilidad elementales.
- $\phi : \{0, 1\}^m \rightarrow \{0, 1\}$ la función *estructura*.

El conjunto S y la función estructura son determinísticos. El vector de probabilidad \vec{p} define la probabilidad elemental $\forall s_i \in S$, determinando un modelo

²Ver definición en [34, 3, 25]

estocástico sobre el sistema. Cada $s_i \in S$ tiene dos estados posibles: operativo o inactivo, regido por una distribución de Bernoulli donde el éxito es medido por la probabilidad de que el elemento esté operativo $P(s_i = \text{operativo}) = p_i$ y el fracaso por la probabilidad de que el elemento se encuentre en falla $P(s_i = \text{inactivo}) = q_i = 1 - p_i$.

Por conveniencia el autor en [3] traduce cada subconjunto $T \subseteq S$ (o evento T) a una palabra binaria, utilizando el vector indicador $\varphi(T) = 1_T \in \{0, 1\}^m$ de manera tal que, $1_T(i) = 1$ si solo si $i \in T$. Su función inversa $\varphi^{-1} : \{0, 1\}^m \rightarrow P(S)$ denotado como $\varphi^{-1}(x) = S_x \subseteq S$, obteniendo así una biyección entre ambos conjuntos. Asumiendo m componentes binarios en S , sea S_x un subconjunto equivalente de S , tal que, $S_x \subseteq S$ entonces S_x es representado por el *vector de estado* o la *configuración del sistema* $\vec{x} = (x_1, \dots, x_m) \in \{0, 1\}^m$ que operan según el vector de probabilidad \vec{p} .

Definición 2.2.2. ² La función lógica llamada *estructura* $\phi(x)$ definida para todo $S_x \subseteq S$, nos indica cuando el sistema se encuentra en inactivo o operativo.

$$\phi(x) = \begin{cases} 1 & S_x \text{ es un estado operativo;} \\ 0 & S_x \text{ es un estado en falla;} \end{cases} \quad (2.2.1)$$

El éxito de la función estructura va a depender si el vector de estados \vec{x} cumple con la lógica definida. El dominio de esta función lógica es el conjunto de todas las palabras binarias $\vec{x} \in \{0, 1\}^m$, es decir el conjunto de todos los posibles estados de los elementos de S , y su codominio el espacio $\{0, 1\}$. Si los elementos operativos que componen S_x cumplen que $\phi(x) = 1$ se obtiene un estado de éxito, en caso contrario genera un estado inactivo en el sistema. En este contexto, el resultado de éxito o fracaso de $\phi(x)$ define una variable aleatoria de Bernoulli con una probabilidad de correcta operación para el sistema, determinada por la esperanza $E(\phi(X))$.

Definición 2.2.3. ² Sea un vector aleatorio $X = (X_1, \dots, X_m)$ compuesto por m variables aleatorias independientes, bajo la distribución de Bernoulli, tal que $P(X_i) = p_i$, entonces la *confiabilidad de un SBS* corresponde a la probabilidad de correcto funcionamiento del sistema donde:

$$\text{Confiabilidad } R_s = E(\phi(X)) = P(\phi(X) = 1) \quad (2.2.2)$$

$$\text{Anti-confiabilidad } Q_s = 1 - R_s = 1 - E(\phi(X)) \quad (2.2.3)$$

Definición 2.2.4. ² Dado un $\mathcal{S} = (S, p, \phi)$ sistema binario estocástico. El vector de estado $\vec{x} \in \{0, 1\}^m$ es un *pathset* si $\phi(x) = 1$, y es un *cutset* si $\phi(x) = 0$.

Definición 2.2.5. ² Sea $\mathcal{S} = (S, p, \phi)$ un sistema binario estocástico. Un componente $s \in S$ es *esencial* si el sistema falla siempre que falla s .

Un componente $s \in S$ es *irrelevante* si $\phi(x|_{s=0}) = \phi(x|_{s=1})$ ³ para todos los posibles valores de $\vec{x} \in \{0, 1\}^m$.

Definición 2.2.6. ² Un sistema binario estocástico es *homogéneo* si la confiabilidad de cada elemento de S es idéntica, o sea $\forall i p_i = p$. Dada la función estructura ϕ , denotaremos a \bar{x} al vector complemento en bit. En particular $\bar{\phi}(x) = 1 - \phi(x)$.

2.3. Sistemas Binarios Estocásticos Monótonos

Los sistemas binarios estocásticos monótonos o SMBS (por sus siglas en inglés), son SBS definidos sobre un orden parcial y determinados por la monotonía de su función estructura. Para definir estos sistemas se toma las notaciones y definiciones del documento [34].

El autor identifica que si un conjunto binario $\{0, 1\}$ es equipado con un orden parcial, cumple las propiedades reflexiva, antisimétrica y transitiva (vea Anexo 1) donde $0 \leq 0$, $0 \leq 1$ y $1 \leq 1$. Consecuentemente el conjunto $\{0, 1\}^m$ hereda un orden natural en el producto cartesiano.

Dadas las coordenadas ordenadas $x = (x_1, \dots, x_m) \in \{0, 1\}^m$ y $y = (y_1, \dots, y_m) \in \{0, 1\}^m$, se cumple que $x \leq y$ si solo si $x_i \leq y_i$ para todos $i = 1, \dots, m$, y $x < y$ si $x \leq y$ y $y \neq x$.

■ Formalmente denota que²:

- $\vec{0}_m$ a la palabra binaria con todos los bits en 0
- $\vec{1}_m$ a la palabra binaria con todos los bits en 1
- δ_i a la palabra binaria con todos los bits en 0 menos el bit de la posición i que toma valor 1.

³la notación hace referencia al reemplazo del bit $s = 0$ o $s = 1$ en la palabra binaria x .

Las notaciones anteriores permiten al autor conferir que, un sistema binario estocástico declarado en el conjunto binario $\{0, 1\}^m$ y equipado con un orden parcial es monótono si cumple la siguiente definición.

Definición 2.3.1. ² La terna $\mathcal{S} = (S, p, \phi)$ define un *Sistema Binario Estocástico Monótono* si la función estructura ϕ cumple,

- $\phi(\vec{0}_m) = 0$
- $\phi(\vec{1}_m) = 1$ y
- $\phi : \{0, 1\}^m \rightarrow \{0, 1\}$ es monótona creciente

Entonces un sistema que se describe como SMBS conserva el orden, en otras palabras todo sistema en correcta operación continuará en estado operativo si éste incorpora nuevos componentes.

El modelo matemático SBS generaliza el modelo SMBS, heredando este último las definiciones de confiabilidad, pathset y cutset entre otros, siendo redefinidos según la monotonía del sistema.

Definición 2.3.2. ² Sea $\mathcal{S} = (S, p, \phi)$ un Sistema Binario Estocástico Monótono.

Un pathset x es un *minpath* si $\phi(y) = 0$ para todo y tal que $y < x$.

Un cutset y es un *mincut* si $\phi(x) = 1$ para todo x tal que $y < x$.

En referencia al documento [23]. Sea $\mathcal{P} = \{P_1, P_2, \dots, P_{|P|}\}$ el conjunto de todos los minpaths de \mathcal{S} y $\{P_i\}$ el evento “todos los elementos de P_i operan”, entonces puede definirse la confiabilidad en términos de $\{P_i\}$. La probabilidad de correcta operación del sistema es la probabilidad de que uno de los eventos $\{P_i\}$ ocurra.

$$R_s = P(\phi(x) = 1) = P(P_1 o P_2 o \dots o P_{|P|}) \quad (2.3.1)$$

Sea $\mathcal{C} = \{C_1, C_2, \dots, C_{|C|}\}$ el conjunto de todos los mincut de \mathcal{S} y $\{C_i\}$ el evento “todos los elementos de C_i fallan”, entonces puede definirse la anti-confiabilidad en términos del evento $\{C_i\}$. La probabilidad de que el sistema falle es la probabilidad de que uno de los eventos $\{C_i\}$ ocurra.

$$Q_s = P(\phi(x) = 0) = P(C_1 o C_2 o \dots o C_{|C|}) \quad (2.3.2)$$

2.4. Medida de Confiabilidad

La medida de confiabilidad a analizar dependerá del conjunto de nodos terminales $K \subseteq V$ que se desea comunicar, del modelo probabilístico que describe su funcionamiento, y la probabilidad de falla de los componentes. Se instanciará este problema a través de sistemas binarios estocásticos, obteniendo así un modelo matemático por intermedio de la terna \mathcal{S} , que permitirá estudiar la confiabilidad de la red de comunicación a partir de la definición 2.2.3.

Entonces, partiendo del conocimiento probabilístico de las fallas de los componentes y la función topología de la red, se llega a determinar un índice o valor correspondiente a la probabilidad con que la red opera. En los siguientes párrafos se proveen diferentes ejemplos de sistemas binarios estocásticos, y sus características.

Dado el grafo $G = (V, E)$ simple no dirigido, un subconjunto $K \subseteq V$ tal que $|K| \geq 2$ y un subgrafo aleatorio $G_x = (V_x, E_x) \subseteq G$. Tomando un orden arbitrario para el conjunto de aristas y nodos tal que, $e_i < e_{i+1} \forall e_i \in E = \{e_1, e_2, \dots, e_{|E|}\}$, y $v_i < v_{i+1} \forall v_i \in V = \{v_1, v_2, \dots, v_{|V|}\}$. Cada subgrafo aleatorio $G_x = (V_x, E_x)$ con $E_x \subseteq E$ y $V_x \subseteq V$ va a ser identificado con una palabra binaria $x \in \{0, 1\}^m$, tal que $x = \{\{x_e\}, \{x_v\}\}$ donde $x_e \in \{0, 1\}^{|E|}$ es la palabra binaria que representa el estado de los enlaces y cumple $x_{e_i} = 1$ si y solo si $e_i \in E_x$, y donde $x_v \in \{0, 1\}^{|V|}$ es la palabra binaria que representa el estado de los nodos y cumple $x_{v_j} = 1$ si y solo si $v_j \in V_x$. Entonces x define una variable aleatoria de Bernoulli gobernada por el vector de probabilidad elemental $p = \{\{p_e\}, \{p_v\}\}$ donde $p_e = \{p_{e_1}, p_{e_2}, \dots, p_{e_{|E|}}\}$ vector de probabilidad de operación elemental de los $e_i \in E$ y $p_v = \{p_{v_1}, p_{v_2}, \dots, p_{v_{|V|}}\}$ vector de probabilidad de operación elemental de los $v_j \in V$.

En consecuencia se define el problema como un SBS coherente (ver 2.2.1), con $G_x \subseteq (V \cup E, p)$ y la función estructura $\phi : \{0, 1\}^m \rightarrow \{0, 1\}$ tal que $\phi(x) = 1$ dependiendo del subconjunto $K \subseteq V$ que se quiera estudiar.

Como brevemente ya se introdujo, los grafos aleatorios G_x , pueden ser clasificados dependiendo de los componentes a los que se les asigna la aleatoriedad de funcionamiento:

- Modelo de Aristas: Nodos perfectos, aristas fallan. Obteniendo el grafo aleatorio $G_x \subseteq (E, p_e)$ donde $p_e : E \rightarrow [0, 1]$.

- Modelo de Nodos: Nodos fallan, aristas perfectas. Obteniendo el grafo aleatorio $G_x \subseteq (V, p_v)$ donde $p_v : V \rightarrow [0, 1]$.
- Modelo Nodos-Aristas: Tanto nodos y aristas en falla. Obteniendo el grafo aleatorio $G_x \subseteq (V \cup E, p)$ donde $p : V \cup E \rightarrow [0, 1]$.
- Modelo Hostil: Nodos terminales perfectos, aristas y nodos no terminales en falla. Entonces $\forall u \in K$ se cumple $p_u = 1$, obteniendo el grafo aleatorio $G_x \subseteq (\{V - K\} \cup E, p)$ donde $p : \{V - K\} \cup E \rightarrow [0, 1]$.

Por otro lado, la función estructura dependerá del tipo de confiabilidad (ver 2.2.3) que se desea obtener. En los siguientes ejemplos⁴ se listan diferentes aplicaciones de SBS asociados a la medida de confiabilidad, sea $x = \{0, 1\}^m$ la palabra binaria que identifica a G_x según la aleatoriedad de sus componentes.

- Confiabilidad All-Terminal, $R_V(G)$: Corresponde a la probabilidad de que entre cualquier par de nodos de la red ($V = K$) exista al menos un camino que los comunique. Por lo que, $\phi(x) = 1$ si y solo si G_x es conexo.
- Confiabilidad K -Terminal, CLR: Se define como $R_K(G)$ a la probabilidad de que todo nodos $u \in K$ este definido en la misma componente conexa. Por lo tanto, $\phi(x) = 1$ si y solo si G_x es K -conexo.
- Confiabilidad Fuente-Terminal: las mismas restricciones del modelo anterior, pero tomando un conjunto de dos terminales $K = \{s, t\}$.
- Confiabilidad Diámetro Acotada, DCR: Se define como $R_{\{K, d\}}(G)$ a la probabilidad resultante de que el modelo K -Terminal quede conectado por caminos de largos d . Por lo que, $\phi(x) = 1$ si y solo si G_x es d - K -conexo.

El análisis de confiabilidad K -Terminal de la red en combinación con el modelo arista, comúnmente llamado clásico, es el más estudiado en la literatura. Este es un SBS monótono, pues el agregado de una arista nunca desconecta a los terminales. Así mismo, las medidas de confiabilidad arriba listadas bajo los distintos modelos probabilísticos son sistemas binarios estocásticos cumpliendo que su función estructura es monótona [3, 34], a excepción bajo el modelo nodo-aristas donde la definición de estado operativo se cumple parcialmente para el encendido de nodos terminales. Para el ejemplo⁵ de la figura 2.2, la

⁴Ver más ejemplos en [34]

⁵Ver mas ejemplos prácticos en [34]

medida de confiabilidad all-terminal en un modelo nodo-arista, obtiene un sistema inicial operacional en 2.2a y al levantar un nodo terminal se obtiene un sistema final no operacional en 2.2b.

Figura 2.2: Confiabilidad all-terminal en modelo nodo-arista



En el contexto de esta tesis, trabajaremos la **Confiabilidad Diámetro Acotada en el Modelo Hostil**.

Bajo las hipótesis del modelo y conjuntos binarios parcialmente ordenados, la función estructura cumple la definición de monotonía siendo modelado matemáticamente como un SMBS [3, 25]. Consecuentemente si el sistema se encuentra en correcta operación va a continuar operativo frente al agregado de nuevos componentes, manteniendo así su correcto estado de operación. Y a su vez si el sistema se encuentra no operativo, el apagado de componentes mantendrá el sistema inactivo.

Como ya se hizo alusión, el modelo hostil ha sido tratado con anterioridad en distintos documentos. Por otro lado la medida confiabilidad diámetro acotada ha sido ampliamente estudiada sobre el modelo probabilístico clásico, no obstante hasta nuestro conocimiento no se ha realizado estudio sobre el modelo hostil planteado.

2.4.1. Confiabilidad Diámetro Acotada

Petingi y Rodriguez en [31] introducen el concepto de “confiabilidad diámetro acotada” como una medida de la confiabilidad en redes que considera una cota superior a la distancia entre los nodos que se desea comunicar. Denotada como $R(G, K, d)$ generaliza el concepto confiabilidad clásica $R(G, K)$ estableciendo que, una red está operativa si los nodos objetivo se comunican por caminos cuyo largo no supera una cota d . Y se demuestra que este es un problema \mathcal{NP} -difícil.

La medida de confiabilidad acotada por diámetro es útil en diferentes contextos, en [13] se menciona que es un indicador de la idoneidad de una topología de red existente, identificando si esta admite aplicaciones de voz sobre IP de buena calidad entre un par de terminales. Y tratan una aplicación particular de esta medida donde la transmisión entre dos nodos de un conjunto K de terminales en una red tiene como requerimiento experimentar un máximo delay dT , donde T es el delay experimentado en un único nodo o enlace. Consecuentemente, la probabilidad de que la red cumpla con la restricción dT es proporcionada por $R(G, K, d)$. En [7] utilizan la medida de confiabilidad diámetro acotada como un modelo natural para modelar la búsqueda de información en redes P2P basado en inundación de paquetes. A partir de estos modelos, buscan elegir la profundidad de reenvío y la cantidad de conexiones simultáneas para optimizar la cantidad de información accesible a los nodos participantes.

Definición 2.4.1. ⁶ Dado un grafo probabilístico $G = (V, E)$, dos nodos $\{s, t\} \in V$ y $d > 0$, se define como $\{s, t\}$ -DCR al problema de computar la probabilidad $R(G, \{s, t\}, d)$ de que los nodos s y t son d -conexos.

Definición 2.4.2. ⁶ Dado un grafo probabilístico $G = (V, E)$ y $d > 0$, se define como All-Terminal-DCR al problema de computar la probabilidad $R(G, V, d)$ donde G es d -conexo.

Definición 2.4.3. ⁶ Dado un grafo probabilístico $G = (V, E)$, sea el subconjunto $K \subseteq V$ donde $|K| \geq 1$ y $d > 0$, se define como K -DCR al problema de computar la probabilidad $R(G, K, d)$ donde el grafo es d - K -conexo.

Desde las definiciones anteriores, resulta que un problema DCR es equivalente al problema CLR cuando $d \geq |V| - 1$. Pero, cuando $d < |V| - 1$, puede haber configuraciones donde el conjunto de terminales está conectado pero no es d -conexo. Esto da como resultado la siguiente propiedad.

Proposición 2.4.1. ⁶ Dado cada grafo probabilístico $G = (V, E)$, un conjunto $K \subseteq V$ y un entero $1 \leq d \leq d^*$, sostiene que $R(G, K, d) \leq R(G, K, d^*)$

La medida de confiabilidad diámetro acotada es una generalización de la clásica, y como consecuencia la medida de confiabilidad clásica es una cota

⁶ Ver definición en [31],[13] y [4]

superior de la diámetro acotada.

Preliminar del documento [31]. Sea Ω el espacio que representa el conjunto de todos los posibles estados del grafo probabilístico definido bajo el modelo hostil, cada $G_x \in \Omega$ tiene una probabilidad de ocurrencia:

$$P(G_x) = \prod_{x_i \in G_x} p(x_i) \prod_{x_i \notin G_x} (1 - p(x_i)) \quad (2.4.1)$$

Definición 2.4.4. Llamaremos $O_k^D(G)$ al conjunto de todos los subgrafos d - K -conexos sobre G . $O_k^D(G) = \{G_x \in G : G_x \text{ es un } d\text{-pathset}\}$

Definición 2.4.5. Un d -pathset P es un d -minpath en el sentido que si al menos un elemento de P está en falla, entonces se asegura un estado inactivo del sistema.

Definición 2.4.6. Llamaremos $\overline{O}_k^D(G)$ al conjunto de todos los subgrafos cuyo K -diámetro es mayor a d . $\overline{O}_k^D(G) = \{G_x \in G : G_x \text{ es estado de falla}\}$

Definición 2.4.7. Un d -cutset C es un conjunto de elemento cuya eliminación asegura un estado inactivo del sistema.

Definición 2.4.8. Un d -mincut C es un d -cutset minimal en el sentido que si al menos un elemento de C esta operativo entonces, se asegura un estado operativo del sistema.

Entonces, identificando los conjuntos de d -pathses o d -cutset, la medida de Confiabilidad Diámetro Acotada para el modelo hostil $R^H(G, K, d)$ queda definida (ver 2.3.1) en función de la probabilidad elemental. Dado el grafo $G = (V, E)$, el conjunto de nodos objetivo K y la cota de diámetro d :

$$R^H(G, K, d) = Prob\left(\bigcup_{P_i \in O_k^d} P_i\right) \quad (2.4.2)$$

$$R^H(G, K, d) = 1 - Prob\left(\bigcup_{C_i \in \overline{O}_k^d} C_i\right) \quad (2.4.3)$$

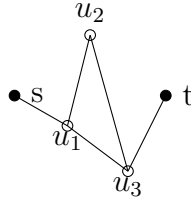
La medida DCR modifica con respecto a la clásica la definición de estado operativo e incorpora los conceptos diámetro, d -pathset y d -minpath. También mantiene el concepto de conjunto de corte, sin embargo en la práctica los

posibles cortes difieren, así como también difiere el procedimiento para identificarlos. Un cutset admite distinciones de acuerdo en como se interrumpe la comunicación, uno separando la red en dos o más componentes conexas interrumpiendo la conexión de los nodos objetivo, o bien sin separar la red evita que la comunicación ocurra en menos de d saltos.

Se exhibirá a través de ejemplos como difiere en la práctica la identificación de cortes para la medida de confiabilidad K -terminal clásica y K -DCR, tanto para el modelo arista como para el modelo hostil.

Sea el grafo de la figura 2.3, donde $G = (V, E)$ con $V = \{s, u_1, u_2, u_3, t\}$, $E = \{su_1, u_3t, u_1u_2, u_1u_3, u_2u_3\}$ y terminales $K = \{s, t\}$.

Figura 2.3: Grafo Modelo



Trabajando sobre el modelo arista y la medida de confiabilidad clásica $R(G, \{s, t\})$, se identifica los siguientes conjuntos de corte minimales. $C = \{C_1, C_2, C_3, C_4\}$ con, $C_1 = \{u_3t\}$, $C_2 = \{su_1\}$, $C_3 = \{u_1u_2, u_1u_3\}$, $C_4 = \{u_1u_3, u_2u_3\}$. Ahora, mismo modelo pero tomando la medida de confiabilidad diámetro acotada $R(G, K, d)$, el conjunto de corte va depender de la cota d tomada. Para el caso de que $d = 2$ la red nunca opera. Para $d = 3$ se agrega al conjunto C el corte $C_5 = \{u_1u_3\}$ el cual no es un conjunto separador en la confiabilidad clásica pero impide que los nodos $\{s, t\}$ se comuniquen por un camino menor o igual a la cota dada. Y para la restricción $d \geq 4$, se está bajo la condición de que todos los caminos operativos que conecten el grafo son estados operativos de las dos medidas de confiabilidad (P^d converge a P). Entonces, a partir de los conjuntos de corte encontrados y de la definición 2.2.3 se observa, que para $d = 3$ la anticonfiabilidad $Q(G, \{s, t\}) < Q(G, \{s, t\}, 3)$ y para cotas $d \geq 4$ se cumple $Q(G, \{s, t\}) = Q(G, \{s, t\}, 4)$. Observando para el caso, que la confiabilidad clásica es cota superior de la confiabilidad diámetro acotada $R(G, \{s, t\}, d) \leq R(G, \{s, t\})$ para todo $d \in \mathbb{Z}$.

Ahora parados sobre el modelo hostil y la medida de confiabilidad clásica $R^H(G, \{s, t\})$, se obtiene los siguientes conjuntos de cortes $C_H = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7\}$ con $C_1 = \{u_1\}$, $C_2 = \{u_3\}$, $C_3 = \{u_2, u_1u_3\}$, $C_4 = \{u_3t\}$, $C_5 = \{su_1\}$, $C_6 = \{u_1u_2, u_1u_3\}$, $C_7 = \{u_1u_3, u_2u_3\}$. En cambio, para la medida de confiabilidad diámetro acotada $R^H(G, \{s, t\}, d)$ la interrupción de la comunicación fuente terminal se dará según la cota dada d . Para el caso de que $d = 2$ la red nunca opera, $C_H^d = \emptyset$. Para $d = 3$ se agrega a C_H el conjunto de corte $C_8 = \{u_1u_3\}$ el cual impide que los nodos se comuniquen por un camino menor igual a la cota. Y para $d \geq 4$ estamos bajo la condición de que $C_H^d = C_H$. Consecuentemente, a partir de los conjuntos de cortes encontrados y de la definición 2.2.3 se observa, que para $d = 3$ la anticonfiabilidad $Q^H(G, \{s, t\}) < Q^H(G, \{s, t\}, 3)$ y para cotas $d \geq 4$ se cumple $Q^H(G, \{s, t\}) = Q^H(G, \{s, t\}, 4)$. Observando que para el caso planteado, la confiabilidad clásica es cota superior a la confiabilidad diámetro acotada bajo el modelo hostil $R^H(G, \{s, t\}, d) \leq R^H(G, \{s, t\})$ para todo $d \in \mathbb{Z}$.

Adicionalmente, si de la figura 2.3 analizamos la confiabilidad pero desde el punto de vista de caminos operativos del grafo. Para la medida DCR en el modelo hostil, existen dos caminos $\{P_1, P_2\}$, que conectan los nodos objetivo $K = \{s, t\}$; donde $P_1 = \{(s, u_1), (u_1, u_3), (u_3, t)\}$ y $P_2 = \{(s, u_1), (u_1, u_2), (u_2, u_3), (u_3, t)\}$. Si se considera $d = 4$ el grafo está operativo si al menos uno de los caminos opera, pues ambos caminos son factibles y se cumple $R^H(G, \{s, t\}) = R^H(G, \{s, t\}, 4)$. Si $d = 3$ sólo es factible el camino p_1 , pues p_2 consta de 4 enlaces. Si $d \leq 2$ el grafo nunca opera.

Qué sucede cuando en vez de querer comunicar nodos fuente-terminal nos interesa estudiar la comunicación d - K -conexo en un conjunto de terminales K , aquí un ejemplo sería más extenso de analizar ya que dependería de la cantidad de terminales y distancia entre nodos [4]. En el documento [13] Cancela y Petingi demuestran que, bajo un modelo clásico el problema DCR pertenece a la clase \mathcal{NP} -difícil variando estas dos variables (K y d). Entendemos que al acotar o reducir el espacio de caminos operativos la confiabilidad disminuye, por lo que la confiabilidad clásica siempre va ser un cota superior para la confiabilidad acotada (para todo $d \in \mathbb{Z}$) bajo el modelo probabilístico hostil.

2.5. Complejidad Computacional

La teoría de complejidad computacional [26] provee un grupo de técnicas para identificar si un problema dado es tan difícil de resolver como un gran número de problemas reconocidos, permite clasificar los problemas a computar de acuerdo a su dificultad inherente, y/o comparar algoritmos respecto a su eficiencia. La noción de eficiencia involucra diversos recursos computacionales necesarios para la resolución de un algoritmo, siendo el tiempo requerido el factor dominante. La llamada *función de complejidad*, expresa los requisitos de tiempo para la resolución de un algoritmo, dando para cada longitud de entrada posible, la cantidad de tiempo que necesita el algoritmo para resolver una instancia del problema de ese tamaño. Computacionalmente se reconoce una simple distinción entre algoritmos de tiempos polinomiales y algoritmos de tiempos exponencial (problemas intratables). Un algoritmo polinomial es aquel cuya función de complejidad queda acotada superiormente por una función polinómica, por otro lado un problema es intratable si es tan difícil que se requiere de un algoritmo exponencial para resolverlo.

En el documento [26] se puede encontrar distintas clases de problemas de acuerdo a su complejidad. Los llamados problemas de decisión, buscan determinar si la configuración de entrada satisface cierta lógica, obteniendo una respuesta del tipo éxito/fracaso. Estos son diferenciados en la clase \mathcal{P} , que se compone por aquellos problemas de decisión para los cuales existe un algoritmo capaz de encontrar soluciones en tiempo polinomial. Y en la clase \mathcal{NP} donde existe un algoritmo capaz de verificar una solución en tiempo polinomial. Se desprende de esto que si un problema puede resolverse en tiempo polinomial, la solución también puede verificarse en tiempo polinomial, es decir $\mathcal{P} \subseteq \mathcal{NP}$. Luego un problema de decisión \mathcal{H} es de la clase \mathcal{NP} -Difícil si todo otro problema de la clase \mathcal{NP} admite una reducción del primero. Si además \mathcal{H} pertenece a \mathcal{NP} , entonces es \mathcal{NP} -Completo. Por lo que un problema de la clase \mathcal{NP} -Completo es resoluble en tiempo polinomial, si y sólo si, se cumple $\mathcal{P} = \mathcal{NP}$. Uno de los problemas abiertos (actualmente planteado) más importantes dentro de la materia refiere a que si $\mathcal{P} = \mathcal{NP}$ o $\mathcal{P} \neq \mathcal{NP}$

Otra clase de interés son los problemas de conteo. Un problema se dice de conteo si su solución es un entero no negativo, incurriendo en la particularidad

de que en cada problema de conteo se tiene un problema de decisión asociado. En [26] indican que Valiant (en 1977) propone una clase de problema polinomial equivalente en tiempo, los llamados $\#P$ -Completo, que incluyen muchos de estos problemas de enumeración y están diseñados para reflejar la dificultad adicional de enumerar.

2.5.1. Complejidad computacional en confiabilidad de redes

En 1986 Ball a través de [1], presentó la complejidad de computar el problema de confiabilidad en redes. Dónde su cálculo tiene asociado el reconocimiento, identificación y conteo de todas las configuraciones de la red que dan lugar a un estado operativo del sistema, identificando tres clases de problemas de análisis de confiabilidad. Determinó que el análisis de la complejidad involucra conceptos utilizados en problemas de reconocimiento y optimización de subredes. A través del siguiente teorema, estableció las condiciones suficientes para determinar la intratabilidad de estimar la confiabilidad de SBS coherente

Teorema 2.5.1. *Dado un SBS coherente si cualquiera de las 5 condiciones se cumple, entonces las tres clases de análisis de la confiabilidad son \mathcal{NP} -Difícil.*

- *El problema de reconocimiento del pathset de mínima cardinal es \mathcal{NP} -Completo.*
- *El problema de reconocimiento del cutset de mínima cardinal es \mathcal{NP} -Completo.*
- *El problema de conteo de pathsets de mínimo cardinal es $\#P$ -Completo,*
- *El problema de conteo de cutsets de mínimo cardinal es $\#P$ -Completo.*
- *El problema de determinar el vector f de la confiabilidad polinomial es $\#P$ -Completo.*

Vinculando las condiciones del anterior teorema, Ball demostró que la complejidad de la medida clásica de confiabilidad (CLR) en sus tres versiones $\{s, t\}$ -terminal, K -terminal y all-terminal es \mathcal{NP} -Difícil.

En particular la dependencia entre la medida clásica de confiabilidad (CLR) y la medida diámetro acotada (DCR) permitió a Cancela y Petingi en [13] investigar la complejidad de computar DCR, haciendo distinción para algunos sub-problemas según el número de nodos terminales y diámetro. Se presenta estos resultados.

Para $d = 1$ todos los terminales deben tener una conexión directa. Por lo cual el subgrafo inducido por K -terminales debe ser un K -clique donde todos sus enlaces se encuentran operacionales, entonces se puede formular la confiabilidad como:

$$R(G, K, 1) = \prod_{\{u,v\} \in K} p(u, v) \quad (2.5.1)$$

donde $p(u, v)$ es la probabilidad de que la arista $\{uv\} \in E$.

Para $d = 2$ y las terminales $K = \{s, t\}$. En este escenario dichos nodos deben tener un enlace directo, o un nodo intermedio que los comunica. Si utilizamos el complemento de la confiabilidad, es decir si falla el enlace directo y todos los caminos de largo 2 entre s y t , la fórmula de la confiabilidad para el caso es:

$$R(G, \{s, t\}, 2) = 1 - (1 - p(s, t)) \prod_{v \in V} (1 - p(s, v)p(v, t)) \quad (2.5.2)$$

Los autores demostraron que para valores fijos $d \geq 3$ y $K = 2$ el subproblema de DCR pertenece a la clase \mathcal{NP} -Difícil. Generalizando el resultado para $K \geq 3$. Inspirados en la demostración de Cancela y Petingi, Eduardo Canale junto con otros autores en [4] prueban que el cómputo de la DCR pertenece a la clase \mathcal{NP} -Difícil para el escenario all-terminal ($K = n$) y $d \geq 2$. La demostración se realiza en partes, en primer lugar desarrollan la intratabilidad de DCR cuando $D \geq 3$ y luego la intratabilidad de $D = 2$. Se muestra el resumen de los resultados obtenidos en la siguiente tabla.

$D \setminus k$	2	3...	n(or free)
2	$O(n)$	$O(n)$	\mathcal{NP} -Difícil
3	\mathcal{NP} -Difícil		\mathcal{NP} -Difícil
\vdots			
$n - 2$	\mathcal{NP} -Difícil		\mathcal{NP} -Difícil
$n - 1$			
\vdots	\mathcal{NP} -Difícil		\mathcal{NP} -Difícil
\vdots			

Tabla 2.1: Complejidad computacional para DCR en términos del diámetro d y el número de terminales $k = |K|$.

2.5.2. Complejidad computacional en un SMBS

Como se ha mencionado en otras secciones los SMBS son SBS donde la función estructura es monótona. Al ser un caso particular de SBS heredan la complejidad ya conocida de éstos, y a su vez introducen otras específicas relacionadas con la monotonía.

En [34, 3] los autores analizan la complejidad para estos sistemas y clasifican que los problemas SMBS son intratables siendo corolarios triviales de dos resultados clásicos de SBS.

Proposición 2.5.1. *Reconocer o contar la cardinalidad mínima de un minpath o mincut es por lo menos tan difícil como computar la confiabilidad de un SMBS.*

Proposición 2.5.2. *Computar la confiabilidad de un SMBS arbitrario pertenece a la clase de problemas NP-Difícil.*

Por medio de una enumeración exhaustiva un cutset (o pathset) se obtiene la expresión exacta de la confiabilidad de un SMBS obtenida como:

$$r = \sum_{x \in \{0,1\}^m: \phi(X)=1} P(X = x) = 1 - \sum_{x \in \{0,1\}^m: \phi(X)=0} P(X = x) \quad (2.5.3)$$

Pero contar o hallar la cardinalidad exacta del conjunto de todos los cutset (o pathset) es en general una exponencial de tamaño m , el cual es prohibitivo desde el punto de vista computacional. Lo que refuerza la afirmación de que la confiabilidad de un SMBS es \mathcal{NP} -Difícil. Siendo lo anterior una de las principales razones que lleva a estudiar la aproximación estadística para la confiabilidad de un SMBS arbitrario.

Adicionalmente, los autores plantearon los siguientes lemas donde el primero es un contraste de la proposición 2.5.2 y es utilizado en [3] como parte de la generalización de RVR, y el segundo enuncia que la monotonía de los SMBS se preserva bajo condiciones de falla y/o operación de algunos de los componentes.

Lema 2.5.1. *Dado un SMBS arbitrario (S, ϕ, p) podemos encontrar un mincut $S' \subseteq S$ en un orden cuadrático de evaluación respecto a $m = |S|$*

Demostración. Ver [3]. Sin pérdida de generalidad, el autor asume que S es un cutset (de otra forma por monotonía se tendría que $\phi = 1$ y no existiría

mincut en el sistema). Buscando el primer componente tal que $S - \{i\}$ no es un cutset. Si no existe tal componente, S en si mismo es un mincut (aplicando m evaluaciones de la función estructura) y S es la evidencia verosímil de la sentencia. Por otro lado, un nuevo paso del mismo proceso toma lugar para el cutset $S_1 = S - \{i\}$. Como m es finito, cada etapa requiere al menos m evaluaciones y hay por lo menos m estados, el proceso en total requiere no menos de m^2 evaluaciones. \square

El Lema 2.5.1 es constructivo y proporciona una interacción con la lógica proposicional. Sin embargo, el autor no sabe si el algoritmo presentado es óptimo.

Lema 2.5.2. *Considere un SMBS (S, ϕ, p) y un subconjunto arbitrario de componentes C , donde algunos de los elementos de C falla y otros operan. Entonces el sub-sistema $(S - C, \phi', p')$ es un SMBS, donde p' es el vector original pero con los índices de C borrados y $\phi' = \phi|_{S-C}$ es la sub-estructura que setea los bit de C correspondientes (setea en 1 los componentes en operación y a 0 los en falla)*

Demostración. la monotonía se hereda desde el SMBS original, ya que el conjunto parcialmente ordenado $(\{0, 1\}^m)$ se mantiene para subconjuntos arbitrarios de $\{0, 1\}$. \square

Definición 2.5.1. Bajo la condición del Lema 2.5.2, la sub-estructura de ϕ asociada al subconjunto C es denotada ϕ_C

El método RVR basa fuertemente su eficiencia computacional en encontrar un cutset (Lema 2.5.1), en el análisis de pequeños subsistemas inspirados en el Lema 2.5.2 y la definición 2.5.1. Entonces mejorando la eficiencia computacional del algoritmo del Lema 2.5.1 se debería observar mejoras en los aspectos recursivos de RVR. En [35] el autor anuncia el Lema 2.5.3 llamado *Minpath*, formalizando una optimización del Lema 2.5.1.

Lema 2.5.3. *Dado un SMBS arbitrario (S, ϕ, p) podemos encontrar un mincut $S' \subseteq S$ en una orden lineal de evaluaciones respecto a $m = |S|$. Ídem, para minpath.*

Demostración. Ver [35]. El autor fundamenta el orden lineal encontrado a partir de la definición de minpath (o mincut) de un SMBS. Asume que $x = \bar{0}_m$

es un cutset (de otra forma por monotonía se tendría que $\phi = 1$ y no existiría mincut en el sistema). Entonces, de 1 hasta m se recorren (o se encienden) los elementos $\{e_i\}$ de S buscando los $x(i) = x + \{e_i\}$ que verifican $\phi(x(i)) = 0$. Si el elemento no verifica la condición, es apagado de x y guardado en S' . Por monotonía y construcción, el conjunto de elementos que componen S' es un cutset y por absurdo se demuestra que es un mincut. Al finalizar el algoritmo S' en sí mismo es un mincut, encontrado en m evaluaciones de la función estructura. Como m es finito, se obtiene en un orden lineal. \square

El autor en [35] continúa con la exploración de cortes que mejoren la eficiencia computacional de RVR, sugiriendo analizar elementos irrelevantes o esenciales, pero dada la siguiente proposición, concluye que no existen algoritmos computacionalmente eficientes disponibles para determinar componentes irrelevantes o estados operativos más baratos.

Proposición 2.5.3. *Dado un SMBS arbitrario (S, ϕ, p) , y sea $s \in S$ un componente arbitrario. El problema de reconocer si s es irrelevante o no pertenece a un problema de clase \mathcal{NP} -Difícil .*

Por otro lado, el autor en [35] anuncia la proposición y el teorema siguiente, preguntándose si existe un estado operativo (o falla) de máxima probabilidad. Bajo la hipótesis de modelo homogéneo, la respuesta es equivalente a encontrar el minpath con mínima cardinalidad. Desafortunadamente, esto es computacionalmente ineficiente, a menos que $\mathcal{P} = \mathcal{NP}$.

Proposición 2.5.4. *La determinación de un pathset con máxima probabilidad en un SMBS arbitrario pertenece a la clase \mathcal{NP} -difícil.*

Teorema 2.5.2. *La determinación de un cutset con máxima probabilidad en un SMBS arbitrario pertenece a la clase \mathcal{NP} -difícil.*

El autor concluyó que este último teorema tiene un impacto negativo en el entendimiento de SMBS, ya que el teorema 2.5.2 descarta cualquier posibilidad de existencia de la teoría de máximo flujo en un SBS, a no ser que se de que $\mathcal{P} = \mathcal{NP}$.

2.6. Cálculo de Confiabilidad

Las posibilidades de cálculo existentes para determinar el valor de la confiabilidad son diversas y dependen de las características del problema en interés. Uno de los principales aspectos condicionantes al calcular la medida de confiabilidad es el tamaño de la red, si es pequeña es posible aplicar métodos de cálculo exacto, mientras que para redes grandes deben considerar en general métodos aproximados. Como fue tratado en 2.5 el problema de calcular la confiabilidad pertenece a la clase \mathcal{NP} -difícil, el tamaño de la red influye directamente sobre el tiempo computacional de los métodos exactos, y salvo en estructuras puntuales, el cálculo exacto es computacionalmente prohibitivo. En las siguientes secciones se analizan métodos de cálculo existentes.

2.6.1. Métodos de Enumeración Completa

El método de Enumeración Completa, también denominado Fuerza Bruta es el método más elemental de cálculo de confiabilidad. Consiste en enumerar todos los estados posibles de G , el valor de la confiabilidad se obtiene sumando la probabilidad de ocurrencia de aquellos que corresponden a un estado operativo global de la red. Esto es equivalente a calcular la probabilidad de todos los posibles pathsets o cutset (véase la ecuación 2.5.3) para un SMBS o SBS, logrando tiempos de cómputo prohibitivos en los peores casos.

En el documento [34] se cita al autor Kishor Trivedi, exponiendo tres métodos de cálculo exacto el Método de Factorización, Principio de inclusión y exclusión, y Suma de productos disjuntos. Recalcando sus características y su posible extensión a SBS.

2.6.1.1. Método de Factorización

El método exacto de factorización, definido en el año 1958 por Moskowitz, propone una fórmula recursiva aplicada sobre subgrafos del grafo original, y cuya topología se obtiene de condicionar G al estado de un enlace e . El cálculo exacto de la medida de confiabilidad $R(G)$, para cualquier enlace e se basa en la siguiente descomposición pivotal:

$$R(G) = p_e R(G - e) + (1 - p_e) R(G * e), \quad (2.6.1)$$

siendo $G - e$ la eliminación de e en el grafo G , y $G * e$ su correspondiente contracción.

Como etapa de pre-procesamiento, normalmente se realizan reducciones serie-paralelas. El rendimiento global de un método de factorización depende de esta etapa de preprocesamiento, y la correcta selección del enlace e , llamado pivote. Satyanarayana y Chang en [37] proporcionaron un método para determinar el pivote usando la teoría de la dominación en grafos. Los autores muestran que los enlaces irrelevantes se pueden encontrar de manera eficiente en el modelo confiabilidad K -terminal. En [10] y [11] proponen un algoritmo de tiempo polinomial para detectar y eliminar enlaces irrelevantes en un modelo clásico, mostrando resultados experimentales que conducen a importantes ganancias computacionales, cuando este algoritmo que preserva la confiabilidad diámetro acotada se integra dentro de un enfoque de factorización exacto basado en el teorema de Moskowitz.

Sin embargo, en un SBS arbitrario determinar componentes irrelevantes pertenecen a la clase NP -Difícil, como se establece en 2.5. El núcleo del método de factorización en un SBS es la siguiente identidad:

$$r_S = p_i r_S|_{i=1} + (1 - p_i) r_S|_{i=0}, \quad (2.6.2)$$

siendo $r_S|_{i=1}$ y $r_S|_{i=0}$ el condicional SBS a la operación/falla del i -ésimo componente.

2.6.1.2. Principio de Inclusión y Exclusión

El método requiere calcular la probabilidad operativa de cada uno de los minpaths (o mincuts) $A = \{x_1, \dots, x_s\}$ del grafo.

Por consecuente, sabemos que $\phi(x) = 1$ si y solo si $x \geq x_k$ para algún $k \in \{1, \dots, s\}$. La probabilidad de la unión de este evento es la fórmula de Poincaré

$$P(\phi(X) = 1) = \sum_{i=1}^s (-1)^{s-1} \sum_{A_i \subseteq A: |A_i|=i} P\left(\bigcap_{j \in A_i} \{X \geq x_j\}\right). \quad (2.6.3)$$

Claramente, el número de minpaths puede ser exponencial, por lo que el cálculo de la confiabilidad mediante la expresión 2.6.3 se torna computacionalmente prohibitivo, a menos que la mayoría de los términos sean nulos.

2.6.2. Métodos aproximados

Como ya se ha hecho mención la obtención del valor exacto de la medida de confiabilidad para distintos modelos implican costos computacionales altos y en general de orden exponencial. Una alternativa a esta evaluación es estimar el valor a través de métodos de aproximación, que en general pertenecen a la familia del método Monte-Carlo, caracterizado por la simulación de múltiples estados del sistema logra la medidas deseada por intermedio de estimadores estadísticos, media, varianza, etc.

El método de Monte Carlo Crudo, ejemplar básico de dicha familia, es muy sencillo de comprender y de llevar a cabo, pero no ofrece márgenes de errores suficientemente pequeños en la mayoría de los casos, por lo que en escenarios de redes altamente confiables se deben utilizar técnicas de reducción de varianza.

Inspirado por el método Monte Carlo (CMC) en [8] proponen un esquema recursivo de reducción de la varianza (RVR-MC) específicamente diseñado para el problema planteado, obteniendo estimaciones más eficiente que en CMC. El autor declara que el método RVR-MC en su recursividad va trasformando el problema original en problemas de evaluación de anti-confiabilidad para redes más pequeñas. El proceso termina cuando el sistema bajo estudio resulta operativo o en falla, independientemente del estado de sus componentes. En el documento [3] se extiende el método RVR para sistemas binarios estocásticos y sistemas binarios estocásticos monótonos, y en [6] se adapta RVR para el modelo de confiabilidad en nodos.

Apoyándonos en los documentos nombrados en las siguientes sub-secciones se describen el método de Monte Carlo y una generalización de RVR, que nos permitirá calcular la confiabilidad de un SMBS arbitrario, para luego extenderlo a nuestro modelo de estudio.

2.6.2.1. Métodos Monte Carlo

Cuando hablamos de estimar una medida, donde no es factible la obtención de un valor exacto el método de Monte Carlo es un digna herramienta computacional a ser tenida en cuenta. La idea general es estimar el valor exacto mediante simulaciones de distintos estados del sistema, tomando N realizaciones independientes de esa simulación estima estadísticamente el rendimiento del sistema [18, 23, 30].

Nomenclatura.

X	variable aleatoria (v.a. discreta o continua)
$F_X(x) = P(X \leq x)$	distribución de probabilidad de X
$E(X)$	esperanza de X
$Var(X) = E(X - E(X))^2$	varianza de X
$\sigma = \sqrt{Var(X)}$	desviación estándar
$\mathcal{X} = (X_1, \dots, X_m)$	vector aleatorio de dimensión m
$\mathcal{X}^1, \dots, \mathcal{X}^N$	N muestras de v.a. independientes con la misma distribución de X

Esquema básico del Método Monte Carlo

Supongamos que se desea calcular un cierto valor ϕ , y se conoce una variable aleatoria X con distribución F_X , tal que $\phi = E(X)$. El método de Monte Carlo en su versión más simple consiste en:

1. Sortear el conjunto $\mathcal{X}^1, \dots, \mathcal{X}^N$ de v.a iid de X
2. Calcular $S_N = \mathcal{X}^1 + \dots + \mathcal{X}^N$ la suma de los N valores sorteados
3. Calcular $\hat{X} = \frac{S_N}{N}$, se dice que \hat{X} es un estimador de ϕ
4. Calcular $\hat{V} = \frac{\sum_{i=1}^N (\mathcal{X}^i)^2 - N(\hat{X})^2}{N(N-1)}$

Por sortear entendemos que, se generan variables aleatoriamente siguiendo la distribución de probabilidad F_X . El conjunto de valores sorteado $\mathcal{X}^1, \dots, \mathcal{X}^N$ se llaman muestras (o en su equivalencia conjunto de replicación de \mathcal{X}) de tamaño N (siendo N número de repeticiones). \hat{X} es en si misma una variable aleatoria o un estimador de ϕ , o v.a de igual esperanza ϕ (ver 2.6.2.1 - Motivación del método). Resulta interesante estimar (si existe) la varianza de \hat{X} , aplicando las hipótesis de equidistribución e independencias de las observaciones \mathcal{X}^i sabemos que:

$$Var(\hat{X}) = Var\left(\frac{S_N}{N}\right) = Var\left(\sum_{i=1}^N \frac{\mathcal{X}^i}{N}\right) = \sum_{i=1}^N \frac{Var(\mathcal{X}^i)}{N^2} = \frac{Var(X)}{N} \quad (2.6.4)$$

Sin embargo en general $Var(X)$ no se conoce, como alternativa podemos empear la propia muestra para obtener un estimador de $Var(X)$, el estimador insesgado más habitual es:

$$s_N^2 = \frac{1}{N-1} \sum_{i=1}^N (\mathcal{X}^i - \hat{X})^2 \quad (2.6.5)$$

realizando manipulaciones el autor llega al siguiente estimador de la varianza

$$\hat{V} = \sum_{i=1}^N \frac{(\mathcal{X}^i)^2 - N\hat{X}^2}{N(N-1)} \quad (2.6.6)$$

Motivación del método

Es claro que el MMC no provee el valor exacto deseado si no una aproximación, con cierto error. La justificación inicial del uso de Monte Carlo [23] proviene de dos teoremas centrales de la probabilidad y la estadística, la ley débil de los grandes números y el teorema central del límite.

Sea $S_N = \mathcal{X}^1 + \dots + \mathcal{X}^N$, si existe la esperanza $\mu = E(X^i)$ entonces la Ley débil de los Grandes Números indica que todo $\varepsilon > 0$

$$\lim_{N \rightarrow \infty} Pr\left(\left|\frac{S_N}{N} - \mu\right| > \varepsilon\right) = 0 \quad (2.6.7)$$

Interpretando que si se suma N muestras independientes de X^i , la probabilidad de que la suma este lejos del valor exacto, es equivalente a estimar que μ tiene a 0 con N . Si adicionalmente existe la varianza $\sigma^2 = E((X^i - \mu)^2)$, el teorema central del límite implica que:

$$\lim_{N \rightarrow \infty} Pr\left(\frac{S_N - N\mu}{\sigma\sqrt{N}} < a\right) = (2\pi)^{-1/2} \int_{-\infty}^a e^{-x^2/2} dx \quad (2.6.8)$$

Dado que el término de la derecha es la distribución de una v.a. normal de media 0 y varianza 1, este teorema indica cuál es el comportamiento asintótico de la distribución del error cometido al emplear S_N como estimador de μ .

Ambos resultados proveen la motivación para aplicar el MMC, ya que indican que con un número suficientemente alto de experimentos es posible estimar el parámetro deseado, incurriendo en un pequeño error con alta probabilidad y permite cuantificar asintóticamente la relación entre estos dos valores (error y probabilidad) a través de la distribución normal [23].

Método Monte Carlo en SBS⁷

Dado un SBS (S, p, ϕ) , se sortean N replicas independientes X^1, \dots, X^N del vector de estado $(X = (x_1, \dots, x_m))$ de S , tal que $P(x_i = 1) = p_i$. Entonces el estimador de la confiabilidad mediante el MMC es:

$$r_{MMC} = \frac{1}{N} \sum_{i=1}^N \phi(X^i) \quad (2.6.9)$$

La varianza de estimador r_{MMC} se rige entonces por la siguiente expresión:

$$Var(r_{MMC}) = \frac{r_{MMC}(1 - r_{MMC})}{N} \quad (2.6.10)$$

En base a la motivación del método, sabemos que r_{MMC} converge casi seguramente a $E(\phi(X))$ obteniendo así una estimación de la confiabilidad del sistema. Al ser el cálculo de la varianza un estimador insesgado de un promedio de muestra se cumple que, el error estándar $\sqrt{(Var(r_{MMC}))}$ es una estimación de la desviación estándar de esta variable aleatoria.

Pseudocódigo 2.1 $r_{MCC} = MCC(S, p, \phi)$

```
1: Sum ← 0
2: for  $i = 1, \dots, N$  do
3:    $X_i \leftarrow 0$ 
4:   for  $j = 1, \dots, m$  do
5:     se sortea  $U \sim \mathcal{U}[0, 1]$ 
6:     if  $U < p_j$  then
7:        $X_i \leftarrow X_i \cup \{x_j\}$ 
8:     end if
9:   end for
10:   $Sum \leftarrow Sum + \phi(X_i)$ 
11: end for
12: return  $r_{MCC} = Sum/N$ 
```

Figura 2.4: Pseudocódigo del Método Monte Carlo⁷.

Sin embargo es preciso ser cauteloso en la aplicación práctica del MMC, ya que las implementaciones reales no verifican las hipótesis de los teoremas, por un lado las limitaciones computacionales limitan la cantidad de muestras a

⁷ Notaciones extraídas de [3]

tomar, y por otro lado las estimaciones de errores a partir del TCL solamente son validas asintóticamente, obteniendo así valores de varianza poco aceptables. Así mismo este método es base de otros algoritmos que obtienen mejor precisión, logran reducir la varianza o el tiempo de cálculo respecto a Monte Carlo Crudo empleando técnicas de muestreo más sofisticadas [30]. Existen diferentes técnicas de reducción de varianza, algunas son generales y pueden aplicarse a distintos problemas de simulación (variables antitética, muestro por importancia, variables de control). Otras son diseñadas particularmente para la confiabilidad de redes, utilizando estructura y propiedades del problema de confiabilidad para disminuir varianza o tiempo de cálculo. Veremos un ejemplo en concreto de este último, la “Reducción Recursiva de la Varianza”.

2.6.2.2. Reducción Recursiva de la Varianza

El método RVR fue presentado en el año 1995 por Héctor Cancela y Mohamed El Khadiri en el documento [8]. Éstos plantearon un esquema recursivo de reducción de la varianza basado en Monte Carlo, específicamente diseñado para el problema de confiabilidad en redes planteado, obteniendo estimaciones más eficientes que con MCC. A partir de un estimación recursiva de la anti-confiabilidad, en cada paso de la iteración el sistema bajo estudio es reducido condicionado a la ocurrencia de un cutset. Según [34], RVR ha sido introducido con éxito a SMBS gracias a 2.5.1, pero no puede ser extendido a SBS debido a que la complejidad de encontrar un mincut pertenece a la clase NP-Completo.

2.6.2.3. RVR en el Modelo Arista

Seguidamente se presenta el modelo clásico de RVR para el modelo arista. Para ello nos basamos en las introducción original del método introducido por Héctor Cancela y Mohamed el Kadiri [8] y documentos complementarios [12], [9], [18]. En los documento [6], [14] el lector puede familiarizarse con la adaptación de RVR para el modelo nodo y redes en general.

RVR basa su algoritmo recursivo en que un estimador insesgado \hat{U} es más exacto que otro estimador insesgado \hat{V} si $Var(\hat{U}) < Var(\hat{V})$. Particularmente se concentran en encontrar una variable aleatoria que en promedio tenga la misma esperanza que la v.a. insesgada pero menor varianza, consiguien-

do así un estimador más exacto que el obtenido por MCC. La v.a planteada trabaja sobre la condicional de la ocurrencia o no del corte encontrado. Y la probabilidad de que ocurra el corte es incluida como término a ser sumada al estimar la anticonfiabilidad.

Motivación del método

El autor plantea una variable basada en una muestra aleatoria Y que tiene la misma esperanza que MCC, pero menor varianza, consiguiendo así un estimador más exacto. Propone una v.a expresada en función de:

- $\Pr\{\text{“todos los componentes de un } K\text{-cutset dado están caídos en } G\text{”}\}$, y
- las $|C|$ v.a $Y_1, Y_2, \dots, Y_{|C|}$,

y demuestra a través de las siguientes proposiciones que éste es un estimador más exacto.

Proposición 2.6.1. *Sea Z la v.a definida*

$$Z = Q_c + (1 - Q_c) \sum_{i=1}^{|C|} 1_{\{U \in J_i\}} Y_i \quad (2.6.11)$$

- i índice tal que $1 \leq i \leq |C|$
- $C = \{l_1, l_2, \dots, l_{|C|}\}$ es el K -cutset dado en G .
- A_C denota el evento, "Todo los elementos de C fallan"
- Q_C es la probabilidad de que el evento A_c ocurra, indicando que todos los componentes de C están en un estado de falla.
- B_i es el evento "el elemento l_i esta encendido pero l_j falla para todo elemento $j < i$ ".
- $G_i = (G - l_1, \dots, -l_j)/l_i$ dado el evento B_i sorteado
- K_i conjunto de terminales de G_i
- desde $1 \leq i \leq |C|$, $Y_i = 1 - \phi\{X_i\}$ y X_i es un vector de estado aleatorio definido por $\Pr\{X_i = x\} = \Pr\{(X = x)|B_i\}$ para $x \in \{0, 1\}^m$.
- U es una variable aleatoria con distribución Uniforme(0,1).
- J_i con $1 \leq i \leq |C|$ es una secuencia de intervalos disjuntos de amplitud $\frac{p(B_i)}{1-Q_C}$ tal que $\bigcup_{i=1}^{|C|} J_i = [0, 1]$

Z tiene la misma esperanza que Y :

$$E(Z) = Q(G, K) = E(Y) \quad (2.6.12)$$

y menor varianza:

$$\begin{aligned} \text{Var}\{Z\} &= [Q(G, K) - Q_c] \hat{Q}(G, K) \leq Q(G, K) \hat{Q}(G, K) \\ &= \text{Var}\{Y\} \end{aligned} \quad (2.6.13)$$

Observación 1

1. $(B_i)_{1 \leq i \leq |C|}$ son una partición de \bar{A}_C en consecuencia

$$\sum_{i=1}^{|C|} \frac{\text{Pr}(B_i)}{1 - Q_C} = \sum_{i=1}^{|C|} \text{Pr}\{\mathcal{U} \in J_i\} = 1 \quad (2.6.14)$$

2. Para cada Y_i definida en Proposición 2.6.1

$$\begin{aligned} E(Y_i) &= E(1 - \phi(X_i)) \\ &= E(1 - \phi(X)|B_i) \\ &= E(Y|B_i) \end{aligned} \quad (2.6.15)$$

Y por definición de (B_i)

$$E(Y_i) = Q(G_i, K_i) \quad (2.6.16)$$

donde

- $Y_i \equiv 0$ implica que G_i siempre está operativo
- $Y_i \equiv 1$ implica que G_i no es K -conexo

En la siguiente proposición el autor muestra que si en Z cada Y_i es remplazada por una v.a Y'_i que tiene la misma exactitud y al menos la misma varianza que Y_i , la v.a Z' resultante verifica que $\text{Var}(Z') \leq \text{Var}(Z)$

Proposición 2.6.2. Sean las $|C|$ v.a $Y'_1, Y'_2, \dots, Y'_{|C|}$ tal que $E(Y_i) = E(Y'_i)$ y $\text{Var}\{Y_i\} \leq \text{Var}\{Y'_i\}$

Y'_i es independiente de \mathcal{U} , para i tal que $1 \leq i \leq |C|$

$$Z' \equiv Q_c + (1 - Q_c) \sum_{i=1}^{|C|} 1_{\{\mathcal{U} \in J_i\}} Y'_i \quad (2.6.17)$$

Entonces

$$E(Z') = Q(G, K) = E(Y) \quad (2.6.18)$$

$$\begin{aligned} Var\{Z'\} \leq Var\{Z\} &= [Q(G, K) - Q_c]\hat{Q}(G, K) \\ &\leq Q(G, K)\hat{Q}(G, K) = Var\{Y\} \end{aligned} \quad (2.6.19)$$

La proposición anterior implica que la muestra promedio \hat{Z}' , basado en Z' es más exacta que \hat{Z} .

La motivación del autor es que bajo la aplicación recursiva de las proporciones (2.6.1 y 2.6.2) se obtiene un estimado cada vez más preciso.

Para $1 \leq i \leq |C|$ define:

$$Y'_i \equiv \begin{cases} 1 & \text{if } G_i \text{ no es } K\text{-conexo;} \\ 0 & \text{if } G_i \text{ siempre esta operativo;} \\ Z_i & \text{en otro caso.} \end{cases} \quad (2.6.20)$$

Donde Z_i es una v.a construida a partir de Y_i ¹.

Y'_i satisface la Proposición 2.6.2 demostrando que $E(Z') = E(Y)$ y $Var\{Z'\} \leq Var\{Z\} \leq Var\{Y\}$.

Entonces, si para cada i en $1 \leq i \leq |C|$, Y' es una constante (0 o 1), el proceso termina. Por otro lado, si para cada v.a $Y'_i = Z_i$ el autor inspira a aplicar la misma idea de forma recursiva construyendo una v.a que tiene la misma esperanza y una varianza más pequeña. Cuando todos los sistemas resultantes están activos o inactivos, independientemente de los estados del enlace, el proceso termina.

El operador recursivo $F()$ ² expuesto por el autor construye una estimación de la anticonfiabilidad Q_k , expresando lo siguiente:

$$F(G) = \begin{cases} 1 & \text{if } G_i \text{ no es } K\text{-conexo;} \\ 0 & \text{if } G_i \text{ siempre esta operativo;} \\ Q_C + (1 - Q_C) \sum_{i=1}^{|C|} 1_{\{U \in J_i\}} F(G_i) & \text{en otro caso.} \end{cases} \quad (2.6.21)$$

¹utilizando la Proposición 2.6.1, obteniendo una v.a Z con la misma esperanza que Y pero menor varianza

²resultante del proceso descrito anteriormente

El estimador de la medida de anticonfiabilidad se define como la media de las observaciones de $F(G)$

$$Q_k(G) = \frac{\sum_{i=1}^N F^i(G)}{N} \quad (2.6.22)$$

y el estimador insesgado de la varianza,

$$Var(Q_k(G)) = \frac{\sum_{i=1}^N (F^i(G) - Q_k(G))^2}{N(N-1)} \quad (2.6.23)$$

Implementación de RVR

El procedimiento principal de RVR consiste en un bucle que llama N veces al operador recursivo $F(G, K)$, y recopila los valores finales para una estimación de la anticonfiabilidad Q_k y su varianza.

Pseudocódigo 2.2 $F(G, K)$

- 1: **if** G es simple opertivo **then**
 - 2: return $Q = 0$
 - 3: **end if**
 - 4: **if** G no es K -conexo **then**
 - 5: return $Q = 1$
 - 6: **else**
 - a) Encontrar conjunto de corte: $C = \{l_1, \dots, l_{|C|}\}$, según criterio
 - b) Calcular $q_c = \prod_{j=1}^{|C|} (1 - p_j)$
 - c) Sortear \mathcal{U} desde una Uniforme(0, 1), seleccione J_i tal que $\mathcal{U} \in J_i$
 - d) Construye $G_i = (G - l_1 - l_2 - \dots - l_{i-1}) * l_i$
 - e) Paso recursivo retorne $Q = q_c + (1 - q_c)RVR(G_i, K)$
 - 7: **end if**
-

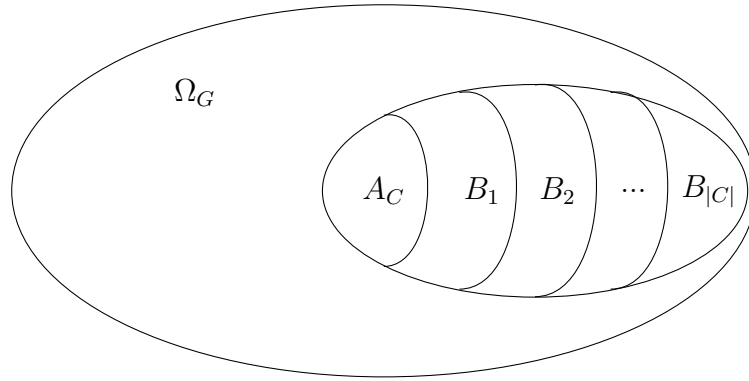
Figura 2.5: Pseudocódigo del operador recursivo $F()$ para RVR

RVR en analogía a MCC sorteas una muestra de G independiente, pero en vez de realizar sorteos independientes para el estado de cada elemento, genera estados dependientes a cierto cutset. Para esto redefine o divide el espacio de

vectores de estados Ω_G del grafo, en términos de cutset, permitiendo al autor calcular la probabilidad de que un vector de estado X pertenezca o no a un mincut de la red, asociado a la probabilidad de falla de los componentes del mincut.

Es de nuestro interés ver como el espacio de vectores de estados de un grafo es particionando al aplicar RVR. Sea un grafo $G = (V, E)$ y su espacio $\Omega_G = \{0, 1\}^m$ con m el número de enlaces. La v.a. definida bajo el método RVR trabaja sobre un espacio reducido de Ω_G (vectores de estados condicionados a la ocurrencia o no de un cutset C dado) definiendo una partición y sus distintos componentes, en la Figura 2.6 se exhibe una representación grafica de la misma.

Figura 2.6: Partición de Ω_G basada en mincut



La probabilidad de que un vector de estado X , pertenezca a una componente de la partición (A_C o B_i , ver 2.6.1) se asocia a la probabilidad de falla de los elementos de C :

1. $Q_C = P_{A_C} = P(X \in A_C) = \prod_{j \in C} (1 - p_j)$, con p_j la confiabilidad elemental del j -ésimo elemento de C .
2. $P(X = i) = P(B_i | \bar{A}_C) = P(B_i) / (1 - Q_C) = \prod_{k=1}^{i-1} p_k (1 - p_k)$, donde los elementos $1, 2, \dots, i - 1$ de C están operativos y el i -ésimo falla.

Entonces, una vez identificado un cutset, en base a lo anterior se construye la v.a. asociada a todos los posibles estados que puede tener el grafo definido bajo las particiones (A_C y B_i), para cada muestra se calcula la probabilidad de ocurrencia definida bajo los intervalos J_i , se sortea \mathcal{U} indicando cuál de los intervalos J_i fue factible, pasando a hallar el nuevo subgrafo G_i donde son fijados los estados correspondientes al sorteo.

Notar que, en cada paso recursivo se considera un único componente de la partición, dado a que en cada sumando $\sum_{i=1}^{|C|} 1_{\mathcal{U} \in J_i} F(G_i)$ un único valor de la indicatriz es válido. El subgrafo G_i es tal que fallan los primeros $i-1$ elementos del cutset, y el i -ésimo está operativo. Al siguiente paso de la iteración, el grafo G_i es condicionado a un nuevo corte, pasando a fijar más elementos del grafo, y así sucesivamente hasta obtener un estado no operativo del sistema, o un subsistema que a lo largo de las iteraciones fue disminuyendo la cantidad de componentes hasta obtener un estado operativo.

2.6.2.4. RVR en SMBS

Entonces, tomando como referencia la sección anterior y las notaciones de los documentos [34, 3], se estima la anticonfiabilidad de un SMBS a través del método RVR. Considere el vector aleatorio $X = (x_1, \dots, x_m)$ tal que x_1, \dots, x_m son variables de Bernoulli independientes y $P(x_i = 1) = p_i$.

Nomenclatura

- Mincut $C = (a_1, \dots, a_{|C|})$ del sistema.
- A_C denota el evento, "Todo los elementos de C fallan"
- $q_C = P(A_C) = \prod_{j \in C} (1 - p_j)$.
- $A_i = \{a_1, \dots, a_i\}$
- B_i es el evento "el elemento a_i esta encendido pero a_j falla para todo elemento $j < i$ ".
- $P(B_i) = p_i \prod_{j < i} (1 - p_j)$.
- V es una v.a discreta tal que, $P(V = i) = P(B_i | \bar{A}_C) = P(B_i) / (1 - q_C)$ donde \bar{A}_C es el complemento de A_C
- $Z_i = 1 - \phi_{B_i}$ es la confiabilidad del sistema condicionado al evento B_i

Se considera la siguiente variable aleatoria:

$$Z = q_c + (1 - q_c) \sum_{i=1}^{|C|} 1_{\{V=i\}} Z_i \quad (2.6.24)$$

Al fijar el estado operacional o en falla de los componentes B_i , un nuevo SMBS es obtenido (al restringir una función monótona se obtiene a cambio otra monótona). A través de un cálculo directo se puede ver que Z es un estimador insesgado de la anticonfiabilidad, y adicionalmente el valor medio de

Z tiene menor varianza que el valor obtenido por MCC, obteniendo así valores más exacto. La demostración es similar a la expuesta en el documento [8] sustituyendo la red por un SMBS, el autor no realiza ninguna especificación más que la monotonía y la existencia de un cutset.

De la expresión 2.6.24 el autor concluye el siguiente operador recursivo de la anticonfiabilidad:

$$Q(X) = \begin{cases} 0 & \text{Si } \phi = 1; \\ 1 & \text{Si } \phi = 0; \\ q_{C'_S} + (1 - q_{C'_S}) \sum_{i=1}^{|C'_S|} 1_{V=i} Q(B_i) & \text{en otro caso.} \end{cases} \quad (2.6.25)$$

Del conjunto de componentes S' , un mincut C'_S es recursivamente encontrado para cada subsistema utilizando el Lema 2.5.1. Observe que la recursividad fija la falla de algunos elementos del conjunto de corte, así como el primer elemento i que está operativo. En otras palabras es un proceso con memoria, que guarda durante el paso recursivo m la condición encontrada del subsistema del paso $m - 1$. Los sucesivos subsistemas tendrán cada vez menos elementos, garantizando el fin de la recursividad.

El estimador de la medida de anticonfiabilidad se define como la media de las observaciones, siendo $\{Q_1, \dots, Q_N\}$ muestras independientes e idénticamente distribuidas de $Q(x)$:

$$q_{RVR} = \frac{1}{N} \sum_{i=1}^N Q_i \quad (2.6.26)$$

En el Pseudocódigo 2.3 se resumen los pasos a seguir para el cálculo de la estimación de la anticonfiabilidad con el método RVR para un SMBS:

Pseudocódigo 2.3 $Q = RVR(S, \phi, p)$

```

1: if  $\phi = 1$  then
2:   return  $Q = 0$ 
3: end if
4: if  $\phi = 0$  then
5:   return  $Q = 1$ 
6: else
   a) Encontrar conjunto de corte:  $C = \{a_1, \dots, a_{|C|}\}$ , según criterio a elección
   b) Calcular  $q_c = \prod_{j=1}^{|C|} (1 - p_j)$ 
   c) Sortear  $v$  de  $V$ : con distribución  $P(V = i) = P(B_i | \bar{A}_C), 1 \leq i \leq |C|$ 
   d) Construir  $(A_i, p')$ 
   e) Paso recursivo  $Q = q_c + (1 - q_c)RVR(\phi_{B_i}, S - A_i, p')$ 
7: end if

```

Figura 2.7: Pseudocódigo del método RVR en SMBS

El método principal consiste en un loop de N muestras del operador $Q(x)$, y la acumulación de este resultado para la estimación de q_{RVR} . Del paso uno al seis se verifica la condición de salida, condicionado a que el sistema cumpla la lógica establecida; En caso contrario se puede encontrar un conjunto de corte en el sistema. En los pasos 6a y 6b, se elige el conjunto de corte C y se computa su probabilidad. Del paso 6c al 6d, se sortea \mathcal{U} para determinar J_i a través de todas las posibilidades de J_k (donde k varia entre $1 \leq k \leq |c|$), obteniendo así el sub-sistema $S - A_i$, con el cual se efectuará una llamada recursiva con este nuevo sub-sistema.

Capítulo 3

Metodología

En el pasado Capítulo 2 se presentó métodos de cálculo exacto y métodos de cálculo aproximado para obtener la medida de confiabilidad diámetro acotada en el modelo hostil. Los métodos exactos permiten determinar el valor de la medida con un error nulo pero en determinados contextos su aplicación está asociada a un costo computacional prohibitivo. Por otro lado, aunque los métodos aproximados también están condicionados al tamaño de la red, estos permiten estimar la medida de confiabilidad a un costo computacional inferior al que impone el cálculo exacto.

Nos preguntamos, ¿cómo podemos disminuir el costo computacional del método aproximado RVR? La reducción recursiva de varianza es un algoritmo de resolución recursiva, que parte de su eficiencia computacional está intrínsecamente relacionada con la estrategia de selección de mincuts. Estos cortes de tamaño mínimo deben ser encontrados en cada etapa recursiva del método, ligando el tiempo de cómputo global de RVR a un método de separación de grafos. Reduciendo la pregunta inicial a ¿Cuál es el rendimiento del método RVR ante distintas estrategias de selección de cortes?

Este Capítulo integra la parte teórica planteada en el Capítulo 2, y se centra en el desarrollo de distintas estrategias de separación de grafos que serán utilizadas en el método aproximado RVR, para luego evaluar cómo es afectada la eficiencia del método. Se busca identificar cortes que mejoren globalmente el algoritmo tanto en tiempos de cómputo como en precisión, o que individualmente destaquen una particularidad.

Las distintas estrategias serán llevadas a la práctica en el Capítulo 4, donde se introducen conceptos que nos ayudarán a evaluar la eficiencia de RVR

comparando los distintos conjuntos de cortes elegidos. La práctica se emplea sobre grafos conocidos dentro de la literatura como son el dodecaedro, grafo de Antel, ARPANET, y otros simples como puente o grilla.

3.1. Implementación de RVR para DCR en el modelo hostil

Se describe brevemente los lineamientos tomados al implementar el Pseudocódigo 2.3 en el contexto de confiabilidad diámetro acotada para el modelo hostil (Anexo 4 muestra un ejemplo gráfico de ejecución). El factor inicial que se tuvo en cuenta fue que en cada iteración de RVR debemos encontrar un corte, para luego sortear el estado de sus elementos. Algunos de estos elementos se fijan como falla, así como se fija el primer elemento i que está operativo (ver 2.6.2.4). Ante la búsqueda de un corte nos preguntamos: ¿Cómo recorrer los elementos del grafo? ¿Qué orden tomar? ¿Qué implica fijar el estado de un elemento en el modelo hostil? ¿Cómo afecta ante una nueva iteración del método?

Teniendo en cuenta que al apagar nodos no terminales (probabilidad de operación en cero) las aristas que le inciden deben ser apagadas, y al encender nodos no terminales (probabilidad de operación en uno) se debe prender todas las aristas incidentes en previa operación que interconectan a nodos previamente operativos o nodos terminales, y/o apagar las aristas que están en falla, imponiendo un orden al sistema. Esto condiciona la implementación del Algoritmo 2.3 según el conjunto cutset encontrado, por lo que se debió identificar si éste está compuesto por nodos no terminales y aristas, sólo aristas, o sólo nodos no terminales. Esta identificación fue clave, y se extendió a las estrategias de separación planteadas, lo que llevó a definir un orden en la búsqueda de un cutset inicial o en la quita de elementos del sistema.

3.2. Estrategias de separación

En [8] el autor remarca que la eficiencia del método RVR depende del cutset tomado en cada iteración recursiva del método, el cutset elegido en este caso fue el conjunto de enlaces incidentes a un nodo terminal. Adicionalmente recomienda tomar conjuntos de cortes con grandes valores de anticonfiabilidad Q_c , y como trabajo futuro proponen cuantificar el efecto de distintos cutset.

En el documento [3] plantean el Lema 2.5.1 donde para un SMBS se halla un mincut en un número cuadrático de evaluaciones de la función estructura ϕ , el cual jugó un papel importante en la eficiencia del método RVR promoviendo un punto de partida para mejorar en éste indicador. En [35] el autor anuncia a partir del Lema 2.5.3 mejoras en los recursos empleados por el Lema 2.5.1 promoviendo encontrar un mincut en un número lineal de evaluaciones.

A su vez el autor en [40] plantea la determinación de un cutset de máxima probabilidad, obtenido a través del teorema de Ford Fulkerson para modelos $s - t$ terminal clásicos, tomando valores $-\log(q_i)$ en la capacidades de los enlaces.

El autor del documento [27] analiza cómo extender el teorema Flujo Máximo - Corte Mínimo de Ford Fulkerson para camino de diámetro acotados. Para eso caracterizó las brechas entre la versiones continuas y enteras, dando un algoritmo de aproximación primal-dual, cuya relación de aproximación cubre estas brechas. Éste también nota que la versión continua del problema es polinómica para cualquier cota, pero encontrar valores enteros de Flujo Máximo - Corte Mínimo es polinomial únicamente cuando los límites son hasta tres ($d \leq 3$).

En base a lineamientos de trabajos previos en esta sección abordaremos distintas estrategias de separación de grafos. Las estrategias elegidas se enfocan en crear cortes a partir de la estructura abstracta de un SMBS, en jugar con la singularidad del camino acotado de nuestro sistema, y/o problemas de optimización como encontrar los cortes de máxima probabilidad. Los cortes centrados en la estructura abstracta de un grafo nos permite trabajar con distintos SMBS sin realizar distinción entre ellos, por otro lado explotar la singularidad de nuestro problema introduciendo información relevante nos proyecta a mejorar la eficiencia del método RVR en forma particular.

¿Qué tanto se puede mejorar la eficiencia de RVR sin afectar la precisión o

exactitud del algoritmo? En el Capítulo 4 juzgaremos las estrategias planteadas en relación a los resultados en performance obtenidos, basadas en supuestos de índole: “ métodos de orden lineal de ejecución deberían mejorar la eficiencia global de RVR”, “ métodos que buscan cortes de máxima probabilidad en cada paso recursivo debería llevar a obtener una solución en menos cantidad de muestras”, “ métodos que incluyen información de la estructura o información particular de diámetro acotado deberían conducir a obtener una solución en menos cantidad de muestras”.

3.2.1. Corte Lineal

Dado un SMBS en [35] el autor propone un algoritmo llamado Minpath que encuentra un mincut en una cantidad lineal de evaluaciones de la función estructura, abarcando a cualquier SMBS lo que independiza la estructura del sistema bajo estudio.

Pseudocódigo 3.1 $x = \text{Minpath}(S, \phi, p)$

```

1:  $m \leftarrow |S|$ 
2:  $x \leftarrow \vec{0}_m$ 
3: for  $i = 1, \dots, m$  do
4:    $x(i) \leftarrow x + \delta_i$ 
5:   if  $\phi(x(i)) = 0$  then
6:      $x \leftarrow x(i)$ 
7:   end if
8: end for
9: return  $x$ 

```

Partiendo de un sistema en falla $\phi(\vec{0}_m) = 0$, el cual es un cutset válido para cualquier SMBS, el algoritmo recorre de 1 hasta m los componentes en orden. En cada paso i se prende el elemento δ_i generando una nueva palabra binaria $x(i) = x + \delta_i$, se pasa a verificar si el nuevo estado generado sigue siendo un estado de falla (punto 5), en caso afirmativo se deja prendido (punto 6), en caso contrario el elemento elegido se deja apagado. Al finalizar el recorrido todos los elementos que encienden el sistema están apagados, definiendo un mincut.

Respecto a la eficiencia, el algoritmo halla un mincut en un número lineal ($m \leftarrow |S|$) de evaluaciones, mejorando el orden cuadrático del Lema 2.5.1.

Implementación DCR para el modelo hostil

En las siguientes secciones se presenta la implementación del pseudocódigo 3.1 y una adaptación de éste (ver 3.3) para el modelo bajo estudio, tomando en cuenta los lineamientos planteados en Sección 3.1.

Pseudocódigo $C = \text{CorteN}(G, K, d)$ Se divide en dos partes, bloque de búsqueda (encontrar el cutset inicial) y bloque de resolución (pauta un orden en el recorrido del cutset inicial):

- Bloque de búsqueda (puntos del 1:12): Se buscan todos los nodos no terminales (*cortev*) y aristas (*cortee*) con probabilidad elemental distinto a 1, estos compondrán el cutset inicial C_0 . Tomando un grafo auxiliar $g_i = G - C_0$ se apagan los elementos respetando el orden esbozado y se verifica si g_i es d - K -conexo.
- Bloque de resolución: En caso de que el sistema esté operativo se entrega un corte vacío. En caso de que resulte en falla, se divide el script según el conjunto inicial C_0 encontrado:
 - si $C_0 = \{\text{cortee}; [\cdot]\}$ es únicamente aristas, se encienden de a una y se busca mincut. Véase puntos desde el 16 al 27.
 - si $C_0 = \{[\cdot]; \text{cortev}\}$ es únicamente nodos no terminales, se encienden de a uno y se busca mincut. Véase puntos desde el 28 al 41.
 - Si $C_0 = \{\text{cortee}; \text{cortev}\}$ conjunto formado por aristas y nodos no terminales. Se opta por recorrer los elementos del grafo primero encendiendo nodos no terminales y luego las aristas restantes, recorriendo así todo el conjunto C_0 . Véase puntos desde el 42 al 65.

Pseudocódigo 3.2 $C = \text{CorteN}(G, K, d)$

```
1:  $cortee = cortev = \emptyset$ ,  $C = \{\emptyset, \emptyset\}$ ,  $gi = \{G_e, G_v\} = G$ ,  $p = [p_{G_e}, p_{G_v}]$ 
2: for  $j_e = 1, \dots, |G_e|$  do
3:   if  $p_{G_e(j_e)} < 1$  then
4:      $cortee = [cortee; G_e(j_e)]$ , y elimino aristas de  $gi$ 
5:   end if
6: end for
7: for  $j_v = 1, \dots, |G_v|$  do
8:   if  $p_{G_e(j_v)} < 1$  then
9:      $cortev = [cortev : G_e(j_v)]$ , y elimino nodo de  $gi$ 
10:  end if
11: end for
12: if  $gi$  es  $d$ - $K$ -Conexo then
13:   No existe corte
14: else
15:    $i_e = i_v = 0$ 
16:   if  $\text{isempty}(cortev)$  then corte de enlaces
17:     while  $i_e \leq |cortee|$  do
18:       Prendo enlace  $cortee(i_e)$  en  $gi$ ;
19:       if  $gi$  no es  $d$ - $K$ -conexo then
20:         Elimino arista de  $cortee(i_e) = \emptyset$ ;
21:          $i_e = i_e - 1$ ;
22:       else
23:         Apago arista  $cortee(i_e)$  de  $gi$ ;
24:       end if
25:        $i_e = i_e + 1$ ;
26:     end while
27:      $C = \{cortee; \emptyset\}$ 
28:   else if  $\text{isempty}(cortee)$  then corte de nodos
29:     while  $i_v \leq |cortev|$  do
30:       Prendo nodo  $cortev(i_v)$  en  $gi$ ;
31:       también se prenden aristas que llegan a  $cortev(i_v)$ ,
32:       condicionado que el nodo extremo este prendido
33:       if  $gi$  no es  $d$ - $K$ -conexo then
34:         Elimino nodo de  $cortev(i_v) = \emptyset$ ;
35:          $i_v = i_v - 1$ ;
36:       else
37:         Apago nodo  $cortev(i_v)$  de  $gi$ ;
38:       end if
39:        $i_v = i_v + 1$ ;
40:     end while
41:      $C = \{\emptyset; cortev\}$ 
```

Pseudocódigo 3.2 $C = \text{CorteN}(G, K, d)$

```
42:   else
43:     Se prenden primero nodos y después aristas.
44:     while  $i_v \leq |\text{cortev}|$  do
45:       Prendo nodo  $\text{cortev}(i_v)$  en  $gi$  y aristas adyacentes,
46:       condicionadas a que  $p_e == 1$  y nodo extremo prendido.
47:       if  $gi$  no es  $d$ - $K$ -conexo then
48:         Elimino nodo de  $\text{cortev}(i_v) = []$ ;
49:       else
50:         Apago nodo  $\text{cortev}(i_v)$  de  $gi$ ;
51:       end if
52:        $i_v = i_v + 1$ ;
53:     end while
54:     while  $i_e \leq |\text{cortee}|$  do
55:       Prendo enlace  $\text{cortee}(i_e)$  en  $gi$ ,
56:       si nodos que lo compone están prendidos;
57:       if  $gi$  no es  $d$ - $K$ -conexo then
58:         Elimino arista de  $\text{cortee}(i_e) = []$ ;
59:          $i_e = i_e - 1$ ;
60:       else
61:         Apago arista  $\text{cortee}(i_e)$  de  $gi$ ;
62:       end if
63:        $i_e = i_e + 1$ ;
64:     end while
65:      $C = \{[\text{cortee}], [\text{cortev}]\}$ ;
66:   end if
67: end if
68: return  $C$ 
```

Las estrategias de cortes planteadas en las secciones 3.2.2, 3.2.3 y 3.2.4 no cuentan con una base matemática formal que nos permita asegurar un mincut, pero sí distintos enfoques para seleccionar un cutset inicial. Bajo la hipótesis y definición de monotonía se adaptó el pseudocódigo 3.2, obteniendo el descripto en el pseudocódigo 3.3 donde se valida la obtención de un mincut, o se reduce un cutset dado hasta encontrar un mincut. Se recalca que 3.3 utiliza la misma lógica que 3.1 para validar que se obtuvo un mincut, pero las condiciones iniciales y orden lineal de cómputo difieren así como su implementación.

Pseudocódigo $x = \text{cutsetCorteN}(S, \phi, p, y)$ Dado un SMBS y un cutset válido (y) el algoritmo obtiene un mincut. A partir del cutset ingresado el algoritmo recorre sus componentes apagados y en cada paso i prende el elementos δ_i , verificando si el nuevo estado generado sigue siendo un cutset, en caso afirmativo se pasa a prender otro componente, en caso contrario el elemento se deja apagado, obteniendo así un mincut del sistema.

Pseudocódigo 3.3 $x = \text{cutsetCorteN}(S, \phi, p, y)$

```

1:  $m \leftarrow |S|$ 
2:  $n \leftarrow |S \setminus y|$ 
3:  $x \leftarrow \vec{y}_m$ 
4: if  $\phi(x) = 0$  then
5:   for  $i = 1, \dots, n$  do
6:      $x(i) = x + \delta_i$ 
7:     if  $\phi(x(i)) = 0$  then
8:        $x = x + \delta_i$ 
9:     end if
10:  end for
11: end if
12: return  $x$ 

```

La implementación del pseudocódigo 3.3 para el modelo bajo estudio también se divide en bloques. Tomando como condición inicial el cutset C_0 ingresado, el bloque de búsqueda se adapta a apagar los elementos correspondientes al cutset dado y creando el grafo auxiliar $g_i = G - C_0$. Para luego verificar si g_i no es d - K -conexo y aplicar el bloque de resolución antes mencionado en el pseudocódigo 3.2.

El algoritmo planteado también responde a órdenes polinomiales, requiriendo una cantidad lineal $|C_0| = (|E_0| + |V_0|)$ de evaluaciones. Por lo que se multiplicará este orden a los algoritmos que lo utilicen.

3.2.2. Corte Trivial

En el documento [8] plantean RVR con un corte trivial o básico para el cálculo de la confiabilidad clásica en el modelo arista. El corte toma todas las aristas incidentes a un nodo terminal. Alineado al documento, se propone una analogía a la estrategia anterior en base al modelo hostil.

El pseudocódigo 3.4 planteado garantiza un cutset o no existe, como RVR trabaja eficientemente bajo mincut se procede a aplicar el Algoritmo 3.3 asegurando que el corte a utilizar es un mincut.

Pseudocódigo 3.4 $C = \text{CorteTrivial}(G, K, d)$

a) \bar{K} : orden aleatorio del conjunto K

b) Sea $v_{k1} \in \bar{K}$ terminal inicial.

c) C_0 , conjunto de todos los nodos no terminales y aristas adyacentes al nodo v_{k1} que cumplen que $c_i \in C_0$ si solo si $p_{c_i} \neq 1$

```

1:  $C = \{\emptyset, \emptyset\}, i = 1$ 
2: while  $i \leq |\bar{K}|$  or FIN do
3:   Hallar  $C_0$  para  $v_{ki}$ 
4:    $g_i = G - C_0$ 
5:   if  $g_i$  no es  $d$ - $K$ -conexo then
6:      $C \leftarrow \text{cutsetCorteN}(C_0)$ 
7:     return  $C$ 
8:   else
9:      $i = i + 1$ 
10:  end if
11: end while

```

Entonces, dado un grafo G , un conjunto de terminales $K \subseteq V$ y la distancia d , con en el pseudocódigo 3.4 se obtiene un mincut C . En los pasos del a) al c) se elige el terminal inicial sorteando el orden del conjunto K , obteniendo así en cada llamada del algoritmo diferentes condiciones iniciales. Considerando v_{k1} el primer terminal del conjunto \bar{K} , se hallan el conjunto C_0 compuesto por todos los vértices y enlaces incidentes a v_{k1} . Sea g_i el grafo auxiliar encontrado apagando todos los componentes de C_0 , si g_i es un estado operativo del sistema se pasa al siguiente terminal, en caso contrario se obtiene un cutset pasando a aplicar el Algoritmo 3.3, retornando éste último un mincut.

Implementación DCR para el modelo hostil

Aquí se presenta la aplicación del pseudocódigo 3.4 para el modelo bajo estudio. Siguiendo lineamientos en 3.1, el script ramifica el flujo según los elementos que componen C_0 .

Pseudocódigo $C = CorteT(G, K, d)$: Se divide en dos grandes partes, bloque de búsqueda (encontrar el cutset inicial) y bloque de resolución (pauta un orden en el recorrido del cutset inicial):

- Bloque de búsqueda (puntos del 1:5): Se sortea un orden para recorrer el conjunto \bar{K} , iniciando por v_{k1} . Se hallan el cutset inicial C_0 compuesto por todos los vértices y enlaces incidentes a v_{k1} , tal que su probabilidad de operación sean distinto de uno.
- Bloque de resolución: Tomando un grafo auxiliar $gi = G - C_0$ se apagan los elementos respetando el orden y se verifica si gi es d - K -conexo. En caso de que el sistema esté operativo (véase punto 6) se pasa el siguiente v_{ki} . En caso de que resulte en falla, se divide el script según el conjunto inicial C_0 encontrado:
 - si $C_0 = \{xe; []\}$ es únicamente aristas, aplicamos el Algoritmo 3.3. Véase puntos del 9 al 16.
 - si $C_0 = \{[]; xn\}$ es únicamente nodos no terminales, aplicamos el Algoritmo 3.3. Véase puntos del 17 al 24.
 - si $C_0 = \{xe, xn\}$ está compuesto tanto de aristas como vértices, tomamos como criterio crear un corte $C_{VE} \subseteq C_0$ compuesto por un nodo no terminal n_c y todas las aristas no incidentes a éste (véase puntos del 26 al 30). Si C_{VE} es un cutset del sistema se aplica el Algoritmo 3.3 para garantizar un mincut (véase puntos del 32 al 34). Por otro lado si el sistema continúa operativo, puede ocurrir que no exista el corte para el nodo v_{ki} (pasando al siguiente nodo terminal, puntos 40 y 41) o que el corte elegido C_{VE} no sea el correcto, por lo que se recorrerá de 1 hasta $|xn|$ todos los nodos no terminales de C_0 verificando los posibles C_{VE} (véase puntos del 35 al 38) que existiesen.

Pseudocódigo 3.5 $C = \text{CorteT}(G, K, d)$

```
1:  $\bar{K}$ : sorteo random del orden del conjunto  $K$ 
2:  $v_{k_1} \in \bar{K}$  terminal inicial.
3:  $C = \{\}$ ,  $i = 1$ 
4: while  $i \leq |\bar{K}|$  or  $\text{fin} = 0$  do
5:   Hallo  $C_0 = \{xe, xn\}$ , conjunto de elementos adyacentes al nodo  $v_{k_i}$ 
6:   if ( $\text{isempty}(xn)$  and  $\text{isempty}(xe)$ ) then
7:     Siguiente terminal  $i = i + 1$ ;
8:   else
9:     if  $\text{isempty}(xn)$  then
10:       $g_i = G - C_0$ 
11:      if  $g_i$  no es  $d$ - $K$ -conexo then
12:         $C \leftarrow \text{cutsetCorteN}(G, C_0, k, d)$ 
13:         $\text{fin} = 1$ 
14:      else
15:         $i = i + 1$ ;
16:      end if
17:    else if  $\text{isempty}(xe)$  then
18:       $g_i = G - C_0$ 
19:      if  $g_i$  no es  $d$ - $K$ -conexo then
20:         $C \leftarrow \text{cutsetCorteN}(G, C_0, k, d)$ 
21:         $\text{fin} = 1$ 
22:      else
23:         $i = i + 1$ ;
24:      end if
25:    else
26:       $\text{finI} = 0$ ;  $x_{nn} = xn$ ;  $x_{ee} = xe$ ;  $\text{fila} = 1$ ;
```

Algorithm 3.5 $C = \text{CorteT}(G, K, d)$

```
27:         while  $\text{size}(xnn) \leq 1$  or  $\text{finI} == 0$  do
28:             Elijo nodo de corte  $n_c = xn(\text{fila}, :)$ ;
29:             Elimino aristas adyacentes de  $n_c$  en  $xe$ 
30:             Creo  $C_{VE} = \{\{xe\}, \{n_c\}\}$ 
31:              $gi = G - C_{VE}$ 
32:             if  $gi$  no es  $d$ - $K$ -conexo then
33:                  $C \leftarrow \text{cutsetCorteN}(G, C_{VE}, k, d)$ 
34:                  $\text{fin} = 0$ ;  $\text{finI} = 0$ ;
35:             else if  $gi$  es  $d$ - $K$ -conexo and  $\text{size}(xnn, 1) \geq 2$  then
36:                 Elimino  $n_c$  de  $xnn$ 
37:                  $xnn(\text{fila}, :) = []$ ;
38:                  $xn = xnn$ ;  $xe = xee$ ;
39:             else
40:                 No existe corte sobre  $n_c$ 
41:                 Siguiete terminal  $\text{finI} = 1$ ;  $i = i + 1$ ;
42:             end if
43:         end while
44:     end if
45: end if
46: end while
47: return  $C$ 
```

El algoritmo planteado también responde a órdenes polinomiales, requiriendo una cantidad $1 + |C_0|$ de evaluaciones para los casos donde $C_0 = \{[], xn\}$ o $C_0 = \{xe, []\}$ ya que estos utilizan el Algoritmo 3.3. Y en el peor de los casos donde $C_0 = \{xe, xn\}$ requiere una cantidad $|C_0|^2$ de evaluaciones.

3.2.3. Corte Estructura

La confiabilidad diámetro acotada cuenta con la singularidad de limitar el largo del camino entre nodos terminales. Basándonos en esta cota y bajo el modelo hostil, se plantea ir quitando enlaces o nodos no terminales aumentando el K -diámetro del grafo obteniendo cortes adaptados a la estructura del SMBS bajo estudio.

La fuente de información de esta estrategia es el algoritmo de búsqueda a lo ancho o por sus siglas en inglés BFS (Breadth First Search), a partir de éste obtenemos un árbol de cubrimiento ¹ del grafo (tomando como raíz uno de los nodos terminales) y la distancia que existe entre el nodo raíz y un nodo cualquiera (Ver anexo 2, BFS).

Pseudocódigo $C = \text{CorteH}(G, K, d)$ Dado un grafo $G = (V, E)$ el conjunto de terminales K y la distancia d , seleccionamos dos nodos terminales que denominados $\{fuente, terminal\}$ e invocamos el pseudocódigo 3.7, el cual implementa la lógica y verifica la existencia de un corte.

Pseudocódigo 3.6 $C = \text{CorteH}(G, K, d)$

```
1:  $finA = 1; i = 1; C = \{\}, \{\}$ ;
2: while  $finA == 1$  and  $i < |K|$  do
3:    $fuente = K(i)$ ;
4:    $j = i + 1$ ;
5:   while  $finA == 1$  and  $j \leq |K|$  do
6:      $terminal = K(j)$ ;
7:      $C \leftarrow \text{corteBFS}(G, K, fuente, terminal, d)$ ;
8:     if  $\text{isempty}(C)$  then
9:        $j = j + 1$ ;
10:    else
11:       $finA = 0$ ;
12:    end if
13:  end while
14:   $i = i + 1$ ;
15: end while
```

En el punto 1 se inicializan las variables. Del punto 2 al 4, el ciclo *while*

¹De teoría de grafos se destacan brevemente las siguientes propiedades. Un árbol G es un grafo en el que cualesquiera dos vértices están conectados por exactamente un camino. Un árbol G es un grafo simple no dirigido, conexo y no tiene ciclos. Y si se le quita alguna arista deja de ser conexo. Para árboles finitos además se cumple que, si un árbol G tiene un número finito de vértices, n , entonces tiene $n - 1$ aristas.

selecciona las terminales, finaliza si existe corte o no. De los puntos 5 al 12, el ciclo *while* anidado recorre los terminales de a par, para invocar el pseudocódigo $C = \text{corteBFS}(G, K, \text{fuente}, \text{terminal}, d)$ determinando si existe corte o no. Si para la pareja de terminales cumple que existe un camino operativo, se pasa a tomar la siguiente pareja en K . Si lo anterior se cumple para cada pareja de terminales que existen en K , el grafo es d - K -conexo (no existe corte).

Implementación DCR para el modelo hostil

Aquí se presenta la aplicación del pseudocódigo 3.6 para el modelo bajo estudio. Siguiendo lineamientos en 3.1.

Pseudocódigo $C = \text{corteBFS}(G, K, \text{fuente}, \text{terminal}, d)$ Dado un grafo $G = (V, E)$, el conjunto de terminales K , dos nodos $\{\text{fuente}, \text{terminal}\}$ y la distancia d , se obtiene un mincut C . Recorremos el grafo de cubrimiento obtenido por BFS, verificamos la distancia entre $\{\text{fuente}, \text{terminal}\}$ creando un pathset. Si la cantidad de saltos a lo largo del pathset cumple que es menor o igual a la cota d (o sea se cumple $D(\text{fuente}, \text{terminal}) \leq d$), apagamos todo el pathset encontrado, y verificamos si el grafo continúa siendo d - K -conexo. En caso afirmativo se continúa aplicando BFS apagando todos los caminos asociados a estos terminales.

Los pathset apagados forman un cutset válido, por lo que se pasa a aplica el Algoritmo 3.3, obteniendo un mincut. Por otro lado, si la pareja de terminales cumple que existe un camino operativo tal que $D(\text{fuente}, \text{terminal}) \leq d$, se entrega un corte vacío.

Al trabajar en el modelo hostil los caminos encontrados están formados por aristas y vértices, por lo que al eliminar cada camino que une fuente y terminal se garantiza el aumento del K -diámetro del grafo, creando cortes basados en la restricción diámetro acotado del SMBS bajo estudio. Particularmente se decidió que para cada camino encontrado, primero apagar todos sus enlaces y a medida que RVR avanza apagar nodos.

Pseudocódigo 3.7 $C = corteBFS(G, K, fuente, terminal, d)$

```
1:  $gi = \{gi_e, gi_v\} = G$ ; grafo auxiliar
2:  $cortee = cortev = []$ ;  $C = \{[], []\}$ ;  $finI = 1$ ;  $D_F = 0$ ;
3: while  $finI == 1$  do
4:    $[marcado, padre, D_F] = bfs(gi_e, fuente)$ ;
5:   if  $D_F(fuente, terminal) \leq d$  and  $D_F! = 0$  then
6:     for  $i = 1, \dots, |padre|$  do
7:       Hallo pathset formado por aristas y vértices
8:        $pathset = \{[path_e], [path_v]\}$ ;
9:     end for
10:     $cuentoPath = 0$ ;
11:    for  $i_{pat} = 1, \dots, |path_e|$  do
12:       $i_e = 1; fin = 1$ ;
13:      while  $fin == 1$  and  $i_e \leq |gi_e|$  do
14:         $en_{off} = path_e(i_{pat})$ ;
15:        if  $gi_e(i_e) == en_{off}$  then
16:          if  $gi_e(i_e)$  no esta prendido then
17:            Apago enlace de  $g_i$ , y lo guardo
18:             $cortee = [cortee; gi_e(i_e)]$ ;
19:          else
20:             $cuentoPath = cuentoPath + 1$ ;
21:          end if
22:           $fin = 0$ ;
23:        end if
24:         $i_e = i_e + 1$ ;
25:      end while
26:    end for
```

Algorithm 3.7 $C = \text{corteBFS}(G, K, \text{fuente}, \text{terminal}, d)$

```
27:     if  $\text{cuentoPath} == |\text{path}_e|$  then
28:          $\text{cuentoNodos} = 0$ ;
29:         for  $i_n = 1, \dots, |\text{path}_v|$  do
30:              $\text{nd}_{\text{off}} = \text{path}_v(i_n)$ ;
31:              $i_v = 1$ ;  $\text{fin} = 1$ ;
32:             while  $\text{fin} == 1$  and  $i_v \leq |g_{i_v}|$  do
33:                 if  $g_{i_v}(i_v) == \text{nd}_{\text{off}}$  then
34:                     if  $g_{i_v}(i_v)$  no esta prendido then
35:                         Apago nodo  $\text{nd}_{\text{off}}$  de  $g_i$  y lo guardo
36:                          $\text{cortev} = [\text{cortev}; g_{i_v}(i_v)]$ 
37:                     else
38:                          $\text{cuentoNodos} = \text{cuentoNodos} + 1$ 
39:                     end if
40:                      $\text{fin} = 0$ ;
41:                 end if
42:                  $i_v = i_v + 1$ ;
43:             end while
44:         end for
45:         if  $\text{cuentoNodos} == |\text{path}_v|$  then
46:              $\text{finI} = 0$ ;
47:         end if
48:     end if
49:     if  $\text{finI} \neq 0$  then
50:         if  $g_i$  no es  $d$ - $K$ -conexo then
51:              $\text{finI} = 0$ ;
52:              $C_0 = \{\text{cortee}, \text{cortev}\}$ ;
53:              $C \leftarrow \text{cutsetCorteN}(G, C_0, k, d)$ 
54:         end if
55:     end if
56:     else
57:         RVR no llama al corte, si  $G$  no es  $d$ - $K$ -conexo,
58:          $\text{finI} = 0$ ;
59:     end if
60: end while
```

En los puntos 1 y 2 se inicializan las variables. En el punto 3, el ciclo *while* finaliza si existe corte, o se cumple que existe un camino prendido para el par de terminales (no existe corte). Del punto 4 al 9, aplicando BFS se construye el camino entre la fuente y el terminal. A partir de aquí se separa el flujo en apagar enlaces o apagar nodos no terminales:

- Desde el punto 10 al 26 se recorre el grafo g_i , apagando los enlaces del pathset que no se encuentran prendidos. Si se encuentra algún enlace a apagar se pasa al punto 49, si g_i es d - K -conexo se vuelve a invocar el método BFS, en caso contrario se invoca al Algoritmo 3.3 obteniendo un mincut.
- Desde los puntos 27 al 48, si todos los enlaces del pathset encontrado están prendidos (*cuentoPath* es un contador de enlaces prendidos) se pasa a apagar nodos, llegando al punto 49 si g_i verifica que no es d - K -conexo se obtiene mincut, en caso contrario se vuelve a invocar el método BFS.

El algoritmo planteado también responde a órdenes polinomiales, requiriendo en el peor de los casos una cantidad $|C_0|^2$ de evaluaciones ya que este utiliza el Algoritmo 3.3.

3.2.4. Corte Optimizado

En esta sección se presenta un criterio de selección de corte de máxima probabilidad, que permita mejorar en cada iteración la eficiencia de RVR. La probabilidad de cualquier cutset está determinada por la probabilidad elemental de falla de las aristas y los vértices que lo compongan. Como la ecuación de probabilidad de un corte es igual a multiplicar la anticonfiabilidad de cada elemento, para maximizar $Pr(C)$ el conjunto C deberá tener la menor cantidad de elementos. Por ejemplo, nos podría interesar encontrar un punto de articulación del grafo, el cual separa a éste en varias partes conexas, y así resolver óptimamente el problema planteado. Pero el anterior es un claro ejemplo de la complejidad que presenta encontrar un corte de mínima cantidad de elementos, los cuales clasifican como problemas \mathcal{NP} -Difícil. Por otro lado, como trabajaremos con grafos de probabilidad homogénea y heterogénea, en este último puede no cumplirse que el corte de menor cantidad de elementos sea el de máxima probabilidad.

Planteamos trabajar con métodos heurísticos, éstos guardan información del óptimo actual escogiendo el estado óptimo entre todos los estados actuales obtenidos en varias etapas del algoritmo. Como la cantidad de d -cutset a evaluar puede ser muy grande, se toma como tiempo máximo N pasos de ejecución. También pautamos que un corte estará formado por vértices y aristas no incidentes a éstos, creando independencia entre los elementos que lo componen. Entonces, dado las pautas anteriores y lineamientos marcados en 3.1 la función a maximizar se define a partir de la probabilidad de falla en aristas (q_e) y en nodos no terminales (q_v) que componen el corte.

Problema de optimización planteado:

$$\begin{cases} \text{máx}\{Pr(C)\} \\ \text{sujeto a: } C \in \{V - K\} \cup E \text{ donde } \phi(C) = 0 \end{cases} \quad (3.2.1)$$

Pseudocódigo $C = \text{CorteMP}(G, K, d)$

Dado un grafo $G = (V, E)$, el conjunto de terminales K , y la distancia d , se obtiene un mincut C de máxima probabilidad de ocurrencia. Se pauta buscar $N = 10$ cortes aleatorios (C_i), calcular su $Pr(C_i)$ y según la probabilidad el código retornará el corte de máxima probabilidad entre los cortes (C_i) encontrados. Dada la naturaleza del sistema bajo estudio y sus restricciones, inicialmente se busca sortear cortes apagando aristas, para luego incorporar nodos.

Pseudocódigo 3.8 $C = \text{CorteMP}(G, K, d)$

```
1:  $N = 10$ ; cantidad de corte encontrados
2: grupo-de-cortes=  $\{\square; \square\}$ ; guardo  $N$  cortes
3:  $C = \{\square; \square\}$ ;
4: prob-corte= 0;
5: cant-corte=  $\square$ ;
6: for  $i = 1, \dots, N$  do
7:   corte $v = \square$ ; corte $e = \square$ ;
8:    $g_i = \{G_e, G_v\}$ ;
9:    $A_r$ : vector de aristas permutadas de  $g_i$ 
10:   $V_r$ : vector de nodos permutados de  $g_i$ 
11:   $i_a = 1$ ; salir = 0;
12:  while  $cutset == 0$  and  $salir == 0$  do
13:    if  $Pr(A_r(i_a)) \neq 1$  then
14:      Guardo arista  $A_r(i_a)$  en corte $e$  y la elimino de  $g_i$  ;
15:      if  $g_i$  no es  $d-K$ -conexo then
16:         $cutset = 1$ ;
17:      end if
18:       $i_a = 1$ ;
19:       $A_r$ :permuto orden de aristas de  $g_i$ 
20:    else
21:      if si todas las aristas que restan están prendidas then
22:         $salir = 1$ ;
23:      end if
24:       $i_a = i_a + 1$ ;
25:    end if
26:  end while
```

Algorithm 3.8 $C = \text{CorteMP}(G, K, d)$

```
27:   if  $cutset == 1$  then
28:        $C \leftarrow \text{cutsetCorteN}(G, \{cortee, cortev\}, K, d)$ 
29:       Guardo  $C$  en grupo-de-cortes
30:       Cálculo  $Pr(C)$  y lo guardo en prob-corte
31:   else
32:        $i_b = 1$ ;  $salir2 = 0$ ;
33:       while  $cutset == 0$  and  $salir2 == 0$  do
34:           if  $Pr(V_r(i_b)) \neq 1$  then
35:                $cortev = [cortev; V_r(i_b)]$  y elimino  $V_r(i_b)$  de  $gi$ 
36:               if  $gi$  no es  $d$ - $K$ -conexo then
37:                    $cutset = 1$ ;
38:               end if
39:                $i_b = 1$ ;
40:                $V_r$ :permuto orden de nodos de  $gi_v$ ;
41:           else
42:               if si todas los nodos que restan están prendidos then
43:                    $salir2 = 1$ ;
44:               end if
45:                $i_b = i_b + 1$ ;
46:           end if
47:       end while
48:       if  $cutset == 1$  then
49:            $C \leftarrow \text{cutsetCorteN}(G, \{cortee, cortev\}, K, d)$ ;
50:           Guardo  $C$  en grupo-de-cortes
51:           Cálculo  $Pr(C)$  y lo guardo en prob-corte
52:       end if
53:   end if
54:    $cant\text{-}cortes = [cant\text{-}cortes, cutset]$ ;
55: end for
56: if Si  $cant\text{-}cortes > 0$  then
57:      $corte = grupo\text{-}de\text{-}cortes(1)$ ;
58:      $max\text{-}prob = prob\text{-}corte(1)$ ;
59:     for  $i_n = 1, \dots, N + 1$  do
60:         Busco en prob-corte y grupo-de-corte  $C$  con mayor probabilidad
61:     end for
62: end if
63: Return  $C$ 
```

De los puntos 1 al 5 se inicializan las variables, en 6 se comienza a buscar los N cortes pautados. Véase los puntos del 7 al 26, en estos sorteamos el orden de las aristas del grafo y se las guarda en un vector A_r , desde 1 hasta $|A_r|$ se van apagando aristas y se verifica que el grafo generado no es d - K -

conexo, en caso de que las aristas encontradas no se puedan apagar (o sea están todas prendidas) se pasa a apagar nodos no terminales. Entre los puntos 31 y 47, se sortea el orden de los vértices del grafo y se los guarda en un vector V_r , desde 1 hasta $|V_r|$ se van apagando y se verifica que el grafo generado no es d - K -conexo obteniendo así un corte C . Para los cortes encontrados se invoca el pseudocódigo 3.3 obteniendo un mincut (véase los puntos del 27 al 30 y del 48 al 53). Los N cortes encontrados son guardados junto con su probabilidad de ocurrencia, para luego iterar entre ellos y obtener el corte de mayor anticonfiabilidad (véase los puntos del 56 al 63).

El algoritmo planteado recorre todo los elementos del grafo sorteando cortes aleatorios, esto lo hace más ineficiente que las otras estrategias elegidas, pero también responde a órdenes polinomiales. Requiere una cantidad $(|V| + |E|)$ de evaluaciones en el peor de los casos, y se adiciona las evaluaciones del Algoritmo 3.3, por lo que el algoritmo responde un orden $N * (|V| + |E|)^2$.

Capítulo 4

Resultados Experimentales

En el Capítulo 2 se presentó el marco teórico que envuelve el problema de confiabilidad en redes, así como los métodos de cálculo exacto y aproximados para obtener la medida de confiabilidad. En el Capítulo 3 se expuso la metodología aplicada describiendo detalladamente los métodos utilizados, así como sus limitaciones para dar solución al problema de la confiabilidad diámetro acotada en el modelo hostil. En el presente capítulo presentamos el marco práctico de la investigación, ambiente de trabajo, análisis de eficiencia, casos de prueba, resultados empíricos de acuerdo a la metodología aplicada de la investigación, y donde los resultados obtenidos serán contrastados o fundamentados sobre la teoría.

4.1. Análisis de eficiencia

Recordemos los objetivos planteados:

- Comparar el desempeño del método RVR con el método Monte Carlo Crudo (MCC), tomando como parámetros de referencia la precisión de las estimaciones, siendo cuantificadas en términos de la varianza y el tiempo de cálculo.
- Estudiar la eficiencia del método RVR respecto a distintas estrategias de separación de grafos. Determinando cuál de estas estrategias nos permite obtener estimaciones de confiabilidad en menos tiempo de cálculo o menos muestras de ejecución de RVR.

Los casos de prueba propuestos se dividen en grafos chicos y grandes. Los primeros fueron utilizados como una fase de comprobación de resultados, por su

fácil acceso al valor de confiabilidad exacta a partir de métodos de fuerza bruta. Para los segundos, como se discutió en otras sesiones el tiempo de cómputo es prohibitivo, por lo que se sigue el lineamiento marcado en los objetivos (marco comparativo entre RVR y MCC).

¿Por qué estudiar distintas estrategias de corte, y cómo afecta a RVR? En el documento [8] identifican que la elección del corte influye en cada llamada recursiva de RVR, por lo que estudiar distintas estrategias o técnicas de separación de grafos nos permite cuantificar como es afectada la eficiencia y/o eficacia de RVR. Este lineamiento nos permite investigar cómo se comporta RVR frente a distintas estrategias de cortes, y/o distintas estructuras de grafos.

Ahora bien, ¿dentro de los cortes planteados para el método RVR cuales arroja mejores resultados? ¿En función de qué parámetros se ejecuta dicha comparación? Concretamente las medidas consideradas para evaluar el desempeño de los casos de estudio son: las estimaciones puntuales de confiabilidad R , su varianza Var , el tiempo de cómputo total t (en segundos), el error relativo (para grafos chicos) y el coeficiente de eficiencia relativa \bar{W} entre RVR y MCC.

Eficiencia relativa, \bar{W}

¿Cómo evaluar el desempeño de dos o más métodos aplicados para resolver un mismo problema? Los resultados empíricos pueden compararse en términos de eficacia y/o eficiencia. Fijando el número de iteraciones, una medida alternativa es comparar la varianza del estimador que proporciona una medida de precisión de la estimación. Otra opción asociada a la eficiencia, es comparar el tiempo de cálculo necesario para obtener una estimación.

La eficiencia relativa \bar{W} es una medida que permite la comparación entre diferentes métodos de estimación tomando como referencia el método MCC (Véase [25, 2, 20]). Para comparar dos métodos $M1$ y $M2$ con esta medida se considera la siguiente relación:

$$\bar{W} = \frac{Var_{M1}T_{M1}}{Var_{M2}T_{M2}} \quad (4.1.1)$$

Donde Var_{M_i} y T_{M_i} son la varianza y el tiempo de cómputo del método M_i , $i = 1, 2$ respectivamente, en nuestro caso el método $M1$ hace referencia a MCC y $M2$ a la implementación de RVR en las distintas estrategias de separación de grafos planteadas. Entonces, $M2$ percibe un mejor desempeño (o demanda una menor cantidad de recursos) respecto a $M1$, si el valor de eficiencia relativa es

mayor a 1. Para cada método esta medida conjunta compromete una relación entre la calidad de la aproximación y la eficiencia en cómputo. Por lo que, se presenta la medida ganancia:

$$G_{Mi} = \log_{10}(Var_{Mi} * T_{Mi}) \quad (4.1.2)$$

la cual nos permite contemplar el orden numérico que reporta la medida conjunta, y estudiar su comportamiento grafico en relación al conjunto de muestras tomadas, adicionando información relevante a la medida de eficiencia relativa.

Error relativo

Si \bar{R} es el estimador de R calculado a partir de n simulaciones. El error relativo Er es una medida que permite conocer la precisión de las estimaciones y se define como:

$$Er = \frac{\bar{R} - R}{R} \quad (4.1.3)$$

Se analizará el error relativo de la confiabilidad para grafos chicos.

4.2. Presentación de datos

En esta sección se describe los casos analizados, datos obtenidos, y sistema de cómputo utilizado para la obtención de los resultados.

4.2.1. Infraestructura de cómputo

La infraestructura de cómputo utilizada fue el servidor Carlitos Gardel el cual es propiedad de la Facultad de Ingeniería. Servidor de 24 CPU - AMD Opteron(tm) Processor 6172, con distribución GNU/Linux. El ingreso al servidor fue por protocolo SSH (Secure Shell) con los permisos correspondientes.

Como aplicación secundaria se trabajó sobre GNU Screen una aplicación de software que multiplexa varias consolas virtuales, lo que permite al usuario acceder a múltiples sesiones de inicio de sesión separadas dentro de una sola ventana de terminal, o separar y volver a conectar sesiones desde un terminal.

Los algoritmos se implementaron en GNU Octave, versión 3.4.3.

4.2.2. Casos de estudio

De la literatura referente se tomaron seis grafos de distintas cantidades de nodos, enlaces y/o topologías. Para cada grafo se analizó la cantidad de nodos terminales y nodos no terminales a considerar, definiendo modelos $s-t$ -terminal o K -terminal.

Para observar cómo influye la cota de diámetro d en la confiabilidad de los grafos seleccionados, se realizó un análisis previo según su topología y modelo definido. En general se tomaron dos valores de cota, y particularmente para grafos simétricos se plantea un único valor.

Para deliberar cómo incide la falla de los elementos del grafo $G = (V, E)$ en el cálculo de la confiabilidad, se plantea estudiar un caso homogéneo y un caso heterogéneo de probabilidad. Se toman valores de probabilidad de operación $p_e = 0.95$ en todos sus enlaces, y $p_v \in \{0.95, 0.97\}$ en los nodos no terminales.

A continuación se detallan las topologías consideradas y sus características:

- Puente, $s-t$ -terminal: $|E| = 5, |V| = 4, |K| = 2$
 - $D \in \{2, 3\}, p_e \in \{0.95\}, p_v \in \{0.95, 0.97\}$, Figura 4.1
- Completo, $s-t$ -terminal: $|E| = 6, |V| = 4, |K| = 2$
 - $D \in \{2, 4\}, p_e \in \{0.95\}, p_v \in \{0.95, 0.97\}$, Figura 4.6
- Grilla $3 * 3$, $s-t$ -terminal: $|E| = 12, |V| = 9, |K| = 2$
 - $D \in \{4\}, p_e \in \{0.95\}, p_v \in \{0.95, 0.97\}$, Figura 4.11
- Grilla $5 * 5$, $s-t$ -terminal: $|E| = 40, |V| = 25, |K| = 2$
 - $D \in \{8\}, p_e \in \{0.95\}, p_v \in \{0.95, 0.97\}$, Figura 4.11
- Arpanet. $s-t$ -terminal: $|E| = 25, |V| = 20, |K| = 2$
 - $D \in \{8\}, p_e \in \{0.95\}, p_v \in \{0.95, 0.97\}$, Figura 4.17
- Dodecaedro. $s-t$ -terminal: $|E| = 30, |V| = 20, |K| = 2$
 - $D \in \{5, 8\}, p_e \in \{0.95\}, p_v \in \{0.95, 0.97\}$, Figura 4.20

Se presentan dieciocho escenarios, con valores de muestras $N \in \{10^2, 10^3, 10^4, 5 * 10^4, 10^5\}$ para cada uno. En las tablas de resultados se desplegará, para cada escenario el valor de diámetro (D), el par de probabilidad (p_e, p_v) , el valor de muestra N , el estimador de confiabilidad (\bar{R}) y de varianza (Var), el tiempo de ejecución (t) y la ganancia (\bar{W}) de RVR respecto a MCC.

En el afán de simplificar los resultados, figuras y tablas, se asigna la siguiente nomenclatura a la implementación de RVR según las estrategias de separación de grafos planteadas en el Capítulo 3:

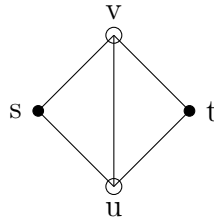
Tabla 4.1: Nomenclatura para RVR según las estrategias de corte

Nombre de estrategias	Referencia	Notación en RVR
Corte Lineal	3.2.1	RVR_N
Corte Trivial	3.2.2	RVR_T
Corte Estructura	3.2.3	RVR_H
Corte Optimizado	3.2.4	RVR_{MP}

4.2.2.1. Grafo Puente

El grafo puente es un caso muy conocido y estudiado dentro de la teoría de grafos. Su topología permite rápidamente obtener resultados, y realizar verificaciones. El valor de la confiabilidad exacta es obtenido fácilmente evaluando todos los estados posibles del grafo. Partiendo del caso de uso donde todos los enlaces están activos, se observa que el camino mínimo es de largo dos, y el subsiguiente es de tres saltos. Se toma un modelo s - t -terminal con cotas $D \in \{2, 3\}$, estos valores nos permiten explorar caminos críticos según la topología.

Figura 4.1: Grafo puente



Se determina la siguiente notación para identificar los distintos escenarios o distintas instancias de corridas:

Tabla 4.2: Notación grafo puente

D	p_e/p_v	Notación
2	"0.95/0.95"	"Puente-PHomgD2"
2	"0.95/0.97"	"Puente-PHetrgD2"
3	"0.95/0.95"	"Puente-PHomgD3"
3	"0.95/0.97"	"Puente-PHetrgD3"

Resultados

Tabla 4.3: Resultados de DCR exacta para grafo puente

D	p_e/p_v	R_{exacta}	t (seg)
2	"0.95/0.95"	0.979658109375000	0.36
2	"0.95/0.97"	0.984481069375000	0.36
3	"0.95/0.95"	0.983527014062500	0.34
3	"0.95/0.97"	0.988514590062500	0.32

Resultados empíricos Puente-PHomgD2 y Puente-PHomgD3

Como características generales observadas en las Tablas 3.3 y 3.4 (ver Anexo 3), el cálculo de la DCR con RVR alcanza valores de varianza dos órdenes menor (mayor precisión) que el obtenido con Monte Carlo Crudo (ver Tabla 3.1 en Anexo 3), y consigue un tiempo de cómputo de uno y dos órdenes mayor a MCC. Esto conlleva a que los valores de eficiencia relativa \bar{W} varíen entre las distintas estrategias planteadas. RVR_N , RVR_H y RVR_T en relación a MCC obtuvieron valores de eficiencia relativa mayores a uno, mientras que para RVR_{MP} menor a uno. Adicionalmente, los valores de varianza obtenidos por RVR para las distintas estrategias de cortes llegan a un orden de 10^{-9} mostrando mayor precisión para una mayor cota, y adicionalmente se registró similares tiempos de cómputo entre los escenarios.

Observando para los escenarios las gráficas de los valores obtenidos en confiabilidad (ver 4.2a y 4.3a) en conjunto con las figuras de varianza (ver 4.2e y 4.3e) y error relativo (ver 4.2b y 4.3b) se sobresalta la necesidad de una mayor cantidad de muestras de MCC para llegar a un valor estimado confiable. De estas tres figuras se puede apreciar que, a partir de muestras mayor o igual a 1000 la DCR calculada con RVR (para todos los cortes) tiende o converge al valor exacto, también para todos los cortes planteados la varianza disminuye (ver 4.2f y 4.3f) al aumentar la cantidad de muestras aumentando la precisión del método.

Contar con el valor exacto de la confiabilidad diámetro acotada nos permite realizar un análisis de exactitud (ver Tabla 3.2 en Anexo 3 y Figuras 4.2b y 4.3b) del método RVR frente a los distintos cortes. Se recalca la buena exactitud del método RVR_N , logrando un orden en error relativo de hasta 10^{-6} , adicionalmente en algunos casos se observa que ante un aumento de muestras el error conserva el orden obtenido (se acentúa a mayor cota). El comportamiento se degrada para los métodos RVR_H , RVR_{MP} y RVR_T , y puede estar asociado a cierto déficit de aleatoriedad en la implementación de las estrategias. Como complemento en el escenario Puente-PHomgD2, para muestras mayor o igual a mil se obtiene un error relativo que varían entre 0.0003875 % y 0.227 %. Para los métodos RVR_N y RVR_H , cada incremento de muestras induce a la disminución del error relativo, en cambio para RVR_{MP} y RVR_T se percibe un leve crecimiento para muestras 50000 y 10000 respectivamente disminuyendo para valores mayores. Se destaca el error relativo de 0.015 % alcanzado para

1000 muestras de RVR_N . Adicionalmente en el escenario Puente-PHomgD3, para muestras mayor o igual a mil se obtiene un error relativo que varían entre 0,00019% y 0.055%. Para los métodos RVR_N y RVR_T , un incremento de muestras induce a disminuir o conservar el orden del error relativo, en cambio para RVR_H y RVR_{MP} se percibe invariante. Se destaca el error relativo de 0.004% alcanzado para 1000 muestras de RVR_{MP} .

En las Figuras 4.2c y 4.3c se visualiza que los métodos mantienen un orden constante de ganancia para todo el rango de muestras. Donde RVR_N , RVR_H y RVR_T muestran una compensación en la medida conjunta (tiempo de cómputo y varianza) en comparación con MCC, mientras que RVR_{MP} consigue un orden más en ganancia debido a su alto tiempo en cómputo, imputando un mejor desempeño a MCC.

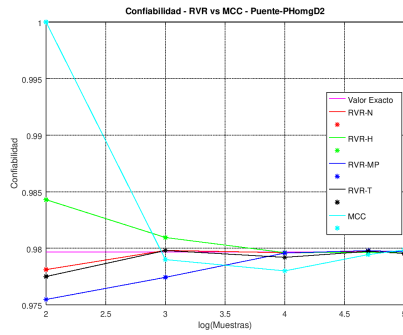
De las Figuras 4.2d y 4.3d se observa que, el tiempo de cómputo de RVR presentó tiempos de ejecución lineal al número de muestras tomadas, en particular RVR_N , RVR_H y RVR_T obtienen menor pendiente en comparación a RVR_{MP} . Se identifica una mejor performance para el RVR_N .

Comparando los escenarios Puente-PHomgD2 y Puente-PHomgD3 contra la teoría, se cumple que los valores de DCR estimados por RVR para la cota $D = 2$ son menores que los alcanzados por $D = 3$. No se advierten vastas diferencias en tiempo de cómputo entre los escenarios. En relación a la cantidad de muestras, se logran valores de varianza de menor unidad para $D = 3$, tendiendo más rápido a su valor estimado y generando un mejor desempeño en la medida conjunta ganancia (tiempo en cómputo por varianza). Esto es atribuido a la disminución en la cardinalidad del espacio de conjuntos de corte para la cota $D = 3$.

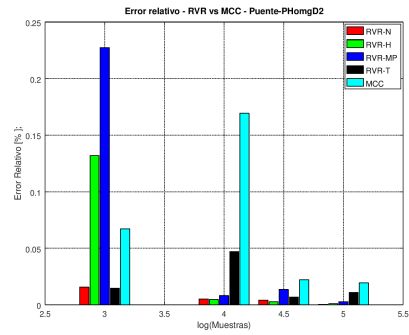
Para los escenarios Puente-PHomgD2 y Puente-PHomgD3 se destaca en general, la performance de RVR_N . Se recomienda 10^3 y 10^4 muestras del método donde observamos una buena precisión, tiempo de cómputo y exactitud.

En la Figura [Análisis estadístico RVR vs MCC en Puente-PHomgD2](#) se grafica el rendimiento de RVR para las distintas estrategias de selección de cortes en función del tamaño de muestra, comparando estos contra la DCR exacta y MCC. En la Figura 4.2a se muestra el valor calculado de la confiabilidad diámetro acotada para el escenario. En la Figura 4.2b se grafica el error relativo En la Figura 4.2c se muestra la ganancia del producto tiempo por varianza. En la Figura 4.2d se grafica el tiempo de ejecución. En las Figuras 4.2e y 4.2f se grafican los valores de varianza.

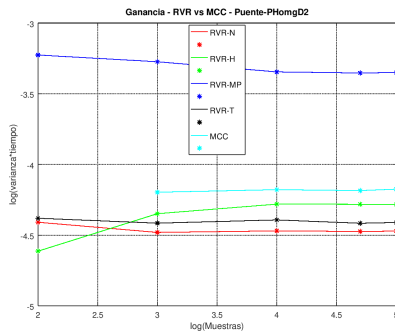
Figura 4.2: Análisis estadístico RVR vs MCC en Puente-PHomgD2



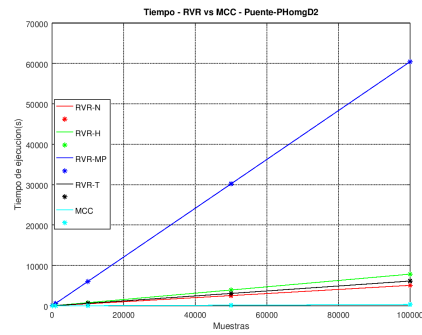
(a) Confiabilidad



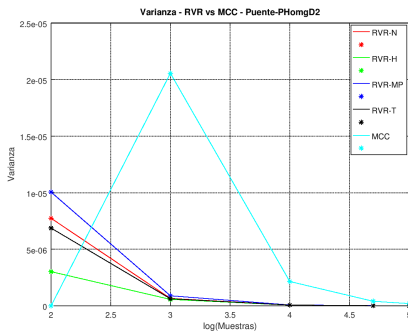
(b) Error relativo



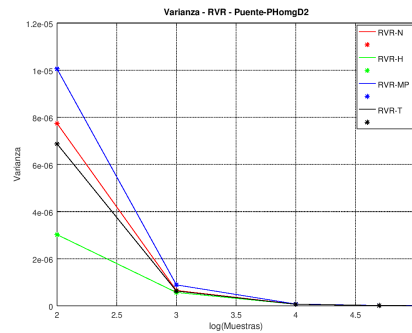
(c) Ganancia



(d) Tiempo de ejecución



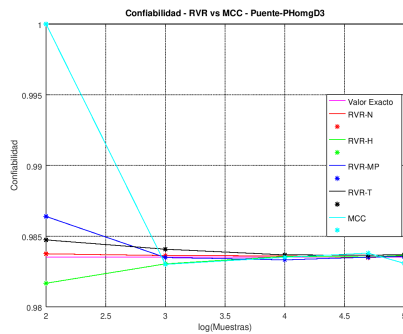
(e) Varianza



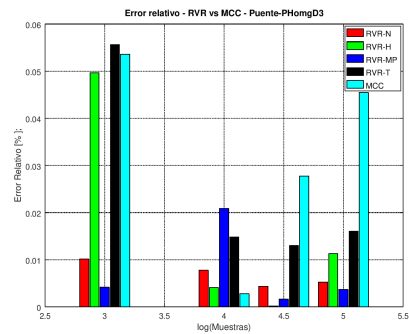
(f) Varianza por corte

En la Figura [Análisis estadístico RVR vs MCC en Puente-PHomgD3](#) se grafica el rendimiento de RVR para las distintas estrategias de selección de cortes en función del tamaño de muestra, comparando estos contra la DCR exacta y MCC. En la Figura 4.3a se muestra el valor calculado de la confiabilidad diámetro acotada para el escenario. En la Figura 4.3b se grafica el error relativo En la Figura 4.3c se muestra la ganancia del producto tiempo por varianza. En la Figura 4.3d se grafica el tiempo de ejecución. En las Figuras 4.3e y 4.3f se grafican los valores de varianza.

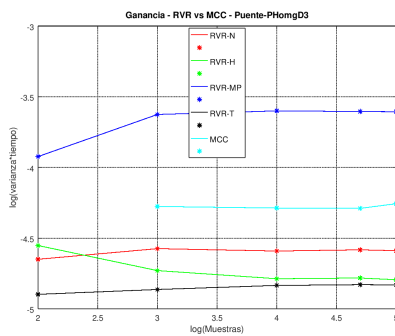
Figura 4.3: Análisis estadístico RVR vs MCC en Puente-PHomgD3



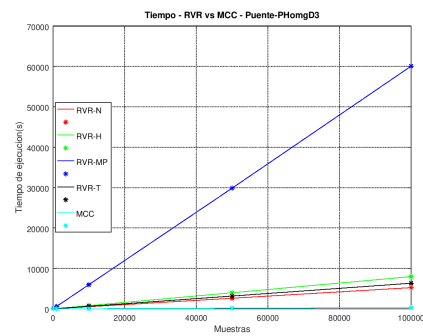
(a) Confiabilidad



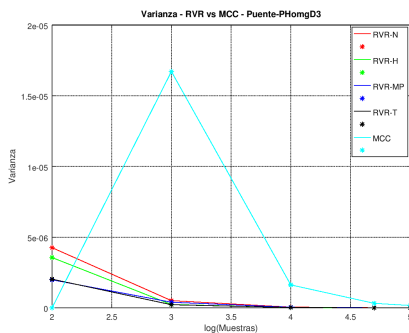
(b) Error relativo



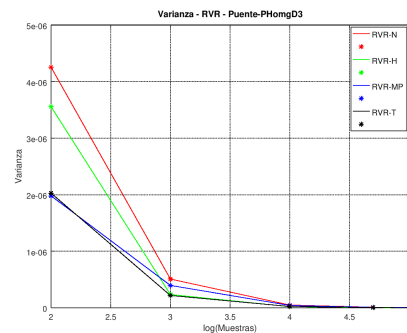
(c) Ganancia



(d) Tiempo de ejecución



(e) Varianza



(f) Varianza por corte

Resultados empíricos Puente-PHetrgD2 y Puente-PHetrgD3

Como características generales observadas en las Tablas 3.5 y 3.6 (ver Anexo 3), el cálculo de la DCR con RVR alcanza valores de varianza dos y tres órdenes menor que el obtenido con Monte Carlo Crudo (ver Tabla 3.1 en Anexo 3), y un tiempo de cómputo uno orden mayor que MCC. Esto conlleva a que los valores de eficiencia relativa \bar{W} varíen entre las distintas estrategias planteadas. RVR_N , RVR_H y RVR_T obtienen valores de eficiencia relativa \bar{W} mayores a uno, mientras que para RVR_{MP} menor a uno. Adicionalmente, los valores de varianza obtenidos por RVR para las distintas selección de cortes llegan a órdenes de 10^{-9} y 10^{-10} menor y mayor cota respectivamente, y adicionalmente se registró similares tiempos de cómputo entre los escenarios.

Observando para los escenarios las gráficas de los valores obtenidos en confiabilidad (ver 4.4a y 4.5a) en conjunto con las figuras de varianza (ver 4.4e y 4.5e) y error relativo (4.4b y 4.5b) se sobresalta la necesidad de una mayor cantidad de muestras de MCC para llegar a un valor estimado confiable. De estas tres figuras se puede apreciar que, a partir de muestras mayor o igual a 1000 la DCR calculada con RVR (para todos los cortes) tiende o converge al valor exacto, también para todos los cortes planteados la varianza disminuye (ver 4.4f y 4.5f) al aumentar la cantidad de muestras aumentando la precisión del método.

Contar con el valor exacto de la confiabilidad diámetro acotada nos permite realizar un análisis de exactitud (ver Tabla 3.2 en Anexo 3 y Figuras 4.4b y 4.5b) del método RVR frente a los distintos cortes. Se recalca la buena exactitud del método RVR_N , logrando un orden en error relativo de hasta 10^{-6} . Adicionalmente, se contempla para todas las estrategias RVR_N , RVR_H , RVR_{MP} y RVR_T , casos aislados donde ante un aumento de muestras el error conserva o aumenta el orden obtenido. El comportamiento se degrada ante una mayor cota, y puede estar asociado a cierto déficit de aleatoriedad en la implementación de las estrategias. Como complemento en el escenario Puente-PHetrgD2, este obtiene un rango de error relativo que varían entre 0.0007841 % y 0.0907 %. Se destaca el error relativo de un 0.0084 % logrado para 1000 muestras de RVR_N . Luego, para el escenario Puente-PHetrgD3 y muestras mayor o igual a mil se obtiene un rango de error relativo que varían entre 0.00037247 % y 0.055787 %. Se destaca el error relativo alcanzado de un 0.0076194 % para 1000 muestras de RVR_H .

En las Figuras 4.4c y 4.5c se visualiza que los métodos mantienen un orden constante de ganancia para todo el rango de muestras. De la Figura 4.4c, se observa que RVR_N , RVR_H y RVR_T muestran una compensación en la medida conjunta (tiempo de cómputo y varianza) en comparación con MCC, mientras que RVR_{MP} consigue un orden más en ganancia debido a su alto tiempo en cómputo, imputando un mejor desempeño a MCC. Luego en la Figura 4.5c se visualiza que, RVR_N y RVR_{MP} logran una compensación en la medida conjunta en comparación con MCC, mientras que RVR_H y RVR_T muestran un orden menor en ganancia, imputando un mejor desempeño a RVR. Comparando los datos anteriores, para una mayor cota los métodos RVR_H , RVR_{MP} y RVR_T registran mejoras en la medida de ganancia, esto es atribuido a las mejoras en precisión (no se aprecia cambios en MCC).

De las Figuras 4.4d y 4.5d se observa que, el tiempo de cómputo de RVR presentó tiempos de ejecución lineal al número de muestras tomadas, en particular RVR_N , RVR_H y RVR_T obtienen menor pendiente en comparación a RVR_{MP} .

Comparando los escenarios Puente-PHetrgD2 y Puente-PHetrgD3 contra la teoría, se cumple que los valores de DCR estimados por RVR para la cota $D = 2$ son menores que los alcanzados por $D = 3$. No se advierten vastas diferencias en tiempo de cómputo entre los escenarios. En relación a la cantidad de muestras, se logran valores de varianza de menor unidad para $D = 3$, tendiendo más rápido a su valor estimado y generando un mejor desempeño en la medida conjunta ganancia (tiempo en cómputo por varianza). Esto es atribuido a la disminución en la cardinalidad del espacio de conjuntos de corte para la cota $D = 3$.

Ahora en un marco general, comparando los valores obtenidos por RVR en las instancias homogéneas contra heterogéneas (para la misma cota de diámetro) no se observan vastas diferencias en tiempo de cómputo, y en relación a la cantidad de muestras se observa valores de varianza de menor unidad para escenarios heterogéneos, tendiendo estos más rápido a su valor estimado, generando un mejor desempeño de la medida conjunta, desencadenando una mayor eficiencia relativa. Las diferencias encontradas pueden ser atribuidas a la sensibilidad del método RVR frente a la diferencia planteada. Cabe notar que en las estrategias de separación de grafos formalizadas no implementan

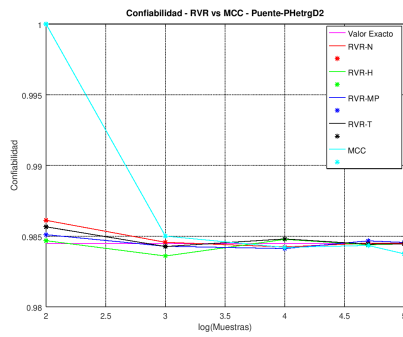
una elección de elementos en base a su probabilidad elemental.

A rasgos generales para los distintos escenarios planteados del grafo puente, concluimos que RVR presentó buen comportamiento asintótico, buena precisión mostrando valores de varianza dos (o tres) órdenes menor y tiempos de cómputo aceptables en comparación con MCC, a través de la medida conjunta se destaca de los métodos RVR_N y RVR_H una buena relación entre la calidad de la aproximación y tiempo de cómputo para todo el conjunto de muestras y escenarios. Adicionalmente, el estudio arroja casos aislados donde ante un aumento de muestras el error relativo conserva o disminuye el orden obtenido. Se acentuó ante una mayor cota, y puede estar asociado a cierto déficit de aleatoriedad en la implementación de las estrategias de cortes.

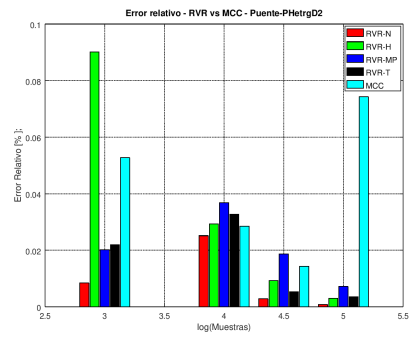
Se destaca en general para los escenarios planteados la performance de RVR_N y RVR_H . Se recomienda 10^4 muestras de los métodos donde observamos una buena precisión, tiempo de cómputo y exactitud.

En la Figura [Análisis estadístico RVR vs MCC en Puente-PHetrgD2](#) se grafica el rendimiento de RVR para las distintas estrategias de selección de cortes en función del tamaño de muestra, comparando estos contra la DCR exacta y MCC. En la Figura 4.4a se muestra el valor calculado de la confiabilidad diámetro acotada para el escenario. En la Figura 4.4b se grafica el error relativo En la Figura 4.4c se muestra la ganancia del producto tiempo por varianza. En la Figura 4.4d se grafica el tiempo de ejecución. En las Figuras 4.4e y 4.4f se grafican los valores de varianza.

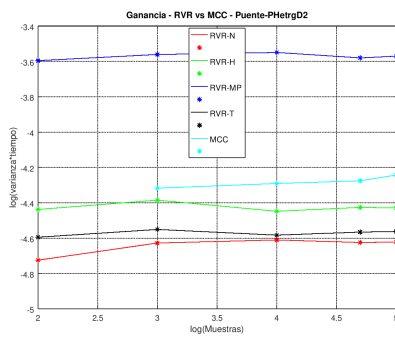
Figura 4.4: Análisis estadístico RVR vs MCC en Puente-PHetrgD2



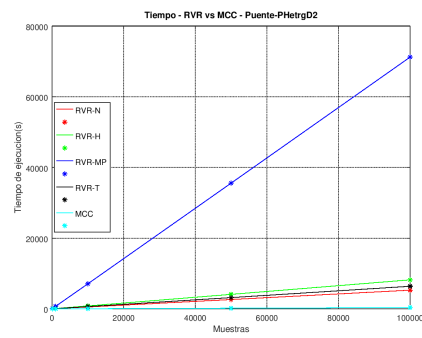
(a) Confiabilidad



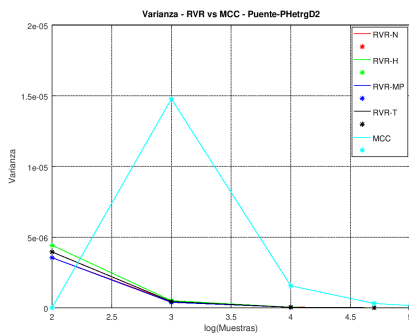
(b) Error relativo



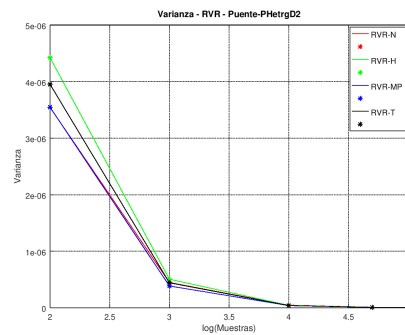
(c) Ganancia



(d) Tiempo de ejecución



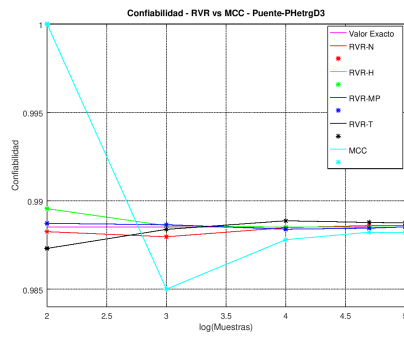
(e) Varianza



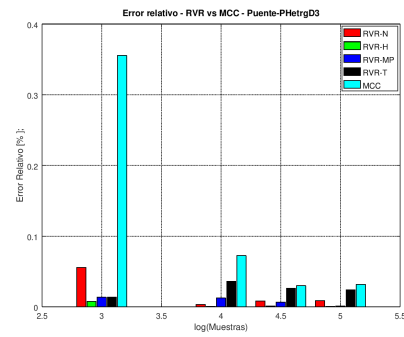
(f) Varianza por corte

En la Figura [Análisis estadístico RVR vs MCC en Puente-PHetrgD3](#) se grafica el rendimiento de RVR para las distintas estrategias de selección de cortes en función del tamaño de muestra, comparando estos contra la DCR exacta y MCC. En la Figura 4.5a se muestra el valor calculado de la confiabilidad diámetro acotada para el escenario. En la Figura 4.5b se grafica el error relativo En la Figura 4.5c se muestra la ganancia del producto tiempo por varianza. En la Figura 4.5d se grafica el tiempo de ejecución. En las Figuras 4.5e y 4.5f se grafican los valores de varianza.

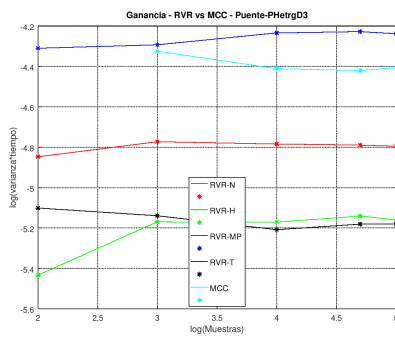
Figura 4.5: Análisis estadístico RVR vs MCC en Puente-PHetrgD3



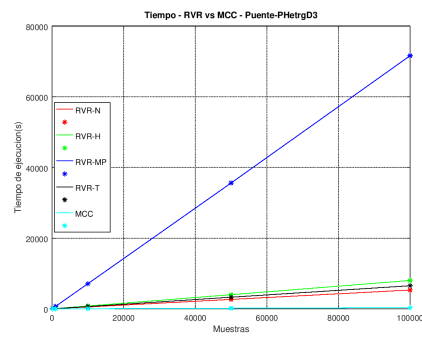
(a) Confiabilidad



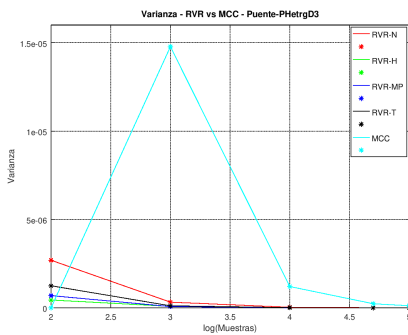
(b) Error relativo



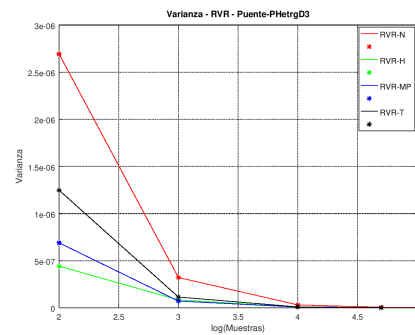
(c) Ganancia



(d) Tiempo de ejecución



(e) Varianza

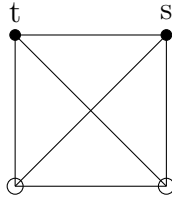


(f) Varianza por corte

4.2.2.2. Grafo Completo K_4

El grafo completo es caracterizado según la siguiente estructura, n vértices tiene $n(n - 1)/2$ aristas y se denota K_n . El valor de la confiabilidad exacta es fácilmente obtenido evaluando todo los estados posibles del grafo, permitiendo obtener resultados en bajo tiempo de cómputo y a su vez realizar verificaciones contra los métodos aproximados. Partiendo del caso de uso donde todos los enlaces están activos, se observa que el camino mínimo es de largo uno, y el siguiente es de dos saltos. Tomamos un modelo s - t -terminal con cota de salto $d \in \{2, 4\}$, estos valores permiten explorar caminos críticos según la topología.

Figura 4.6: Grafo Completo K_4



Se determina la siguiente notación para identificar los distintos escenarios o distintas instancias de corridas:

Tabla 4.4: Notación grafo completo

D	p_e/p_v	Notación
2	"0.95/0.95"	"Completo-PHomgD2"
2	"0.95/0.97"	"Completo-PHetrgD2"
4	"0.95/0.95"	"Completo-PHomgD4"
4	"0.95/0.97"	"Completo-PHetrgD4"

Tabla 4.5: Resultados de DCR exacta para grafo completo

Escenario	R_{exacta}	t (seg)
"Completo-PHomgD2"	0.998982905468750	0.51
"Completo-PHetrgD2"	0.999224053468749	0.6
"Completo-PHomgD4"	0.999176350703125	0.47
"Completo-PHetrgD4"	0.999425729503124	0.54

Resultados empíricos Completo-PHomgD2, Completo-PHomgD4

Como características generales observadas en las Tablas 3.9 y 3.7 (en Anexo 3), el cálculo de la DCR con RVR alcanza valores de varianza uno y dos órdenes menor (mayor precisión) que el obtenido con Monte Carlo Crudo, y muestra tiempos de cómputo uno y dos órdenes de magnitud mayor a MCC. Esto conlleva a que los valores de eficiencia relativa \bar{W} varíen entre las distintas estrategias planteadas. RVR_N , RVR_H y RVR_T obtienen valores de eficiencia relativa mayores a uno, mientras que RVR_{MP} menor a uno. Para todos los cortes elegidos RVR muestra valores de varianza que llega a órdenes de 10^{-10} y adicionalmente se registró similares tiempos de cómputo entre los escenarios.

Observando las gráficas de los valores obtenidos en confiabilidad (ver 4.7a, 4.8a) en conjunto con las figuras de varianza y error relativo (ver 4.7e, 4.8e 4.7b y 4.8b) se sobresalta la necesidad de una mayor cantidad de muestras de MCC para llegar a un valor estimado confiable. De estas tres figuras se puede apreciar que, a partir de muestras mayor o igual a 1000 la DCR calculada con RVR (para todos los cortes) tiende o converge al valor exacto, también para todos los cortes planteados la varianza disminuye (ver 4.7f y 4.8f) al aumentar la cantidad de muestras aumentando la precisión del método.

Se recalca la buena exactitud del método RVR_N (ver Tabla 3.8 en Anexo 3 y Figuras 4.7b y 4.8b), logrando un orden en error relativo de hasta 10^{-7} , adicionalmente en algunos casos se observa que ante un aumento de muestras el error conserva el orden obtenido (se acentúa a mayor cota), esto puede ser atribuido a la topología del grafo el cual cuenta con un camino mínimo de largo uno (corte con alta probabilidad de ocurrencia), lo que nos lleva cuestionarnos la importancia de implementar aleatoriedad entre los elementos elegidos en las distintas estrategias de separación planteadas, y preguntarnos si este factor puede ser mejorado aplicando técnicas especializadas en aleatoriedad. Por ejemplo, en las estrategias planteadas luego de encontrar el cutset inicial se podría haber intercambiar la posición de los elementos, para dar cierta aleatoriedad, antes de reducir este a un mincut. El comportamiento es más notorio para los métodos RVR_H , RVR_{MP} y RVR_T . Como complemento para muestras mayor o igual a mil, en el escenario Completo-PHomgD2, se obtiene un error relativo que varían entre $6.28 * 10^{-5} \%$ y 0.249% . Se destaca el error relativo de 0.00339% logrado para 1000 muestras de RVR_{MP} . Luego, para el escenario Completo-PHomgD4 se obtiene un rango de error relativo que varían entre

$1.488 * 10^{-4} \%$ y 0.0158% . Se destaca el error relativo de 0.00754% obtenido para 1000 muestras de RVR_H .

En las Figuras 4.7c y 4.8c se visualiza que para muestras mayores a 1000 todos los métodos mantienen un orden constante de ganancia. De la Figura 4.7c, se observa que, RVR_N consigue un orden menor en ganancia en contraste con MCC, imputando un mejor desempeño a RVR_N , luego RVR_H y RVR_T muestran una compensación en la medida conjunta (tiempo de cómputo y varianza) en comparación con MCC, mientras que RVR_{MP} logra un orden más en ganancia debido a su alto tiempo en cómputo, imputando un mejor desempeño a MCC. En la Figura 4.8c se visualiza que, RVR_N , RVR_H y RVR_T imputan mejor desempeño, mientras que RVR_{MP} sigue siendo menos eficiente en contraste con MCC. Comparando los datos anteriores, para una mayor cota los métodos RVR_H y RVR_T registraron una mejora en la medida de ganancia, esto es atribuido a las mejoras en precisión.

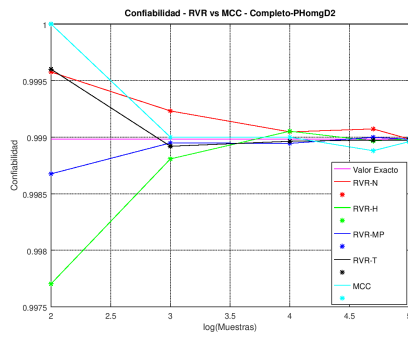
De las Figuras 4.7d y 4.8d se observa que, el tiempo de cómputo de RVR (para las distintas estrategias elegidas) presentó tiempos de ejecución lineal al número de muestras tomadas, en particular RVR_N , RVR_H y RVR_T obtienen menor pendiente en comparación a RVR_{MP} . Se identifica una mejor performance para el RVR_N .

Comparando los escenarios Completo-PHomgD2 y Completo-PHomgD4 (ver Tabla 3.9) contra la teoría, se cumple que los valores de DCR estimados por RVR para la cota $D = 2$ son menores que los alcanzados por $D = 4$. No se advierten vastas diferencias en tiempo de cómputo entre los escenarios. En relación a la cantidad de muestras, se logran valores de varianza de menor unidad para $D = 4$, tendiendo más rápido a su valor estimado y generando un mejor desempeño en la medida conjunta (tiempo en cómputo por varianza). Esto es atribuido a la disminución en la cardinalidad del espacio de conjuntos de corte para la cota $D = 4$.

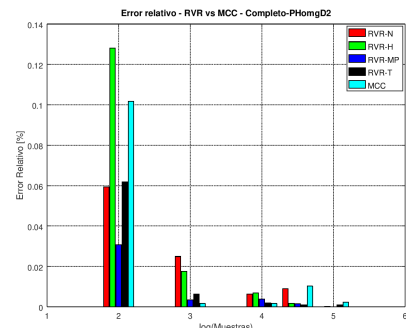
Para los escenarios Completo-PHomgD2 y Completo-PHomgD4 se destaca en general, la performance de RVR_N y RVR_H . Se recomienda 10^4 muestras de los métodos donde observamos una buena precisión, tiempo de cómputo y exactitud.

En la Figura [Análisis estadístico RVR vs MCC en Completo-PHomgD2](#) se grafica el rendimiento de RVR para las distintas estrategias de selección de cortes en función del tamaño de muestra, comparando estos contra la DCR exacta y MCC. En la Figura 4.7a se muestra el valor calculado de la confiabilidad diámetro acotada para el escenario. En la Figura 4.7b se grafica el error relativo En la Figura 4.7c se muestra la ganancia del producto tiempo por varianza. En la Figura 4.7d se grafica el tiempo de ejecución. En las Figuras 4.7e y 4.7f se grafican los valores de varianza.

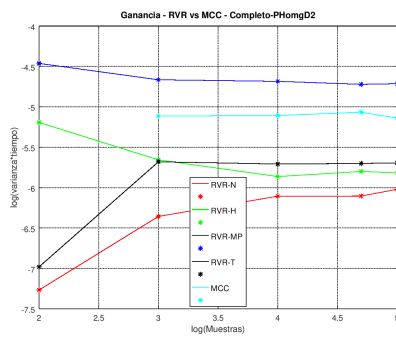
Figura 4.7: Análisis estadístico RVR vs MCC en Completo-PHomgD2



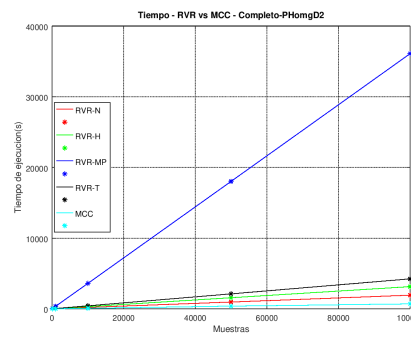
(a) Confiabilidad



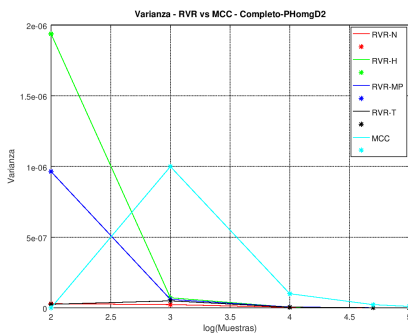
(b) Error relativo



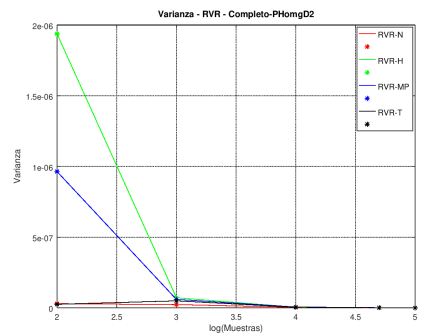
(c) Ganancia



(d) Tiempo de ejecución



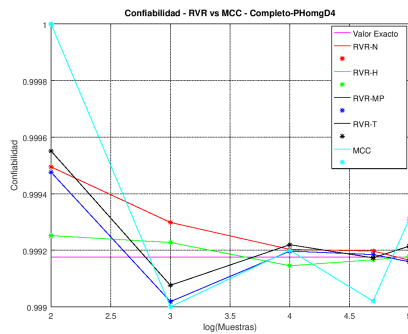
(e) Varianza



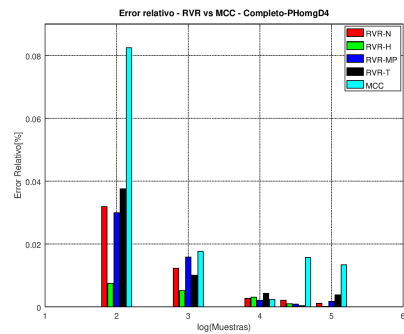
(f) Varianza por corte

En la Figura [Análisis estadístico RVR vs MCC en Completo-PHomgD4](#) se grafica el rendimiento de RVR para las distintas estrategias de selección de cortes en función del tamaño de muestra, comparando estos contra la DCR exacta y MCC. En la Figura 4.8a se muestra el valor calculado de la confiabilidad diámetro acotada para el escenario. En la Figura 4.8b se grafica el error relativo En la Figura 4.8c se muestra la ganancia del producto tiempo por varianza. En la Figura 4.8d se grafica el tiempo de ejecución. En las Figuras 4.8e y 4.8f se grafican los valores de varianza.

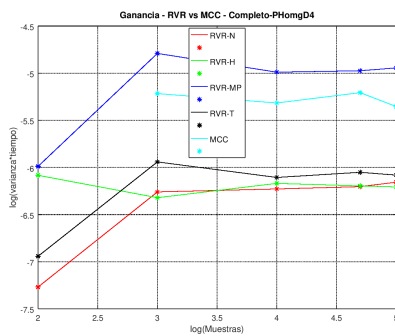
Figura 4.8: Análisis estadístico RVR vs MCC en Completo-PHomgD4



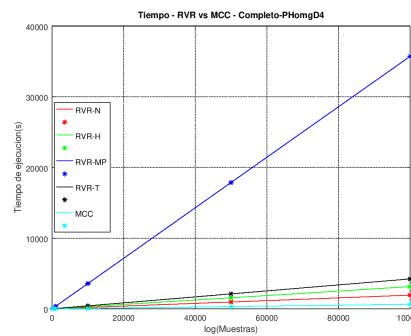
(a) Confiabilidad



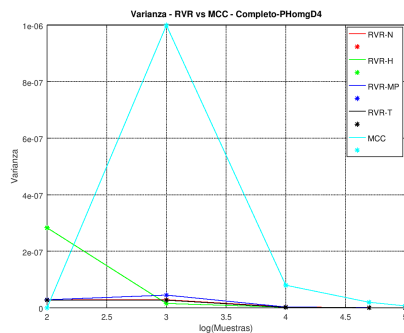
(b) Error relativo



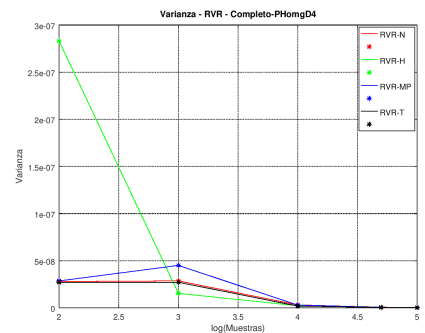
(c) Ganancia



(d) Tiempo de ejecución



(e) Varianza



(f) Varianza por corte

Resultados empíricos Completo-PHetrgD2, Completo-PHetrgD4

Como características generales observadas, el cálculo de la DCR con RVR logra valores de varianza uno y dos órdenes menor que el obtenido con Monte Carlo Crudo (ver Tablas 3.10 y 3.7 en Anexo 3), así como un tiempo de cómputo de uno y dos órdenes mayor a MCC. Esto conlleva a que los valores de eficiencia relativa \bar{W} varíen entre las distintas estrategias planteadas. RVR_N , RVR_H y RVR_T obtuvieron valores de eficiencia relativa mayores a uno, mientras que para RVR_{MP} los valores de \bar{W} se muestran variantes. Para todos los cortes elegidos RVR alcanza valores de varianza que llegan a órdenes de 10^{-10} y 10^{-11} menor y mayor cota respectivamente, y adicionalmente se registró similares tiempos de cómputo entre los escenarios.

Comparando los escenarios Completo-PHetrgD2 y Completo-PHetrgD4 (ver Tabla 3.10) contra la teoría, se llega a conclusiones similares en comparación a escenarios posteriores.

Observando las gráficas de los valores obtenidos en confiabilidad (ver 4.9a, 4.10a) en conjunto con las figuras de varianza y error relativo (ver 4.9e, 4.10e, 4.9b y 4.10b), se observan resultados similares en comparación a escenarios posteriores. Se recalca la buena exactitud del método RVR_N (ver Tabla 3.8 en Anexo 3 y Figuras 4.9b y 4.10b), logrando un orden en error relativo de hasta 10^{-6} , se llega conclusiones similares en comparación a escenarios posteriores.

En las Figuras 4.9c y 4.10c se visualiza que para muestras mayores a 1000 todos los métodos mantienen un orden constante de ganancia. De la Figura 4.9c, se observa que, RVR_N y RVR_H consiguen un orden menor en ganancia imputando mayor eficiencia que MCC, luego RVR_T muestra una compensación en la medida conjunta en comparación con MCC, mientras que RVR_{MP} logra un orden más en ganancia debido a su alto tiempo en cómputo, imputando un mejor desempeño a MCC. En la Figura 4.10c se visualiza que, RVR_N , RVR_H y RVR_T consiguen un mejor desempeño, mientras que RVR_{MP} logra menor eficiencia en contarse con MCC. Comparando los datos anteriores, para una mayor cota el método RVR_T registra una mejoría en la eficiencia de la medida conjunta, esto es atribuido a la mayor precisión registrada.

De las Figuras 4.9d y 4.10d, se observan resultados similares en tiempo de ejecución comparando escenarios posteriores.

Ahora en un marco general, comparando los valores obtenidos por RVR en

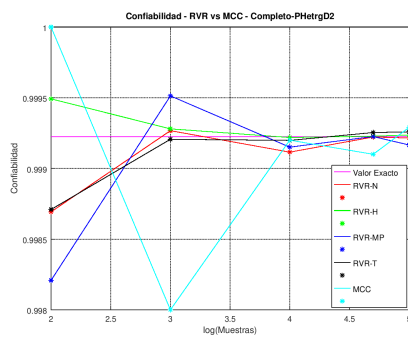
las instancias homogéneas contra heterogéneas (para la misma cota de diámetro) no se observan vastas diferencias en tiempo de cómputo, y en relación a la cantidad de muestras se observa valores de varianza de menor unidad para escenarios heterogéneos, tendiendo estos más rápido a su valor estimado, generando un mejor desempeño de la medida conjunta, desencadenando una mayor eficiencia relativa. Las diferencias encontradas pueden ser atribuidas a la sensibilidad del método RVR frente a la diferencia planteada. Cabe notar que en las estrategias de separación de grafos formalizadas no implementan una elección de elementos en base a su probabilidad elemental.

Al contar con una estructura particular el grafo nos permitió observar la sensibilidad que presenta RVR frente a la selección de cortes, construcción de cortes y topología del grafo. Se puede tener un algoritmo eficiente en tiempo pero la forma o estilo de implementación puede jugar un papel importante en otras variables (como por ejemplo la eficacia). El corte $x = \text{Minpath}(S, \phi, p)$ establece mejoras en tiempos de cómputo para RVR, pero a trabajo futuro se podría realizar un enfoque en exactitud aplicando técnicas especializadas en aleatoriedad.

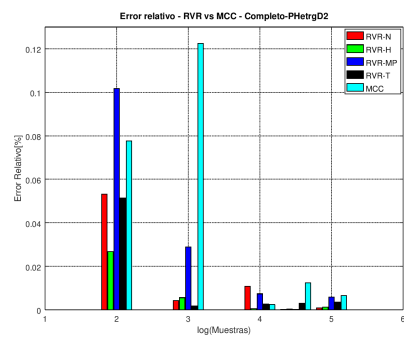
Se destaca en general para los escenarios planteados la performance de RVR_N y RVR_H . Se recomienda 10^4 muestras de los métodos donde observamos una buena precisión, tiempo de cómputo y exactitud.

En la Figura [Análisis estadístico RVR vs MCC en Completo-PHetrgD2](#) se grafica el rendimiento de RVR para las distintas estrategias de selección de cortes en función del tamaño de muestra, comparando estos contra la DCR exacta y MCC. En la Figura [4.9a](#) se muestra el valor calculado de la confiabilidad diámetro acotada para el escenario. En la Figura [4.9b](#) se grafica el error relativo En la Figura [4.9c](#) se muestra la ganancia del producto tiempo por varianza. En la Figura [4.9d](#) se grafica el tiempo de ejecución. En las Figuras [4.9e](#) y [4.9f](#) se grafican los valores de varianza.

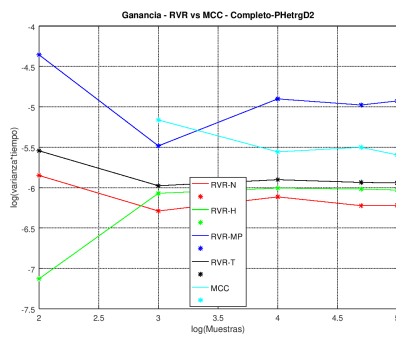
Figura 4.9: Análisis estadístico RVR vs MCC en Completo-PHetrgD2



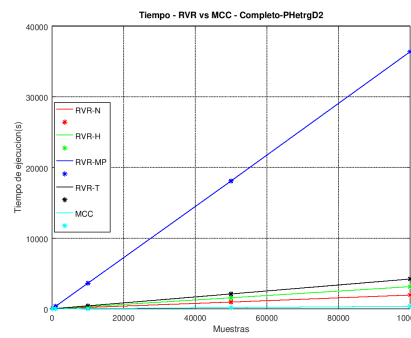
(a) Confiabilidad



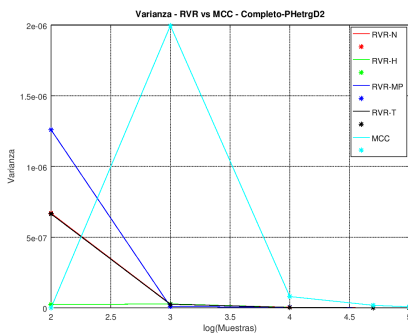
(b) Error relativo



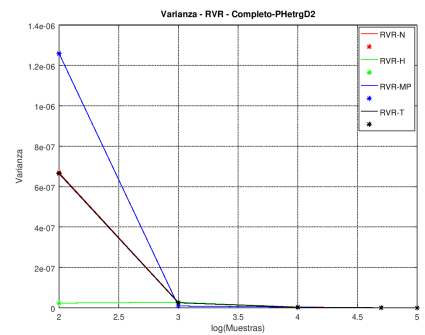
(c) Ganancia



(d) Tiempo de ejecución



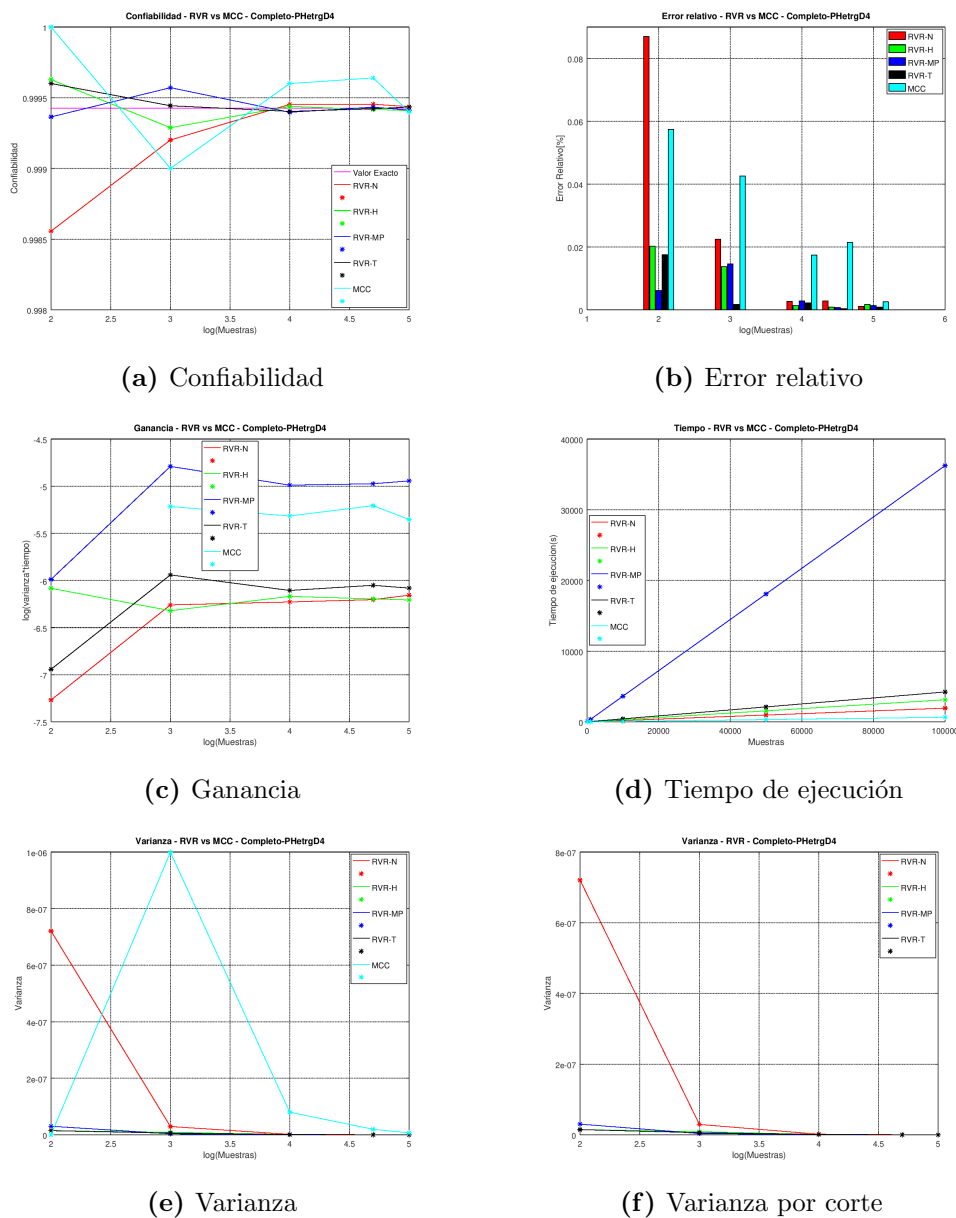
(e) Varianza



(f) Varianza por corte

En la Figura [Análisis estadístico RVR vs MCC en Completo-PHetrgD4](#) se grafica el rendimiento de RVR para las distintas estrategias de selección de cortes en función del tamaño de muestra, comparando estos contra la DCR exacta y MCC. En la Figura 4.10a se muestra el valor calculado de la confiabilidad diámetro acotada para el escenario. En la Figura 4.10b se grafica el error relativo En la Figura 4.10c se muestra la ganancia del producto tiempo por varianza. En la Figura 4.10d se grafica el tiempo de ejecución. En las Figuras 4.10e y 4.10f se grafican los valores de varianza.

Figura 4.10: Análisis estadístico RVR vs MCC en Completo-PHetrgD4



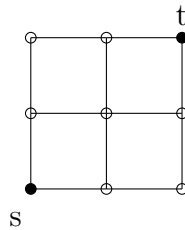
4.2.2.3. Grafo Grilla

Se considera el grafo grilla para dos casos según $n \in \{3, 5\}$. La topología del grafo en conjunto con el modelo s - t -terminal tiene la particularidad de tener varios caminos del mismo largo, permitiendo estudiar el desempeño de RVR ante las distintas estrategias planteadas.

Grafo Grilla, $n = 3$

El valor de la DCR exacta es obtenido fácilmente evaluando todos los estados posibles del grafo, esto nos permite comparar y verificar los resultados obtenidos por métodos aproximados. Dada su topología y partiendo del caso de uso donde todos los enlaces están activos, observamos varios caminos de largo cuatro. En la Figura 4.11 se muestra el grafo en el modelo s - t -terminal.

Figura 4.11: Grafo Grilla, $n = 3$



Se determina la siguiente notación para identificar los distintos escenarios o distintas instancias de corridas:

Tabla 4.6: Notación grafo grilla $n = 3$

D	p_e/p_v	Notación
4	"0.95/0.95"	"GrillaN3-PHomgD4"
4	"0.95/0.97"	"GrillaN3-PHetrgD4"

Tabla 4.7: Resultados de DCR exacta para grafo grilla $n = 3$

Escenario	R_{exacta}	t (seg)
"GrillaN3-PHomgD4"	0.973736522447238	53
"GrillaN3-PHetrgD4"	0.983384097586643	52

Resultados empíricos GrillaN3-PHomgD4,GrillaN3-PHetrgD4

Como características generales observadas, el cálculo de la DCR con RVR (para los distintos cortes) alcanza valores de varianza uno o dos órdenes menor que los obtenidos por MCC (ver Tablas 3.13, 3.14 y 3.11 en Anexo 3) y consigue un tiempo de cómputo de dos y tres órdenes mayor a MCC. Esto conlleva a que los valores de eficiencia relativa \bar{W} varíen entre las distintas estrategias planteadas. RVR_N , RVR_H y RVR_{MP} en relación a MCC obtuvieron valores de eficiencia relativa menores a uno, mientras que para RVR_T mayor a uno. Adicionalmente se registró similares tiempos de cómputo entre los escenarios, a excepción de RVR_{MP} donde se percibe un aumento para el caso heterogéneo.

Observando las gráficas de los valores obtenidos en confiabilidad (ver 4.12a, 4.13a) se puede apreciar la sensibilidad que presenta RVR ante estrategias que recorren todos los posibles conjuntos de corte de un grafo, contra una estrategia de corte que se centra en aislar nodos terminales. Al aumentar la cantidad de muestras en estos primeros la estimación de la DCR tiende al valor exacto calculado, mientras que para RVR_T la confiabilidad estimada se distancia sin acercarse a su valor exacto. Sumando a la anterior las figuras de varianza y error relativo (ver 4.12e, 4.13e, 4.12b y 4.13b) se sobrealta la necesidad de una mayor cantidad de muestras de MCC para llegar a un valor estimado confiable. Por otro lado, para todos los cortes planteados (ver 4.12b, 4.13b 4.12f y 4.13f) la varianza disminuye al aumentar la cantidad de muestras aumentando la precisión del método, logra valores de varianza del orden de 10^{-8} y para algunos casos hasta 10^{-9} , RVR_T muestra la menor varianza en relación a las muestras indicando una buena precisión, pero un vasto resultado en exactitud (ver Tabla 3.12 en Anexo 3).

Contar con el valor exacto de la confiabilidad diámetro acotada nos permite realizar un análisis de exactitud (ver Tabla 3.12 en Anexo 3 y Figuras 4.12b y 4.13b) del método RVR frente a los distintos cortes. Se recalca la buena exactitud del método RVR_N , logrando un orden en error relativo de hasta 10^{-5} , adicionalmente en algunos casos se observa que ante un aumento de muestras el error conserva el orden obtenido. El comportamiento se degrada para los métodos RVR_H , RVR_{MP} y RVR_T , donde se observa casos aislados de aumento de orden ante un aumento de muestras. Como se concluyó en grafos anteriores, esto puede estar asociado a cierto déficit de aleatoriedad en la implementación de las estrategias.

En las Figuras 4.12c y 4.13c se visualiza que todos los métodos presentan un incremento de ganancia ante el aumento de muestras, induciendo un mayor consumo de recursos. RVR_T consigue un orden menor en ganancia en contraste con MCC, imputando un mejor desempeño. Luego RVR_N y RVR_H logran el mismo orden que MCC, alcanzando una compensación en la medida conjunta, mientras que RVR_{MP} debido a su alto tiempo en cómputo sigue siendo menos eficiente en comparación con MCC.

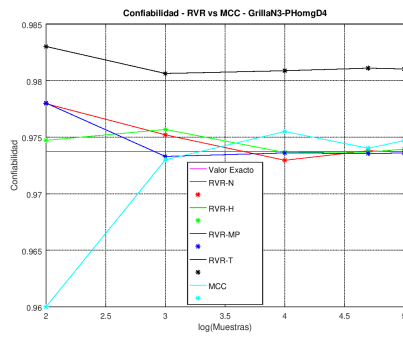
De las Figuras 4.12d y 4.13d se observa que, el tiempo de cómputo de RVR (para las distintas estrategias elegidas) presentó tiempos de ejecución lineal al número de muestras tomadas, en particular RVR_N , RVR_H y RVR_T obtienen menor pendiente en comparación a RVR_{MP} . Se identifica una mejor performance para el RVR_N .

Ahora en un marco general, comparando los valores obtenidos por RVR en las instancias homogéneas contra heterogéneas (para la misma cota de diámetro) no se observan vastas diferencias en tiempo de cómputo, y en relación a la cantidad de muestras se observa valores de varianza de menor unidad para escenarios heterogéneos, tendiendo estos más rápido a su valor estimado, generando un mejor desempeño de la medida conjunta, desencadenando una mayor eficiencia relativa. Las diferencias encontradas pueden ser atribuidas a la sensibilidad del método RVR frente a la diferencia planteada. Cabe notar que en las estrategias de separación de grafos formalizadas no implementan una elección de elementos en base a su probabilidad elemental.

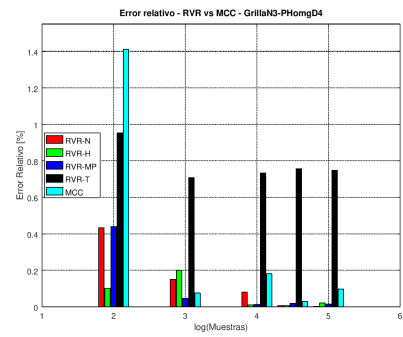
Se destaca en general para los escenarios planteados la performance de RVR_N y RVR_H . Se recomienda 10^3 y 10^4 muestras de los métodos donde observamos una buena precisión, tiempo de cómputo y exactitud.

En la Figura [Análisis estadístico RVR vs MCC en GrillaN3-PHomgD4](#) se grafica el rendimiento de RVR para las distintas estrategias de selección de cortes en función del tamaño de muestra, comparando estos contra la DCR exacta y MCC. En la Figura [4.12a](#) se muestra el valor calculado de la confiabilidad diámetro acotada para el escenario. En la Figura [4.12b](#) se grafica el error relativo En la Figura [4.12c](#) se muestra la ganancia del producto tiempo por varianza. En la Figura [4.12d](#) se grafica el tiempo de ejecución. En las Figuras [4.12e](#) y [4.12f](#) se grafican los valores de varianza.

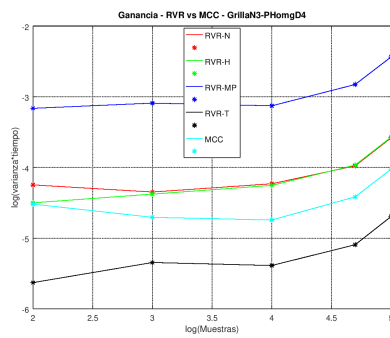
Figura 4.12: Análisis estadístico RVR vs MCC en GrillaN3-PHomgD4



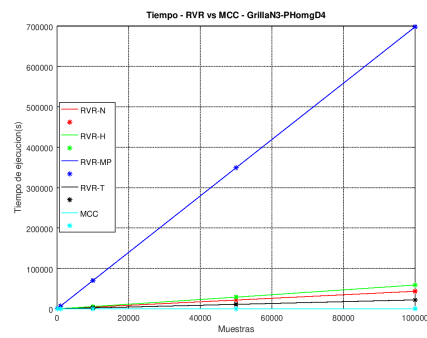
(a) Confiabilidad



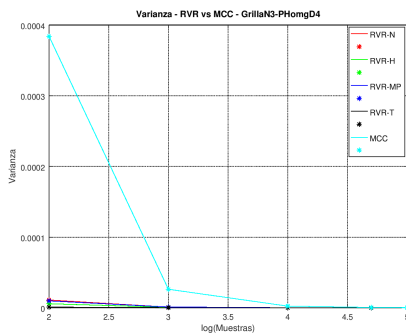
(b) Error relativo



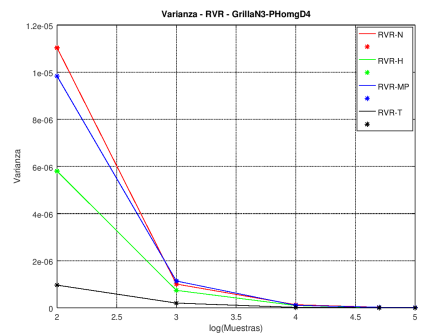
(c) Ganancia



(d) Tiempo de ejecución



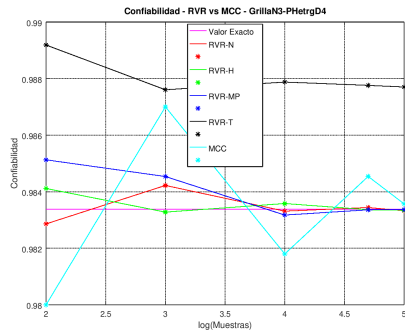
(e) Varianza



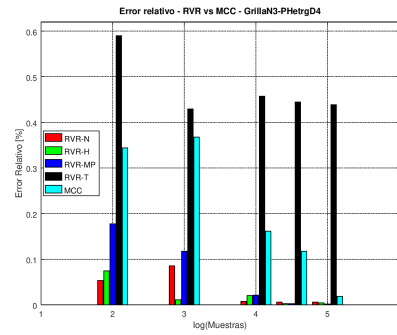
(f) Varianza por corte

En la Figura [Análisis estadístico RVR vs MCC en GrillaN3-PHetrgD4](#) se grafica el rendimiento de RVR para las distintas estrategias de selección de cortes en función del tamaño de muestra, comparando estos contra la DCR exacta y MCC. En la Figura 4.13a se muestra el valor calculado de la confiabilidad diámetro acotada para el escenario. En la Figura 4.13b se grafica el error relativo En la Figura 4.13c se muestra la ganancia del producto tiempo por varianza. En la Figura 4.13d se grafica el tiempo de ejecución. En las Figuras 4.13e y 4.13f se grafican los valores de varianza.

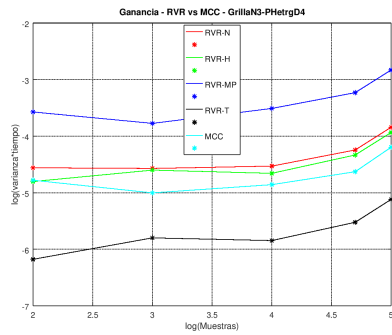
Figura 4.13: Análisis estadístico RVR vs MCC en GrillaN3-PHetrgD4



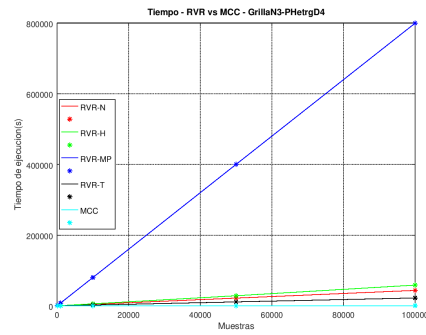
(a) Confiabilidad



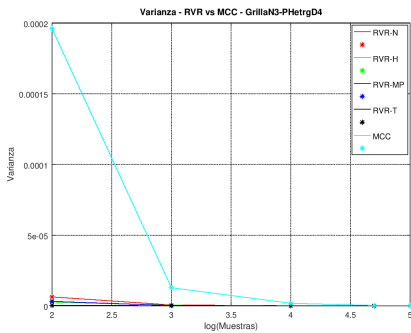
(b) Error relativo



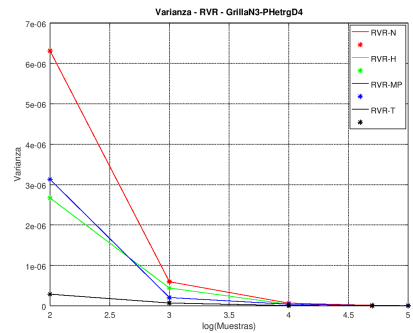
(c) Ganancia



(d) Tiempo de ejecución



(e) Varianza

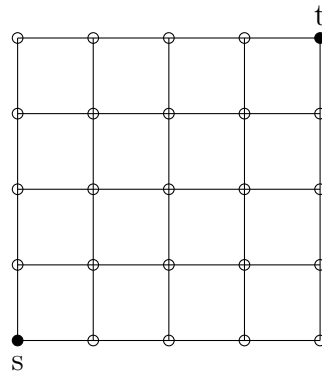


(f) Varianza por corte

Grafo Grilla, $n = 5$

Las dimensiones del grafo hacen que el cálculo de la confiabilidad exacta por métodos de fuerza bruta sea computacionalmente prohibitivo. Dada la topología simétrica y partiendo del caso de uso donde todos los enlaces y nodos no terminales están activos, observamos que el camino mínimo entre fuente terminal es de largo ocho. El escenario bajo estudio es un modelo s - t -terminal con cota de salto $D = 8$. Dada la simetría del grafo no se estudian otros casos de diámetro.

Figura 4.14: Grafo Grilla $n = 5$



Se determina la siguiente notación para identificar los distintos escenarios o distintas instancias de corridas:

Tabla 4.8: Notación grafo grilla $n = 5$

D	p_e/p_v	Notación
8	"0.95/0.95"	"GrillaN5-PHomgD8"
8	"0.95/0.97"	"GrillaN5-PHetrgD8"

Resultados empíricos GrillaN5-PHomgD8, GrillaN5-PHetrgD8.

Como características generales observadas para los distintos cortes RVR alcanza valores de varianza uno y dos ordenes menor que el obtenido por Monte Carlo Crudo (ver Tablas 3.16, 3.17 y 3.15 en Anexo 3), sin embargo RVR obtiene un tiempo de cómputo de tres y cuatro órdenes mayor a MCC. Esto conlleva a que la eficiencia relativa \bar{W} muestre valores menores a uno para todos los cortes planteados, influyendo en el indicador el tiempo de procesamiento de RVR. Para 10 muestras del método RVR_{MP} se observaron tiempos de cómputo de una hora, limitando el procesamiento para muestras grandes. Por otro lado, no se aportan datos del método RVR_T dada la ineficacia mostrada en grafos anteriores.

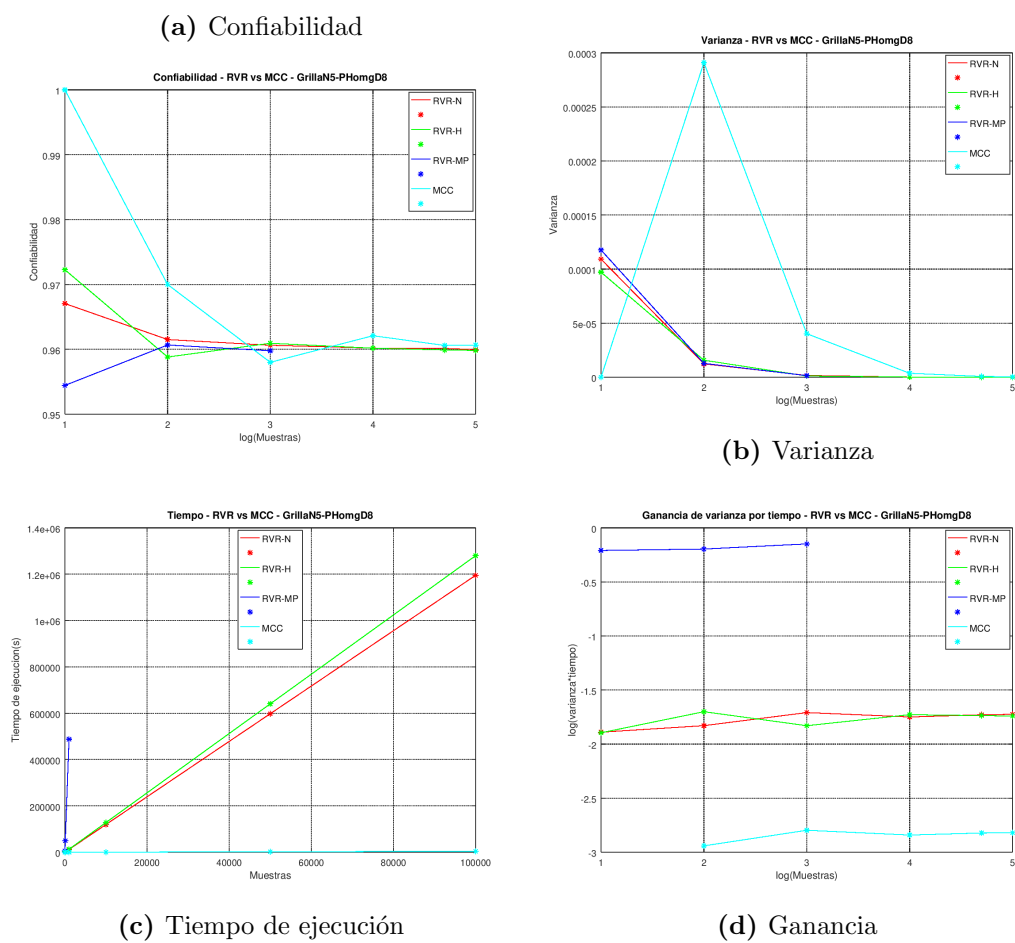
De las Figuras 4.15a y 4.16a se puede observar que, todos los métodos de aproximación tienden a un valor estimado de la confiabilidad para los escenarios planteados. Sumando a la anterior las Figuras 4.15b y 4.16b se sobresaalta la necesidad de una mayor cantidad de muestras de MCC para llegar a un valor estimado confiable. De las Figuras 4.15c y 4.16c se observa tiempos lineales de ejecución sin diferencias entre caso homogéneo y heterogéneo. En las Figuras 4.15d y 4.16d se visualiza que para el caso homogéneo de probabilidad RVR_N y RVR_H superan en un orden a MCC, mientras que para el caso heterogéneo RVR_N y RVR_H este indicador muestra una compensación entre tiempo de cómputo y la varianza que se obtiene en comparación con MCC, esto es atribuido a la disminución de la varianza de RVR en este escenario.

En conjunto se evidencia que MCC es menos demandante desde el punto de vista computacional pero la calidad de la aproximación es inestable requiriendo una mayor cantidad de muestras. En cambio RVR muestra buena exactitud, precisión y un buen comportamiento en la medida conjunta “tiempo por varianza” para todo el conjunto de muestras y escenarios.

Para este escenario se destaca la performance en precisión de RVR en comparación con MCC, compensando el tiempo de cómputo obtenido. Se recomienda RVR_N y RVR_H con un valor de 10^4 muestras.

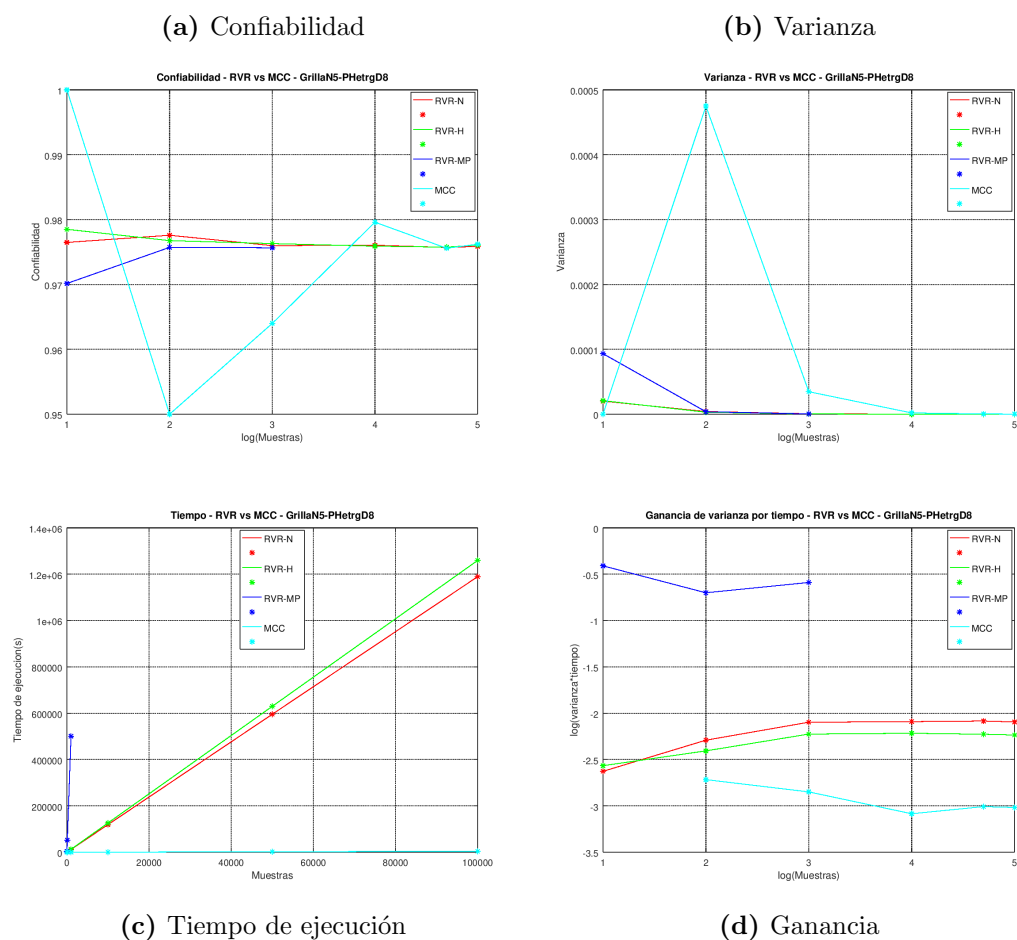
En la Figura [Análisis estadístico RVR vs MCC en GrillaN5-PHomgD8](#) se grafica el rendimiento de las distintas estrategias de selección de cortes en función del tamaño de muestra, y se compara los resultados obtenidos de RVR contra MCC. En la Figura 4.15a se muestra el valor calculado de la confiabilidad diámetro acotada, en la Figura 4.15b se grafican los valores de varianza, en la Figura 4.15c se grafica el tiempo de ejecución de RVR y en la Figura 4.15d se muestra la ganancia del producto tiempo por varianza.

Figura 4.15: Análisis estadístico RVR vs MCC en GrillaN5-PHomgD8



En la Figura [Análisis estadístico RVR vs MCC en GrillaN5-PHetrgD8](#) se grafica el rendimiento de las distintas estrategias de selección de cortes en función del tamaño de muestra, y se compara los resultados obtenidos de RVR contra MCC. En la Figura 4.16a se muestra el valor calculado de la confiabilidad diámetro acotada, en la Figura 4.16b se grafican los valores de varianza, en la Figura 4.16c se grafica el tiempo de ejecución y en la Figura 4.16d se muestra la ganancia del producto tiempo por varianza.

Figura 4.16: Análisis estadístico RVR vs MCC en GrillaN5-PHetrgD8

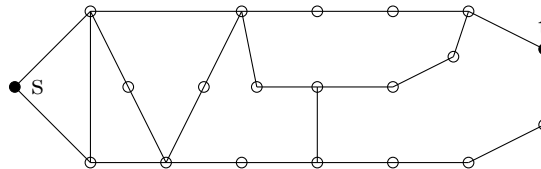


Cabe destacar que en las estrategias de separación de grafos formalizadas no se implementa una elección de elementos en base a su probabilidad elemental, por lo que las diferencias encontradas entre los casos homogéneo y heterogéneo (del mismo diámetro) puede ser atribuido a la sensibilidad de los métodos RVR y MCC frente a la diferencia planteada.

4.2.2.4. Grafo Arpanet

El grafo arpanet es un caso conocido y estudiado dentro de teoría de grafos. Las dimensiones del grafo hacen que el cálculo de la confiabilidad exacta por métodos de fuerza bruta sea computacionalmente prohibitivo. Partiendo del caso de uso donde todos los enlaces están activos, se observa que el camino mínimo es de largo seis, y el subsiguiente de largo ocho. Para el grafo se toma un modelo s - t -terminal con cota de saltos $D = 8$, valor que permite explorar caminos críticos según la topología.

Figura 4.17: Grafo Arpanet



Se determina la siguiente notación para identificar los distintos escenarios o distintas instancias de corridas:

Tabla 4.9: Notación grafo arpanet

D	p_e/p_v	Notación
8	"0.95/0.95"	"Arpanet-PHomgD8"
8	"0.95/0.97"	"Arpanet-PHetrgD8"

Resultados empíricos Arpanet-PHomgD8, Arpanet-PHetrgD8.

Como características generales observadas en los escenarios planteados, para los distintos cortes RVR alcanza valores de varianza un orden menor que el obtenido por Monte Carlo Crudo (ver Tablas 3.19, 3.20 y 3.18 en Anexo 3), sin embargo RVR consigue un tiempo de cómputo de dos y tres órdenes mayor a MCC. Esto conlleva a que la eficiencia relativa \bar{W} muestre valores menor a uno para todos los cortes planteados, influyendo en el indicador el tiempo de procesamiento de RVR, que se acentúa en RVR_{MP} . Por otro lado, no se aportan datos del método RVR_T dada la ineficacia mostrada en grafos anteriores.

De las Figuras 4.18a y 4.19a se puede observar que todos los métodos de aproximación (RVR y MCC) tienden a un valor estimado de la confiabilidad en los escenarios planteados. En las Figuras 4.18b y 4.19b se sobresa la necesidad de una mayor cantidad de muestras de MCC para llegar a un valor estimado confiable. De las Figuras 4.18c y 4.19c se observa tiempos lineales de ejecución respecto a la cantidad de muestras. En las Figuras 4.18d y 4.19d se visualiza que todos los métodos obtienen un orden constante en la medida conjunta, y RVR_N y RVR_H supera en un orden a MCC.

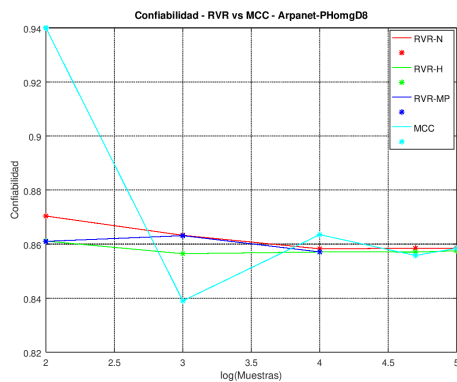
En conjunto se observa que MCC es menos demandante desde el punto de vista computacional, pero la aproximación es imprecisa llevando a necesitar una mayor cantidad de muestras. En cambio RVR muestra buena exactitud, precisión y un buen comportamiento en la medida conjunta “tiempo por varianza” para todo el conjunto de muestras.

Para los escenarios se destaca la performance en precisión de RVR para pocas muestras del método, compensando el tiempo de cómputo obtenido. Se recomienda RVR_N y RVR_H con un valor de 10^4 muestras.

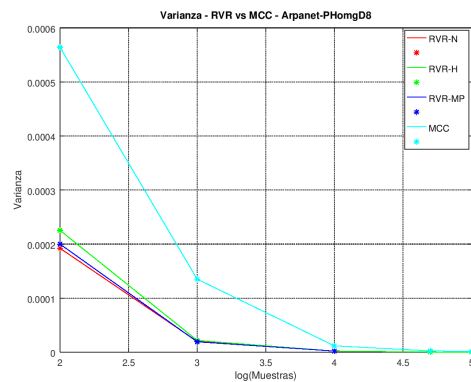
En la Figura [Análisis estadístico RVR vs MCC en Arpanet-PHomgD8](#) se grafica el rendimiento de las distintas estrategias de selección de cortes en función del tamaño de muestra, y se compara los resultados obtenidos de RVR contra MCC. En la Figura 4.18a se muestra el valor calculado de la confiabilidad diámetro acotada, en la Figura 4.18b se grafican los valores de varianza, en la Figura 4.18c se grafica el tiempo de ejecución y en la Figura 4.18d se muestra la ganancia para la medida conjunta “tiempo por varianza”.

Figura 4.18: Análisis estadístico RVR vs MCC en Arpanet-PHomgD8

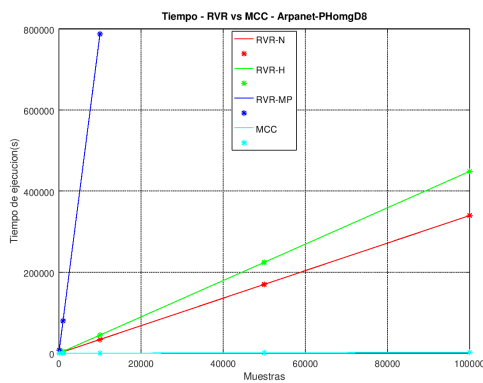
(a) Confiabilidad



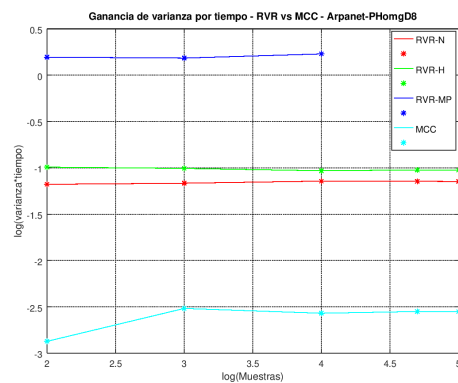
(b) Varianza



(c) Tiempo de ejecución



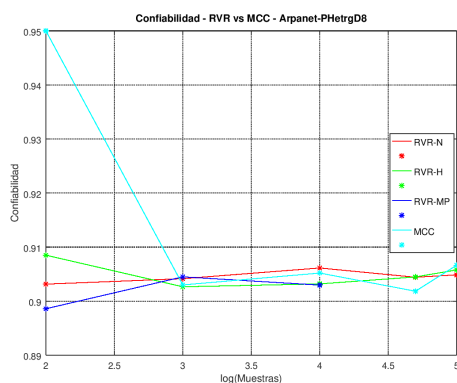
(d) Ganancia



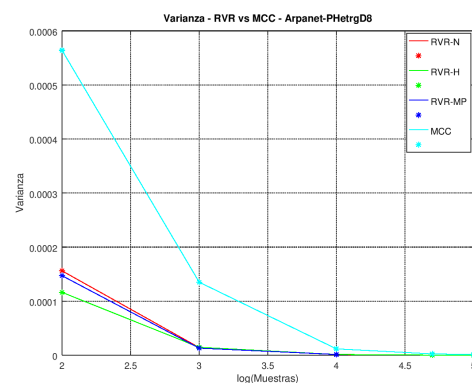
En la Figura [Análisis estadístico RVR vs MCC en Arpanet-PHetrgD8](#) se grafica el rendimiento de las distintas estrategias de selección de cortes en función del tamaño de muestra, y se compara los resultados obtenidos de RVR contra MCC. En la Figura 4.19a se muestra el valor calculado de la confiabilidad diámetro acotada, en la Figura 4.19b se grafican los valores de varianza, en la Figura 4.19c se grafica el tiempo de ejecución y en la Figura 4.19d se muestra la ganancia para la medida conjunta ‘tiempo por varianza’.

Figura 4.19: Análisis estadístico RVR vs MCC en Arpanet-PHetrgD8

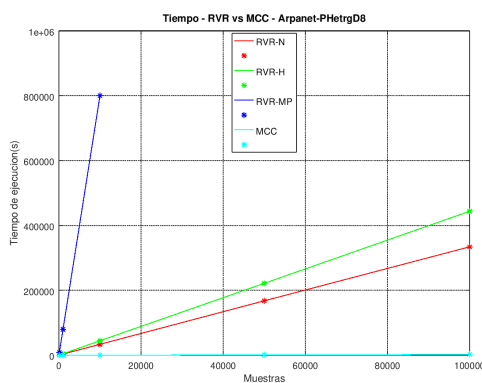
(a) Confiabilidad



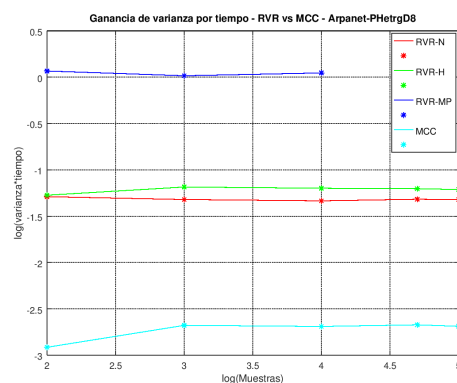
(b) Varianza



(c) Tiempo de ejecución



(d) Ganancia

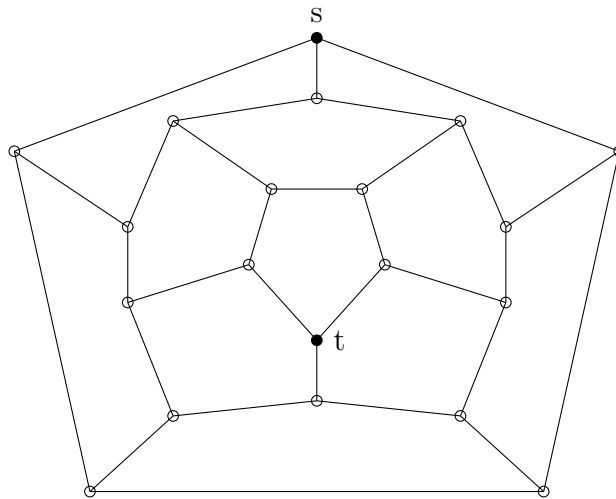


Cabe destacar que en las estrategias de separación de grafos formalizadas no se implementa una elección de elementos en base a su probabilidad elemental, por lo que las diferencias que puedan ser encontradas entre el caso homogéneo y el heterogéneo (del mismo diámetro) puede ser atribuido a la sensibilidad del método RVR y MCC frente a la diferencia planteada.

4.2.2.5. Grafo Dodecaedro

El grafo dodecaedro es un caso conocido y estudiado dentro de la literatura. Las dimensiones del grafo hacen que el cálculo de la confiabilidad exacta por métodos de fuerza bruta sea computacionalmente prohibitivo. Dada su topología y partiendo del caso de uso donde todos los enlaces están activos, se observa múltiples caminos de largo cinco y de largo ocho. Se toma un modelo s - t -terminal (ver Figura 4.20) con cota de saltos $D \in \{5, 8\}$, valores que permite explorar caminos críticos según la topología.

Figura 4.20: Grafo Dodecaedro



Se determina la siguiente notación para identificar los distintos escenarios o distintas instancias de corridas:

Tabla 4.10: Notación grafo dodecaedro

D	p_e/p_v	Notación
5	"0.95/0.95"	"Dodecaedro-PHomgD5"
5	"0.95/0.97"	"Dodecaedro-PHetrgD5"
8	"0.95/0.95"	"Dodecaedro-PHomgD8"
8	"0.95/0.97"	"Dodecaedro-PHetrgD8"

Resultados empíricos.

Como características generales observadas en todos los escenarios planteados, RVR_N y RVR_H obtuvieron valores de varianza uno y dos órdenes menor que el obtenido por Monte Carlo Crudo (ver Tablas 3.22, 3.23 y 3.21 en Anexo 3), y obtienen un tiempo de cómputo de uno y dos órdenes mayor a MCC.

Esto conlleva a que la eficiencia relativa \bar{W} muestre valores menores a uno para todas las muestras tomadas. Para RVR_{MP} el tiempo de cómputo es tres a cuatro órdenes mayor, mostrando una degradación en eficiencia relativa, para 100 muestras del método se observan tiempos de cómputo de casi 4 horas, limitando el procesamiento para muestras grandes. No se aportan datos de RVR_T dada la ineficacia mostrada en grafos anteriores.

Observando las gráficas de los valores obtenidos en confiabilidad (ver 4.21a, 4.22a, 4.23a, 4.24a) y comparando RVR contra MCC, se puede observar que todos los métodos de aproximación tienden a un valor estimado de la confiabilidad. Sumando a la anterior las figuras de varianza (ver 4.21b, 4.22b, 4.23b, 4.24b) se sobresaalta la necesidad de una mayor cantidad de muestras de MCC para llegar a un valor estimado confiable. De las Figuras 4.21c, 4.22c, 4.23c y 4.24c se observa tiempos lineales de ejecución para todos los escenarios, y una inferior magnitud para casos de menor cotas. De las figuras de ganancia (ver 4.21d, 4.22d, 4.23d, 4.24d) se evidencia que MCC es menos demandante desde el punto de vista computacional pero la calidad de la aproximación es inestable y en algunos casos muestra nefasta medida conjunta “tiempo por varianza” (dependiendo del escenario) acentuándose para pocas muestras del método. Por otro lado RVR marca una buena relación entre la calidad de la aproximación y tiempo de cómputo para todo el conjunto de muestras y escenarios.

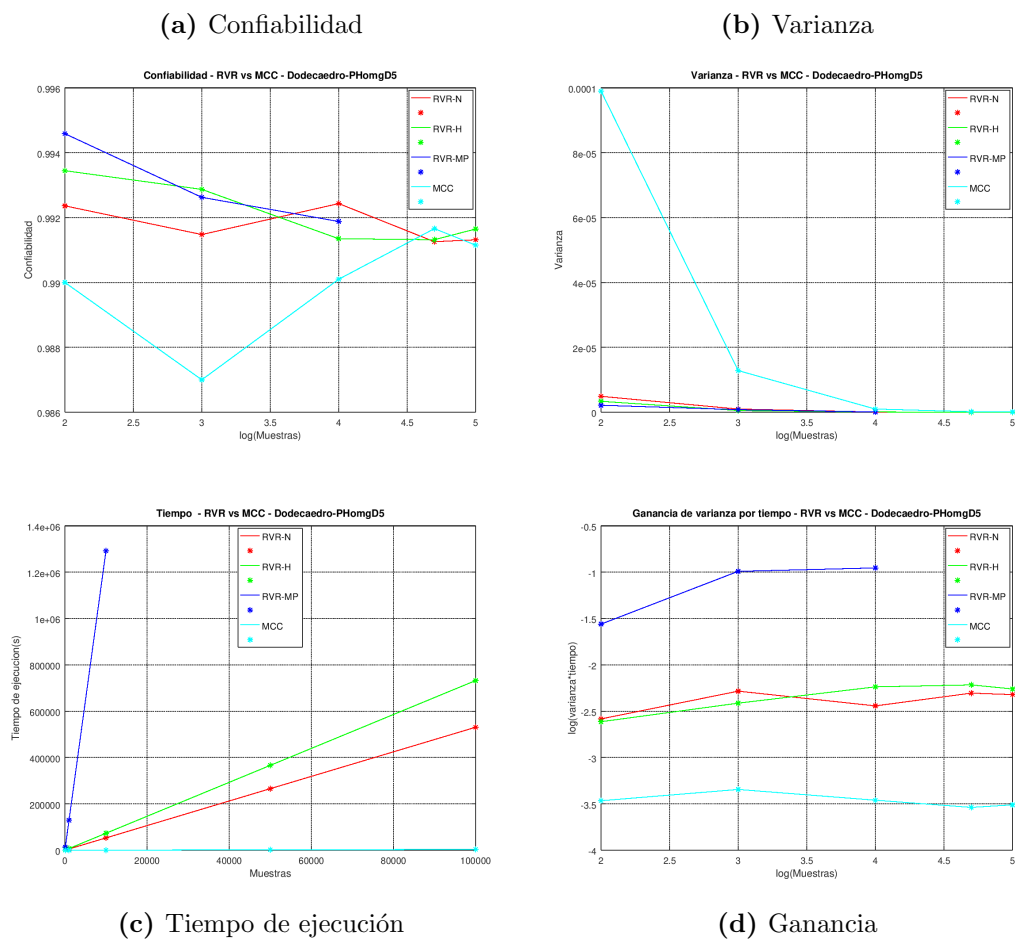
Comparando los escenarios contra la teoría, se cumple que los valores de DCR estimados por RVR para la cota $D = 5$ son menores que los alcanzados para $D = 8$. Se llega conclusiones similares a grafos anteriores.

Ahora en un marco general, comparando los valores obtenidos por RVR en las instancias homogéneas contra heterogéneas (para la misma cota de diámetro), se llega conclusiones similares a grafos anteriores.

Para los escenarios se destaca la performance en precisión de RVR en comparación con MCC, compensando el tiempo de cómputo obtenido. Se recomienda RVR_N y RVR_H con un valor de 10^4 muestras.

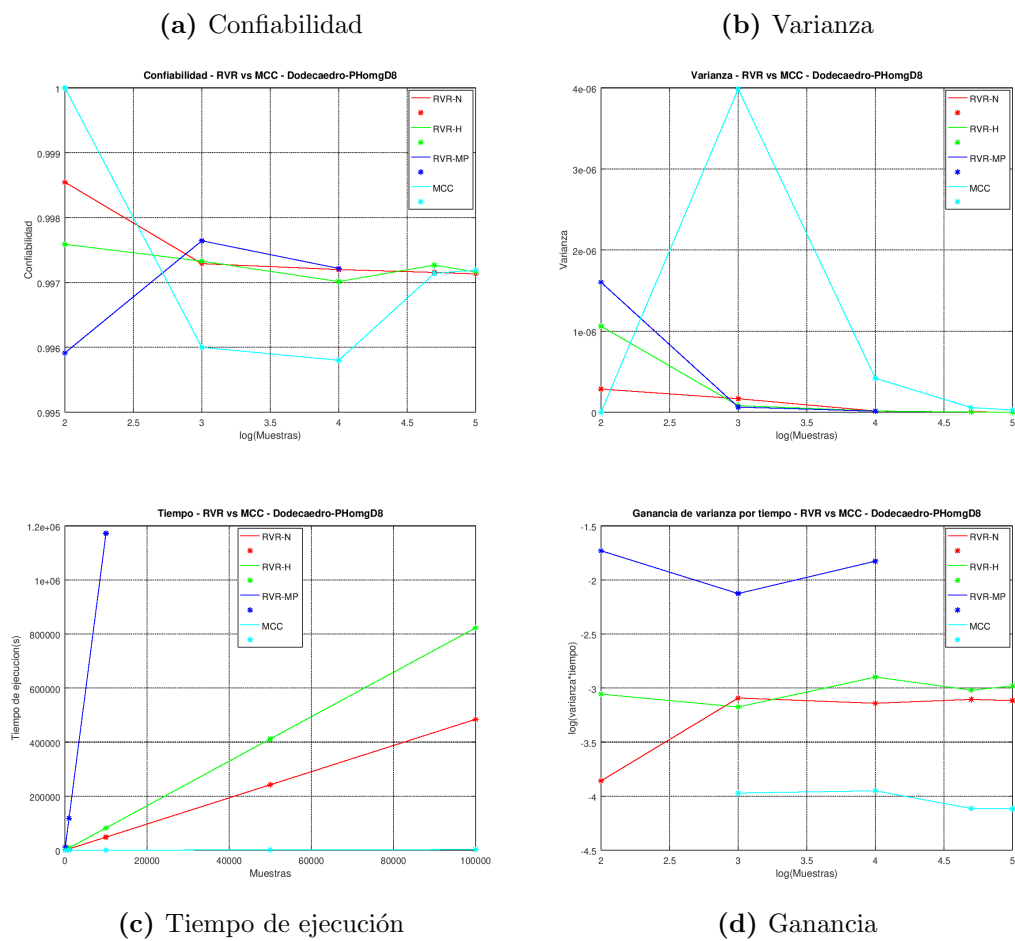
En la Figura [Análisis estadístico RVR vs MCC en Dodecaedro-PHomgD5](#) se grafica el rendimiento de las distintas estrategias de selección de cortes en función del tamaño de muestra, y se compara los resultados obtenidos de RVR contra MCC. En la Figura 4.21a se muestra el valor calculado de la confiabilidad diámetro acotada, en la Figura 4.21b se grafican los valores de varianza, en la Figura 4.21c se grafica el tiempo de ejecución, y en la Figura 4.21d se muestra la ganancia para la medida conjunta “tiempo por varianza”.

Figura 4.21: Análisis estadístico RVR vs MCC en Dodecaedro-PHomgD5



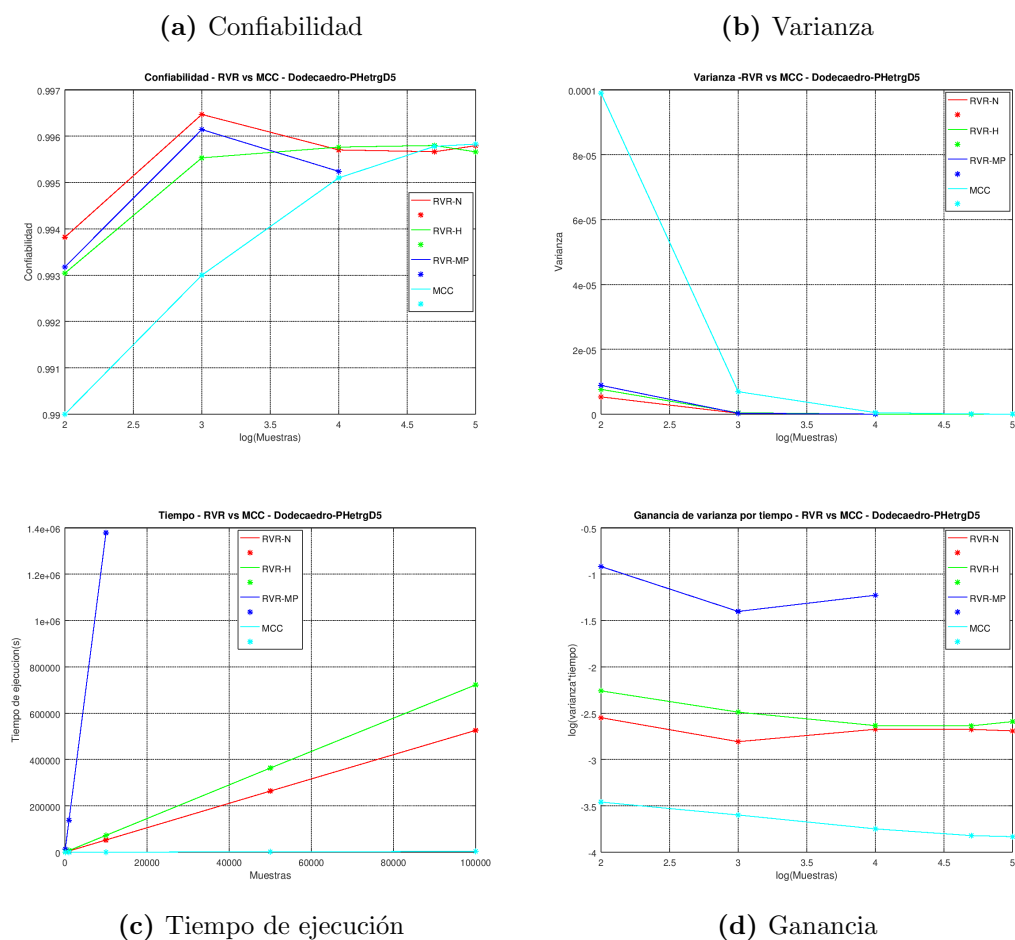
En la Figura [Análisis estadístico RVR vs MC en Dodecaedro-PHomgD8](#) se grafica el rendimiento de las distintas estrategias de selección de cortes en función del tamaño de muestra, y se compara los resultados obtenidos de RVR contra MCC. En la Figura 4.22a se muestra el valor calculado de la confiabilidad diámetro acotada, en la Figura 4.22b se grafican los valores de varianza, en la Figura 4.22c se grafica el tiempo de ejecución, y en la Figura 4.22d se muestra la ganancia para la medida conjunta “tiempo por varianza”.

Figura 4.22: Análisis estadístico RVR vs MC en Dodecaedro-PHomgD8



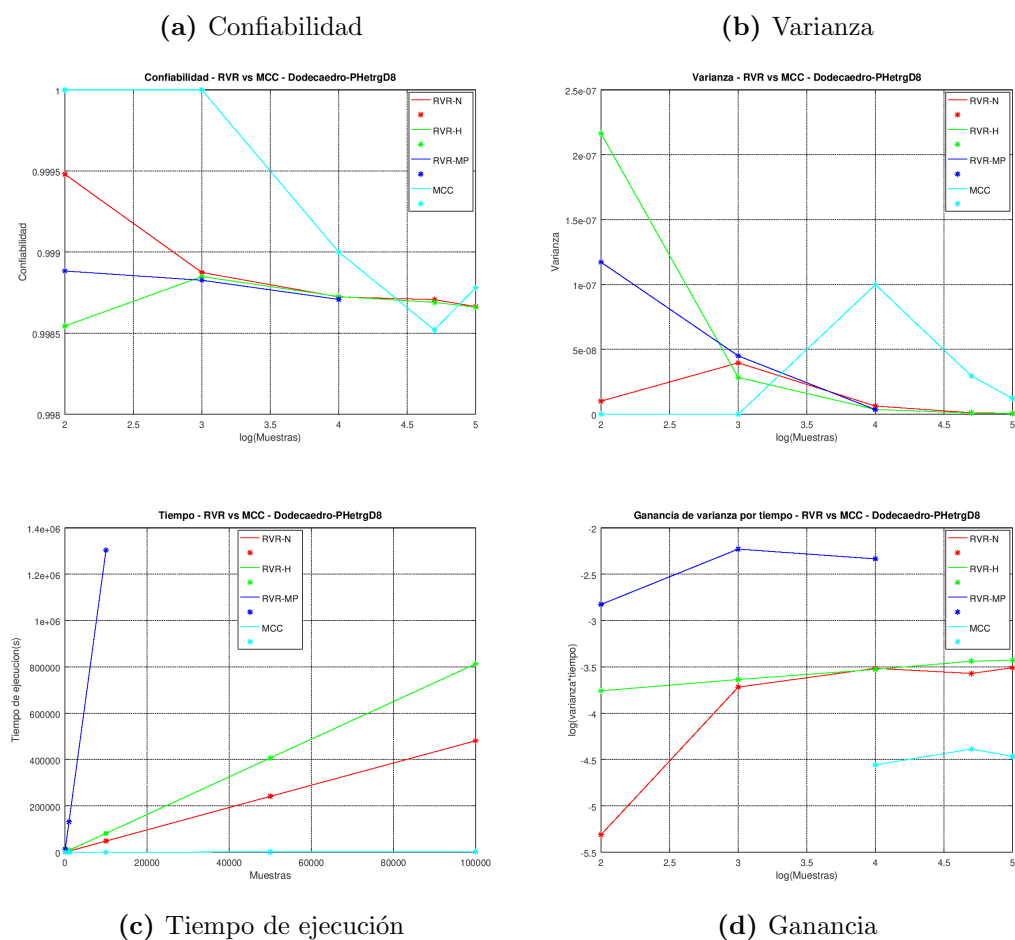
En la Figura [Análisis estadístico RVR vs MCC en Dodecaedro-PHetrgD5](#) se grafica el rendimiento de las distintas estrategias de selección de cortes en función del tamaño de muestra, y se compara los resultados obtenidos de RVR contra MCC. En la Figura 4.23a se muestra el valor calculado de la confiabilidad diámetro acotada, en la Figura 4.23b se grafican los valores de varianza, en la Figura 4.23c se grafica el tiempo de ejecución, y en la Figura 4.23d se muestra la ganancia para la medida conjunta “tiempo por varianza”.

Figura 4.23: Análisis estadístico RVR vs MCC en Dodecaedro-PHetrgD5



En la Figura [Análisis estadístico RVR vs MCC en Dodecaedro-PHetrGD8](#), se grafica el rendimiento de las distintas estrategias de selección de cortes en función del tamaño de muestra, y se compara los resultados obtenidos de RVR contra MCC. En la Figura 4.24a se muestra el valor calculado de la confiabilidad diámetro acotada, en la Figura 4.24b se grafican los valores de varianza, en la Figura 4.24c se grafica el tiempo de ejecución y en la Figura 4.24d se muestra la ganancia para la medida conjunta ‘tiempo por varianza’.

Figura 4.24: Análisis estadístico RVR vs MCC en Dodecaedro-PHetrGD8.



Capítulo 5

Consideraciones finales

En este capítulo se sintetizan las posturas expuestas en los capítulos anteriores resaltando los aportes de la investigación, y discutiendo los resultados obtenidos.

En esta tesis hemos reunido varias líneas generales referidas al problema del cálculo de la confiabilidad diámetro acotada para el modelo hostil. Se estudió y profundizó en conceptos fundamentales del problema como, teoría de grafos, confiabilidad diámetro acotada, sistemas binarios estocásticos, y sistemas binarios estocásticos monótonos. Formalmente se examinó la complejidad computacional y la intratabilidad de su cálculo. Se estudiaron y adaptaron los métodos de resolución aproximada Reducción Recursiva de la Varianza y Monte Carlo Crudo, siguiendo lineamientos de orden impuestos por la monotonía del sistema bajo estudio.

En este contexto, los aportes prácticos de esta tesis fueron estimar la confiabilidad del problema para distintos grafos reconocidos de la literatura, aplicando el método Reducción Recursiva de la Varianza. Realizamos un análisis de sensibilidad y eficiencia computacional del método en base a distintas estrategias de separación de grafos. Y finalmente, se brindó un estudio comparativo entre el método Monte Carlo Crudo y RVR, analizando la performance de estos.

Con la finalidad de evaluar la sensibilidad del método RVR, abordamos distintas estrategias de separación de grafos teniendo en cuenta, el concepto de conjunto separador en la DCR y las restricciones de orden bajo el modelo

hostil. Se formalizaron distintas estrategias de separación, siendo el algoritmo planteado en [35] base principal para encontrar mincuts en una cantidad lineal de evaluaciones de la función estructura (abarcando a cualquier SMBS). También, se utilizó como fuente de información el algoritmo BFS y métodos heurísticos.

Los casos de prueba propuestos se dividieron en grafos chicos y grandes. Los primeros fueron utilizados como una fase de comprobación de resultados por su fácil acceso al valor de confiabilidad exacta. Se pudo comprobar el funcionamiento del método RVR implementado a partir de los resultados empíricos de las medidas de desempeño de eficacia (y/o eficiencia), varianza, ganancia y tiempo de cómputo, logrando discriminar estrategias ineficientes para el método. Para grafos grandes se realizó un marco comparativo entre RVR y MCC.

Resultados particulares cosechados por grafo:

- **Grafo puente:** A rasgos generales este grafo nos permitió observar y concluir que, para muestras 10^4 o mayores RVR presentó buen comportamiento asintótico, buena precisión mostrando valores de varianza de menor orden en comparación con MCC.
- **Grafo completo:** Este grafo al contar con una estructura particular, nos permitió profundizar en el análisis de exactitud alcanzado por RVR, donde se destaca buenos resultados en orden numérico. Adicionalmente, ante un aumento de muestras se observó algunos casos donde el error conservaba el orden obtenido, y casos aislados donde la medida aumenta en un orden. Este comportamiento nos llevó a cuestionarnos la importancia de implementar aleatoriedad entre los elementos elegidos en las distintas estrategias de separación planteadas, y preguntarnos si esta medida de error puede ser mejorada aplicando técnicas especializadas en aleatoriedad.
- **Grafo grilla, $n = 3$:** Este grafo cuenta con la particularidad de tener varios caminos del mismo largo, nos permitió apreciar la sensibilidad que presenta RVR ante estrategias que recorren todo los posibles conjuntos de cortes de un grafo, contra una estrategia de corte que se centra en aislar nodos terminales. Al aumentar la cantidad de muestras en estos primeros la estimación de la DCR converge al valor exacto calculado,

mientras que para RVR_T la confiabilidad estimada se distanciaba sin acercarse de este valor. Llevándonos a descartar los resultados obtenidos de la implementación RVR con estrategia Corte trivial (Ver 3.2.2) para los siguientes grafos.

- **Grafo grilla $n = 5$, arpanet y dodecaedro:** Estos grafos nos permitieron constatar el desempeño que presenta RVR respecto a la medida de tiempo en cómputo, la cual aumenta considerablemente frente a estos grafos de gran tamaño. Ejemplo para 100 muestras de la implementación RVR_{MP} se llegó a constatar tiempos de cómputo de casi 4 horas, limitando el procesamiento para muestras grandes. Comparando RVR contra MCC en todos los escenarios y grafos, RVR obtuvo valores de varianza menores a MCC, sin embargo este último obtiene un mejor desempeño en tiempo de cómputo. Evidenciamos que MCC es menos demandante desde el punto de vista computacional pero la calidad de la aproximación es inestable y en algunos casos muestra deficiente medida conjunta (tiempo por varianza) acentuándose para pocas muestras del método. Por otro lado RVR_N y RVR_H marcan una buena relación entre la calidad de la aproximación y tiempo de cómputo para todo el conjunto de muestras y escenarios.

Resultados generales observados:

- Se destaca la performance en precisión de RVR. Y recalcamos la necesidad de una mayor cantidad de muestras de MCC para llegar a un valor estimado confiable.
- El tiempo de ejecución de RVR se presentó lineal al número de muestras tomadas y se mostró dependiente al tamaño del grafo, en particular RVR_N y RVR_H obtienen menor pendiente en comparación a RVR_{MP} . En esta medida identificamos una mejor performance para la implementación RVR_N .
- RVR exhibió sensibilidad ante instancias homogéneas y heterogéneas de probabilidad elemental (para la misma cota de diámetro). Entre estas no se contempló vastas diferencias en tiempo de cómputo, pero en relación a la cantidad de muestras se observó valores de varianza de menor unidad para escenarios heterogéneos, tendiendo estos más rápido a su valor estimado, generando un mejor desempeño de la medida conjunta, y desencadenando una mayor eficiencia relativa.

- Al contrastar los resultados contra la teoría, se constató que al restringir la cota el valor estimado de la confiabilidad estudiada disminuye.
- En general se recomienda las implementaciones de RVR_N y RVR_H , donde destacamos una buena relación entre la calidad de la aproximación y tiempo de cómputo para todo el conjunto de muestras y escenarios.

El marco práctico de la investigación nos permite concluir que se logró los objetivos planteados, y los resultados obtenidos son coherentes a la teoría. Ejemplo, se comprobó en forma práctica para el sistema bajo estudio la Proposición 2.4.1, llevando a afirmar que la medida de confiabilidad diámetro acotada es una generalización de la medida clásica bajo el modelo hostil, y como consecuencia la medida de confiabilidad clásica es una cota superior de la medida diámetro acotada. También, se comprobó que la varianza de RVR es siempre menor que la de MCC, bajo cualquier modelo monótono.

A rasgos generales en esta tesis se evaluó la calidad del estimador RVR, concluyendo que esta está fuertemente ligada a la selección y construcción de la estrategia de corte elegida, así como al tamaño del grafo. Esto se ve reflejado en los resultados tanto en tiempo de cómputo como en precisión. La estrategia planteada en 3.1 proporciona mejoras a nivel de procesamiento computacional obteniendo tiempos lineales a la cantidad de muestras. Adicionalmente, se puede tener una estrategia de separación eficiente en tiempo de cómputo pero la forma de implementación puede jugar un papel importante en otras variables, ejemplo la eficacia.

5.1. Trabajo futuro

En esta tesis se abordó la estimación estadística de la confiabilidad diámetro acotada bajo el modelo hostil. No obstante, existen diversos modelos de confiabilidad y su cálculo presenta distintos desafíos en términos de complejidad computacional, así como también de la algorítmica exacta y aproximada.

Como se ha hecho mención en otros capítulos, encontrar componentes operativos de mínima cardinalidad es un problema perteneciente a la clase \mathcal{NP} -Difícil (ver Teorema 2.5.1), pero esta búsqueda es de importancia en modelos económicos, donde el sistema funcional de menor cantidad de componentes puede significar un ahorro. La detección de componentes irrelevantes es también un problema perteneciente a esta misma clase. Estas son preguntas con valor práctico fundamental en sistemas reales (que son usualmente modelos monótonos), pero de las cuales no se dispone de una teoría lo suficientemente madura para su resolución general. El problema de detección de componentes irrelevantes admite un algoritmo polinomial para el caso de la confiabilidad diámetro acotada en modelo clásico de redes, y cuando tenemos exactamente dos terminales (ver [5]). No obstante, la complejidad de la detección de estos componentes es un problema abierto en general si se consideran sistemas con una cantidad arbitraria de terminales, y diámetro libre.

También, el estudio de eventos raros ha cobrado gran importancia en la confiabilidad de sistemas, y el desarrollo de estimadores robustos para la confiabilidad de eventos raros es un tema actual de investigación (ver [11]).

Esta tesis ha dado lugar a nuevas preguntas que son meritorias de abordar en un trabajo futuro:

- Realizar instancias de corrida sobre modelo K -terminales.
- Investigar virtualización o paquetes de octave que proporcionan funciones para el procesamiento paralelo en varios núcleos de CPU.
- Investigar como mejorar la estrategia de separación de grafos de máxima probabilidad (ver Corte 3.2.4).
- Analizar si es posible extender la investigación realizada en [5], con el objetivo de detectar o eliminar elementos irrelevantes que preserven la

DCR bajo el modelo hostil.

- Investigar el comportamiento asintótico de RVR para el problema planteado, aplicando estrategias de separación de grafos eficiente en tiempo de cómputo (ver Corte [3.2.1](#)) y técnicas especializadas en aleatoriedad (ver [\[40\]](#), [\[24\]](#)).

Referencias bibliográficas

- [1] Ball, M. O. (1986). Computational Complexity of Network Reliability Analysis: An Overview. *IEEE Transactions on Reliability*, 35(3):230–239.
- [2] Bertinat, M. E. (2011). Universidad de La República, UdelaR. Master’s thesis.
- [3] Canale, E., Cancela, H., Piccini, J., Robledo, F., Romero, P., Rubino, G., and Sartor, P. (2015a). Recursive Variance Reduction method in stochastic monotone binary systems. In *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 135–141.
- [4] Canale, E., Cancela, H., Robledo, F., Romero, P., and Sartor, P. (2015b). Full complexity analysis of the diameter-constrained reliability. *International Transactions in Operational Research*, 22(5):811–821.
- [5] Canale, E., Romero, P., and Rubino, G. (2017). Factorization and exact evaluation of the source-terminal diameter-constrained reliability. *Networks*, 70(4):283–291.
- [6] Cancela, H. (1999). Adapting RVR simulation for residual connectedness network reliability models. Technical report.
- [7] Cancela, H., Coll, J., Costanzo, D., and Rodríguez, M. (2006). Modelos de confiabilidad diámetro-acotada para optimizar la profundidad y amplitud de búsqueda en redes P2P. In *CLEI 2006, Santiago, Chile*, page 10.
- [8] Cancela, H. and El Khadiri, M. (1995). A Recursive Variance-Reduction Algorithm For Estimating Communication-Network Reliability. *IEEE Transactions on Reliability*, 44(4):595–602.

- [9] Cancela, H. and El Khadiri, M. (2003). The recursive variance-reduction simulation algorithm for network reliability evaluation. *IEEE Transactions on Reliability*, 52(2):207–212.
- [10] Cancela, H., El Khadiri, M., and Petingi, L. (2010a). Polynomial-time topological reductions that preserve the diameter constrained reliability of a communication network :un ambiente para el desarrollo de experimentos científicos. Technical report.
- [11] Cancela, H., El Khadiri, M., and Rubino, G. (2009). *Rare Event Analysis by Monte Carlo Techniques in Static Models*, chapter 7, pages 145–170. John Wiley & Sons, Ltd.
- [12] Cancela, H., Khadiri, M. E., Rubino, G., and Tuffin, B. (2010b). Recursive Variance Reduction Estimators for the Static Communication Network Reliability Problem. page 34.
- [13] Cancela, H. and Petingi, L. (2004). Reliability of communication networks with delay constraints: computational complexity and complete topologies. *International Journal of Mathematics and Mathematical Sciences*, 2004(29):1551–1562.
- [14] Cancela, H. and Urquhart, M. E. (2002). Adapting RVR simulation techniques for residual connectedness network reliability models. *IEEE Transactions on Computers, Computers*, 51(4):439–443.
- [15] Chiappara, N., Lacordelle, G., Robledo, F., and Romero, P. (2016). CADILAC: Caminos disjuntos de largo acotado. *Investigacion Operacional*, 37(3):226–246.
- [16] Colbourn, C. J. (1987). *The Combinatorics of Network Reliability*. Oxford University Press, Inc., New York, NY, USA.
- [17] Dash, R. K., Barpanda, N. K., Tripathy, P. K., and Tripathy, C. R. (2012). Network reliability optimization problem of interconnection network under node-edge failure model. *Applied Soft Computing*, 12(8):2322–2328.
- [18] E. Canale, H. Cancela, F. Robledo, P. and Romero (2014). Reporte Técnico RT 14-15. Technical report, Instituto de Computación – Facultad de Ingeniería Universidad de la República Montevideo, Uruguay.

- [19] Gomes, T., Tapolcai, J., Esposito, C., Hutchison, D., Kuipers, F., Rak, J., de Sousa, A., Iossifides, A., Travanca, R., Andre, J., Jorge, L., Martins, L., Ugalde, P. O., Pasic, A., Pezaros, D., Jouet, S., Secci, S., and Tornatore, M. (2016). A survey of strategies for communication networks to protect against large-scale natural disasters. *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pages 11–22.
- [20] Gonzalez, F. (2013). *Muestreo Por Importancia Y Reducción Recursiva De Varianza Para La Evaluación De La Confiabilidad Diámetro Acotada*. PhD thesis.
- [21] Haesik, K. (2020). Ultra-Reliable and Low Latency Communication Systems. In *Design and Optimization for 5G Wireless Communications*, chapter 8, pages 303–342. Wiley-IEEE Press.
- [22] Haverkort, B. R. and Niemegeers, I. G. (1996). Performability modelling tools and techniques. *Performance Evaluation*, 25(1):17–40.
- [23] J Faulin, AA Juan, SSM Alsina, J. E. R.-M. (2010). *Simulation Methods for Reliability and Availability of Complex Systems*, pages 65–83. 1st edition.
- [24] L’Ecuyer, P. and Tuffin, B. (2011). Approximating zero-variance importance sampling in a reliability setting. *Annals of Operations Research*, 189(1):277–297.
- [25] Lena, D., Franco, R., and Romero, P. (2017). Hostile Model for Network Reliability Analysis. *Croatian Operational Research Review*, 8(2):489–498.
- [26] Lewis Harry, R. (1983). Computers and Intractability. A Guide to the Theory of NP-Completeness Michael R. Garey David S. Johnson. *The Journal of Symbolic Logic*, 48(2):498–500.
- [27] Mahjoub, A. and McCormick, S. (2010). Max Flow and Min Cut with bounded-length paths: complexity, algorithms, and approximation. *Mathematical Programming*, 124(1):271–284.
- [28] Mathematics, D. A. (1996). A decomposition algorithm for network evaluation ” reliability. *Discrete Applied Mathematics*, 65(95):141–156.

- [29] Meyer, J. F. (1980). On Evaluating the Performability of Degradable Computing Systems. *IEEE Transactions on Computers, Computers, IEEE Transactions on, IEEE Trans. Comput*, C-29(8):720–731.
- [30] Murray, L. (2007). Comparación entre cinco métodos de Monte Carlo para estimar la confiabilidad diámetro acotada de redes de comunicaciones. Technical report, UR. FI – INCO.
- [31] Petingi, L. (2008). A diameter-constrained network reliability model to determine the probability that a communication network meets delay constraints. *WSEAS Transactions on Communications*, 7(6):574–583.
- [32] Petingi, L. and Ingeneria, F. D. (1997). Generalized Edge-Toughness Edge-Toughness. pages 174–181.
- [33] Robin J. Wilson (1996). *Introduction to Graph Theory*. Addison Wesley Longman Limited, 4th edition.
- [34] Romero, P. (2016). Duality in stochastic binary systems. *Proceedings of 2016 8th International Workshop on Resilient Networks Design and Modeling, RNDM 2016*, pages 85–91.
- [35] Romero, P. (2019). Challenges in Stochastic Binary Systems. *Proceedings of 2019 11th International Workshop on Resilient Networks Design and Modeling, RNDM 2019*, pages 1–8.
- [36] Sartor, P. (2013). *Diameter-constrained network reliability: properties and computation methods*. PhD thesis.
- [37] Satyanarayana, A. and Chang, M. K. (1983). Network reliability and the factoring theorem. *Networks*, 13(1):107–120.
- [38] T. H. Cormen, C. E. Leiserson, R. L. Riverst, C. S. (2001). *Introduction to algorithms*, volume 53. Section 22.
- [39] Tarjan, R. (1972). Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, 1(2):146–160.
- [40] Tuffin, B., Saggadi, S., and L’Ecuyer, P. (2014). An adaptive zero-variance importance sampling approximation for static network dependability evaluation. *Computers and Operations Research*, 45:51–59.

ANEXOS

Anexo 1

Conceptos Preliminares

1.1. Fundamentos básicos

1.1.1. Concepto de generalización

Dados dos conceptos relacionados, A y B, el concepto A es una generalización del concepto B si y sólo si:

- Cada instancia del concepto B es también una instancia del concepto A.
- Existen instancias del concepto A que no son instancias del concepto B.

1.1.2. Teoría del orden

Esta sección es de carácter general y queda a cargo del lector informarse sobre teoría de conjuntos y aritmética y relación binaria.

La teoría del orden es una rama de la matemática que estudia varias clases de relaciones binarias, que capturan la noción intuitiva del orden matemático.

El orden aparece por todas partes - por lo menos, si se trata de matemática y áreas relacionadas tales como la informática. El primer orden que uno típicamente encuentra en la educación matemática de la escuela primaria es el orden \leq de los números naturales. Este concepto intuitivo es fácilmente extendido a otros conjuntos de números, tal como los enteros y reales. Los tipos antedichos de orden tienen una propiedad especial: cada elemento se puede comparar con cualquier otro elemento, es decir es o mayor, o menor, o igual. Sin embargo, esto no siempre es un requisito deseable. Un ejemplo bien conocido es el orden de los subconjuntos de un conjunto.

Además, la teoría del orden no se restringe a las varias clases de relaciones de orden, sino que también considera funciones apropiadas entre ellas. Un ejemplo simple de una propiedad orden teórica viene del análisis donde encontramos con frecuencia a las funciones monótonas.

1.1.2.1. Conjuntos parcialmente ordenados

Como ya se hizo alusión arriba, un orden es una relación binaria especial. Por lo tanto consideremos algún conjunto P y una relación binaria \leq en P . Entonces \leq es un orden parcial si es reflexiva, antisimétrica, y transitiva, es decir, para todo a, b y c en P , tenemos que:

- $a \leq a$ (reflexividad)
- si $a \leq b$ y $b \leq c$ entonces $a \leq c$ (transitividad)
- si $a \leq b$ y $b \leq a$ entonces $a = b$, (antisimetría).

Un conjunto con un orden parcial se llama conjunto parcialmente ordenado, o, en breve, poset (del inglés partially ordered set). Comprobando esta propiedad, se ve inmediatamente que los bien conocidos órdenes de los naturales, enteros, racionales y reales son todos órdenes en el antedicho sentido. Sin embargo, tienen la propiedad adicional de ser total, es decir, para todo a, b en X .

- $a \leq b$ o $b \leq a$ (totalidad)

este orden se puede también llamar orden lineal o cadena. mientras que muchos órdenes clásicos son lineales, el orden entre subconjuntos de un conjunto proporciona un ejemplo donde éste no es el caso. De hecho, muchas propiedades avanzadas de los posets son interesantes principalmente para un orden no lineal.

1.1.2.2. Par ordenado

En matemáticas, un par ordenado es una pareja de objetos matemáticos, en la que se distingue un elemento y otro. El par ordenado cuyo primer elemento es A y cuyo segundo elemento es B se denota como (a, b) .

Un par ordenado (a, b) no es el conjunto que contiene a A y B , denotado por $\{a, b\}$. Un conjunto está definido únicamente por sus elementos, mientras que en un par ordenado el orden de estos es también parte de su definición. Por

ejemplo, los conjuntos $\{0, 1\}$ y $\{1, 0\}$ son idénticos, pero los pares ordenados $(0, 1)$ y $(1, 0)$ son distintos.

Del siguiente par ordenado (a, b) podemos decir que:

- a es el primer componente del primer conjunto y;
- b como el segundo componente del segundo conjunto.

Matemáticamente esto se expresa:

$$A \times B = \{(a, b) | a \in A \wedge b \in B\} \quad (1.1.1)$$

1.1.2.3. Producto Cartesiano

Dados dos conjuntos A y B , su producto cartesiano es el conjunto de todos los pares ordenados que pueden formarse con estos dos conjuntos:

El producto cartesiano de A y B es el conjunto $A \times B$ cuyos elementos son los pares ordenados (a, b) , donde a es un elemento de A y b un elemento de B :

$$A \times B = \{(a, b) : a \in A \quad y \quad b \in B\} \quad (1.1.2)$$

Definimos los conjuntos:

$$A = \{1, 4, 6\} \quad , \quad B = \{2, 3, 5\} \quad (1.1.3)$$

La enumeración de los elementos, del conjunto de pares ordenados, sería la siguiente:

$$A \times B = \{(1, 2), (1, 3), (1, 5), (4, 2), (4, 3), (4, 5), (6, 2), (6, 3), (6, 5)\} \quad (1.1.4)$$

1.1.3. Teoría de grafos

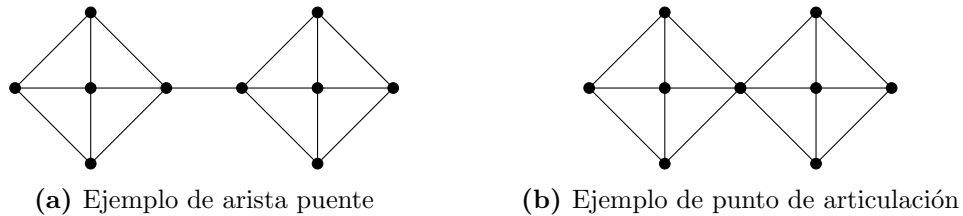
Se presentan a continuación definiciones básicas de Teoría de Grafos así como versiones del Teorema de Menger, las cuales pueden ser encontradas en [?].

Definición 1.1.1. Sea $G = (V, E)$ un grafo decimos $X \subseteq V \cup E$ "separa" G si $G - X$ es no conexo.

Definición 1.1.2. Sea $G = (V, E)$ un grafo, decimos que un nodo $v \in V$ es un **punto de articulación** si separa a dos nodos de la misma componente; i.e. $G \setminus v$ es desconexo

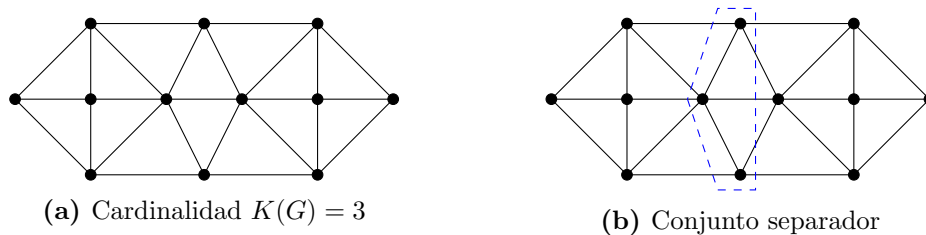
Definición 1.1.3. Sea $G = (V, E)$ un grafo, decimos que una arista $e \in E$ es un puente si separa a dos nodos de la misma componente; i.e. $G \setminus e$ es desconexo

Figura 1.1: Separadores de cardinalidad 1



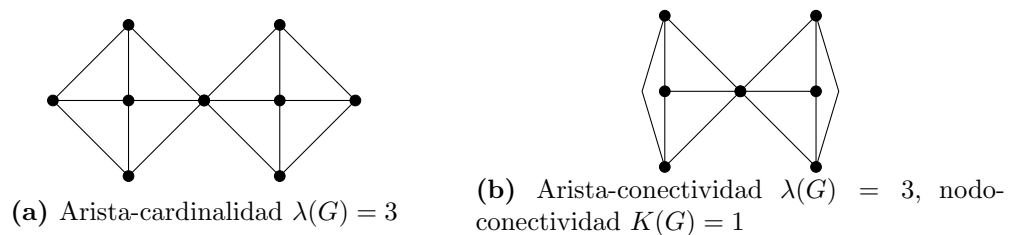
Definición 1.1.4. Un grafo $G = (V, E)$ es k -nodo-conexo, si $|V| > k$ y $G \setminus X$ es conexo para todo $X \subseteq V$ con $|X| < k$. Es decir, ningún par de nodos de G están separados por menos de k nodos. Y se denota **conectividad de G** $K(G)$ al mayor valor de k para el cual un grafo G es k -nodo-conexo.

Figura 1.2: Separador de cardinalidad 3



Definición 1.1.5. Un grafo $G = (V, E)$ es k -arista-conexo, si $|V| > k$ y $G \setminus F$ es conexo para todo $F \subseteq E$ con $|F| < k$. Es decir, ningún par de nodos de G están separados por menos de k aristas. Y se denota **arista-conectividad de G** $\lambda(G)$ al mayor valor de k para el cual un grafo G es k -arista-conexo.

Figura 1.3: Separadores arista-cardinalidad y nodo-cardinalidad



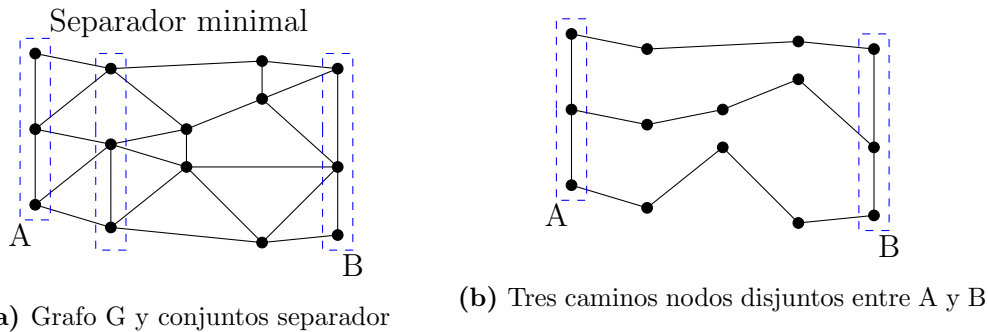
Definición 1.1.6. Llamaremos árbol a un grafo acíclico conexo. Decimos que un subgrafo $G_x \subseteq G$ es un árbol de cubrimiento de G , si es árbol y $V(H) = V$

Uno de los resultados fundamentales de la teoría de conectividad de grafos es el resultado conocido como Teorema de Menger, el cual establece una relación directa entre conjuntos separadores de cardinalidad mínima y la cantidad máxima de caminos disjuntos entre dos conjuntos de nodos predeterminados.

Teorema 1.1.1. (Menger 1927): Sea $G = (V, E)$ un grafo y $A, B \subseteq V$ entonces el mínimo número de nodos separadores A de B en G es igual al máximo número de caminos disjuntos de A a B en G .

En el siguiente ejemplo podemos ver que el grafo tiene 3 caminos nodos disjuntos, equivalente al conjunto de corte de cardinalidad mínima, para el conjunto de nodos A y B .

Figura 1.4: Ejemplo Teorema de Menger camino nodo disjunto

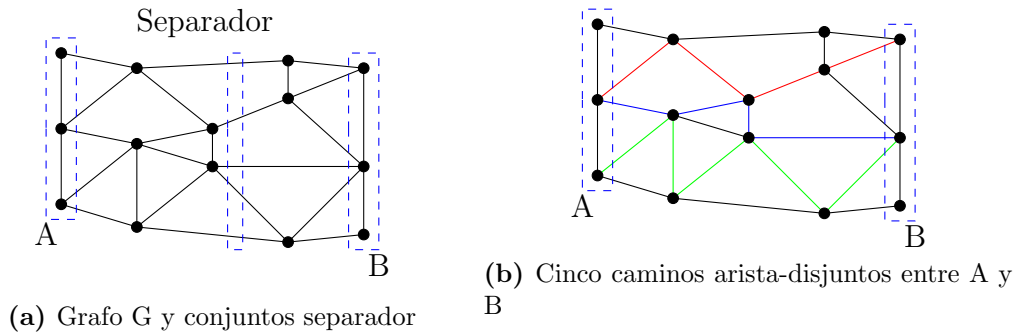


Teorema 1.1.2. (Versión global de Teorema de Menger)

- Un grafo es k -nodo-conexo si contiene k caminos independientes entre cualquier par de nodos.
- Un grafo es k -arista-conexo si contiene k caminos arista-disjuntos entre cualquier par de nodos.

En la figura 1.5 podemos observar cinco caminos aristas disjuntos el cual proporciona un conjunto separador de 5 aristas entre los conjuntos A y B .

Figura 1.5: Ejemplo Teorema de Menger camino arista disjunto



Una forma eficiente para determinar la cantidad de caminos arista disjuntos de un grafo es a través del teorema de Ford-Fulkerson, en particular cuando la capacidad de la red es igual a 1, el valor del corte mínimo nos da la cantidad de caminos arista disjuntos entre el nodo fuente y el nodo destino (ver mas en 2.4). Por lo que a través de este algoritmo podemos determinar el conjunto arista separador de un grafo. En el caso s - t -terminal la aplicación del algoritmo es directa. En el caso K -terminal se eligen dos nodos del conjunto objetivo y se busca un cutset que los separe. Si no existe corte para el par elegido se continúa con otro par de nodos hasta encontrar un corte que separe al menos dos nodos de K .

También existen varios trabajos relacionados al problema de hallar K caminos nodo o arista disjuntos entre dos nodos de un grafo. En el artículo [15] se destacan los autores Suurballe, Bhandari, Fleischer, y el trabajo de Beshir, Kuipers y Guo. Dado que el problema en su formulación más sencilla, tiene una solución conocida, muchos de ellos tratan variantes que imponen restricciones adicionales sobre los caminos, que transforman el problema original en uno de mayor complejidad computacional. En base a los distintos autores y el algoritmo GRASP (Greedy Randomized Adaptive Search Procedure) en [15] formalizan el algoritmo CADILAC para la búsqueda de caminos K -nodos disjuntos con diámetro acotado.

Entonces, a partir de la teoría tradicional de grafos podemos obtener la nodo conectividad y arista conectividad del grafo, jugar con estos dos valores obteniendo la exacta cantidad de vértices y aristas a remover del grafo para obtener conjuntos de cortes.

Anexo 2

Algoritmos auxiliares

Entre los algoritmos centrales [38] para el cálculo de la medida de confiabilidad diámetro acotada se consideran operaciones en forma reiterada. Entre éstas, evaluar si una red está operativa, identificar el conjunto de corte de una red y encontrar el camino más corto entre un par de nodos.

El algoritmo de búsqueda en profundidad se aplica para evaluar el estado de una red, en el contexto de este trabajo se evalúa si los nodos objetivo están conectados. El algoritmo de búsqueda en anchura también se utiliza para evaluar el estado de una red para el caso de confiabilidad diámetro acotado, pero en nuestro caso se utiliza para obtener un árbol de cobertura y los distintos pathset entre nodos terminales. El Algoritmo Ford-Fulkerson permite identificar conjuntos de corte mínimo de un grafo, de forma que la eliminación de los elementos del corte hacen que la red quede en estado de falla. El algoritmo de Dijkstra se aplica para identificar el camino más corto entre dos nodos y una adaptación sencilla permite identificar el camino más corto con una cota superior que restringe el largo de los caminos buscados.

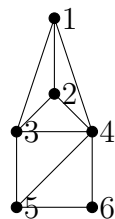
2.1. Algoritmo de búsqueda en profundidad

El algoritmo de búsqueda en profundidad, DFS por sus siglas en inglés Depth First Search, permite recorrer todos los nodos que integran un grafo generando caminos en forma recurrente. De esta manera permite distinguir si un grafo es conexo o no. Dado un grafo $G(V, E)$, la idea es recorrer los nodos iniciando en un nodo cualquiera de G y construir caminos agregando en cada paso un nuevo nodo hasta que todos los nodos alcanzables hayan sido visitados. La búsqueda se inicia en $v \in V$ y cada nodo tiene dos estados posibles: Visitado o NoVisitado. Si v tiene vecinos sin visitar, se elige uno de ellos, w que desde este momento es Visitado. Desde w , se identifican sus vecinos no visitados. Si existe al menos uno se elige uno de ellos a visitar, x . En cambio si w no tiene nodos vecinos por visitar, se vuelve a v y se buscan los vecinos sin visitar de v .

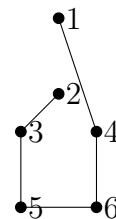
Se construye así un camino simple a un nodo de $Visitados = v, w, x, \dots, t$. El último nodo del camino es una hoja del árbol recubridor en construcción y no tiene vecinos sin visitar. Si aún quedan nodos de G sin visitar se retrocede al vértice anterior en el camino, x y se construye otro camino simple que nace en x e incorpora nodos aún no visitados. Si desde x no es posible construir ningún camino con nuevos nodos se sigue retrocediendo, en este caso hacia w . La aplicación del algoritmo continúa hasta construir un árbol recubridor de G . El tiempo de búsqueda empleado es lineal en V y E .

En la figura 2.1 se muestra un ejemplo de búsqueda para el grafo, con condición inicial en el nodo 1, y el orden de visita es el mostrado en (b) o sea [146532]. En el contexto de esta tesis, el algoritmo DFS será utilizado para determinar si el grafo sorteado es DK-conexo o en otras palabras la red opera o falla.

Figura 2.1: Algoritmo de búsqueda en profundidad



(a) Grafo



(b) árbol de cubrimiento de G en DFS

```

function marcado=dfs(x,marcado,origen)
#Comprueba si el grafo x, es conexo.
#para eso busca un camino entre todos los nodos del grafo,
#este camino va marcando nodos ya visitados

##marco nodo
marcado=[marcado origen];

#busco nodos adjacentes del grafo maracdo recientemente
nodosadj=[];
for i=1:size(x,1)
if x(i,1)==origen  nodosadj=[nodosadj x(i,2)]; endif
if x(i,2)==origen  nodosadj=[nodosadj x(i,1)]; endif
endfor

for j=1:size(nodosadj,2)
if sum(marcado==nodosadj(j))==0
#si el nodo no se encuentra marcado vuelo a llamar a dfs
marcado=[dfs(x,marcado,nodosadj(j))];
endif

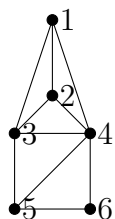
endfor

```

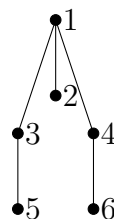

2.2. Algoritmo de búsqueda en anchura

El Algoritmo de Búsqueda en Anchura, en inglés BFS-Breadth First Search, permite recorrer los nodos que integran un grafo, y al igual que el DFS es un algoritmo de búsqueda “ciega” (sin información). Comenzando desde un nodo raíz el objetivo es visitar todos los nodos incorporando en forma consecutiva los vecinos de un nodo actual. Cada nodo del grafo G puede ser “visitado” o “no visitado”. Si se inicia de un nodo raíz o fuente éste se considera visitado y luego se identifican los nodos vecinos del nodo raíz que aún no fueron visitados. Como éstos vecinos son alcanzables desde la raíz se cambia su estado a “visitado” y se procede a buscar vecinos de éstos nodos que aún no fueron visitados. El proceso se repite hasta que todos los nodos sean alcanzados o bien no hayan nodos alcanzables desde los nodos visitados. En cada paso además calcula la distancia desde el nodo raíz a los nodos visitados. Por la forma de proceder en la búsqueda sólo se visitan los nodos a distancia L si fueron visitados los nodos a distancia $L - 1$ desde el nodo raíz. Además cada nodo es visitado una única vez. El tiempo de ejecución del Algoritmo BFS es $O(|V| + |E|)$. En la figura 2.2 se muestra un ejemplo de búsqueda BFS para el grafo (a), con nodo raíz en 1, y el grafo de cubrimiento obtenido se muestra en (b). Esto fue programado sobre octave en la función bfs.m.

Figura 2.2: Algoritmo de búsqueda en profundidad



(a) Grafo



(b) árbol de cubrimiento de G en BDF

```
function [ marcado , padre , distancia ]= bfs (x, origen )
##iniciando variables
marcado=[origen ]; #nodos visitados
A=[origen ]; #nodos visitados variables
padre=[origen ; 0]; #matrix: [nodo; nodo_padre]
distancia=padre; #matrix: [nodo; distancia a raiz]
```

```

while any(A)==1
    u=A(1);
    A(find(A==u))=[];
    #busco nodos adjacentes del grafo marcado recientemente
    nodosadj=[];
    for i=1:size(x,1)
        if x(i,1)==u nodosadj=[nodosadj x(i,2)]; endif
        if x(i,2)==u nodosadj=[nodosadj x(i,1)]; endif
    endfor
    for j=1:size(nodosadj,2)
        if sum(marcado==nodosadj(j))==0
            marcado=[marcado nodosadj(j)];
            distancia=[distancia [nodosadj(j); distancia(2,find(distancia(1,:))=
            padre=[padre [nodosadj(j); u]];
            A=[A nodosadj(j)];
        endif
    endfor
endwhile

```

Se obtiene la siguiente salida en consola:

```
[marcado , nodo_padre , distancia_a_nodo_raiz]=bfs(grafoE1 ,1)
```

```
marcado =
```

```
1   3   2   4   5   6
```

```
nodo_padre =
```

```
1   3   2   4   5   6
0   1   1   1   3   4
```

```
distancia_a_nodo_raiz =
```

```
1   3   2   4   5   6
0   1   1   1   2   2
```

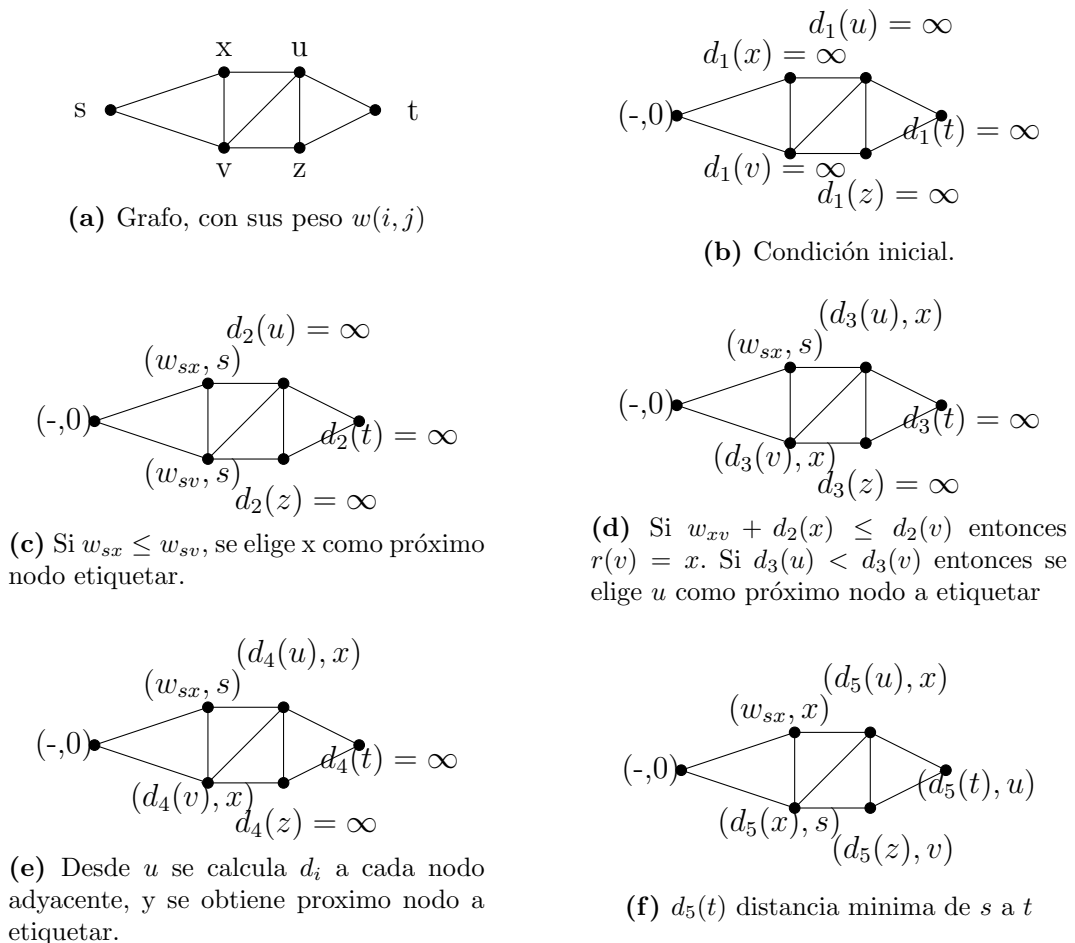
2.3. Algoritmo Dijkstra

Este algoritmo de Dijkstra calcula el camino mínimo de un nodo A a otro nodo Z en particular, a la vez que calcula los caminos mínimos desde el nodo inicial A dado hasta cada uno de los otros nodos del grafo.

En nuestro caso además de calcular del camino mínimo queremos restringir que estos sean de un largo D , por lo que debemos utilizar una adaptación de Dijkstra. Según las notas [39], dado un grafo $G = (V, E)$ y matriz de costos asociado a los enlaces, donde $w(u, v)$ indica el costo asociado al enlace que une los nodos $u, v \in V$. Si los costos o las restricciones de largo son iguales $\forall v \in V$ entonces el problema es soluble en tiempo polinomial.

Dado el ejemplo de la figura donde, cada nodo es etiquetado con la siguiente etiqueta (d_i, r) , r :predecesor, d_i : distancia de s a $v \in V$.

Figura 2.3: Algoritmo Dijkstra



Sea L la cota superior para el largo de los caminos entre dos nodos de G . $\forall v \in V$ se define $d_i(v)$ la distancia más corta entre un nodo origen s y v usando a lo sumo i enlaces en la iteración actual del algoritmo. Se define además el conjunto U de nodos no etiquetados, inicializando este como $U - \{s\}$ y definiendo $R = \{v \in U : \exists k \leq L \text{ donde } d_k(v) \leq \infty\} \subseteq U$, donde R es el conjunto de todos los nodos sin etiquetar que pueden ser alcanzados desde s utilizando todos los nodos intermedios (vecinos) etiquetados. En cada iteración se le asigna al nodo corresponde la etiqueta con la distancia mas corta desde el nodo s a v .

Pseudocódigo 2.1 Dijkstra de largo acotado

Input: grafo de red G , largo L y nodo s

Output: Camino mas corto entre s y cualquier nodo $V - s$ de largo menor a L

```

1:  $U = V - \{s\}$ 
2:  $R = \{v \in V \mid (s, v) \in E\}$ 
3: for  $v \in V$  do
4:    $d_i(v) = \begin{cases} 0 & \text{si } v = s \\ w_e & \text{si } e = (s, v) \\ \infty & \text{en otro caso} \end{cases}$ 
5: end for
6: while  $R \neq \emptyset$  do
7:   Encontrar  $v \in R$  y  $1 \leq i \leq L$  tal que  $\forall w \in R, 1 \leq j \leq L$ 
      a)  $d_i(v) \leq d_j(w)$  y
      b)  $d_i(v) = d_j(w) \rightarrow j \leq i$ 
8:    $U = U - \{v\}, R = R - \{v\}$ 
9:
10:  for all los vecinos  $w \in U$  do
11:
12:    for all  $j$  con  $i + 1 \leq j \leq L$  do
13:       $d_j(w) = \min\{d_j(w), d_{j-1}(v) + \mu(v, w)\}$ 
14:    end for
15:    if  $d_L(w) < \infty$  then
16:       $R = R \cup w$ 
17:    end if
18:  end for
19: end while

```

2.4. Método Ford-Fulkerson

El método de Ford-Fulkerson [38] tiene aplicaciones diversas. Una de ellas identificar el flujo máximo de una red a través de las capacidades asociadas a los enlaces. Para esto define el teorema corte mínimo-flujo máximo, el cual caracteriza el valor del flujo máximo en términos del corte de capacidad mínima de la red.

Este método trabaja sobre grafos orientados, y cada arco de la red se lo pondera asignándole una capacidad $c(e)$ y un flujo $f(e)$. Donde $c(e)$ es la capacidad o cota superior a la que puede trasportar el enlace y $f(e)$ cantidad de datos reales a trasportar.

El conjunto de nodos V puede particionar en 2 subconjuntos P y P^c , tal que $P \cup P^c = V$ y $P \cap P^c = \emptyset$.

Definición 2.4.1. Un corte en la red es el conjunto de arcos incidentes hacia el exterior de P .

$$(P, P^c) = \{(x, y), x \in P, y \in P^c\} \quad (2.4.1)$$

Definición 2.4.2. corte fuente terminal es aquel que $s \in P$ y $t \in P^c$

Definición 2.4.3. Capacidad de un corte (P, P^c) es aquel que

$$c(P, P^c) = \sum_{e \in (P, P^c)} c(e) \quad (2.4.2)$$

Corte mínimo – flujo máximo

Definición 2.4.4. Flujo fuente-terminal: Sea $G = (V, E)$ una red con vértices V y enlaces E , capacidad de los enlaces $c : E \rightarrow R^+$, $s, t \in V$ nodos fuente y terminal respectivamente. Diremos que $\phi : E \rightarrow R^+ \cup \{0\}$ es un flujo en G si:

$$0 \leq f(e) \leq c(e), \forall e \in E \quad (2.4.3)$$

$$f(e) = 0 \quad \forall u \in I^-(s) \quad \forall u \in I^+(t) \quad (2.4.4)$$

$$\forall x \neq s \neq t \quad \sum_{e \in I^+(x)} f(e) = \sum_{e \in I^-(x)} f(e) \quad (2.4.5)$$

La segunda condición nos asegura que el sentido del flujo es de s a t y no al revés. La tercera condición cumple la ley de conservación de kirchoff. Y a partir de esta conservación se cumple, que para un conjunto P de nodos que no contenga $\{s, t\}$:

$$\sum_{u \in (P, P^c)} f(e) = \sum_{u \in (P^c, P)} f(e) \quad (2.4.6)$$

Teorema 2.4.1. Flujo $f_{\{s,t\}}$. *El flujo que sale de la fuente s es igual al flujo que entra a la terminal t .*

Entonces sea $|f_{\{s,t\}}|$ y se define como la suma de flujos que salen de s , por lo que una cota superior de este es la suma de las capacidades de los arcos salientes(I^+):

$$|f_{\{s,t\}}| = \sum_{e \in I^+(x)} f(e) \leq \sum_{e \in I^+(x)} c(e) \quad (2.4.7)$$

De la misma manera, para los arcos incidentes en t .

Teorema 2.4.2. *Para cualquier flujo $f_{\{s,t\}}$ y corte (P, P^c) se cumple: $f_{\{s,t\}} \leq c(P, P^c)$*

Corolario 2.4.1. *Entonces $f_{\{s,t\}} = c(P, P^c)$ si y solo si:*

$$f(e) = 0, \forall e \in (P^c, P) \quad (2.4.8)$$

$$f(e) = c(e), \forall e \in (P, P^c) \quad (2.4.9)$$

En este caso decimos que $f_{\{s,t\}}$ es un flujo máximo y (P, P^c) es un corte de capacidad mínima.

Bases Para la Construcción de un Flujo Máximo

Llamaremos **arco saturado** a un arco u tal que $c(e) = f(e)$. Y **holgura de una arco** a la capacidad libre, que es posible utilizar $g(e) = c(e) - f(e)$ y se define la **holgura mínima** de un camino que inicia en s y termina en t como la holgura mínima entre todos los arcos de este camino. Sea L_i el camino de desde la fuente a la terminal. Entonces si la unidad de flujo $\{s, t\}$ en L es f_L tal que

$$f_L(e) = 1, \forall e \in L \quad (2.4.10)$$

$$f_L(e) = 0, \forall e \neq L \quad (2.4.11)$$

En base a la unidad de los camino se define la suma de flujos,

$$f_{\{s,t\}} = f_{\{s,t\}} + \sum_{i=1}^N f_{L_i}(e), \forall e \in E \quad (2.4.12)$$

por lo tanto si g_i es la holgura mínima del camino L_i , surge el mecanismo de construir un flujo máximo: aumentando el flujo de la red mediante la suma de unidades de $f_{\{s,t\}}$ equivalentes a la holgura mínima de los caminos L_i que así lo permitan.

$$f_{\{s,t\}} = f_{\{s,t\}} + \sum_{i=1}^N g_i * f_{L_i}(e), \forall e \in E \quad (2.4.13)$$

Algoritmo de flujo máximo

El algoritmo de aumento que a continuación planteamos, de Ford y Fulkerson, construye un corte mínimo y un flujo máximo, corrigiendo e incrementando, mediante el aumento del valor del flujo $f_{\{s,t\}}$ con unidades de flujo en cadenas. Iterativamente va construyendo cadenas de flujos desde s , de una manera similar a la del camino mínimo de Dijkstra. A cada nodo q visitando se le asigna una etiqueta: $(p \pm, D(q))$ compuesta de dos etiquetas:

- p es el nodo precedente a q en la cadena de flujo $f_{\{s,q\}}$, el índice superior de p será el signo de $+$ si el último arco en la cadena es (p, q) y el signo de $-$ si el último arco es (q, p) .
- $D(q)$ es la holgura mínima entre todos los arcos de la cadena que van de s a q . En un arco u de orientación contraria al de la cadena, la holgura es la cantidad de flujo que puedo quita

Pseudocódigo 2.2 Algoritmo de Flujo máximo

Input: grafo de red G , matiz de costos, nodos objetivos: s, t Output: mincut de $G:C$

```
1: Marcar la fuente  $s$  con  $(\infty)$ 
2: Sea  $p$  el vértice a examinar cuya etiqueta es  $D(p)$ 
3: while  $\exists u = (q, p) \in I^-(p)$  do
4:   if  $f(u) > 0$  y  $q$  no marcado then
5:     marcar  $q$  con  $(p^-, D(q))$ 
6:      $D(q) = \min(D(p), f(u))$ 
7:   end if
8: end while
9: while  $\exists u = (p, q) \in I^-(p)$  do
10:  if  $g(u) = c(u) - f(u) > 0$  y  $q$  no marcado then
11:    marcar  $q$  con  $(p^+, D(q))$ 
12:     $D(q) = \min(D(p), g(u))$ 
13:  end if
14: end while
15: if  $T$  marcado then
16:   Ir al punto 21
17: else if buscar otro vértice marcado para examinar then
18:   Ir al punto 2
19: else Si no existen mas vértices marcados para examinar se ha determinado
    un corte  $s, t$  saturado  $(P, P^c)$ , con  $P$  el conjunto de los vértices marcados.
     $|f\{s, t\}| = (P, P^c)$  y por eso  $f\{s, t\}$  maximal  $\rightarrow$  FIN
20: end if
21: Encuentre una cadena  $L_{\{s, t\}}$  con holgura retrocediendo desde  $t$  como en
    el algoritmo Dijkstra. El flujo a incrementar  $f_L$  es cadena flujo del  $f_{\{s, t\}}$ ,
    de valor  $D(t)$  unidades  $f_{\{s, t\}}' = f_{\{s, t\}} + D(t) * f_L$ . Reasignar el flujo, y
    se aplica nuevamente el algoritmo a  $f_{\{s, t\}}'$ , es decir volver a 1)
```

Teorema 2.4.3. *Dado cualquier $f_{\{s, t\}}$, aplicando un numero finitos de veces el algoritmo de Ford y Fulkerson, se obtiene un flujo maximo. Además si P es el conjunto de vértices marcados en la etapa final del algoritmo, el corte (P, P^c) es un corte $\{s, t\}$ minimo.*

Corolario 2.4.2. *En cualquier red orientada, el valor de un flujo máximo $\{s, t\}$ es igual a al capacidad de un corte $\{s, t\}$ minimo.*

Anexo 3

Resultados auxiliares

3.1. Resultados grafo puente

Tabla 3.1: Resultados de Monte Carlo para grafo puente

D	Método	Muestras	p_e/p_v	R	V	$t(\text{seg})$
2	<i>MCC</i>	10	"0.95/0.95"	1	0	0.038
	<i>MCC</i>	100	"0.95/0.95"	1	0	0.3123
	<i>MCC</i>	1000	"0.95/0.95"	0.9790000000000000	2.055900000000000e-05	3.0935
	<i>MCC</i>	10000	"0.95/0.95"	0.9780000000000000	2.151600000000000e-06	30.8095
	<i>MCC</i>	50000	"0.95/0.95"	0.9794400000000000	4.027457280000000e-07	162.07
	<i>MCC</i>	100000	"0.95/0.95"	0.9798500000000000	1.974397750000000e-07	338.9462
2	<i>MCC</i>	10	"0.95/0.97"	1	0	0.0491
	<i>MCC</i>	100	"0.95/0.97"	1	0	0.3256
	<i>MCC</i>	1000	"0.95/0.97"	0.9850000000000000	1.477500000000000e-05	3.2593
	<i>MCC</i>	10000	"0.95/0.97"	0.9842000000000000	1.555036000000000e-06	32.9215
	<i>MCC</i>	50000	"0.95/0.97"	0.9843400000000000	3.082952880000000e-07	171.85
	<i>MCC</i>	100000	"0.95/0.97"	0.9837500000000000	1.598593750000000e-07	357.5746
3	<i>MCC</i>	10	"0.95/0.95"	1	0	0.0423
	<i>MCC</i>	100	"0.95/0.95"	1	0	0.3211
	<i>MCC</i>	1000	"0.95/0.95"	0.9830000000000000	1.671100000000000e-05	3.166
	<i>MCC</i>	10000	"0.95/0.95"	0.9835000000000000	1.622775000000000e-06	31.818
	<i>MCC</i>	50000	"0.95/0.95"	0.9838000000000000	3.187512000000000e-07	161.039
	<i>MCC</i>	100000	"0.95/0.95"	0.9830800000000000	1.663371360000000e-07	333.0477
3	<i>MCC</i>	10	"0.95/0.97"	1	0	0.0332
	<i>MCC</i>	100	"0.95/0.97"	1	0	0.3223
	<i>MCC</i>	1000	"0.95/0.97"	0.9850000000000000	1.477500000000000e-05	3.2045
	<i>MCC</i>	10000	"0.95/0.97"	0.9878000000000000	1.205116000000000e-06	32.1966
	<i>MCC</i>	50000	"0.95/0.97"	0.9882200000000000	2.328246320000000e-07	163.241
	<i>MCC</i>	100000	"0.95/0.97"	0.9882000000000000	1.166076000000000e-07	336.2068

Tabla 3.2: Error relativo RVR vs MCC, para grafo puente

Puente-PHomgD2					
Muestra	Er[%] RVR_N	Er[%] RVR_H	Er[%] RVR_{MP}	Er[%] RVR_T	Er[%] MCC
100	1.5752e-01	4.7328e-01	4.2693e-01	2.1996e-01	2.0764e+00
1000	1.5649e-02	1.3196e-01	2.2749e-01	1.4564e-02	6.7177e-02
10000	5.2485e-03	4.7428e-03	8.1611e-03	4.7111e-02	1.6925e-01
50000	4.1394e-03	2.5741e-03	1.3603e-02	6.7033e-03	2.2264e-02
100000	3.8753e-04	1.1862e-03	2.8034e-03	1.1042e-02	1.9588e-02
Puente-PHomgD3					
Muestra	Er[%] RVR_N	Er[%] RVR_H	Er[%] RVR_{MP}	Er[%] RVR_T	Er[%] MCC
100	2.2826e-02	1.8905e-01	2.9137e-01	1.2248e-01	1.6749e+00
1000	1.0127e-02	4.9623e-02	4.1436e-03	5.5661e-02	5.3584e-02
10000	7.7359e-03	4.1266e-03	2.0866e-02	1.4828e-02	2.7467e-03
50000	4.3707e-03	1.9114e-04	1.6301e-03	1.2979e-02	2.7756e-02
100000	5.2601e-03	1.1323e-02	3.6714e-03	1.6043e-02	4.5450e-02
Puente-PHetrgD2					
Muestra	Er[%] RVR_N	Er[%] RVR_H	Er[%] RVR_{MP}	Er[%] RVR_T	Er[%] MCC
100	1.6612e-01	1.9985e-02	6.3184e-02	1.1978e-01	1.5764e+00
1000	8.4082e-03	9.0147e-02	2.0237e-02	2.2006e-02	5.2711e-02
10000	2.5154e-02	2.9292e-02	3.6857e-02	3.2732e-02	2.8550e-02
50000	2.8294e-03	9.2435e-03	1.8701e-02	5.3477e-03	1.4329e-02
100000	7.8416e-04	2.9741e-03	7.2129e-03	3.5988e-03	7.4259e-02
Puente-PHetrgD3					
Muestra	Er[%] RVR_N	Er[%] RVR_H	Er[%] RVR_{MP}	Er[%] RVR_T	Er[%] MCC
100	2.6574e-02	1.0518e-01	2.0818e-02	1.2239e-01	1.1619e+00
1000	5.5787e-02	7.6194e-03	1.3492e-02	1.3456e-02	3.5554e-01
10000	3.4346e-03	3.7247e-04	1.2454e-02	3.5770e-02	7.2289e-02
50000	8.2843e-03	1.2695e-03	6.7217e-03	2.6198e-02	2.9801e-02
100000	8.9436e-03	7.5145e-04	9.3791e-04	2.4229e-02	3.1825e-02

Tabla 3.3: Resultados numéricos de RVR según cortes, escenario Puente-PHomgD2.

Puente-PHomgD2					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.978114969692889	7.74093337563396e-06	5.038	0
RVR_N	1000	0.979811417288669	6.51896434882287e-07	50.698	1.924343839529332
RVR_N	10000	0.979606691605584	6.70703493731884e-08	505.374	1.955702191071696
RVR_N	50000	0.979698661115316	1.32645768884034e-08	2530.83	1.944362333455266
RVR_N	100000	0.979654312868976	6.65980419102756e-09	5070.749	1.981672317542302
RVR_H	100	0.984294665485462	3.02040959494662e-06	8.065	0
RVR_H	1000	0.980950835744488	5.67857798468493e-07	78.96	1.418421719801517
RVR_H	10000	0.979611646610482	6.69000931523849e-08	782.918	1.265619581148032
RVR_H	50000	0.979632892156775	1.33321174007055e-08	3921	1.248641033811158
RVR_H	100000	0.979669729875159	6.65028591180131e-09	7845.364	1.282661329046443
RVR_{MP}	100	0.975475658557374	1.00549474489813e-05	59.06	0
RVR_{MP}	1000	0.977429494773385	8.88922440695140e-07	598.82	0.1194790992867278
RVR_{MP}	10000	0.979578158721736	7.47415929445279e-08	6035.43	0.1469520398992265
RVR_{MP}	50000	0.979791369291353	1.46759612593072e-08	30194.1	0.1473007400828812
RVR_{MP}	100000	0.979685572810698	7.39111893093582e-09	60430.2	0.149830833180179
RVR_T	100	0.977503226494799	6.87283021593951e-06	6.056	0
RVR_T	1000	0.979800787773121	6.28531491958985e-07	61.134	1.655168649585153
RVR_T	10000	0.979196578456155	6.62208083088268e-08	611.55	1.636890913294109
RVR_T	50000	0.979723779141952	1.25065716451213e-08	3068.27	1.700989863058875
RVR_T	100000	0.979549932026965	6.32886200595487e-09	6136.035	1.723264582014786

Tabla 3.4: Resultados numéricos de RVR según cortes, escenario Puente-PHomgD3.

Puente-PHomgD3					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.983751509776871	4.25482559580076e-06	5.267	0
RVR_N	1000	0.983626616667922	5.08867227555428e-07	52.362	1.985604010633752
RVR_N	10000	0.983603098312035	4.90359689501767e-08	522.529	2.015143722999493
RVR_N	50000	0.983484027391635	9.96947395028312e-09	2621.93	1.963765191921067
RVR_N	100000	0.983578748407413	4.90994890968513e-09	5246.764	2.1504391358704
RVR_H	100	0.981667674748318	3.55680018824869e-06	7.881	0
RVR_H	1000	0.983038959309204	2.35292606992057e-07	79.207	2.838843841343281
RVR_H	10000	0.983567600220427	2.05669901303156e-08	793.994	3.161864290208387
RVR_H	50000	0.983525134112998	4.15064925564163e-09	3981.2	3.106367705693352
RVR_H	100000	0.983638375159892	2.01202135571261e-09	7979.148	3.450694818153297
RVR_{MP}	100	0.986392719695250	1.97935805689520e-06	60.29	0
RVR_{MP}	1000	0.983486260326679	3.96261900862944e-07	597.69	0.2233855318907592
RVR_{MP}	10000	0.983321792679153	4.19596920491056e-08	5986.37	0.2055584517774311
RVR_{MP}	50000	0.983510981895885	8.32469985958468e-09	29858.6	0.2065117982158342
RVR_{MP}	100000	0.983563123066770	4.12694866040600e-09	60066.94	0.2234761001208299
RVR_T	100	0.984731680514777	2.02779524366574e-06	6.245	0
RVR_T	1000	0.984074459421842	2.18295007364523e-07	62.77	3.861156618003238
RVR_T	10000	0.983672848886148	2.34347474862454e-08	624.784	3.526476616591353
RVR_T	50000	0.983654662303700	4.68396336737877e-09	3164.13	3.463499148619177
RVR_T	100000	0.983684806039962	2.33079242777168e-09	6338.331	3.749878053674991

Tabla 3.5: Resultados numéricos de RVR según cortes, escenario Puente-PHetrgD2.

Puente-PHetrgD2					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.986116501435495	3.54299061998589e-06	5.32	0
RVR_N	1000	0.984563846674931	4.48819415656699e-07	52.51	2.043328403185379
RVR_N	10000	0.984233434741919	4.67239783149534e-08	524.847	2.08760116014211
RVR_N	50000	0.984508923890250	9.01994387831209e-09	2631.04	2.232467459496089
RVR_N	100000	0.984488789322017	4.52026265484937e-09	5276.938	2.396398890737749
RVR_H	100	0.984677813457674	4.42171531142172e-06	8.255	0
RVR_H	1000	0.983593587147684	5.06126264692077e-07	81.408	1.168761415994723
RVR_H	10000	0.984769447306007	4.34285013023744e-08	820.432	1.436820874383513
RVR_H	50000	0.984390068487267	9.15828973088481e-09	4093.58	1.41318416560488
RVR_H	100000	0.984451789767451	4.54003155150906e-09	8198.624	1.535694863198775
RVR_{MP}	100	0.985103107427991	3.54738028875824e-06	71.48	0
RVR_{MP}	1000	0.984281836542105	3.85982823560607e-07	711.9	0.1752527552496836
RVR_{MP}	10000	0.984118217974818	3.98289713407595e-08	7081.46	0.1815090007699955
RVR_{MP}	50000	0.984665175992408	7.39544317546128e-09	35550.7	0.2015134612142916
RVR_{MP}	100000	0.984552079344806	3.77287516598861e-09	71208.83	0.2127641664056643
RVR_T	100	0.985660310615239	3.95096216696268e-06	6.435	0
RVR_T	1000	0.984264425090009	4.42928721466999e-07	63.493	1.712348413534027
RVR_T	10000	0.984803307111739	4.10769235990893e-08	635.862	1.960014482749118
RVR_T	50000	0.984428422051704	8.54927961103183e-09	3178.36	1.949771939053886
RVR_T	100000	0.984445640359301	4.29008972082386e-09	6382.188	2.087703427496157

Tabla 3.6: Resultados numéricos de RVR según cortes, escenario Puente-PHetrgD3.

Puente-PHetrgD3					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.988251905387883	2.69318085198632e-06	5.278	0
RVR_N	1000	0.987963129915534	3.21789477700127e-07	52.437	2.805938122127213
RVR_N	10000	0.988480638846402	3.11650902469889e-08	527.426	2.360526828591455
RVR_N	50000	0.988596481522529	6.09405750924534e-09	2664.4	2.340734727513313
RVR_N	100000	0.988602999070845	3.01679167829994e-09	5314.814	2.445118762454453
RVR_H	100	0.989554279387329	4.41998647311680e-07	8.344	0
RVR_H	1000	0.988589909401507	8.43436152289841e-08	80.26	6.99417344228189
RVR_H	10000	0.988510908159833	8.44695124016259e-09	798.704	5.751127792665069
RVR_H	50000	0.988502041155564	1.80747976166333e-09	4001.04	5.255473315676764
RVR_H	100000	0.988522018243649	8.63058921315679e-10	8026.998	5.659000717721758
RVR_{MP}	100	0.988720377103448	6.89407580924738e-07	71.04	0
RVR_{MP}	1000	0.988647963515458	7.16675693899388e-08	710.82	0.9294059554786595
RVR_{MP}	10000	0.988391475947493	8.20665138599440e-09	7108.81	0.6650831921145799
RVR_{MP}	50000	0.988448144736920	1.66339524773802e-09	35588.8	0.6420212027145153
RVR_{MP}	100000	0.988523861478149	8.07063396725606e-10	71602.08	0.6784222150446921
RVR_T	100	0.987304712421328	1.24785092172671e-06	6.352	0
RVR_T	1000	0.988381574059324	1.14536388764474e-07	63.359	6.52433068786569
RVR_T	10000	0.988868183308335	9.78593479955041e-09	632.609	6.267598680840641
RVR_T	50000	0.988773557800255	2.00605957513735e-09	3288.17	5.761825253011703
RVR_T	100000	0.988754098031409	1.00947251532461e-09	6532.437	5.945161010290224

3.2. Resultados grafo completo

Tabla 3.7: Resultados de Monte Carlo para grafo completo

D	Método	Muestras	p_e/p_v	R	V	$t(\text{seg})$
2	<i>MCC</i>	10	0.95/0.95	1	0	0.0881052
	<i>MCC</i>	100	0.95/0.95	1	0	0.799854
	<i>MCC</i>	1000	0.95/0.95	0.9990000000000000	9.99000000000001e-07	7.6761
	<i>MCC</i>	10000	0.95/0.95	0.9990000000000000	9.99000000000001e-08	78.1588
	<i>MCC</i>	50000	0.95/0.95	0.9988800000000000	2.23749120000002e-08	382.885
	<i>MCC</i>	100000	0.95/0.95	0.9989600000000000	1.03891840000004e-08	698.174
2	<i>MCC</i>	10	0.95/0.97	1	0	0.0339
	<i>MCC</i>	100	0.95/0.97	1	0	0.3418
	<i>MCC</i>	1000	0.95/0.97	0.9980000000000000	1.99600000000000e-06	3.4427
	<i>MCC</i>	10000	0.95/0.97	0.9992000000000000	7.99360000000023e-08	34.681
	<i>MCC</i>	50000	0.95/0.97	0.9991000000000000	1.79838000000002e-08	175.286
	<i>MCC</i>	100000	0.95/0.97	0.9992900000000000	7.09495899999988e-09	358.266
4	<i>MCC</i>	10	0.95/0.95	1	0	0.166966
	<i>MCC</i>	100	0.95/0.95	1	0	0.735375
	<i>MCC</i>	1000	0.95/0.95	0.9990000000000000	9.99000000000001e-07	6.07827
	<i>MCC</i>	10000	0.95/0.95	0.9992000000000000	7.99360000000023e-08	60.3746
	<i>MCC</i>	50000	0.95/0.95	0.9990200000000000	1.95807919999996e-08	317.183
	<i>MCC</i>	100000	0.95/0.95	0.9993100000000000	6.89523899999968e-09	641.928
4	<i>MCC</i>	10	0.95/0.97	1	0	0.088728
	<i>MCC</i>	100	0.95/0.97	1	0	0.706907
	<i>MCC</i>	1000	0.95/0.97	0.9990000000000000	9.99000000000001e-07	6.64239
	<i>MCC</i>	10000	0.95/0.97	0.9996000000000000	3.99839999999956e-08	71.6179
	<i>MCC</i>	50000	0.95/0.97	0.9996400000000000	7.19740800000054e-09	390.369
	<i>MCC</i>	100000	0.95/0.97	0.9994000000000000	5.99640000000045e-09	783.526

Tabla 3.8: Error relativo RVR vs MCC, para grafo completo

Completo-PHomgD2					
Muestra	Er[%] RVR_N	Er[%] RVR_H	Er[%] RVR_{MP}	Er[%] RVR_T	Er[%] MCC
100	5.9386e-02	1.2808e-01	3.0744e-02	6.1869e-02	1.0181e-01
1000	2.4908e-02	1.7534e-02	3.3932e-03	6.3515e-03	1.7112e-03
10000	6.3842e-03	6.9506e-03	3.8356e-03	1.9320e-03	1.7112e-03
50000	8.9770e-03	1.6607e-03	1.5889e-03	9.3248e-04	1.0301e-02
100000	8.1743e-05	2.2887e-04	6.2839e-05	9.3748e-04	2.2929e-03
Completo-PHomgD4					
Muestra	Er[%] RVR_N	Er[%] RVR_H	Er[%] RVR_{MP}	Er[%] RVR_T	Er[%] MCC
100	3.1869e-02	7.5414e-03	2.9994e-02	3.7500e-02	8.2433e-02
1000	1.2214e-02	5.1150e-03	1.5803e-02	1.0035e-02	1.7650e-02
10000	2.6862e-03	3.1109e-03	2.0486e-03	4.3355e-03	2.3669e-03
50000	2.0937e-03	1.0106e-03	8.4718e-04	3.1698e-04	1.5648e-02
100000	1.1329e-03	1.4881e-04	1.7031e-03	3.8106e-03	1.3376e-02
Completo-PHetrgD2					
Muestra	Er[%] RVR_N	Er[%] RVR_H	Er[%] RVR_{MP}	Er[%] RVR_T	Er[%] MCC
100	5.3152e-02	2.6808e-02	1.0165e-01	5.1476e-02	7.7655e-02
1000	4.2903e-03	5.5853e-03	2.8908e-02	1.7812e-03	1.2250e-01
10000	1.0891e-02	5.3633e-04	7.3543e-03	2.5809e-03	2.4072e-03
50000	1.9047e-04	4.0241e-04	1.3489e-04	2.9756e-03	1.2415e-02
100000	8.5055e-04	1.2189e-03	5.8179e-03	3.4604e-03	6.5998e-03
Completo-PHetrgD4					
Muestra	Er[%] RVR_N	Er[%] RVR_H	Er[%] RVR_{MP}	Er[%] RVR_T	Er[%] MCC
100	8.6972e-02	2.0296e-02	6.1526e-03	1.7492e-02	5.7460e-02
1000	2.2486e-02	1.3771e-02	1.4545e-02	1.7682e-03	4.2597e-02
10000	2.7171e-03	1.2995e-03	2.8624e-03	2.2349e-03	1.7437e-02
50000	2.8357e-03	8.9105e-04	7.7659e-04	3.8816e-04	2.1439e-02
100000	1.1145e-03	1.6622e-03	1.3279e-03	8.8458e-04	2.5744e-03

Tabla 3.9: Resultados numéricos de RVR según cortes, escenarios Completo-PHomgD2 y Completo-PHomgD4.

Completo-PHomgD2					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.999576160485781	2.91669709093701e-08	1.853	0
RVR_N	1000	0.999231727733226	2.30083469307716e-08	19.138	17.41502669766626
RVR_N	10000	0.999046682582504	3.87903656199134e-09	201.876	9.970910248854006
RVR_N	50000	0.999072584411362	8.19322031499128e-10	960.245	10.88912544362011
RVR_N	100000	0.998983722071022	4.93683176042503e-10	1930.585	7.610406501063394
RVR_H	100	0.997703457727500	1.93733246691518e-06	3.309	0
RVR_H	1000	0.998807743612672	7.13805616029584e-08	30.902	3.476478623831908
RVR_H	10000	0.999052340382773	4.41337298393129e-09	310.622	5.695613716691536
RVR_H	50000	0.998966315397307	1.00866086052616e-09	1571.33	5.40526657676446
RVR_H	100000	0.998985191799015	4.86208252319446e-10	3138.309	4.753648676872317
RVR_{MP}	100	0.998675774141113	9.63653920011008e-07	35.77	0
RVR_{MP}	1000	0.998949008435799	5.97555366247317e-08	361.6	0.3548947210597931
RVR_{MP}	10000	0.998944588931462	5.74722623598620e-09	3596.28	0.3777735785484353
RVR_{MP}	50000	0.998998778287457	1.05277851555518e-09	18026.1	0.4514305010997027
RVR_{MP}	100000	0.998982277719690	5.35426004016877e-10	36089.64	0.3753730594062938
RVR_T	100	0.999600970352519	2.49395755025325e-08	4.191	0
RVR_T	1000	0.998919454864910	4.92773857828956e-08	42.348	3.674730954279989
RVR_T	10000	0.998963604719930	4.62944624723389e-09	423.45	3.983017257047206
RVR_T	50000	0.998973590113503	9.41488736439881e-10	2116.21	4.299874673836978
RVR_T	100000	0.998973540219357	4.77029707946794e-10	4238.265	3.587662453227032
Completo-PHomgD4					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.999494781009976	2.77450820922091e-08	1.941	0
RVR_N	1000	0.999298386955255	2.85954767831143e-08	19.222	11.04713244870291
RVR_N	10000	0.999203190307103	3.04786256083393e-09	193.941	8.164539020852082
RVR_N	50000	0.999197270439929	6.47997752570392e-10	964.847	9.933635178167883
RVR_N	100000	0.999165030572354	3.60025660926792e-10	1935.945	6.35051830678693
RVR_H	100	0.999251702297344	2.83114416087663e-07	2.914	0
RVR_H	1000	0.999227458891937	1.53650200839797e-08	31.071	12.71912140614168
RVR_H	10000	0.999145267017187	2.17398179716378e-09	311.526	7.126010806432124
RVR_H	50000	0.999166253293265	4.07397550124974e-10	1562.33	9.757733919179195
RVR_H	100000	0.999177837601232	1.97422396278569e-10	3144.607	7.129726127035577
RVR_{MP}	100	0.999476048625937	2.85115837106368e-08	35.99	0
RVR_{MP}	1000	0.999018452723670	4.50064999180060e-08	359.84	0.3749391564999873
RVR_{MP}	10000	0.999196819840173	2.86776675849905e-09	3571.8	0.4711570777604787
RVR_{MP}	50000	0.999184815557035	5.93625635058793e-10	17871.7	0.585412031267778
RVR_{MP}	100000	0.999159334161044	3.18099282704431e-10	35712.33	0.3896321345212301
RVR_T	100	0.999551039094707	2.71562889390360e-08	4.204	0
RVR_T	1000	0.999076087957466	2.70823952927906e-08	42.24	5.308043749762708
RVR_T	10000	0.999219669995308	1.85594918594964e-09	421.764	6.165397611880493
RVR_T	50000	0.999173183508886	4.20042802932922e-10	2113.21	6.996872362238477
RVR_T	100000	0.999214425413103	1.95916584337164e-10	4229.865	5.341188881256092

Tabla 3.10: Resultados numéricos de RVR según cortes, escenarios Completo-PHetrgD2 y Completo-PHetrgD4.

Completo-PHetrgD2					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.998692941999725	6.68872050711677e-07	2.114	0
RVR_N	1000	0.999266923238865	2.68627568936101e-08	19.13	13.37193216273961
RVR_N	10000	0.999115228045937	3.82715153989865e-09	200.686	3.609452144563667
RVR_N	50000	0.999222150223702	6.25536152821434e-10	955.579	5.273630323222833
RVR_N	100000	0.999215554576171	3.09178301100643e-10	1952.005	4.211778786116843
RVR_H	100	0.999491929026162	2.32220149389848e-08	3.207	0
RVR_H	1000	0.999279862878788	2.74816373236347e-08	30.982	8.070634191329011
RVR_H	10000	0.999218694317491	3.17040325875610e-09	310.814	2.813318782351745
RVR_H	50000	0.999228074457682	6.12714296320024e-10	1566.87	3.2835052536666351
RVR_H	100000	0.999236232716695	2.97622000177798e-10	3134.402	2.724807074592641
RVR_{MP}	100	0.998208366637799	1.25909228992340e-06	34.98	0
RVR_{MP}	1000	0.999512906469742	9.10980829149762e-09	360.67	2.091416157776101
RVR_{MP}	10000	0.999150567564670	3.46009657767519e-09	3622.41	0.2211811943596193
RVR_{MP}	50000	0.999225401356564	5.82629911115587e-10	18087.7	0.2991249036362888
RVR_{MP}	100000	0.999165919312054	3.24889267377791e-10	36365.24	0.2151461912302661
RVR_T	100	0.998709695834781	6.64260091896584e-07	4.297	0
RVR_T	1000	0.999206255255724	2.49449527715407e-08	42.278	6.515722744733725
RVR_T	10000	0.999198264653190	2.96922476078659e-09	422.287	2.21097197815568
RVR_T	50000	0.999253786381762	5.50515969048107e-10	2106.96	2.717706059472784
RVR_T	100000	0.999258631028330	2.71713892616770e-10	4215.697	2.219086605451368
Completo-PHetrgD4					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.998556508473492	7.20512740085163e-07	2.405	0
RVR_N	1000	0.999200995743035	2.93926305712254e-08	19.718	11.44955324000714
RVR_N	10000	0.999452885204679	1.72488693892394e-09	189.976	8.73873255656707
RVR_N	50000	0.999454070527530	3.46367090827492e-10	957.865	8.468578505367555
RVR_N	100000	0.999436868094060	1.82869385531039e-10	1935.973	13.27100174061314
RVR_H	100	0.999628568819650	1.47826950914258e-08	3.05	0
RVR_H	1000	0.999288094622517	9.00752723405877e-09	31.906	23.08936103385541
RVR_H	10000	0.999438716840074	8.50012897091337e-10	307.61	10.9517080075433
RVR_H	50000	0.999416824104041	1.69789149453667e-10	1551.41	10.66632777034393
RVR_H	100000	0.999409117195527	9.05458823677123e-11	3122.118	16.61980750860769
RVR_{MP}	100	0.999364238701549	3.03039060196682e-08	37.8	0
RVR_{MP}	1000	0.999571096407665	3.65786982271268e-09	357.67	5.071998296266971
RVR_{MP}	10000	0.999397121771890	8.46123779218664e-10	3622.23	0.9343248449436411
RVR_{MP}	50000	0.999433490964966	1.52848191903166e-10	18088.3	1.016233217290262
RVR_{MP}	100000	0.999412457635519	8.32601728068558e-11	36240.51	1.557085280442576
RVR_T	100	0.999600548421587	1.47260552355812e-08	4.236	0
RVR_T	1000	0.999443401740608	6.06103288619354e-09	42.423	25.80725671276343
RVR_T	10000	0.999403393632852	1.01353254419095e-09	421.109	6.709275098842982
RVR_T	50000	0.999421850141623	1.80318086318765e-10	2110.31	7.383561521467495
RVR_T	100000	0.999434570210754	8.97928332773119e-11	4224.957	12.38454453916949

3.3. Resultados grafo grilla $n = 3$

Tabla 3.11: Resultados de Monte Carlo para grafo grilla $n = 3$

D	Método	Muestras	p_e/p_v	R	V	$t(\text{seg})$
4	<i>MCC</i>	10	0.95/0.95	0.9000000000000000	0.009000000000000000	0.0788
	<i>MCC</i>	100	0.95/0.95	0.9600000000000000	3.84000000000000e-04	0.7496
	<i>MCC</i>	1000	0.95/0.95	0.9730000000000000	2.62710000000000e-05	7.5544
	<i>MCC</i>	10000	0.95/0.95	0.9755000000000000	2.38997500000000e-06	75.7485
	<i>MCC</i>	50000	0.95/0.95	0.9740200000000000	5.06100792000000e-07	383.249
	<i>MCC</i>	100000	0.95/0.95	0.9746900000000000	2.46694039000001e-07	778.2324
4	<i>MCC</i>	10	0.95/0.97	1	0	0.0848
	<i>MCC</i>	100	0.95/0.97	0.9800000000000000	1.96000000000000e-04	0.7737
	<i>MCC</i>	1000	0.95/0.97	0.9870000000000000	1.28310000000000e-05	7.7654
	<i>MCC</i>	10000	0.95/0.97	0.9818000000000000	1.78687600000000e-06	77.2032
	<i>MCC</i>	50000	0.95/0.97	0.9845400000000000	3.04419768000001e-07	393.911
	<i>MCC</i>	100000	0.95/0.97	0.9835700000000000	1.61600550999999e-07	802.5133

Tabla 3.12: Error relativo RVR vs MCC, para grafo grilla $n = 3$

GrillaN3-PHomgD4					
Muestra	Er[%] RVR_N	Er[%] RVR_H	Er[%] RVR_{MP}	Er[%] RVR_T	Er[%] MCC
100	0.4335852	0.1007755	0.4390779	0.9527378	1.4107022
1000	0.1508270	0.1997718	0.0455110	0.7076632	0.0756388
10000	0.0801897	0.0096424	0.0135390	0.7327236	0.1811042
50000	0.0059928	0.0054216	0.0185909	0.7557941	0.0291123
100000	0.0010809	0.0208641	0.0142751	0.7488344	0.0979195
GrillaN3-PHetrgD4					
Muestra	Er[%] RVR_N	Er[%] RVR_H	Er[%] RVR_{MP}	Er[%] RVR_T	Er[%] MCC
100	0.0530651	0.0742632	0.1772419	0.5901033	0.3441278
1000	0.0851374	0.0107328	0.1172033	0.4291584	0.3676999
10000	0.0072632	0.0200877	0.0214030	0.4569033	0.1610864
50000	0.0056899	0.0021607	0.0024586	0.4449926	0.1175433
100000	0.0058452	0.0039100	0.0016720	0.4389850	0.0189044

Tabla 3.13: Resultados numéricos de RVR según cortes, escenario GrillaN3-PHomgD4.

GrillaN3-PHomgD4					
Método	Muestras	R	V	$t(\text{seg})$	W
$RV R_N$	100	0.977958499903957	1.10349345102042e-05	44.55	0.5855221566764959
$RV R_N$	1000	0.975205179630208	1.00780582421062e-06	444.73	0.4427955939363074
$RV R_N$	10000	0.972955685579500	1.32086568322689e-07	4389.07	0.3122742908812341
$RV R_N$	50000	0.973794876456226	2.40101586359066e-08	21736.8	0.3716442418506659
$RV R_N$	100000	0.973725997519989	1.20671302562833e-08	43440.74	0.3662408201540796
$RV R_H$	100	0.974717810058248	5.79978634872236e-06	56.63	0.8763997504172641
$RV R_H$	1000	0.975681773777557	7.41158133437526e-07	572.59	0.4676510816609123
$RV R_H$	10000	0.973642630708902	9.72507158361709e-08	5664.94	0.3286088640722372
$RV R_H$	50000	0.973683730795474	1.90468151340760e-08	28899.5	0.352375213262614
$RV R_H$	100000	0.973939683784700	9.26934616066970e-09	58923.35	0.3515049394086779
$RV R_{MP}$	100	0.978011984425463	9.83223493745021e-06	714	0.04100250055578338
$RV R_{MP}$	1000	0.973293365569707	1.13746416918861e-06	6954.1	0.02508984189257762
$RV R_{MP}$	10000	0.973604687803603	1.06980843547346e-07	69961.9	0.02418799008529965
$RV R_{MP}$	50000	0.973555496024960	2.12811110763489e-08	349256	0.02609635628404311
$RV R_{MP}$	100000	0.973597520452872	1.04688680751390e-08	698031.2	0.02627201609325076
$RV R_T$	100	0.983013678815781	9.62732635289152e-07	22.55	13.25893257114924
$RV R_T$	1000	0.980627297332895	2.00515288385423e-07	220.55	4.487681512514594
$RV R_T$	10000	0.980871319855073	1.85874248486584e-08	2210.21	4.406711613968181
$RV R_T$	50000	0.981095965679361	3.64960644087593e-09	11071.8	4.800138636681369
$RV R_T$	100000	0.981028196753592	1.82500566516843e-09	22149.6	4.749390275080975

Tabla 3.14: Resultados numéricos de RVR según cortes, escenario GrillaN3-PHetrgD4.

GrillaN3-PHetrgD4					
Método	Muestras	R	V	$t(\text{seg})$	W
$RV R_N$	100	0.982862263673357	6.31133319889372e-06	45	0.5339431823887898
$RV R_N$	1000	0.984221325566276	5.96360932524231e-07	443.3	0.3768924345904117
$RV R_N$	10000	0.983312672907840	6.65847924547998e-08	4425.57	0.468150458131928
$RV R_N$	50000	0.983440050738464	1.28699024867391e-08	21861.2	0.4262081062817848
$RV R_N$	100000	0.983326616399084	6.56073151109555e-09	43650.53	0.4528488711195092
$RV R_H$	100	0.984114390456097	2.66920421457451e-06	56.84	0.9995230113586667
$RV R_H$	1000	0.983278552674023	4.41266770139623e-07	569.86	0.3962369196178188
$RV R_H$	10000	0.983581637288610	3.86156255241435e-08	5693.84	0.6274243713961131
$RV R_H$	50000	0.983362849438125	8.15553869018761e-09	28531.5	0.5153398210720123
$RV R_H$	100000	0.983345647273829	4.06534198282098e-09	58656.53	0.5438531189036848
$RV R_{MP}$	100	0.985127066432104	3.12828573937205e-06	820.5	0.05908043277307092
$RV R_{MP}$	1000	0.984536656071471	2.04993413653891e-07	8029.7	0.06053201192960642
$RV R_{MP}$	10000	0.983173624185462	3.84724756821705e-08	80049.7	0.04479400303394323
$RV R_{MP}$	50000	0.983359920381978	7.33800309458068e-09	400200	0.04083344150438553
$RV R_{MP}$	100000	0.983367655801732	3.68903662581332e-09	799527.9	0.04396919206708462
$RV R_T$	100	0.989187079769509	2.87712705832736e-07	22.68	23.23948911795167
$RV R_T$	1000	0.987604372686889	7.01145326152082e-08	222.45	6.388279114313347
$RV R_T$	10000	0.987877212428830	6.40401848403803e-09	2225.7	9.678554835992689
$RV R_T$	50000	0.987760083874331	1.34728910520180e-09	11105.6	8.014346853827625
$RV R_T$	100000	0.987701006257567	6.85319458561192e-10	22197.95	8.524897069401645

3.4. Resultados grafo grilla $n = 5$

Tabla 3.15: Resultados de Monte Carlo para grafo grilla $n = 5$

D	Método	Muestras	p_e/p_v	R	V	$t(\text{seg})$
8	<i>MCC</i>	10	0.95/0.95	1	0	0.695
	<i>MCC</i>	100	0.95/0.95	0.9700000000000000	2.91000000000000e-04	3.936
	<i>MCC</i>	1000	0.95/0.95	0.9580000000000000	4.02360000000000e-05	39.66
	<i>MCC</i>	10000	0.95/0.95	0.9621000000000000	3.64635900000000e-06	395.304
	<i>MCC</i>	50000	0.95/0.95	0.9606000000000000	7.56952800000000e-07	1989.16
	<i>MCC</i>	100000	0.95/0.95	0.9606600000000000	3.77923644000000e-07	3998.476
8	<i>MCC</i>	10	0.95/0.97	1	0	0.641
	<i>MCC</i>	100	0.95/0.97	0.9500000000000000	4.75000000000000e-04	4.042
	<i>MCC</i>	1000	0.95/0.97	0.9640000000000000	3.47040000000000e-05	40.781
	<i>MCC</i>	10000	0.95/0.97	0.9796000000000000	1.99838400000000e-06	411.62
	<i>MCC</i>	50000	0.95/0.97	0.9755800000000000	4.76473272000000e-07	2060.81
	<i>MCC</i>	100000	0.95/0.97	0.9762100000000000	2.32240359000000e-07	4132.686

Tabla 3.16: Resultados numéricos de RVR según cortes, escenario GrillaN5-PHomgD8.

GrillaN5-PHomgD8					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	10	0.967062065953647	1.09187055778926e-04	118	0
RVR_N	100	0.961512376478487	1.23620655214386e-05	1198	0.07733929690807358
RVR_N	1000	0.960626976602956	1.64022918094331e-06	11928	0.08156340443803008
RVR_N	10000	0.960191027370488	1.48908799025915e-07	119688	0.08087599981590661
RVR_N	50000	0.960144040858126	3.12235102397580e-08	597815	0.08066589991376415
RVR_N	100000	0.959953691668830	1.57677743237541e-08	1195169	0.08018605390582524
RVR_H	10	0.972268361376017	9.71373978473407e-05	131	0
RVR_H	100	0.958817755676263	1.57208523238298e-05	1268	0.05745829531297754
RVR_H	1000	0.960937421344590	1.15546170945618e-06	12805	0.107853034438518
RVR_H	10000	0.960182886386751	1.45935294074653e-07	128222	0.07703139259301352
RVR_H	50000	0.959900655912783	2.86125143402746e-08	640446	0.08216748161836585
RVR_H	100000	0.959832664187997	1.42211148947404e-08	1280073	0.08300995427186837
RVR_{MP}	10	0.954440882648640	1.17623127823184e-04	5.25e+03	0
RVR_{MP}	100	0.960670666209915	1.28267757471420e-05	4.965e+04	0.001798503651729326
RVR_{MP}	1000	0.959778522812719	1.45661798414037e-06	4.8763e+05	0.002246629334498756

Tabla 3.17: Resultados numéricos de RVR según cortes, escenario GrillaN5-PHetrGD8.

GrillaN5-PHetrGD8					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	10	0.976482308828009	2.01670422553515e-05	117	0
RVR_N	100	0.977585937389854	4.13169977126720e-06	1238	0.3753535416464043
RVR_N	1000	0.975989307746411	6.70244744855434e-07	11929	0.1770108752763077
RVR_N	10000	0.976049392694324	6.80916922637637e-08	119102	0.1014290201742276
RVR_N	50000	0.975708708543794	1.38346358715308e-08	595453	0.1191958887438466
RVR_N	100000	0.975862768933226	6.76813756482742e-09	1189275	0.1192390854163984
RVR_H	10	0.978503046997738	2.08908987563768e-05	130	0
RVR_H	100	0.976757031682116	3.09273649262179e-06	1268	0.4895845976635308
RVR_H	1000	0.976301126754376	4.73121788885439e-07	12585	0.2376901783549442
RVR_H	10000	0.975904092515316	4.81125652275149e-08	126248	0.1354229881750868
RVR_H	50000	0.975752750906069	9.41463879661076e-09	629654	0.1656421533370054
RVR_H	100000	0.976074238623107	4.61832417782299e-09	1259367	0.1650187829218065
RVR_{MP}	10	0.970153015968474	9.32781683520247e-05	4.17e+03	0
RVR_{MP}	100	0.975716574610537	3.83534946889851e-06	5.195e+04	0.009636058431639989
RVR_{MP}	1000	0.975656778478381	5.14307360149369e-07	5.0056e+05	0.005497415014350056

3.5. Resultados grafo arpanet

Tabla 3.18: Resultados de Monte Carlo para grafo arpanet

D	Método	Muestras	p_e/p_v	R	V	$t(\text{seg})$
8	<i>MCC</i>	10	0.95/0.95	0.7000000000000000	0.0210000000000000	0.423
	<i>MCC</i>	100	0.95/0.95	0.9400000000000000	5.64000000000000e-04	2.382
	<i>MCC</i>	1000	0.95/0.95	0.8390000000000000	1.35079000000000e-04	22.517
	<i>MCC</i>	10000	0.95/0.95	0.8635000000000000	1.17867750000000e-05	229.003
	<i>MCC</i>	50000	0.95/0.95	0.8557800000000000	2.46841183200000e-06	1141.89
	<i>MCC</i>	100000	0.95/0.95	0.8584500000000000	1.21513597500000e-06	2308.715
8	<i>MCC</i>	10	0.95/0.97	0.9000000000000000	0.0090000000000000	0.615
	<i>MCC</i>	100	0.95/0.97	0.9500000000000000	4.75000000000000e-04	2.564
	<i>MCC</i>	1000	0.95/0.97	0.9030000000000000	8.75910000000000e-05	23.965
	<i>MCC</i>	10000	0.95/0.97	0.9052000000000000	8.58129600000000e-06	238.152
	<i>MCC</i>	50000	0.95/0.97	0.9018400000000000	1.77049228800000e-06	1200.98
	<i>MCC</i>	100000	0.95/0.97	0.9066400000000000	8.46439104000000e-07	2423.384

Tabla 3.19: Resultados numéricos de RVR según cortes, escenario Arpanet-PHomgD8.

Arpanet-PHomgD8					
Método	Muestras	R	V	$t(\text{seg})$	\bar{W}
RVR_N	10	0.846744770320000	0.00143013699449815	34.2	0.1816167563698382
RVR_N	100	0.870384551688620	1.92223613533224e-04	345.8	0.02021106203480804
RVR_N	1000	0.863253564044588	1.99893796735799e-05	3416.4	0.04453796139892532
RVR_N	10000	0.858259474857843	2.11715967915175e-06	33980.3	0.03751935311109129
RVR_N	50000	0.858463346380045	4.21486140280362e-07	169772	0.03939059364227345
RVR_N	100000	0.858525304526836	2.09942301146224e-07	339627.7	0.03934523522903562
RVR_H	10	0.848908835703224	0.00220526781566171	44.6	0.09031574404385034
RVR_H	100	0.861119018601443	2.25369469557065e-04	451.6	0.01319993660281434
RVR_H	1000	0.856450072404487	2.18213468417823e-05	4521.4	0.03082789580237216
RVR_H	10000	0.857078215165752	2.07838087962941e-06	44759.2	0.02901540995541487
RVR_H	50000	0.857053252857345	4.23349351808302e-07	224157	0.02970234131841577
RVR_H	100000	0.857513388156288	2.11062658702432e-07	448536.2	0.02963372861865563
RVR_{MP}	10	0.852572082014491	0.00221964494437453	782	0.005117636073801929
RVR_{MP}	100	0.861015212896882	1.99972820852548e-04	7824	0.0008586596329691539
RVR_{MP}	1000	0.863121795935774	1.92458350402822e-05	79496	0.001987999765422245
RVR_{MP}	10000	0.857098458961462	2.15472205631576e-06	787207	0.001591314281793432

Tabla 3.20: Resultados numéricos de RVR según cortes, escenario Arpanet-PHetrgD8.

Arpanet-PHetrgD8					
Método	Muestras	R	V	$t(\text{seg})$	W
$RV R_N$	10	0.938900465492784	2.78296108492530e-04	32.8	0.6063685220540177
$RV R_N$	100	0.903148616537558	1.56078292832464e-04	329.1	0.02371052891829123
$RV R_N$	1000	0.904157565587314	1.41447104728925e-05	3382.2	0.04387767129655841
$RV R_N$	10000	0.906135978930050	1.37804484577469e-06	33642	0.04408206779477259
$RV R_N$	50000	0.904422569469940	2.87192459199832e-07	167707	0.04414744679495271
$RV R_N$	100000	0.904860594220795	1.42973243530668e-07	334159.6	0.04293478008917199
$RV R_H$	10	0.916032897937833	4.04322302252167e-04	48.6	0.2816784734715397
$RV R_H$	100	0.908504137447458	1.16303497240831e-04	458.2	0.02285408148380585
$RV R_H$	1000	0.902659741651874	1.46675339007182e-05	4450.7	0.03215522076697007
$RV R_H$	10000	0.903194139454144	1.42043159289305e-06	44600.8	0.03225849812072402
$RV R_H$	50000	0.904481232561038	2.82402112663118e-07	221606	0.03397663450229325
$RV R_H$	100000	0.905778063440806	1.38375123198097e-07	443815.3	0.03340086118573733
$RV R_{MP}$	10	0.923250711704262	3.68845657654480e-04	791	0.0189712726984267
$RV R_{MP}$	100	0.898595272879237	1.47109668845918e-04	7923	0.00104491450211814
$RV R_{MP}$	1000	0.904497757800504	1.30189909865440e-05	79622	0.002025006908975156
$RV R_{MP}$	10000	0.902972129814028	1.38878414300013e-06	800053	0.001839304394659447

3.6. Resultados grafo dodecaedro

Tabla 3.21: Resultados de Monte Carlo para grafo dodecaedro

D	Método	Muestras	p_e/p_v	R	V	$t(\text{seg})$
5	<i>MCC</i>	10	0.95/0.95	0.9000000000000000	0.0090000000000000	0.406
	<i>MCC</i>	100	0.95/0.95	0.9900000000000000	9.90000000000001e-05	2.693
	<i>MCC</i>	1000	0.95/0.95	0.9950000000000000	4.9750000000000e-06	26.862
	<i>MCC</i>	10000	0.95/0.95	0.9909000000000000	9.0171900000000e-07	268.588
	<i>MCC</i>	50000	0.95/0.95	0.9906000000000000	1.8623279999999e-07	1350.42
	<i>MCC</i>	100000	0.95/0.95	0.9909600000000000	8.95827840000005e-08	2724.406
5	<i>MCC</i>	10	0.95/0.97	1	0	0.439
	<i>MCC</i>	100	0.95/0.97	0.9900000000000000	9.90000000000001e-05	2.736
	<i>MCC</i>	1000	0.95/0.97	0.9980000000000000	1.9960000000000e-06	27.731
	<i>MCC</i>	10000	0.95/0.97	0.9954000000000000	4.57884000000005e-07	275.641
	<i>MCC</i>	50000	0.95/0.97	0.9954400000000000	9.07841280000002e-08	1382.25
	<i>MCC</i>	100000	0.95/0.97	0.9958300000000000	4.15261110000001e-08	2783.766
8	<i>MCC</i>	10	0.95/0.95	1	0	0.418
	<i>MCC</i>	100	0.95/0.95	1	0	2.721
	<i>MCC</i>	1000	0.95/0.95	0.9960000000000000	3.9840000000000e-06	26.795
	<i>MCC</i>	10000	0.95/0.95	0.9958000000000000	4.18235999999998e-07	268.171
	<i>MCC</i>	50000	0.95/0.95	0.9971400000000000	5.70364079999995e-08	1348.85
	<i>MCC</i>	100000	0.95/0.95	0.9971900000000000	2.80210389999998e-08	2722.126
8	<i>MCC</i>	10	0.95/0.97	1	0	0.48
	<i>MCC</i>	100	0.95/0.97	1	0	2.741
	<i>MCC</i>	1000	0.95/0.97	1	0	27.647
	<i>MCC</i>	10000	0.95/0.97	0.9990000000000000	9.99000000000001e-08	277.012
	<i>MCC</i>	50000	0.95/0.97	0.9985200000000000	2.95561920000007e-08	1389.34
	<i>MCC</i>	100000	0.95/0.97	0.9987800000000000	1.21851160000000e-08	2799.113

Tabla 3.22: Resultados numéricos de RVR según cortes, escenarios Dodecaedro-PHomgD5 y Dodecaedro-PHomgD8.

Dodecaedro-PHomgD5					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.992362337237879	4.89763248151055e-06	532.9	0.1310345931094835
RVR_N	1000	0.991477197750138	9.73716512318653e-07	5345.9	0.08693341497845193
RVR_N	10000	0.992432034343813	6.78988105892296e-08	53147.3	0.09601235628028787
RVR_N	50000	0.991256315777714	1.86509528846714e-08	265644	0.05849338849660871
RVR_N	100000	0.991315667016554	8.99497976258269e-09	530749.4	0.06462085960934393
RVR_H	100	0.993446356583963	3.35797263771031e-06	725.3	0.1404180852887663
RVR_H	1000	0.992869788595311	5.25775900037317e-07	7344.8	0.1171816086958314
RVR_H	10000	0.991348029820296	7.90377388375155e-08	73237.1	0.05985560912027218
RVR_H	50000	0.991318840864841	1.65984008354156e-08	366207	0.04767765942094753
RVR_H	100000	0.991648295096543	7.50747794395144e-09	732325.9	0.05611306280671436
RVR_{MP}	100	0.994586359399006	2.14197438061631e-06	1.285e+04	0.0124251167301098
RVR_{MP}	1000	0.992624724855487	7.91571425089362e-07	1.2904e+05	0.004430223567350408
RVR_{MP}	10000	0.991879613451966	8.60128356084743e-08	1.29208e+06	0.003117581613314535
Dodecaedro-PHomgD8					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.998545558580739	2.85691966413419e-07	484.5	0
RVR_N	1000	0.997290200459905	1.66631635511022e-07	4850.3	0.1320830329357
RVR_N	10000	0.997198831045864	1.49350339493918e-08	48387.3	0.1552013949575489
RVR_N	50000	0.997154399732272	3.24188988024221e-09	242165	0.09799551273359772
RVR_N	100000	0.997132766530842	1.58032940529365e-09	484366.3	0.09964853490906435
RVR_H	100	0.997588197565107	1.06085558911704e-06	828.2	0
RVR_H	1000	0.997328193155849	8.14158282058915e-08	8200.3	0.1598948607423085
RVR_H	10000	0.997014303521140	1.53787411201871e-08	82215.7	0.08870695644176188
RVR_H	50000	0.997267892580370	2.32458466035870e-09	411777	0.08037266527746902
RVR_H	100000	0.997160742342967	1.26908332733116e-09	822596.1	0.07306605736870139
RVR_{MP}	100	0.995914364906519	1.60252980009320e-06	1.161e+04	0
RVR_{MP}	1000	0.997642345343174	6.33977623167845e-08	1.1789e+05	0.01428309070211435
RVR_{MP}	10000	0.997216593238235	1.27146052877414e-08	1.17217e+06	0.007525575883516843

Tabla 3.23: Resultados numéricos de RVR según cortes, escenarios Dodecaedro-PHetrgD5 y Dodecaedro-PHetrgD8.

Dodecaedro-PHetrgD5					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.993820300748625	5.34989365102794e-06	528.5	0.1228801232655921
RVR_N	1000	0.996469133413433	2.92736670945928e-07	5325.8	0.1617642354493845
RVR_N	10000	0.995703715987427	4.01347275709610e-08	52828.3	0.08421292920060068
RVR_N	50000	0.995663889464930	8.01934701210109e-09	263963	0.07139083397928141
RVR_N	100000	0.995794111912085	3.86928538175025e-09	525869.1	0.07229515923930972
RVR_H	100	0.993043430334485	7.66733957008263e-06	718	0.06311064764952085
RVR_H	1000	0.995532511826908	4.44375047641810e-07	7307	0.07767043195932617
RVR_H	10000	0.995764587345599	3.19608797449023e-08	72720	0.07682333090172314
RVR_H	50000	0.995799559559027	6.35770118129654e-09	363220	0.06544171191858576
RVR_H	100000	0.995661221207980	3.55386265981770e-09	722371.2	0.05730025708286206
RVR_{MP}	100	0.993179259540902	8.89802078844656e-06	1.359e+04	0.002873154009614712
RVR_{MP}	1000	0.996143458405029	2.86821657307284e-07	1.3815e+05	0.006364754580425738
RVR_{MP}	10000	0.995240276085987	4.29953066553412e-08	1.37888e+06	0.003011745051242174
Dodecaedro-PHetrgD8					
Método	Muestras	R	V	$t(\text{seg})$	W
RVR_N	100	0.999479572359225	1.00777957746410e-08	484.7	0
RVR_N	1000	0.998874022605519	3.96023767592125e-08	4828.1	0
RVR_N	10000	0.998723117172145	6.32375099825193e-09	48238.8	0.09071786368296053
RVR_N	50000	0.998706541048707	1.11330768233651e-09	241076	0.1529987334959919
RVR_N	100000	0.998663312476282	6.41997269774738e-10	481082.5	0.1104326156820188
RVR_H	100	0.998543909777552	2.16414552299098e-07	806.1	0
RVR_H	1000	0.998848734132255	2.83391052830255e-08	8147.1	0
RVR_H	10000	0.998725756939106	3.64827719708079e-09	81361.6	0.09323023133161082
RVR_H	50000	0.998689596159160	8.94315941520212e-10	406837	0.1128614441651089
RVR_H	100000	0.998659915622506	4.59163237426283e-10	812833.7	0.09138634094407873
RVR_{MP}	100	0.998883127743664	1.17168979979897e-07	1.273e+04	0
RVR_{MP}	1000	0.998825923569098	4.49758523065761e-08	1.311e+05	0
RVR_{MP}	10000	0.998708020934953	3.54786189676418e-09	1.30327e+06	0.005984983624896768

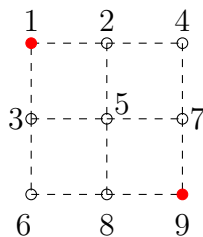
Anexo 4

Algoritmo RVR en acción

RVR en analogía a MCC sorteos estados del sistema pero estos son dependientes a cierto corte que debe ser encontrado. Bajo un enfoque de descomposición recursiva de la ecuación de la anticonfiabilidad, utilizando conceptos de mincut el autor define una v.a que es un estimador de la anticonfiabilidad. Esta v.a sorteos estados sobre un espacio conocido de vectores de estados. Estos vectores de estados son definidos sobre la condicional de la ocurrencia o no del corte, en caso de que no ocurra el corte uno de los elementos del mincut debe estar operativo. En cada paso de la iteración el estado sorteado fija elementos en falla y el primer operativo por lo que el sistema es reducido recursivamente a un estado operativo o en falla, obteniendo una estimación de la anticonfiabilidad. Realizando N observaciones del método la medida de anticonfiabilidad queda determinada por la media de las muestras obtenidas

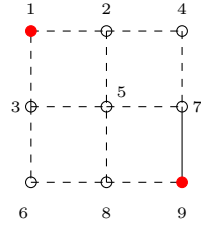
En el ejemplo se grafica una muestra de RVR con la estrategia Corte Lineal. En la secuencia de cortes del 1 al 16 se puede observar los distintos cortes elegidos por la estrategia planteada y como RVR a través del sorteo va fijando elementos.

Figura 4.1: Grafo Grilla 3×3 , $d = 4$ - DCR $s-t$ -terminal



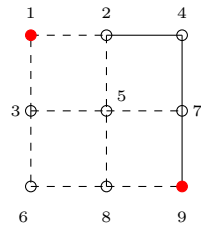
(a) Corte 1

$$\begin{cases} C = \{(7, 9), (8, 9)\} \\ Bi = 1 \\ Tipo = 001 \end{cases}$$

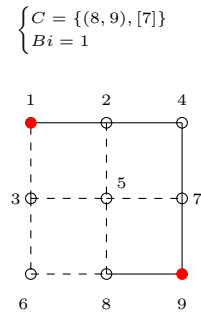


(c) Corte 3

$$\begin{cases} C = \{(2, 4), (5, 7), (8, 9)\} \\ Bi = 1 \\ Tipo = 001 \end{cases}$$

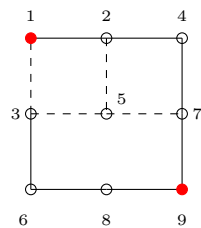


(e) Corte 5



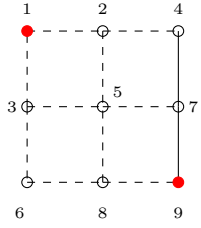
(g) Corte 7

$$\begin{cases} C = \{(3, 6), [7]\} \\ Bi = 1 \end{cases}$$



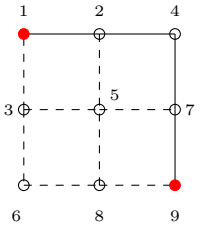
(b) Corte 2

$$\begin{cases} C = \{(4, 7), (5, 7), (8, 9)\} \\ Bi = 1 \\ Tipo = 001 \end{cases}$$



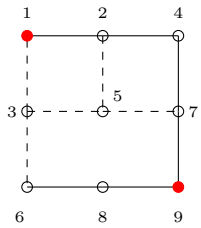
(d) Corte 4

$$\begin{cases} C = \{(1, 2), (5, 7), (8, 9)\} \\ Bi = 1 \end{cases}$$



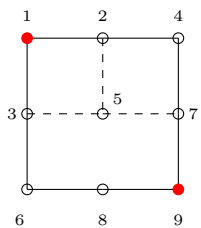
(f) Corte 6

$$\begin{cases} C = \{(5, 8), (6, 8), [7]\} \\ Bi = [01] \end{cases}$$



(h) Corte 8

$$\begin{cases} C = \{(1, 3), [7]\} \\ Bi = 1 \end{cases}$$



(1)

(2)

(3)

(4)

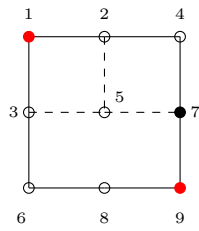
(7)

(8)

(a) Corte 9

$$\begin{cases} C = \{[7], [8]\} \\ Bi = 1 \end{cases}$$

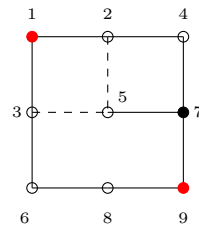
(9)



(b) Corte 10

$$\begin{cases} C = \{(5, 7), [4], [8]\} \\ Bi = 1 \end{cases}$$

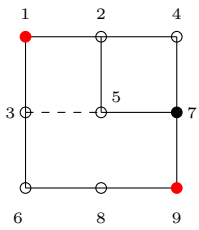
(10)



(c) Corte 11

$$\begin{cases} C = \{(2, 5), (3, 5), [4], [8]\} \\ Bi = 1 \end{cases}$$

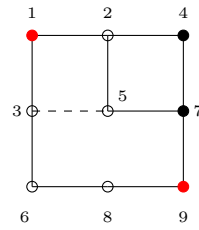
(11)



(d) Corte 12

$$\begin{cases} C = \{[4], [5], [8]\} \\ Bi = 1 \end{cases}$$

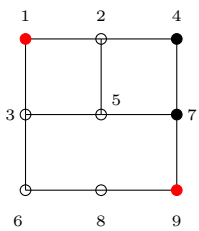
(12)



(e) Corte 13

$$\begin{cases} C = \{(3, 5), [2], [8]\} \\ Bi = 1 \\ Tipo = 000 \end{cases}$$

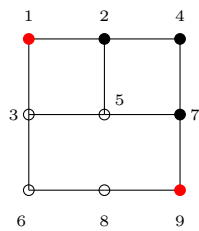
(13)



(f) Corte 14

$$\begin{cases} C = \{[2], [5], [8]\} \\ Bi = 1 \\ Tipo = 010 \end{cases}$$

(14)



(g) Corte 15

$$\begin{cases} C = \{\} \\ Bi = 1 \\ Tipo = 001 \end{cases}$$

(15)

Corte vacío, FIN: Estado operativo del sistema