

**ALGORITMOS EVOLUTIVOS APLICADOS A
PROBLEMAS DE DISEÑO DE REDES
CONFIABLES**

Autor: Ing. Héctor Enrique Magnago

Director Local: Lic. Francisco Javier Díaz

Director Externo: Mg. José Luís Hernández

Tesis presentada para obtener el grado de

Magíster en Redes de Datos

FACULTAD DE INFORMÁTICA

UNIVERSIDAD NACIONAL DE LA PLATA

Febrero de 2006

Agradecimientos

Esta es una excelente oportunidad para manifestar mi más sincero agradecimiento a todos aquellos que de una forma u otra me ha motivado a seguir adelante, a no rendirme y ser cada vez mejor.

A mis directores Javier Díaz y José Hernández por su invaluable guía y ayuda en la elaboración del mismo, especialmente a José por su incondicional apoyo durante la escritura de esta tesis y su disposición a colaborar con las dificultades.

Al Centro de Cómputos y a la Facultad de Ingeniería de la Universidad Nacional de Río Cuarto, que brindaron los recursos necesarios para la realización de esta maestría y para la ejecución del proyecto de investigación del cual formó parte esta tesis.

A mis compañeros de trabajo por sus palabras de aliento y confianza.

Por último quiero destacar mi agradecimiento a mi familia; a mis padres, a quienes les debo mucho de lo bueno que he aprendido en la vida y, muy especialmente a mi hija Gabriela, por su comprensión, y compañía, por las penas y alegrías vividas juntos y sobre todo, por ser la fuerza que me mantuvo siempre en pie.

Índice

<u>1</u>	<u>INTRODUCCIÓN</u>	9
1.1	ANTECEDENTES Y MOTIVACIÓN	9
1.2	ÁREAS DE APLICACIÓN	10
1.2.1	REDES DE CONMUTACIÓN DE CIRCUITOS A NIVEL DE BACKBONE	11
1.2.2	INTERCONEXIÓN DE REDES	11
1.2.3	REDES DE FIBRA ÓPTICA PARA TELEFONÍA	12
1.2.4	COMPUTADORAS TOLERANTES A FALLAS Y ARQUITECTURA DE SWITCH	12
1.3	OBJETIVOS DE LA TESIS	12
1.4	METODOLOGÍA PROPUESTA	13
1.5	CONTENIDO DE LA TESIS	14
1.6	RESUMEN	15
<u>2</u>	<u>DISEÑO DE REDES</u>	16
2.1	DESCRIPCIÓN DEL PROBLEMA	16
2.2	FORMULACIÓN GENERAL DEL PROBLEMA	17
2.2.1	DISEÑOS BASADOS EN LA UTILIZACIÓN UN ALGORITMO DETERMINÍSTICO	18
2.2.2	DISEÑOS ÓPTIMOS UTILIZANDO ALGORITMOS ESTOCÁSTICOS	19
2.3	PROBLEMA DE DISEÑO DE REDES	20
2.3.1	REPRESENTACIÓN MATRICIAL DE GRAFOS	21
2.3.2	MODELADO CON VARIABLES BINARIAS	24
2.3.3	ÁRBOL	25
2.3.4	CONJUNTO DE CORTE Y NODOS DE ARTICULACIÓN	26
2.4	RESUMEN	27
<u>3</u>	<u>EL PROBLEMA DE OPTIMIZACIÓN EN DISEÑO DE REDES CONFIABLES</u>	29
3.1	INTRODUCCIÓN	29
3.2	CONFIABILIDAD EN REDES	30
3.2.1	DEFINICIONES BÁSICAS	30
3.2.2	ANÁLISIS DE LA CONFIABILIDAD DE UNA RED	32
3.2.3	ESTIMACIÓN DE LA CONFIABILIDAD	35

3.2.4	EVALUACIÓN DE LA CONFIABILIDAD POR SIMULACIÓN	37
3.3	DISEÑO ÓPTIMO DE REDES CONFIABLES.....	43
3.4	RESUMEN	46
4	<u>ALGORITMOS EVOLUTIVOS.....</u>	47
4.1	INTRODUCCIÓN.....	47
4.2	REPRESENTACIÓN GENÉTICA DE LAS SOLUCIONES	52
4.3	CREACIÓN DE LA POBLACIÓN INICIAL	53
4.4	FUNCIÓN DE EVALUACIÓN O FUNCIÓN DE ADAPTABILIDAD (<i>FITNESS</i>)	54
4.5	SOLUCIONES FACTIBLES Y NO FACTIBLES: MANEJO DE LAS RESTRICCIONES DEL PROBLEMA.....	55
4.6	MECANISMOS DE SELECCIÓN DE PADRES	58
4.6.1	MÉTODOS PROPORCIONALES	59
4.6.2	MÉTODO DE SELECCIÓN POR JERARQUÍAS.....	61
4.7	SELECCIÓN LOCAL	63
4.8	OPERADORES GENÉTICOS	64
4.8.1	OPERADOR DE CRUZAMIENTO.....	64
4.8.2	MUTACIÓN	71
4.8.3	CRUZAMIENTO VS. MUTACIÓN	72
4.9	ELITISMO	74
4.10	MODELOS DE POBLACIÓN	74
4.10.1	MODELO GLOBAL (PANMÍTICO).....	75
4.10.2	MODELO REGIONAL	75
4.10.3	MODELO DE DIFUSIÓN LOCAL	77
4.10.4	REINSERCIÓN.....	78
4.11	PARÁMETROS DE LOS ALGORITMOS EVOLUTIVOS.....	80
4.12	CONCLUSIONES.....	82
5	<u>DISEÑO DE UN ALGORITMO EVOLUTIVO PARA LA OPTIMIZACIÓN DE REDES CONFIABLES</u>	83
5.1	INTRODUCCIÓN.....	83
5.2	FORMULACIÓN DEL PROBLEMA	83
5.3	ENFOQUE EVOLUTIVO	85
5.4	REPRESENTACIÓN DE LAS SOLUCIONES.....	86

5.5	GENERACIÓN DE LA POBLACIÓN INICIAL	86
5.6	FUNCIÓN OBJETIVO Y FUNCIÓN DE APTITUD	88
5.7	CONOCIMIENTO ESPECÍFICO DEL PROBLEMA EN EL CÁLCULO DE LA CONFIABILIDAD	
	88	
5.8	OPERADORES GENÉTICOS	89
5.8.1	MECANISMO DE SELECCIÓN	89
5.8.2	OPERADOR DE CRUZAMIENTO	92
5.8.3	OPERADOR DE MUTACIÓN.....	93
5.8.4	BÚSQUEDA LOCAL	93
5.8.5	ALMACENAMIENTO DE SOLUCIONES EVALUADAS	95
5.8.6	POST OPTIMIZACIÓN.....	96
5.8.7	EXPERIMENTOS Y RESULTADOS	97
5.9	CONCLUSIONES.....	100
6	<u>EJEMPLOS DE APLICACIÓN.....</u>	<u>101</u>
6.1	INTRODUCCIÓN.....	101
6.2	DESCRIPCIÓN DE LOS EXPERIMENTOS	101
6.2.1	CASO 1.....	101
6.2.2	CASO 2.....	103
6.2.3	CASO 3.....	105
6.2.4	CASO 4.....	107
6.3	CONCLUSIONES.....	109
6.4	APÉNDICE.....	110
6.4.1	REDES TOTALMENTE CONECTADAS	110
	Caso 1: problemas 1 y 2. Matriz de 5 nodos y 10 enlaces.....	110
	Caso 2: problemas 3, 4 y 5. Matriz de 7 nodos y 21 enlaces.....	111
	Caso 3: problemas 6, 7 y 8. Matriz de 8 nodos y 28 enlaces.....	112
	Caso 4: problemas 9, 10 y 11: Matriz de 9 nodos y 36 enlaces.....	114
	Caso 5: problema 12, 13 y 14. Matriz de 10 nodos y 45 enlaces	116
	Caso 6: problema 15. Matriz 15 nodos y 105 enlaces	118
	Caso 7: problema 16. Matriz de 20 nodos y 190 enlaces	119
	Caso 8: problema 17. Matriz de 25 nodos y 300 enlaces	120
6.4.2	REDES NO TOTALMENTE CONECTADAS	122
	Caso 9: problema 18. Matriz de 14 nodos y 21 enlaces	122
	Caso 10: problema 19. Matriz de 16 nodos y 24 enlaces	123

Caso 11: problema 20. Matriz de 20 nodos y 30 enlaces	124
6.4.3 TABLA DE COMPARACIÓN DE RESULTADOS.....	126
<u>7 CONCLUSIONES</u>	<u>127</u>
<u>8 REFERENCIAS Y BIBLIOGRAFÍA.....</u>	<u>131</u>

Resumen

En la actualidad existe un conjunto de problemas para los cuales no se conoce ningún método eficiente para resolverlos computacionalmente. Este conjunto pertenece a la clase NP-completa o *NP-hard*. Muchos de dichos problemas tratados mediante la utilización de meta-heurísticas, han sido resueltos con resultados alentadores en muchos de los casos. Entre las meta-heurísticas más utilizadas se encuentran los algoritmos evolutivos, los cuales están inspirados en los mecanismos de la evolución natural.

En la planificación de sistemas de telecomunicaciones el diseño de la configuración necesaria para prestar un servicio de manera óptima respecto de algún criterio de desempeño es fundamental. Por ejemplo, si el criterio de desempeño es el costo, un problema a resolver es encontrar una topología de red que interconecte sus nodos al menor costo y que tenga la propiedad de asegurar la comunicación confiable de datos.

En esta tesis se analizan e implementan estrategias para el Diseño Óptimo de Redes de Datos para la obtención de topologías de mínimo costo y de confiabilidad admisible. En otras palabras se busca lograr, en forma eficiente, una asignación óptima de recursos económicos que asegure una confiabilidad especificada del sistema bajo diseño. La focalización está puesta en el diseño de redes de gran escala, tales como un *backbone* de telecomunicaciones, donde la métrica relevante es la confiabilidad total, es decir, la probabilidad de que cada par de nodos pueda comunicarse.

En este trabajo se implementa un Algoritmo Evolutivo híbrido, para resolver un problema de diseño de redes confiables, con estructura poblacional por vecindades, utilización de límites superiores para la confiabilidad, el cálculo de ésta a través de un algoritmo de reducción recursiva de la varianza (RVR), con población sembrada y un mecanismo de búsqueda local incorporado a modo de post-optimizador.

Los resultados obtenidos en este trabajo indican, por un lado, que el algoritmo propuesto es superior a los utilizados anteriormente en la literatura, y que las soluciones generadas son aceptables para un diseño óptimo de redes confiables.

Se incluyen ejemplos de aplicación de las estrategias propuestas a un conjunto de redes propuestas en la literatura y se realiza un análisis comparativo de resultados.

CAPITULO 1

1 Introducción

1.1 Antecedentes y motivación

En los últimos años las redes de comunicaciones han adquirido una enorme importancia en una gran variedad de aplicaciones relacionadas con la mayoría de las actividades que se desarrollan en el mundo moderno. Esto ha producido un rápido desarrollo de la infraestructura de redes, del software y de los servicios de Internet, lo que ha renovado el interés por los problemas de diseño de redes de comunicaciones, generando asimismo un enorme desafío para el diseño, por cuanto son necesarias redes cada vez más complejas, más confiables, más eficientes y más económicas.

Uno de los más importantes objetivos de la planificación de sistemas de telecomunicaciones es el diseño de la configuración necesaria para prestar un servicio de manera óptima respecto de algún criterio de desempeño. Por ejemplo, si el criterio de desempeño es el costo, un problema a resolver es encontrar una topología de red que interconecte sus nodos al menor costo y que tenga la propiedad de asegurar la comunicación confiable de datos.

La optimización topológica de redes es un problema clásico de investigación operativa con aplicabilidad práctica en diversos campos de las telecomunicaciones, diseño de redes de computadoras, etc.

Inicialmente los trabajos relativos a redes altamente confiables se concentraron en sistemas cuyas fallas podrían causar daños masivos o pérdida de vidas humanas, por ejemplo centrales nucleares o sistemas aeronáuticos. Actualmente se consideran también sistemas cuya necesidad de alta confiabilidad tiene un sentido económico. En ese sentido, el uso de las redes de comunicaciones se ha incrementado

significativamente en los últimos años debido al dramático crecimiento de la demanda de las comunicaciones para uso personal o de negocios, lo que está transformado a la sociedad actual. En este cuadro, el diseño de estructuras de redes confiables, como el primer recurso necesario para mantener esta sociedad de información, se está convirtiendo en una actividad muy crítica.

En esta tesis se analizan e implementan estrategias para el Diseño Óptimo de Redes de Datos para la obtención de topologías de mínimo costo y de confiabilidad admisible. En otras palabras se busca lograr, en forma eficiente, una asignación óptima de recursos económicos que asegure una confiabilidad especificada del sistema bajo diseño. La focalización está puesta en el diseño de redes de gran escala, tales como un *backbone* de telecomunicaciones, donde la métrica relevante es la confiabilidad total, es decir, la probabilidad de que cada par de nodos pueda comunicarse (Dengiz, 1997).

En este contexto, se analiza un problema de optimización discreto restringido. Se trata de optimización porque el objetivo es minimizar el costo asociado a la red. Es discreto porque hay número determinado (o indeterminado pero numerable) de estados de la red y, finalmente restringido, porque la o las soluciones admisibles están condicionadas a cumplir ciertas imposiciones de diseño, en este caso, de confiabilidad.

1.2 Áreas de aplicación

Las primeras redes de telecomunicaciones definían su confiabilidad únicamente a través de mediciones de conectividad (un enlace, físicamente, funciona o no funciona). En la actualidad, si bien la tecnología permite contar con enlaces mucho más confiables, el concepto de confiabilidad pasa no sólo por la determinación del funcionamiento físico del mismo, sino además por consideraciones de situaciones derivadas del enrutamiento dinámico, tales como congestión y retardos, que pueden llegar a producir el colapso de la red completa.

Las redes actuales son bastante densas, lo que resulta en una menor probabilidad de desconexión. Sin embargo, el aumento de la densidad es ocasionado por el incremento del tráfico. Esto significa que la capacidad y congestión deberían ser

incluidas como criterios de desempeño en el diseño alterando la métrica de costos o la lista de restricciones impuesta en el problema de optimización que se resuelve.

Esto permite utilizar las técnicas y métodos desarrollado para redes cuya medida de la confiabilidad, por ejemplo, tiene que ver con la probabilidad de conexión o desconexión de los enlaces, a redes más complejas donde puede definirse la operatividad como la capacidad de mantener un desempeño admisible medido en otros términos, tales como retardos, capacidad de los enlaces, probabilidad de conexión de enlaces, etc.

En todos estos casos las entidades relevantes son nodos y conexiones entre nodos, y en general el principal objetivo buscado es lograr una comunicación segura en toda la red al menor costo. La solución a este problema tiene un amplio rango de aplicación en algunas áreas como las que se describen a continuación.

1.2.1 Redes de conmutación de circuitos a nivel de backbone

En este tipo de redes, tal como es el caso de las redes públicas de telefonía, un canal de comunicaciones está dedicado a un par de usuarios durante el tiempo en que tiene lugar la llamada. La capacidad de trabajo de la red puede verse reducida por la falla de sus componentes y podría resultar que cierto número de usuarios finales o nodos no puedan conectarse, esto se denomina "*call blocking*". A diferencia con las redes de conmutación de paquetes donde la falla incrementa los retardos en la transmisión, en este caso la performance de la red está definida en términos de bloqueo en lugar de retardos.

1.2.2 Interconexión de redes

Son casos especiales de redes de conmutación de circuitos para la interconexión de redes donde el modelo de conectividad esté basado en modelos de falla debido a la congestión y al mal funcionamiento de sus componentes. En estos casos es particularmente importante el diseño de redundancia en la interconexión de redes.

1.2.3 Redes de fibra óptica para telefonía

Los enlaces de fibra óptica tienen alta capacidad, esto motiva el rápido reemplazo de redes de cable tradicionales. Pero, es justamente por esa mayor capacidad que las redes de fibras ópticas son menos densas, más bien de pocos enlaces, y si la confiabilidad no es tomada en cuenta en el diseño, una simple falla en un enlace puede dejar fuera de servicio una gran parte de la red. Debido a los pocos enlaces de una red de fibra óptica, nuevamente puede considerarse apropiada la medición de conectividad para cuantificar la confiabilidad.

1.2.4 Computadoras tolerantes a fallas y arquitectura de *switch*

Son sistemas de computadoras que tienen la capacidad de continuar operando en presencia de una falla de uno o más componentes del mismo tipo con el propósito de incrementar la performance del sistema. Para el análisis de la confiabilidad de estos sistemas se los modela como redes, tales como las que estamos considerando.

1.3 Objetivos de la tesis

El objetivo principal de esta tesis es modelar, desarrollar e implementar estrategias aplicadas al Diseño Óptimo de Redes de Información de tal manera de obtener una topología que minimice un criterio de desempeño (por ejemplo el costo) a la vez que satisface restricciones adicionales impuestas (por ejemplo de confiabilidad).

Los objetivos específicos son:

- Estudiar diferentes alternativas para solucionar problemas de tipo combinatorio relacionados con el objetivo principal planteado, haciendo énfasis en las técnicas que aporta la Computación Evolutiva.
- Estudiar las diferentes formas de evaluar o estimar la confiabilidad de una red compleja y la aplicación de formas de reducir el costo computacional asociado.

1.4 Metodología propuesta

En este tipo de problemas deben resolverse dos cuestiones básicas: la determinación de la mejor topología y el cálculo de otros parámetros de desempeño como retardos o confiabilidad de la red. Esta última implica, junto con el problema de determinar la mejor topología, grandes esfuerzos de cálculo por cuanto deben trabajar con espacios de búsqueda muy grandes, dependiendo del tamaño de la red.

Se examinarán las diversas formulaciones existentes para su diseño, así como las metodologías y algoritmos de solución propuestos. Entre ellos, se seleccionan los Algoritmos Evolutivos (AE) ya que muestran ventajas para resolver estos problemas de optimización combinatorios, cuyas características hacen necesaria la implementación de heurísticas capaces de proporcionar buenas soluciones en tiempos razonables.

Se estudiará el Diseño Óptimo de Redes de mínimo costo, sujeto a restricciones de confiabilidad con el supuesto de fallas en los enlaces y se propone como estrategia de resolución un AE con operadores genéticos ad-hoc enriquecidos con conocimiento específico del problema, lo cual permite acotar el espacio de búsqueda del algoritmo.

Un proceso evolutivo sobre una población (soluciones posibles) corresponde a una búsqueda dentro de un espacio de soluciones potenciales. Tal búsqueda sugiere encontrar un equilibrio entre dos objetivos aparentemente en conflicto: explotar las mejores soluciones y explorar el espacio de búsqueda en forma global. Existen estrategias que benefician totalmente uno de esos aspectos pero desestiman el otro. Por ejemplo, puede hacerse una buena exploración del espacio de búsqueda ignorando la explotación de regiones promisorias, o explotar la mejor solución para obtener una posible mejora, sin buscar otras soluciones en otras regiones.

Finalmente se incluyen ejemplos de aplicación de las estrategias propuestas a sistemas reportados en la literatura con el objeto de comparar los resultados y analizar las ventajas de los métodos propuestos en esta tesis.

1.5 Contenido de la Tesis

Los problemas de diseño topológico de redes de información caen dentro de la categoría de problemas de optimización combinatoria. En muchos casos, cuando el tamaño de la instancia del problema crece, los tiempos de resolución del mismo a través de técnicas exhaustivas pueden resultar inadmisibles. Por esta razón las heurísticas en general y la computación evolutiva en particular, son una alternativa válida para atacar este tipo de problemas.

Este trabajo se centra en la resolución de problemas de diseño óptimo de redes confiables utilizando técnicas evolutivas. A continuación se muestra un esquema del desarrollo del mismo.

En el capítulo 2 se realiza una revisión de las metodologías existentes aplicadas para resolver problemas de diseño de redes.

La complejidad matemática asociada a la resolución de la obtención de la topología óptima se discute en el Capítulo 3. El abordaje del diseño de una red de comunicaciones cuando la función objetivo es el costo de implementación y la restricción recae sobre la confiabilidad del sistema, está plasmado en este capítulo, ubicando al problema dentro de una clase de problemas de optimización combinatoria. Este diseño se utiliza como problema test para analizar el comportamiento de un algoritmo estocástico.

En el capítulo 4 se realiza una revisión de las técnicas proporcionadas por la computación evolutiva para resolver problemas de optimización. El mismo contiene los conceptos básicos relacionados con la aplicación de Algoritmos Evolutivos (AE), basados en AG, para resolver problemas de optimización.

En el capítulo 5 se describe el diseño de un algoritmo evolutivo con conocimiento específico del problema para encontrar una topología de mínimo costo sujeto a restricciones de confiabilidad entre todas las terminales de la red.

El Capítulo 6 contiene ejemplos de diseño de redes resueltos a través de las metodologías propuestas en esta tesis. Las conclusiones y futuras líneas de

investigación se presentan en el Capítulo 7.

1.6 Resumen

En este primer capítulo se han descrito las motivaciones por las cuales se ha iniciado este trabajo. Se enuncian las áreas donde se aplican los algoritmos propuestos en esta tesis. Finalmente se plantea una metodología de trabajo para el diseño del algoritmo proponiendo métodos de computación evolutiva. En el último punto se describen los contenidos de cada capítulo que conforman esta tesis.

CAPÍTULO 2

2 Diseño de redes

2.1 Descripción del Problema

Las redes de comunicaciones han experimentado un enorme crecimiento en la última década debido, entre otros factores, al uso progresivo de la red de redes o Internet. Los requerimientos de calidad de servicio y confiabilidad de las redes modernas, acompañadas por grandes inversiones en ellas, han tornado críticos los problemas de diseño de las mismas siendo necesario el diseño óptimo de redes que reúnan características determinadas.

Esto conlleva la resolución de problemas cada vez más complejos y más grandes. Complejos desde un punto de vista de complejidad computacional y grandes respecto del tamaño de las entradas de instancias de dichos problemas. Desde el conocido problema del *Minimum Spanning Tree* resuelto a través de un algoritmo determinístico, varios tipos de problemas confluyen cuando se trata de optimizar una red de comunicaciones.

Además, son diferentes los criterios de optimización que pueden ser considerados: el costo de instalación, costo de operación, confiabilidad de la red, o incluso más de un objetivo a optimizar simultáneamente.

La mayoría de los problemas involucrados en diseño de redes están catalogados como problemas de tipo *NP-Hard* o *NP-Completo*, es decir que no existen a la fecha algoritmos que puedan encontrar solución en un tiempo acotado polinomialmente respecto del tamaño de la entrada.

No obstante, la necesidad de encontrar soluciones basada en las cuestiones

mencionadas anteriormente, han conducido las investigaciones a la utilización de heurísticas y metaheurísticas que permitan, cuando no obtener el óptimo global, encontrar buenas soluciones. Así, algoritmos como *simulated annealing*, *tabú search*, redes neuronales y algoritmos evolutivos entre otros han experimentado un notable adelanto e interés por parte de los investigadores como técnicas alternativas a las técnicas clásicas que permiten la obtención de buenas soluciones a los problemas planteados.

Esta tesis pretende estudiar y resolver algunos problemas de diseño de redes utilizando algoritmos evolutivos cuando se desea el diseño óptimo de una red de comunicaciones desde distintos criterios de optimización: diseño de redes de mínimo costo, diseño de redes de máxima confiabilidad, u otros. En particular en esta tesis se abordará el diseño de redes de mínimo costo con restricciones de confiabilidad. No obstante las metodologías propuestas permiten el abordaje de los otros problemas mencionados. El diseño de redes con otros criterios de optimización, por ejemplo la confiabilidad con restricciones económicas, simplemente puede ser resuelto con las herramientas presentadas en este trabajo mediante la adaptación de la función objetivo. Con algún esfuerzo adicional, pueden atacarse problemas con criterios de optimización multiobjetivo; por ejemplo, optimizar simultáneamente costo y confiabilidad.

2.2 Formulación general del problema

En el diseño de redes de comunicaciones, una etapa importante consiste en encontrar la mejor forma de interconectar los componentes de una red, optimizando alguna variable respetando ciertas restricciones impuestas. La variable a optimizar suele ser el costo mientras que las restricciones más comunes están aplicadas sobre flujos de datos, retardos de transmisión o confiabilidad, entre otras.

El problema del diseño de redes óptimas puede representarse a través de un modelo matemático o conjunto de ecuaciones que relaciona las variables involucradas y consiste en determinar una topología definida por la cantidad y posición de los enlaces que transportarán los datos a través de la red y sus características.

A fin de obtener una distribución óptima de enlaces, con respecto a algún

criterio específico, se define el Diseño Optimo de la Red de Datos como un problema de optimización, cuya formulación general es la siguiente (Nemhauser, 1999):

$$\begin{aligned} & \text{Min / Max } f(\mathbf{q}) \\ & \text{st.} \\ & g(\mathbf{q}) \leq g^* \end{aligned} \quad (1.1)$$

Siendo g^* el valor máximo admitido y \mathbf{q} es un vector de variables binarias tal que:

$$\begin{cases} q_i = 1, \text{ si el enlace } i \text{ existe} \\ q_i = 0 \text{ en caso contrario.} \end{cases} \quad (1.2)$$

Los criterios de desempeño $f(\mathbf{q})$, pueden ser diversos, por ejemplo: de tipo económico, relacionados con la capacidad del enlace, los retardos, la confiabilidad del sistema, etc. En cuanto al conjunto de las restricciones $g(\mathbf{q})$, el diseño debe garantizar también la satisfacción de las condiciones impuestas, por ejemplo, sobre el *throughput*, confiabilidad o disponibilidad.

Cualquiera sea la formulación dada, el problema resultante es un *problema de optimización combinatoria*.

La resolución o forma de ataque de esta clase de problemas, sin embargo, depende fuertemente del caso particular que se trate. Un intento de clasificación de los métodos de resolución propuestos puede ser la siguiente:

2.2.1 Diseños basados en la utilización un algoritmo determinístico

Para algunos problemas de diseño, existen algoritmos determinísticos bien probados y bien conocidos que arrojan resultados satisfactorios. El problema del *Minimum Spanning Tree*, por ejemplo, puede resolverse a través de un algoritmo *greedy* con buena performance. El algoritmo de Dijkstra, permite encontrar el camino de mínimo costo entre dos nodos de una red, para construir las tablas de ruteo en cada nodo. Estos son algunos ejemplos de algoritmos determinísticos que, además, garantizan la obtención del óptimo global en tiempo polinomial.

Para otro tipo de formulaciones no se han reportado métodos de resolución que garanticen la obtención del óptimo global para cualquier tamaño de una instancia. En estos casos pueden utilizarse varias alternativas (Papadimitriou y Steiglitz, 1982), algunas de las cuales se listan a continuación:

- 1- Algoritmos Exponenciales: entre los que se incluyen las técnicas de enumeración implícitas, como el algoritmo de *Branch and Bound*, *Branch and Cut* y *Programación Dinámica*. Estas técnicas pueden ser aplicadas con éxito a instancias de tamaño razonable. Algunos de los algoritmos exponenciales aplicados hasta la actualidad que pueden citarse son: 1) métodos basados en la enumeración implícita con criterio de terminación y 2) *Branch and Bound* aplicado a la resolución de problemas de optimización de tamaño pequeño con funciones objetivo lineales y restricciones convexificadas (Jan, 1993).
- 2- Algoritmos de búsqueda Local: Estos son métodos de ataque ampliamente utilizado. Son análogos en optimización discreta a la técnica de *hill climbing*. Se basa en la búsqueda de óptimos locales en la *vecindad* de un punto. Un método de esta clase es el Tabu Search (Glover, 2002).

2.2.2 Diseños óptimos utilizando algoritmos estocásticos

Desde el punto de vista de la optimización son métodos probabilísticos de orden cero (sólo requieren los valores de la función a ser optimizada). La búsqueda del óptimo está guiada por conceptos probabilísticos y pueden resolver cualquier tipo de problemas de optimización.

Las desventajas residen en que, dada su naturaleza estocástica, no ofrecen ninguna garantía de convergencia para una corrida dada, y además el costo computacional puede ser elevado; aunque esta última característica puede ser mejorada sustancialmente, incorporando conocimiento específico del problema a resolver (Schoenauer, 1995).

En contraste con los algoritmos de búsqueda local éstas son técnicas de búsqueda global que proveen una forma de salir de un óptimo local aceptando

selectivamente soluciones de mayor costo que la corriente.

El problema que se plantea en esta tesis es uno de un gran grupo de problemas para los cuales no han podido desarrollarse algoritmos que sean lo suficientemente rápidos para encontrar el óptimo global. Sin embargo es posible diseñar algoritmos eficientes que arrojen una solución cuasi-óptima. Estos algoritmos no garantizan la obtención del óptimo global, pero pueden encontrar buenas soluciones, teniendo en cuenta los tiempos de cálculo.

2.3 Problema de diseño de redes

Una red puede ser modelada a través de un grafo simple (sin lazos ni redundancias) (Brecht, 1995) $G = (V, E)$ donde $V = \{v_1, v_2, \dots, v_n\}$ es el conjunto de n nodos o vértices que representan los centros de comunicaciones de la red y $E = \{e_1, e_2, \dots, e_m\}$ es un conjunto de m vínculos o enlaces que permiten la conexión entre los nodos. Cada enlace e_k se identifica con un par no ordenado (v_i, v_j) . Los vínculos pueden ser orientados o no-orientados representando enlaces de comunicaciones que son unidireccionales o bidireccionales respectivamente, pero no se permiten lazos cerrados ni enlaces redundantes (un par de nodos cualesquiera están directamente vinculados, como máximo, por un único enlace). Un enlace es incidente sobre dos nodos, si los dos nodos están unidos por el enlace.

En la figura 2.1 se muestra un ejemplo de un grafo no orientado que representa una red

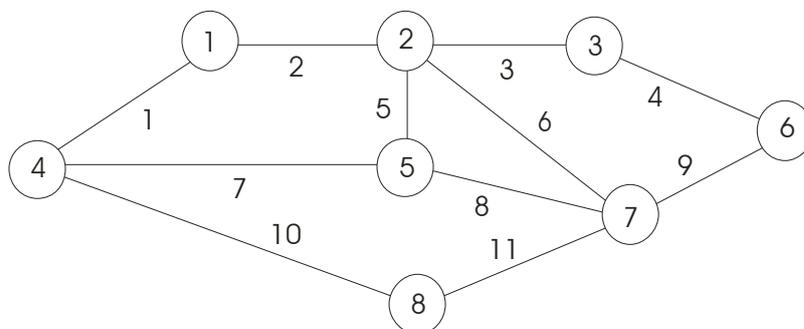


Figura 2.1 - Grafo simple representando el modelo de una red

Donde

$$G = (V, E)$$

$$V = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

$$E = \{(1, 2), (1, 4), (2, 3), (2, 5), (2, 7), (3, 6), (4, 5), (4, 8), (5, 7), (6, 7), (7, 8)\}$$

La cardinalidad de un conjunto H , denotada como $|H|$ es la cantidad de elementos que dicho conjunto posee. Así el número de nodos en un grafo G se representa como $n = |V|$ mientras que el número de enlaces es $e = |E|$. Para el grafo de la figura 2.1 $n = |V| = 8$ y $e = |E| = 11$.

Un problema de diseño de redes consiste en la elección de enlaces de comunicaciones, de diferentes características o tecnologías, entre un conjunto de nodos dado, de forma tal que la red resultante alcance un conjunto de valores (tales como costo o confiabilidad) y cumpla con otro conjunto de requerimientos que operan como restricciones al problema.

Existen dos aspectos a tener en cuenta en el diseño óptimo de redes confiables: la obtención de la topología de mínimo costo y el cálculo de la confiabilidad. El segundo aspecto limita el espacio de búsqueda factible entre el conjunto de todas las topologías posibles. Además la confiabilidad depende de la topología de la red.

2.3.1 Representación Matricial de Grafos

Los grafos constituyen una herramienta útil para el estudio y modelado topológico de redes. Ahora bien, para analizarlos y procesarlos computacionalmente es factible representarlos en forma matricial. A continuación se presentan algunas variantes de las mismas, las cuales se utilizarán en este trabajo (Deo, 1974).

2.3.1.1 Matriz incidente

Sea un grafo G con n nodos y m enlaces, sin lazos. Se define la matriz \mathbf{A} de dimensión n por m tal que $A = [a_{ij}]$, donde la n filas corresponden a los n nodos y las m

columnas corresponden a los m enlaces. Por ende, $a_{ij} = 1$ si el j -ésimo enlace e_j es incidente al i -ésimo nodo v_i ; o será igual a cero en caso contrario.

Esta matriz \mathbf{A} se denomina matriz de incidencia nodo-enlace o simplemente matriz de incidencia. A veces, si la matriz \mathbf{A} representa el grafo G se la simboliza como $\mathbf{A}(G)$. En la figura 2.2 se muestra la matriz de incidencia del grafo representado en la figura 2.1.

$$A(G) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Figura 2.2 - Matriz de incidencia del grafo de la figura 2.1

En esta representación matricial se puede observar que dado que los enlaces son incidentes en dos nodos, cada columna de \mathbf{A} tiene exactamente dos unos. La cantidad de unos en cada fila representa la cantidad de enlaces que inciden sobre el nodo asociado la cual se define como *grado de un nodo*. Una fila con todos 0, representa un nodo aislado y finalmente, la permutación de dos filas (o columnas) cualesquiera de $\mathbf{A}(G)$, simplemente resulta en un reenumeración de los nodos (o enlaces) del mismo grafo G .

2.3.1.2 Matriz de adyacencia y matriz de costo

Sean n el número de nodos y m la cantidad de enlaces de una red de datos representada por un grafo simple G . Luego, la matriz de conexión $\mathbf{M} = [m_{ij}]$ es una matriz binaria de dimensión n por n y donde:

$$m_{ij} = \begin{cases} 1 & \text{si el enlace que une los nodos } i \text{ y } j \text{ está disponible} \\ 0 & \text{en caso contrario} \end{cases}$$

La figura 2.3 muestra la matriz de adyacencia del grafo representado en la figura 2.1.

$$C(G) = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Figura 2.3 - Matriz de adyacencia del grafo de la figura. 2.1

La matriz de costo $C = [c_{ij}]$ tiene la misma dimensión que la matriz de adyacencia donde cada elemento c_{ij} representa el costo del enlace entre los nodos i y j . Así, mientras que la matriz de adyacencia representa un grafo simple, la matriz de costo representa un grafo simple ponderado

Los ceros se mantienen invariante, asumiendo que un costo nulo representa un enlace que no está disponible (tiene mayor coherencia matemática si un enlace inexistente se representara por ∞ , pero habitualmente se utiliza la representación con 0 dado que permite al algoritmo interpretar redes no totalmente conectadas).

La figura 2.4 muestra la matriz de costo asociada al grafo representado en la figura 2.1, cuyos valores ha sido elegido arbitrariamente.

$$C(G) = \begin{bmatrix} 0 & 18 & 0 & 21 & 0 & 0 & 0 & 0 \\ 18 & 0 & 32 & 0 & 10 & 0 & 12 & 0 \\ 0 & 32 & 0 & 0 & 0 & 17 & 0 & 0 \\ 21 & 0 & 0 & 0 & 11 & 0 & 0 & 18 \\ 0 & 10 & 0 & 11 & 0 & 0 & 18 & 0 \\ 0 & 0 & 17 & 0 & 0 & 0 & 15 & 0 \\ 0 & 12 & 0 & 0 & 18 & 15 & 0 & 13 \\ 0 & 0 & 0 & 18 & 0 & 0 & 13 & 0 \end{bmatrix}$$

Figura 2.4 - Matriz de costo del grafo de la figura 2.1

Las matrices $\mathbf{M}(G)$ y $\mathbf{C}(G)$ son siempre cuadradas dado que tanto las filas como las columnas representan el mismo parámetro (los nodos del grafo) y son simétricas porque el enlace que une el i -ésimo con el j -ésimo nodo es el mismo que une el j -ésimo con el i -ésimo nodo. Al no estar permitidos los lazos, la diagonal es siempre nula.

2.3.2 Modelado con variables binarias

El grafo que modela la red puede ser representado en un arreglo unidimensional teniendo en cuenta todos los enlaces posibles entre los nodos. Si se considera la red de 5 nodos representada por la figura 2.5, la cantidad de enlaces posibles está dada por la cardinalidad del conjunto E la cual puede calcularse como

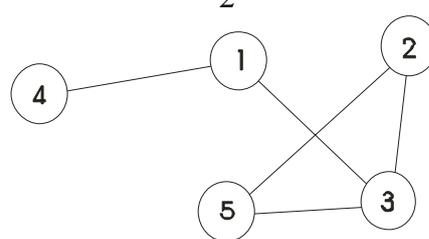
$$|E| = \frac{|V| \cdot (|V| - 1)}{2} = 10$$


Figura 2.5 - Red de ejemplo

Todos los enlaces posibles (i,j) entre cada par de nodos i y j , numerando los nodos de 1 a 5 conforman el conjunto $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\} = \{(1,2) (1,3) (1,4) (1,5) (2,3) (2,4) (2,5) (3,4) (3,5) (4,5)\}$. El arreglo que representa la topología es \mathbf{x} tal que $x_i=1$ si e_i existe y $x_i=0$ en caso contrario. En definitiva, la representación de \mathbf{x} con el esquema de conexión de la figura 2.5 será:

$$\mathbf{x} = [0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0]$$

El modelado con variables binarias está relacionado con la forma de representar soluciones que se utilizarán al aplicar algoritmos evolutivos para resolver el problema.

2.3.3 Árbol

Un árbol es un grafo simple conectado que no contiene ciclos. Cualquier grafo contiene por lo menos un árbol, y dado que se consideran únicamente grafos finitos, también hay una cantidad finita de árboles en un grafo.

Si en un grafo G existe un y sólo un camino entre cada par de vértices, entonces G es un árbol y un grafo es un árbol si está mínimamente conectado.

2.3.3.1 Spanning Tree

Se dice que un árbol T es un *Spanning Tree* de un grafo G conectado, si T es un subgrafo de G y contiene todos los nodos de G . La figura 2.6 muestra uno de los posibles *Spanning Tree* del grafo G .

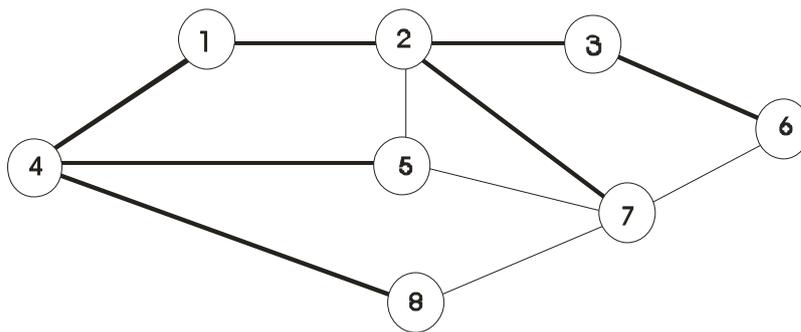


Figura 2.6 - Un grafo con un spanning tree remarcado

Dado que los *spanning Trees* son los árboles más grandes (con mayor cantidad de enlaces) que pueden concebirse dentro de un grafo determinado, a veces se los denomina subgrafo de árbol máximo.

Encontrar un *Spanning Tree* de un grafo conectado es relativamente simple. Si G no tiene ciclos, G es su propio *Spanning Tree*, es decir: un grafo G conectado siempre tendrá, al menos, un *Spanning Tree*.

La importancia de este tipo de árbol, en el problema que se intenta resolver en esta tesis es que una red se considera operativa si y sólo si todos sus nodos son alcanzables desde cualquier otro, a través de enlaces disponibles. Un *Spanning Tree*

concede la conectividad mínima necesaria ($n-1$ enlaces) para que esta condición se cumpla. Reducir la cantidad de enlaces manteniendo la red operativa es una de las formas posibles de alcanzar el objetivo de minimizar costos, sin dejar necesariamente de cumplir con la confiabilidad pretendida.

2.3.4 Conjunto de corte y nodos de articulación

En un grafo G conectado, un *conjunto de corte* (*cut-set*) es un conjunto de enlaces (que vinculados con sus nodos limitantes constituyen un subgrafo) cuya remoción de G lo desconecta.

Un *conjunto de corte* siempre corta un grafo en dos. Dado que en un árbol la extracción de cualquier enlace parte lo desconecta, cada enlace constituye un *conjunto de corte* por sí mismo y no existe ningún *conjunto de corte* con más de un elemento.

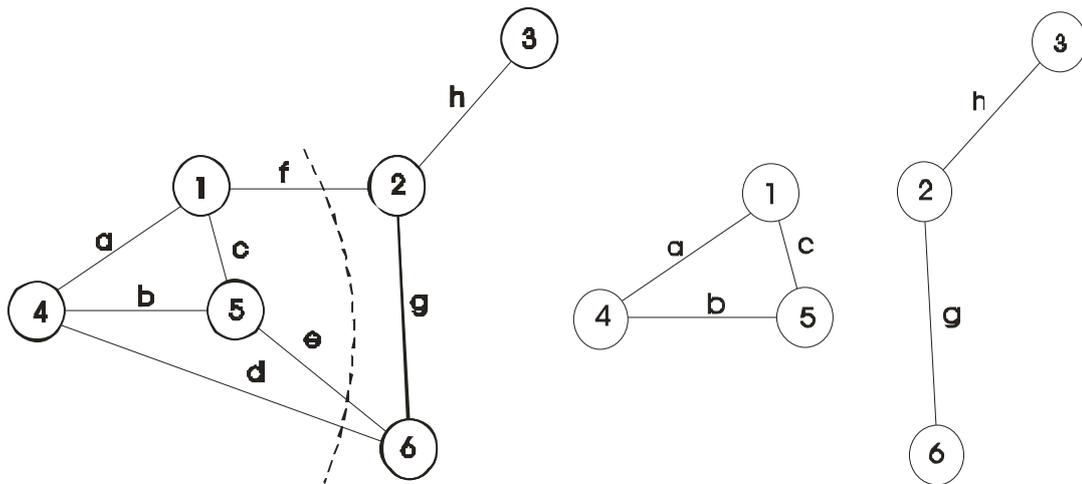


Figura 2.7 - Remover el conjunto de corte $\{d, e, f\}$ divide en dos al grafo

Si se busca todos los *conjuntos de corte* de la red, el o los conjuntos que tienen el menor número de elementos representarán el punto más vulnerable. En el ejemplo de la figura 2.7, la caída del enlace h aísla completamente al nodo 3, mientras que para que otro nodo quede aislado, deberán cortarse o caerse más de un enlace simultáneamente.

A partir de las definiciones anteriores se puede afirmar que, dado el grafo conectado G y un conjunto de corte S arbitrariamente elegido, éste contendrá al menos

un enlace de cada posible *spanning tree* de G . La inversa, es decir, cualquier conjunto mínimo de enlaces que contenga al menos un enlace de cada *spanning tree* es un *conjunto de corte*, también es cierta.

Se define también el mínimo conjunto de corte (MCC) como el *conjunto de corte* con menor cantidad de enlaces que desconecta el grafo G (en otras palabras el conjunto mínimo de enlace que, removidos de G , lo desconecta), por ejemplo el MCC de un árbol es uno.

Los *nodos de articulación* son nodos cuya “caída” desconecta el grafo, es decir, lo separa en dos o más componentes. El grafo de la figura 2.8 tiene dos *nodos de articulación*, los nodos 2 y 3.

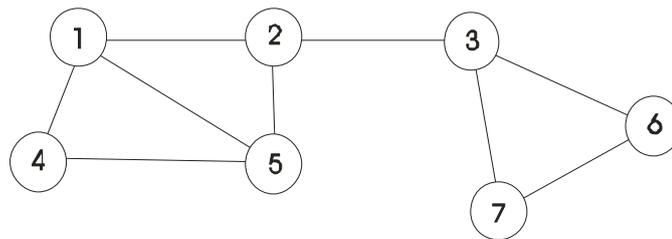


Figura 2.8 - Red con dos nodos de articulación

Por otra parte, se dice que un grafo es *biconectado* si y sólo si no contiene *nodos de articulación*.

Análogamente se define la conectividad nodo como el mínimo número de nodos que removidos (conjuntamente con sus enlaces incidentes) dejan el grafo desconectado.

El grado de un nodo v , es el número de enlaces que inciden sobre v , si todos los nodos del grafo tienen el mismo grado, se dice que es un *grafo regular*.

2.4 Resumen

En este capítulo se han introducido algunos conceptos básicos de diseño topológico de redes y se justifica la importancia de modelar las redes mediante grafos y sus representaciones matriciales, así como la elección de métodos estocásticos para la

búsqueda de buenas soluciones en tiempos razonables.

También se presentan algunos conceptos de teoría de grafos, que serán de interés en el desarrollo de los algoritmos. Entre ellos, tiene especial interés los grafos que mantienen mínimamente conectados a todos los nodos del grafo original y que se definió como *spanning tree*.

CAPÍTULO 3

3 EL PROBLEMA DE OPTIMIZACIÓN EN DISEÑO DE REDES CONFIABLES

3.1 Introducción

Los problemas de optimización pueden dividirse en forma natural, en dos categorías: aquellos con variables continuas, y aquellos con variables discretas o problemas combinatorios. En los primeros, la atención se centra en un conjunto de números reales o de funciones reales; en los segundos, la atención está puesta en un objeto de un conjunto finito o infinito contable, tal como un entero, una permutación, una combinación o un grafo.

Estas dos clases de problemas tienen características diferentes, y los métodos de resolución aplicados a ellos resultan muy disímiles.

Los problemas de diseño de redes confiables de datos pertenecen a la segunda categoría. El problema de diseño de red, estudiado en este trabajo, consiste en encontrar una red de comunicaciones que minimice el costo total de la misma, bajo ciertas condiciones de confiabilidad dadas.

En este contexto se analiza en profundidad un problema de optimización discreta restringido. Es optimización porque el objetivo es minimizar el costo asociado a una red de telecomunicaciones, es discreta porque las variables sólo toman valores de un conjunto finito de números, lo que permite su simulación exacta; finalmente, es restringido debido a que el objetivo está condicionado a cumplir con un conjunto de exigencias. En este caso la restricción es la confiabilidad.

Tanto el problema de diseño de redes de comunicaciones confiables como el

cálculo de la confiabilidad de una red dada, son problemas NP-hard (Garey, 1979). Esto significa que el esfuerzo computacional crece exponencialmente con el número de nodos y enlaces en la red y dado que, durante el proceso de diseño debe calcularse la confiabilidad de un gran número de topologías candidatas, el tiempo de cálculo resulta prohibitivo aún con redes de tamaño relativamente pequeño (Srivaree-Ratana, 2000).

Una instancia de un problema de optimización es un par (F,c) , donde F es el conjunto de puntos factibles, y c es la función de costo (función objetivo), una aplicación tal que:

$$c: F \rightarrow \mathbb{R}^1 \quad (3.1)$$

El problema es encontrar un $f \in F$ para el cual:

$$c(f) \leq c(y) , \text{ para todo } y \in F \quad (3.2)$$

El punto f es llamado un *óptimo global*.

Un *problema de optimización* es un conjunto I de instancias asociadas al mismo.

Informalmente, en una instancia se tienen los “datos de entrada” y suficiente información como para obtener una solución; un problema de optimización es una colección de instancias, todas generadas en forma similar (Garey, 1999).

3.2 Confiabilidad en redes

3.2.1 Definiciones básicas

Debido a la inherente dificultad de calcular la confiabilidad en una red independientemente del tiempo, se emplea un modelo de probabilidad discreto en el análisis de la misma. Esto es, los componentes de la red (nodos y enlaces) pueden estar en uno de dos estados: operativo o en falla, en forma independiente de los estados de los otros componentes.

Entonces, el análisis de confiabilidad del problema es: dada la probabilidad de que cada componente esté operativo, calcular una medida de la confiabilidad de la red.

Esta definición de confiabilidad no incluye consideraciones de reparación; en lugar de eso se dice que: dado un tiempo suficientemente extenso t , se define la confiabilidad como la probabilidad que un componente no falle en el tiempo t .

El punto de arranque es una red $G = (V, E)$, donde V es un conjunto de nodos o vértices y E es un conjunto de enlaces no dirigidos o aristas. Cuando se estudia el modelo de conectividad de cada $e \in E$ se define p_e como $\Pr[e \text{ esté operativo}]$, que es la confiabilidad de e . Este modelo puede servir también para estudios de flujos (camino más corto) asociando una capacidad C_e con cada $e \in E$ y se interpreta p_e como la probabilidad de que e esté operativo y tiene una capacidad C_e . También definimos $1-p_e$ como la probabilidad de que e falle o tenga una capacidad cero.

Es conveniente para estudios de confiabilidad de redes definir un contexto para un análisis más general. Un “sistema binario estocástico” representa las fallas aleatorias como una función Ψ donde sus componentes T pueden tener dos estados: operativo o fallado. Si S es el conjunto de componentes operativos entonces puede definirse Ψ como sigue:

$$\Psi(S) = \begin{cases} 1 & \text{si cuando } S \text{ está operativo y suceden } T - S, \text{ el sistema está operativo} \\ 0 & \text{si cuando } S \text{ está operativo y suceden } T - S, \text{ el sistema falla} \end{cases}$$

El problema computacionalmente se puede definir

$$R(S, p) = \Pr[\Psi(S) = 1] \quad (3.3)$$

Dado el modelo de red descrito antes, el problema general del análisis de la confiabilidad de una red es, ¿Cual es la probabilidad de que una red esté operativa, dada la probabilidad de que cada uno de los enlaces estén operativos?

3.2.2 Análisis de la confiabilidad de una red

Un sistema está formado por componentes los cuales pueden ser susceptibles de falla. La falla de un componente cualquiera se da a un tiempo desconocido pero estimable a través de una variable aleatoria T . La figura 3.1 muestra una curva típica de confiabilidad de un componente en función del tiempo. La forma de la curva tiene relación directa con la distribución de probabilidades de falla del componente.

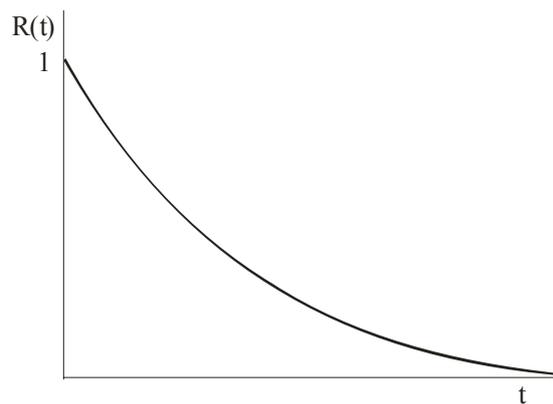


Figura 3.1 – Confiabilidad de un componente

De esta forma la confiabilidad del mismo está dada por la probabilidad que no falle en un tiempo no inferior a un valor especificado. Esto es

$$R = \Pr[\text{no falla en } T \leq t] \quad (3.4)$$

Tomando un tiempo t admisible, la confiabilidad del componente será un valor escalar representado por un punto en la curva de la figura 3.1.

Interesa ahora definir una medida (o índice) de confiabilidad para una red, que esté basada en su topología y en la confiabilidad de sus componentes. La forma de obtener dichos índices varía dependiendo de los datos con que se cuente. Un dato imprescindible es el referente a la topología de la red, pero según se disponga o no con datos sobre las confiabilidades elementales de los componentes, se pueden adoptar dos enfoques (Cancela, 1993):

- Solidez o resistencia a fallas: Este enfoque se basa en la posibilidad de comparar dos topologías y determinar cual de ellas es más confiable, para lo cual es necesario considerar algunos parámetros en ellas. Intuitivamente, una red que tiene una topología de árbol (sólo un *spanning tree*) resulta menos confiable que otra que posee una topología en anillo. Reichelt (2003), propone el conteo de todos los posibles *spanning trees* en una red para dar un índice de confiabilidad relativa sobre la misma. Otra alternativa es contar la cantidad de puntos de articulación que la red tiene (la topología con menor número de puntos de articulación será más confiable).
- Confiabilidad: Se calcula la probabilidad de funcionamiento de la red sobre la base de las confiabilidades elementales de cada componente, dadas mediante probabilidades de funcionamiento de los mismos.

El enfoque a tomar en este trabajo es el segundo, considerándose un modelo probabilístico basado en un grafo estocástico donde se asignan probabilidades de funcionamiento a nodos y enlaces. Sobre grafos de este tipo, existen básicamente tres medidas de interés:

- a) La confiabilidad entre dos nodos, denotada $R_2(G)$, es la probabilidad de que para un par específico de nodos s y t , hay al menos un conjunto de enlaces operacionales: una ruta que conectan los nodos s y t .
- b) La confiabilidad de k -nodos, que se denota $R_k(G)$, es la probabilidad de que para un conjunto específico K de k nodos, hay un conjunto de enlaces operacionales conectando cada par de nodos en K .
- c) La confiabilidad de todos los nodos, $R_A(G)$ es la probabilidad que haya un conjunto de vínculos operacionales, o una ruta, entre todo par de nodos de la red.

Dado un conjunto de nodos K y un nodo $s \in K$ ($k = |K|$) y dada una red G con enlaces de confiabilidad p_e , la confiabilidad de k -nodo está definida por

$$R(G,s,K,p_e) = \Pr[\text{Existe un camino operativo desde } s \text{ a cada nodo en } k]$$

De aquí se deducen los dos casos especiales mencionados. Para $k=2$, que es el caso de dos terminales, $R_2(G,s,t,p_e)$ y para $k=V$ para todos los nodos, $R_A(G,s,p_e)$. s es el nodo fuente y $\{K \setminus s\}$ los nodos terminales (Ball, 1992).

La confiabilidad del sistema dependerá de la confiabilidad de sus componentes y de la interconexión entre ellos. Por lo tanto, la confiabilidad de la red, definida como la probabilidad de que ésta sea operativa, dependerá de las confiabilidades de los enlaces y de la topología de la misma.

En este trabajo se estudia la medida de confiabilidad de todos los nodos modelando a la red mediante un grafo probabilístico, considerando que los enlaces fallan con cierta probabilidad q_e y los nodos son confiables ($p_{\text{nodo}}=1$).

Para un enlace $e \in E$, denotándose con p_e a la probabilidad de que el enlace esté operativo y con $q_e=1-p_e$ a la probabilidad que el mismo falle y, si se considera que en cualquier instante sólo un conjunto de enlaces de G estará operacional, un estado de G es un subgrafo (N,E') donde $E' \subseteq E$ es el conjunto de enlaces que está operacional. Si Ω es el conjunto de todos los posibles estados operacionales, la confiabilidad exacta de la red esta dada por la siguiente expresión (Dengiz, 1997):

$$R(G) = \sum_{\Omega} \left[\prod_{e \in E'} p_e \right] \cdot \left[\prod_{e \in E \setminus E'} q_e \right] \quad (3.5)$$

El tamaño de Ω crece exponencialmente en función del tamaño de la red haciendo impracticable su cálculo aún tratándose de redes de tamaño moderado (Colbourn, 1991). Se han presentado métodos de cálculo exacto de la confiabilidad basados en enumeración de estados, caminos mínimos (min-path) o conjuntos de corte mínimos (cut-set) (Aggarwal, 1981), (Aggarwal, 1982), (Nakawaza, 1981), (Rai, 1982), (Cavers, 1975). No obstante, las mejoras relativas al tiempo computacional no han sido importantes. Así en el cálculo de la confiabilidad para redes de tamaño considerable, una alternativa a utilizar es estimarla con métodos basados en simulación.

Cabe aclarar que el modelo utilizado se adecua a realidades donde la información a comunicarse entre dos nodos se rutea de forma dinámica a través de la red. Esto significa que si bien puede existir un camino predeterminado para comunicar dos nodos, se puede elegir un camino alternativo en caso de falla de algún componente intermedio del camino predeterminado (por ejemplo, el ruteo de paquetes entre dos hosts de Internet). No se consideran aquí detalles como el control de congestión y retrasos en la comunicación (problemas clásicos en redes de datos y de transporte) (Ball, 1992).

3.2.3 Estimación de la confiabilidad

A raíz de los problemas derivados de la complejidad computacional asociada al cálculo exacto de la confiabilidad, se han propuesto métodos que, o bien intentan reducir el espacio explorado o bien, basados en modelos de simulación, permiten estimar un valor para la confiabilidad de una red dada las probabilidades de falla de sus componentes.

Un método ampliamente conocido es el algoritmo de Jan (Jan, 1993), propuesto para encontrar una topología de mínimo costo con una confiabilidad no inferior a un valor dado R_0 . Jan sugiere calcular un límite inferior de la cantidad de enlaces e^* , que combinados de alguna manera permitan que la red posea una confiabilidad mayor o igual a R_0 . De esta forma evita la inspección de aquellas redes con una cantidad de enlaces tal que se puede asegurar que su confiabilidad será inferior a la requerida.

En realidad, esta propuesta tiende a evitar el cálculo de la confiabilidad cuando a priori puede determinarse que su valor será inferior al requerido y por tanto descartable (o penalizable) en lo referente al problema de diseño de la red. De hecho, una red de n nodos con menos de $n-1$ enlaces tiene confiabilidad 0. La red más confiable con n enlaces debe ser un *spanning tree*, mientras que si se consideran $n+1$ enlaces la red más confiable tiene una topología de anillo. Finalmente si se cuentan con $n+2$ nodos, la red más confiable se dispone como muestra la figura 3.2.

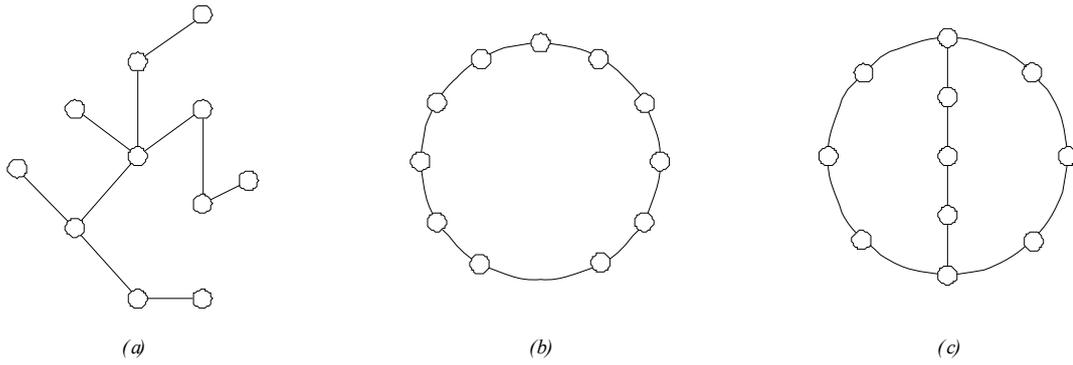


Figura 3.2 - Topologías más confiables para una red de 11 nodos.

a) 10 enlaces, b) 11 enlaces, c) 12 enlaces.

Para estos casos Jan ha propuesto expresiones sencillas que permiten el cálculo exacto de la confiabilidad. Para el caso a) de la figura 3.2, la confiabilidad de la red, si todos los enlaces tienen la misma confiabilidad p_e está dada por:

$$R(n-1) = p_e^{n-1} \tag{3.6}$$

Una expresión para el caso b de la figura 3.2 es:

$$R(n) = p_e^n + np_e^{n-1}q_e \tag{3.7}$$

Donde $q_e=1-p_e$. El mejor de los casos para $n+1$ enlaces activos es un grafo de tres ciclos, donde la cantidad de enlaces que forma cada ciclo no difiere en más de uno. La expresión para este caso ya es más compleja y no se ha generalizado, sólo es precisa cuando $n+1$ es múltiplo de 3.

$$R(n+1) = p_e^{n+1} + (n+1)p_e^n q_e + \frac{1}{3}(n+1)^2 p_e^{n-1} q_e^2 \tag{3.8}$$

Finalmente Jan propone una expresión, basada en los grados de cada nodo de la red, para establecer un límite superior (*upper bound*) sobre la confiabilidad:

$$R(G) \leq H(d) \tag{3.9}$$

Donde

$$H(d) = 1 - \left\{ \sum_{i=1}^n q_e^{d_i} \prod_{k=1}^{m_i} (1 - q_e^{d_k-1}) \prod_{k=m_i+1}^{i-1} (1 - q_e^{d_k}) \right\} \quad (3.10)$$

Y donde $m_i = \min(d_i, i-1)$, para toda i y d_k es el grado del nodo k .

Este límite superior será un umbral de referencia para la confiabilidad máxima posible que puede alcanzar cualquier red.

No obstante, y como se ha mencionado en párrafos anteriores, esta propuesta está orientada a evitar el cálculo cuando a priori puede establecerse que la topología propuesta no alcanza el valor prefijado como admisible. Cuando la cantidad de nodos en la red se incrementa, el tiempo de cómputo nuevamente se hace prohibitivo. Una alternativa está dada por la aplicación de métodos de simulación para obtener un valor, ahora aproximado, de la confiabilidad.

3.2.4 Evaluación de la confiabilidad por simulación

Los métodos más conocidos de este tipo están basados en la técnica de simulación Monte Carlo, y proporcionan estimaciones (con un cierto intervalo de confianza) de las medidas de seguridad de funcionamiento. El método Monte Carlo estándar (o “Monte Carlo crudo”) también sufre de inconvenientes cuando los eventos de interés son raros (ocurren con muy baja probabilidad), lo que es común en los sistemas altamente confiables. Es necesario entonces recurrir a otras técnicas, tales como la de la reducción de la varianza, que pueden calcular estimadores más precisos utilizando muestras del mismo tamaño (Fishman, 1997).

3.2.4.1 Algoritmo Monte Carlo Crudo (MCC)

Dado un grafo G , un estado o subgrafo G' de G en un instante particular queda definido por los componentes originales de G que en ese instante funcionan. La idea de este algoritmo se basa en obtener una estimación del valor de la confiabilidad de la red mediante la selección de N estados G'_i del grafo original G . En definitiva, la

confiabilidad de la red se calcula como la frecuencia de ocurrencia del suceso “*todos los nodos están conectados entre sí*”. Formalmente se puede decir que se estima la medida original R mediante el estimador R_{est} definido a partir de la muestra G'_1, G'_2, \dots, G'_N de la siguiente forma¹:

$$R_{est} = \frac{1}{N} \sum_{i=1}^N 1(G'_i \text{ operativo}) \quad (3.11)$$

La calidad de la estimación se evalúa mediante el cálculo de intervalos de confianza cuya amplitud es proporcional a la varianza del estimador. La expresión para dicha varianza es la siguiente:

$$Var(R_{est}) = \frac{R \cdot (1 - R)}{N} \quad (3.12)$$

Donde el numerador corresponde a la varianza de una variable aleatoria con distribución de Bernoulli. Al no disponerse del valor de R se construye un estimador para $Var(R_{est})$ de la siguiente forma:

$$Var(R_{est}) = \frac{1}{N(N-1)} \sum_{i=1}^N (1_{(G'_i \text{ operativo})} - R_{est})^2 \quad (3.13)$$

En el caso particular de las variables aleatorias de Bernoulli, la expresión anterior se simplifica a:

$$Var(R_{est}) = \frac{R_{est} (1 - R_{est})}{N - 1} \quad (3.14)$$

El algoritmo para la determinación de la confiabilidad de una red mediante el método de Monte Carlo consiste en eliminar aleatoriamente algunas ramas de la red con una

¹ Notación: $1A$ es la función indicatriz del conjunto (evento) A . Se define como $1A(\omega) = 1$ si $\omega \in A$ y $1A(\omega) = 0$ si $\omega \notin A$, o de otra forma $1A(\text{true}) = 1$ y $1A(\text{false}) = 0$.

probabilidad p_e y se determina si la red queda conectada, aun después de la eliminación de esas ramas. Si se repite este experimento un número suficientemente elevado de veces, la relación entre redes desconectadas y el número de ensayos realizados da una estimación de la confiabilidad.

```
* algoritmo MCC
  Calcular n veces
    Desconectar enlaces con probabilidad p
    Calcular cantidad de desconexión
  Fin
  Confiabilidad = (1-desconec)/n
```

Figura 3.3 – Algoritmo de Monte Carlo Crudo

3.2.4.2 Algoritmo de Reducción Recursiva de la Varianza

El algoritmo Monte Carlo estándar produce resultados aceptables para redes de mediana confiabilidad, pero para redes altamente confiables se deben realizar muchas repeticiones para obtener estados no operativos. Los tiempos de procesamiento crecen linealmente con el número de ensayos, sin mejorar la exactitud en la misma proporción. En pruebas realizadas se observa que con 6000 ensayos se obtienen precisiones del orden de 0,001, por debajo de ese valor la precisión cae notoriamente, por otra parte incrementando el número de ensayo en un orden de magnitud; la varianza mejora muy poco. Estos resultados hacen necesario buscar métodos que equilibren tiempo de procesamiento y precisión.

En (Cancela, 1997) se propone un algoritmo que reduce el problema original a un problema en una red más pequeña construida a partir de la original, condicionada a uno de sus cortes. El proceso es recursivo y se detiene cuando la red se encuentra siempre en estado operativo o no operativo, independientemente del estado de sus componentes. La construcción del método se realiza para obtener una estimación de la medida Q_k (anti-confiabilidad para un conjunto K de terminales), pero se puede generalizar para todas las restantes.

Los métodos de reducción recursiva de la varianza (Cancela, 1995) garantizan una varianza menor que los métodos de Monte Carlo estándar, y están basados en el uso

recursivo de un procedimiento de particionamiento. A continuación se describe brevemente el método.

3.2.4.2.1 Definiciones y notación

Para el desarrollo del método, previamente se define lo siguiente:

- Red K -conexa: una red $G=(V,E)$ con un conjunto de terminales asociado $K \subseteq V$ es K -conexa cuando existe por lo menos un camino entre todo par de nodos del conjunto K (red en estado operativo).
- $Q(G)$: anti-confiabilidad de G (probabilidad de que la red G no sea K -conexa).
- Un conjunto $D \subset V \cup E$ es un K -corte extendido de G si la subred $G'=(V - D, E - D)$ no es K -conexa.
- Para un enlace $e \in E$ en $G=(V, E)$ con nodos en K , $G - e$ es la red cuyo conjunto de nodos es V y cuyo conjunto de enlaces se obtiene a partir de E eliminando e . El conjunto de nodos de $G - e$ es igual a K .
- Para un nodo $v \in V$ en $G=(V, E)$ con nodos en K , $G - v$ es la red cuyo conjunto de nodos es $V - \{v\}$ y cuyo conjunto de enlaces se obtiene a partir de E eliminando todos los enlaces incidentes a v . El conjunto de nodos de $G - v$ es igual a $K - \{v\}$.
- Sea $G=(V, E)$ con terminales en K y d un componente (nodo o enlace) de la misma. Se denota $G | d$ a la red derivada de G asignando probabilidad de funcionamiento de d a 1 (d será un componente perfecto).
- Sea $G=(V, E)$ con nodos en K y d un componente (nodo o enlace) de la misma. Se denota $G * d$ a la red derivada de G asignando la probabilidad de funcionamiento de d a 1. Si luego se halla un enlace $e=(v_1, v_2)$ (si d es una arista será $e=d$; si d es un nodo será $v_1=d$ ó $v_2=d$) tal que $r_e=r_{v_1}=r_{v_2}=1$ se debe realizar la contracción de la misma, esto es, eliminando e , fundiendo sus extremidades v_1 y v_2 en un nuevo nodo

w y asignando el nuevo conjunto de enlaces a $K - \{v_1, v_2\} \cup \{w\}$ si v_1 o v_2 pertenece a K , o simplemente K si ninguno es nodo.

El objetivo ahora, es construir un estimador con igual esperanza que el utilizado en MCC pero con menor varianza. Para ello se consideran las siguientes propiedades:

3.2.4.2.2 Propiedad 1

Sea $G = (V, E)$ con terminales en $K = \{v_1, v_2, \dots, v_{|K|}\}$, entonces se cumple:

$$R(G) = \left(\prod_{v \in K} r_v \right) R(G | v_1 | v_2 | \dots | v_{|K|}) \tag{3.15}$$

y

$$Q(G) = \left(1 - \prod_{v \in K} r_v \right) + \left(\prod_{v \in K} r_v \right) Q(G | v_1 | v_2 | \dots | v_{|K|}) \tag{3.16}$$

Significa que la medida de confiabilidad puede obtenerse considerando los nodos terminales como perfectos, multiplicando el resultado de la confiabilidad de esa red resultante por los productos de las confiabilidades de los nodos terminales. Esta propiedad permite reducir el caso de fallas en enlaces y nodos al modelo de nodos perfectos.

3.2.4.2.3 Propiedad 2

Sea $G = (V, E)$ con terminales en K , y d un componente de G , entonces se cumple $R(G | d) = R(G * d)$.

Significa que si al considerarse un componente como perfecto en la red original se puede realizar contracción, la confiabilidad de la red original es igual a la confiabilidad de la red "contraída". Utilizando esta propiedad, el algoritmo RVR (que se presenta más adelante) reduce el problema original a un problema en una red cada vez de menor tamaño.

Las demostraciones de las propiedades 1 y 2 pueden encontrarse en (Cancela, 1997).

El método define la variable aleatoria $Y(G)$ como $Y(G) = 1 - \Phi(X_G)$ (cuya esperanza es el estimador de la confiabilidad para MCC).

Donde:

X_G es el vector de estados aleatorios de los componentes

Φ es la función de estructura definida como $\Phi : G \rightarrow \{0,1\}$, que vale 1 si la red está en estado operativo y 0 en caso contrario

El objetivo de la técnica RVR es construir una variable aleatoria $Z(G)$ con la misma esperanza de $Y(G)$ y menor varianza. Dicha variable aleatoria se construye a partir de un K-corte extendido D , y se expresa en términos de $|D|$ variables aleatorias $Y(G_i)$ correspondientes a estados de la red original.

3.2.4.2.4 Propiedad 3

Para una red G tal que $r_v = 1 \forall v \in K$ (red donde los nodos no fallan) se define entonces la variable aleatoria

$$Z(G) = Q_D + (1 - Q_D) \sum_{v=i} 1_{v=i} Y(G_i) \quad (3.17)$$

Que tiene la misma esperanza, y menor varianza que la original Y . Para la cual Q_D se define como la probabilidad de que todos los enlaces en el corte fallen, y

$$Y(G) = 1 - \Phi(X_G) \quad (3.18)$$

Sobre esta base, se construye en forma recursiva la siguiente variable aleatoria F :

$$F(G) = \left\{ \begin{array}{l} 1 \text{ si } G \text{ no es } K\text{-conexo} \\ 0 \text{ si } K \text{ está formado por un solo nodo (} G \text{ siempre } K\text{-conexo)} \\ Z(G) = Q_D + (1 - Q_D) \sum_{v=i} 1 Y(G_i) \text{ en otro caso} \end{array} \right\}$$

El algoritmo RVR se muestra en la figura 3.4

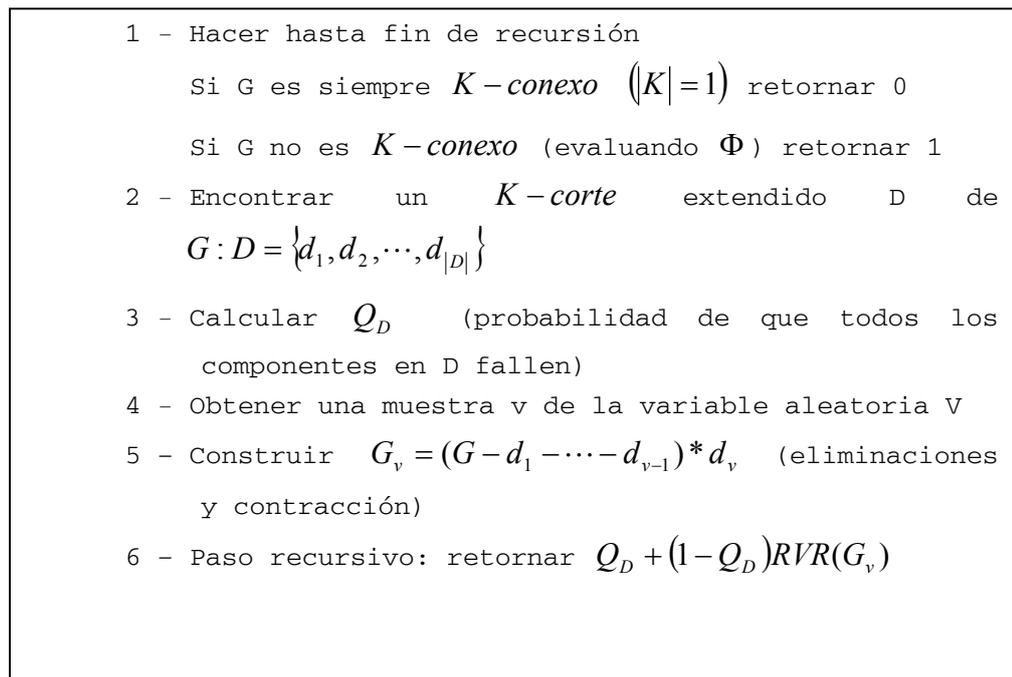


Figura 3.4 – Algoritmo RVR

En este trabajo, la evaluación de la confiabilidad se realiza mediante el método de reducción recursiva de la varianza por cuanto permite obtener una esperanza igual a la que proporciona el método de Monte Carlo pero garantiza una varianza menor respecto de aquel.

3.3 Diseño óptimo de redes confiables

Una vez decidida la forma de asignar un valor de confiabilidad a una red se aborda a continuación el problema de diseño óptimo de redes confiables. Esto es, determinar para una determinada cantidad y ubicación de nodos a interconectar, una topología que tenga el mínimo costo posible y que a la vez posea una confiabilidad no inferior a un valor prefijado como admisible.

Una de las primeras propuestas se deben a Jan (1993) quien, como se ha mencionado, propuso un método basado en *branch and bound* para la exploración de las diferentes topologías e hizo uso de la técnica de *backtracking* propuesta por Ball (1992). Adicionalmente estableció estrategias para determinar límites superiores para la confiabilidad de una topología propuesta, de forma tal de evitar el cálculo innecesario de la confiabilidad exacta.

Todo grafo conectado tiene al menos $n-1$ enlaces activos y $n_{max}=n(n-1)/2$ enlaces posibles. Pueden existir problemas en los cuales no todos los enlaces están disponibles. Esto reduce la cantidad de enlaces activos posibles a un valor $n^* \leq n_{max}$. Entonces, podrían encontrarse todas las combinaciones posibles de enlaces activos, descomponiendo el problema en $n^*-(n-1)+1 = n^*-n+2$ subproblemas (*sp*) de manera tal que cada uno de ellos involucre una cantidad fija de estos enlaces activos

La solución al problema principal menos elaborada consistiría en encontrar la mejor solución de cada subproblema para luego optar por la mejor de todas. Sin embargo, este enfoque puede resultar ineficiente desde el punto de vista de procesamiento, porque es necesario resolver todos y cada uno de los subproblemas.

El trabajo de Jan sugiere calcular primeramente un límite inferior de cantidad de enlaces a^* , que combinados de alguna manera permitan que la red posea una confiabilidad mayor o igual a la admisible R_0 . De esta forma se evita inspeccionar aquellos subproblemas cuya cantidad de enlaces se encuentre entre $n-1$ y a^* .

Luego se resuelve el subproblema correspondiente a a^* , $sp(a^*)$, donde se busca la mejor solución posible con a^* enlaces activos y se establece temporalmente la solución encontrada como la mejor solución global.

Seguidamente se procede a evaluar el menor costo esperado para el siguiente nivel (a^*+1), mediante la suma de los costos de los a^*+1 enlaces más baratos. Si esta combinación es más cara que la ya encontrada en el nivel a^* , aquella era la mejor de todas las posibles. En caso contrario, habrá que resolver el subproblema $sp(a^*+1)$. Se procede así sucesivamente hasta encontrar la solución del problema principal. De esta manera es posible evitar el registro de cierta cantidad de niveles, dependiendo de las

características de cada caso.

Sin embargo existen muchos casos en los cuales el algoritmo presenta problemas que dependen de la rama del árbol que inspecciona *branch and bound* está la solución. Además, la función que fija el límite superior para la confiabilidad no resulta todo lo eficiente que se quisiera porque en general ni siquiera es capaz de posicionarse en el subproblema que contiene la solución óptima. Alcanzarlo puede significar la exploración completa de muchos de los niveles anteriores, haciendo que los tiempos se vuelvan impracticables en dichos casos.

Incluso ya en el subproblema correcto, este *límite superior* no logra la precisión necesaria como para evitar, en general, el cálculo de la restricción sobre topologías poco confiables.

Si bien estas apreciaciones justifican investigar otras formas de búsqueda, algunas de las estrategias mencionadas hasta aquí son factibles de ser utilizadas dado que es necesario valerse de técnicas que no inspeccionen todo el espacio de búsqueda. Por eso muchas veces se han utilizado heurísticas que permiten una exploración parcial del espacio en la resolución de problemas combinatorios. Entre ellas, los Algoritmos Evolutivos (AEs) han demostrado ser una interesante alternativa dada su relativamente fácil implementación, su robustez y su posible balance, a través de una correcta determinación de sus parámetros, entre la exploración y explotación del espacio de búsqueda.

Koide (2001) considera una extensión del método de Jan con el objetivo de mejorar los tiempos de cómputo en algunos casos y propone otro método para calcular el límite superior para la confiabilidad. No obstante aunque presenta buenas mejoras respecto del algoritmo de Jan, se reportan resultados para redes de unos pocos nodos.

Dengiz y Co. (1997) propusieron la implementación de un Algoritmo Genético para resolver el problema sobre una buena cantidad de instancias con espacios de búsqueda de hasta $2 \cdot 10^{90}$. Para reducir el tiempo de cálculo requerido se impuso la restricción adicional de que las redes candidatas debían reunir condiciones de biconectividad.

Reichelt y Co (2003) resolvieron un problema de diseño óptimo de redes confiables utilizando heurísticas de reparación de redes no factibles. La propuesta utiliza una medida de la confiabilidad entre todos los nodos basada en el conteo de *spanning trees* que posee la red candidata (Fard, 2001).

3.4 Resumen

En este capítulo se han esbozado los conceptos involucrados en la resolución del problema de diseño de redes confiables. Se han enfocado dos aspectos muy importantes a tener en cuenta: el problema de encontrar una topología de mínimo costo sujeto a restricciones de confiabilidad y el cálculo de la misma. Ambos problemas son de difícil resolución y se han presentado, a modo de revisión, distintas estrategias para atacarlo.

En general, estos problemas de diseño pueden verse como una búsqueda dentro de un espacio de soluciones potenciales. Dado que se busca la mejor entre todas las soluciones posibles, se puede decir que se está en presencia de un proceso de optimización. En estos casos se sabe que no se puede garantizar una solución óptima global, pero es posible hallar una que sea lo suficientemente buena, si se puede evaluar su calidad con cierta precisión.

En el próximo capítulo se analizan técnicas para inspeccionar el espacio de búsqueda, constituido por todas las posibles combinaciones topológicas, desde un enfoque no exhaustivo, mucho más apropiado para abordar este tipo de problema. Sin embargo, no es posible eliminar los tiempos de procesamiento propios del cálculo de la confiabilidad, los cuales crecen muy rápidamente con el tamaño de la red.

CAPÍTULO 4

4 Algoritmos Evolutivos

4.1 Introducción

Los Algoritmos Evolutivos (AE) son técnicas de búsqueda que intentan simular los mecanismos de la evolución natural, resultantes de la adaptación de los seres vivos del medio ambiente, para solucionar problemas de optimización. Esta simulación establece analogías entre los AE y los mecanismos que provee la naturaleza buscando adaptar soluciones potenciales (cromosomas o individuos) a un medio en función de su *fitness* o ajuste (dado por el mérito del individuo en la población). Esto implica que, en general, un AE puede resolver problemas de optimización de funciones sin la exigencia necesaria de continuidad ni derivabilidad.

El funcionamiento de los AE se basa en la modificación iterativa de una población de individuos candidatos a solución mediante el uso de operadores genéticos. La teoría en la cual se apoyan los AE está explicitada en Goldberg (1989) y Michalewicz (1995).

El término Algoritmo Evolutivo, en realidad engloba una serie de técnicas de búsquedas inspiradas en la biología (teoría neo-Darwiniana de la evolución natural) basada en los mecanismos de la naturaleza, de acuerdo a los cuales los individuos más aptos de una población sobreviven al adaptarse mejor a los cambios que se producen en su entorno. Es por esto que se encuentra literatura relativa a estos algoritmos, que utiliza términos biológicos, tales como la denominación de cromosomas a los individuos y gen a los bits que conforman esos individuos o cromosomas. En términos generales, para simular el proceso evolutivo en una computadora se requiere:

- Estructuras codificadas que puedan replicarse

- Operaciones que afecten a los individuos
- Una función de aptitud
- Un mecanismo de selección

El mecanismo de resolución de problemas evolutivos puede esquematizarse como en la Figura 4.1.

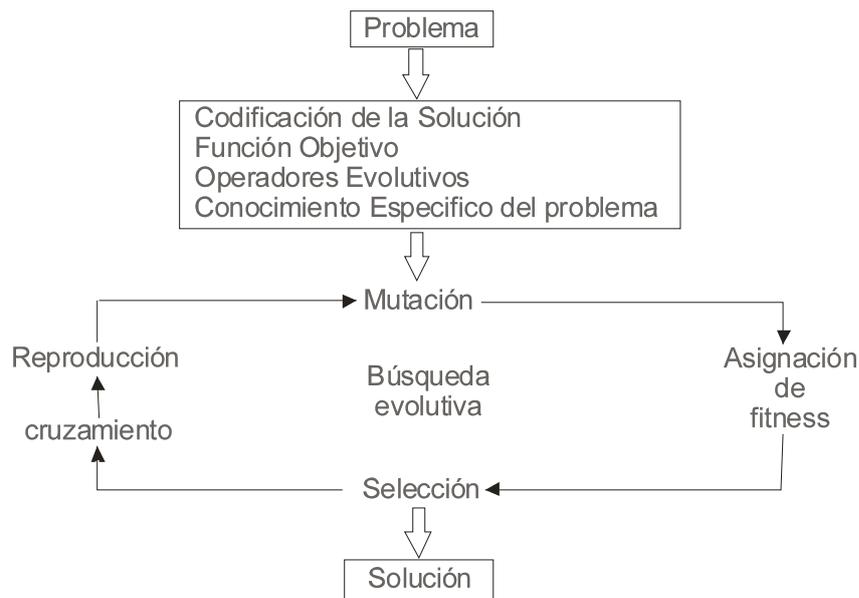


Figura 4.1 - Esquema de funcionamiento de un AE

Los AE son algoritmos estocásticos que mantienen una población de n individuos, $P(t) = [x_1^t, \dots, x_n^t]$ para cada iteración t . Cada individuo representa una solución potencial al problema a tratar, la cual es implementada como alguna (posiblemente compleja) estructura de datos \mathbf{S} .

Cada solución x_i^t es evaluada con alguna medida de su aptitud. Posteriormente se seleccionan individuos de la población para conformar una población intermedia (o conjunto de apareamiento) que es transformada por la acción de los llamados operadores genéticos para luego aplicar un mecanismo de reinserción que arrojará como resultado una nueva población (iteración $t+1$). Los operadores genéticos son transformaciones que tienen lugar sobre los individuos. Algunos miembros sufren transformaciones unarias \mathbf{m}_i (mutación), que crean un nuevo individuo ($\mathbf{m}_i: \mathbf{S} \rightarrow \mathbf{S}$) y

transformaciones de orden más alto c_j (tipo cruzamiento) que crean nuevos individuos combinando parte de algunos (dos o más) individuos ($c_j: S \times S \dots \times S \rightarrow S$).

El algoritmo se ejecuta para cierto número de generaciones o hasta que se satisface algún criterio de detención. En la figura 4.2 se muestra la estructura de un AE simple, (Michalewicz, 1996).

```
* AE simple.  
t=0;  
iniciar P(t);  
hacer_mientras t<tmax  
    t=t+1;  
    selección (P(t)) desde P(t-1);  
    cruzamiento(P(t));  
    mutación(P(t));  
fin_de_hacer  
fin
```

Figura 4.2 - Estructura de un AE simple

Dentro de estos AE se encuentran los Algoritmos Genéticos (AG), con algunas modificaciones al AG puro, que son ampliamente utilizados en la resolución de diversos problemas de optimización combinatorial con restricciones, tales como el problema del viajante (TSP), el problema de coloreo de un grafo, el empaqueo de objetos, búsqueda de un ciclo hamiltoniano, problema de programación de tareas, problemas de diseño de topologías, problemas de ubicación óptima de sensores en procesos continuos, ubicación óptima de radio bases en sistemas de telefonía celular o redes inalámbricas, entre otros.

Está generalmente aceptado que cualquier Algoritmo Evolutivo debe tener al menos cinco componentes básicos (Michalewicz, 1996):

- Una representación genética de las soluciones
- Una forma de crear una población inicial de soluciones
- Una función de evaluación que "clasifica" las soluciones en términos de su adaptabilidad o *fitness*.

- Operadores genéticos que alteran la composición genética de los hijos durante la reproducción.
- Valores de los parámetros (tamaño de la población, probabilidad de aplicar los operadores genéticos, etc).

También es conocido que para lograr una implementación exitosa de una técnica evolutiva en un problema concreto, los componentes básicos listados anteriormente requieren de la incorporación de alguna estrategia adicional como por ejemplo la provisión de conocimiento específico del problema.

Los métodos de búsqueda de los AE modelan un fenómeno natural, es decir, tratan de simular lo que hace la propia naturaleza en la búsqueda de la o las soluciones óptimas para un problema de evolución dado. Pertenecen, además, a los algoritmos probabilísticos pero no son simplemente aleatorios porque combinan elementos de búsqueda directa y estocástica.

Una característica fundamental es que mantienen una población de soluciones potenciales, llevando a cabo una búsqueda multidireccional e intercambiando información entre esas soluciones, mientras que otros métodos procesan sólo un punto del espacio de búsqueda.

Para modelar los procesos de la naturaleza los AE deben representar procesos de selección, recombinación, mutación, y estructuras de población que se describirán a continuación.

La figura 4.3 muestra la estructura de un algoritmo evolutivo de una única población. A diferencia de otros métodos de optimización que trabajan sobre una solución inicial y van refinándola sucesivamente, un AE trabaja con una población de soluciones (individuos) que evolucionan. Esta característica le da al AE posibilidades de paralelizar más fácilmente la búsqueda.

Al comienzo del cálculo se crea una cierta cantidad de individuos (población) y se evalúa la función objetivo para esos individuos, entre los cuales generalmente no se

encontrará la solución buscada. Comienza entonces la generación de una nueva población, donde los individuos son seleccionados de acuerdo a sus *fitness* para la producción de descendientes. Los padres se recombinan para generar hijos y todos los individuos generados sufren un proceso de mutación con una cierta probabilidad, calculándose un *fitness* de los descendientes que luego se reinsertan en la población reemplazando a los padres, produciendo la nueva generación. Este ciclo se realiza hasta que se satisface algún criterio de optimización.

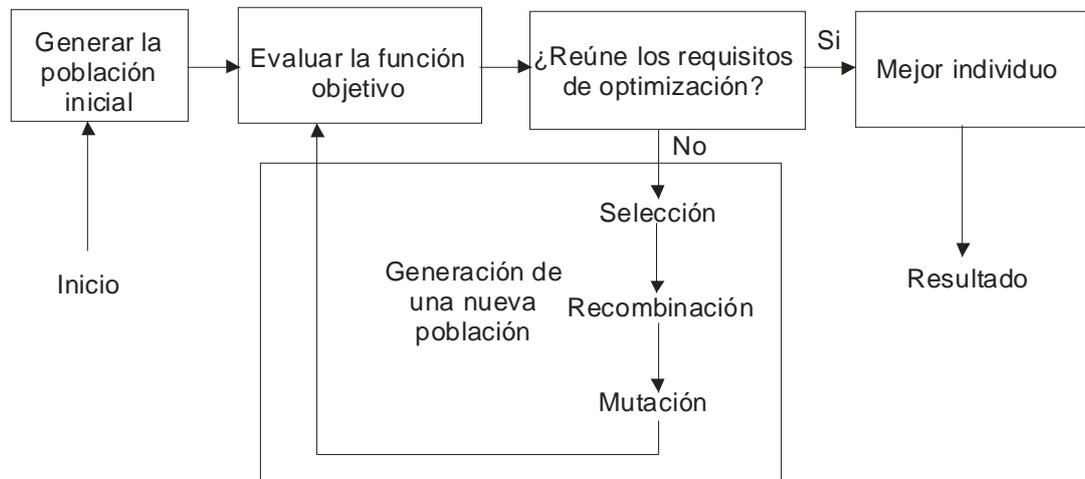


Figura 4.3 - Estructura de un AE de una única población

Un algoritmo evolutivo de una única población es potente y funcionará bien en una amplia variedad de problemas, pero es posible obtener mejores resultados introduciendo múltiples subpoblaciones. Estas desarrollan algunas generaciones (igual que en el caso de población única) y luego intercambian individuos entre las subpoblaciones. Este modelo multipoblacional es más parecido a los procesos naturales. La figura 4.4 muestra la estructura de un AE con multipoblación.

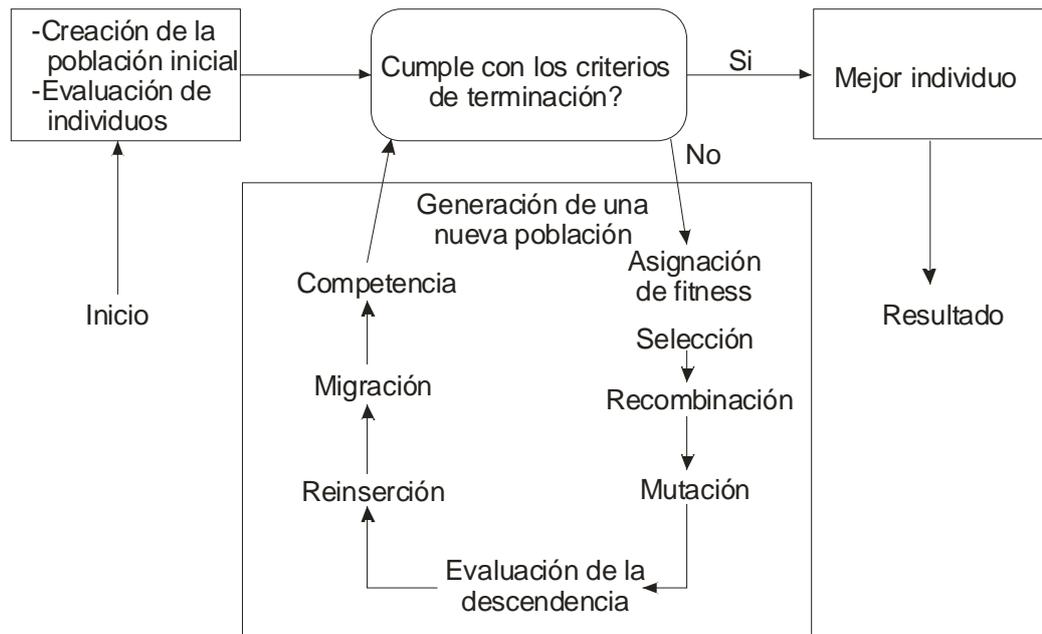


Figura 4.4 - Estructura de un AE con múltiples poblaciones

4.2 Representación genética de las soluciones

La aplicación más común de los algoritmos evolutivos ha sido la solución de problemas de optimización, donde han demostrado ser muy eficientes y confiables. Sin embargo, no es aplicable para la resolución de todos los problemas de este tipo, por lo cual es necesario, antes de su aplicación, tener en cuenta las siguientes características:

- El espacio de búsqueda (es decir, las posibles soluciones) debe estar delimitado dentro de un cierto rango.
- Debe poder definirse una función de aptitud que nos indique qué tan buena o mala es una propuesta de solución.
- Las soluciones deben codificarse de una forma que resulte relativamente fácil de implementar en una computadora.

Para diseñar un AE es necesario codificar una solución del espacio de búsqueda real en una solución del espacio de búsqueda del AE. Esto implica una transformación, entre ambos espacios de tal forma que la codificación no admita duplicación de

soluciones. Por otro lado, como apunta Goldberg (1989), una solución debe ser codificada en la forma más natural.

Una de las formas de representar soluciones es la utilización de vectores binarios, también llamados cadenas binarias. Por ejemplo, en una red de n nodos, en la cual pueden existir $n*(n-1)/2$ enlaces entre ellos, la existencia o no de un enlace puede representarse con una variable binaria $x_i \in \{0,1\}$ de tal manera que $x_i = 1$ si el enlace i existe y $x_i = 0$ en caso contrario. De esa manera un vector \mathbf{x} podrá representar los enlaces presentes para una cantidad de nodos determinada n .

Otra forma de representar soluciones puede llevarse a cabo mediante la utilización de vectores de números enteros. En el tratamiento de problemas como el del viajante de comercio (TSP o *Traveller Salesman Problem*) o el problema de ruteo de vehículos (VRP o *Vehicle Routing Problem*) resulta conveniente representar las rutas mediante la identificación de los nodos que van siendo visitados. De esta forma el vector $r = [1\ 3\ 7\ 2\ 5\ 4\ 6]$ representa la ruta que pasa por los nodos (clientes, localidades) 1, 3, 7, 2, 5, 4, 6 en ese orden.

También es posible utilizar vectores de números reales donde cada una de las componentes del vector está representando una magnitud determinada. En este caso cada uno de los genes componentes del cromosoma posee como valor un número real.

Cada problema tiene asociado una forma de representación que mejor se adapta para llevar a cabo el proceso evolutivo. La elección de una representación conveniente, generalmente la más natural, conduce a un mejor tratamiento de los individuos que forman la población.

4.3 Creación de la población inicial

Se ha propuesto varias alternativas de creación de la población inicial: se puede comenzar con una población generada en forma aleatoria, lo cual puede implicar la existencia de individuos no válidos, en caso de problemas con restricciones. Otra alternativa consiste en sembrar la población utilizando la salida de algún algoritmo que

incluya aspectos tales como conocimiento específico del problema, diversidad de las soluciones, factibilidad de las mismas, etc.

En algunos casos, es posible contar con algún conocimiento sobre la forma que pueden tener las soluciones y es muy conveniente hacer uso de él. Por ejemplo, es conocido que las redes biconectadas son más confiables que las que tienen nodos con un único enlace. Una forma de incorporar este hecho es construir la población inicial teniendo en cuenta estas sub-combinaciones, aunque a veces resulta conveniente determinar que porcentaje de la población debe ser sembrada para que el AE trabaje aprovechando esta información sin provocar problemas, por ejemplo de convergencia prematura. También es posible valerse de los resultados de algún proceso determinista como los presentados en el capítulo anterior.

Una buena población inicial resulta un buen comienzo siempre y cuando se asegure la diversidad genética en la población.

4.4 Función de evaluación o función de adaptabilidad (*fitness*)

La función de adaptabilidad o *fitness* es una medida de la calidad de la solución. En ella está contenida la principal fuente de conocimiento específico del problema y debe medir las diferencias en la calidad de las soluciones.

En los métodos de computación evolutiva la función de evaluación sirve no sólo como nexo entre el problema y el algoritmo, sino que mide la calidad de las soluciones: los mejores individuos tienen mayores probabilidades de supervivencia y reproducción. Por lo tanto es esencial definir una función de adaptabilidad que caracterice al problema con precisión.

En la mayoría de los casos, el proceso de selección de una función de adaptabilidad parece ser directo, sin embargo tal proceso puede ser más complejo cuando se trabaja con soluciones factibles y no factibles. El tratamiento de problemas restringidos se esboza en el siguiente punto.

4.5 Soluciones factibles y no factibles: manejo de las restricciones del problema.

Los algoritmos evolutivos, tal como están descritos, no contemplan la resolución de problemas con restricciones. Sin embargo los problemas del mundo real casi siempre están sujetos a restricciones las cuales pueden ser de igualdad, desigualdad, lineales y/o no lineales. Por ejemplo, para un problema de minimización que incluya restricciones, la ecuación 1.1 puede expresarse de la siguiente manera:

$$\begin{aligned} & \text{Min } f(\mathbf{x}) \\ & \text{st.} \\ & g_i(\mathbf{x}) \geq 0 \quad i = 1, \dots, k \\ & h_j(\mathbf{x}) = 0 \quad j = 1, \dots, p \end{aligned} \tag{4.1}$$

Donde \mathbf{x} es el vector de estado de enlaces f evalúa la solución con algún criterio de desempeño y g y h representan las restricciones al problema.

Para considerar la resolución del problema restringido planteado es necesario implementar estrategias que permitan el manejo de las restricciones.

En particular, el problema de manejar individuos factibles y no factibles, debe llevarse a cabo cuidadosamente: a menudo la población contiene individuos no factibles, pero siempre se busca un *óptimo factible*. Encontrar una medida de adaptabilidad para ambos tipos de individuos es de gran importancia, por cuanto esto influye directamente en el resultado del algoritmo.

En el caso de problemas de programación no lineal, con dominios continuos, se han realizado diferentes aplicaciones utilizando varias funciones de prueba (De Jong, 1975). En los dominios discretos el problema de las restricciones fue reconocido más tarde. Problemas tales como el TSP, de scheduling, el de localización de sensores, o el de optimización de topología de redes, son problemas de optimización combinatorial con restricciones.

Las estrategias de manejo de restricciones utilizando AEs puede clasificarse en (Gen, 1996):

1. **Estrategias de rechazo:** Dentro de esta categoría la estrategia de *pena de muerte* ha sido una opción común en varias técnicas evolutivas. El método consiste en eliminar las soluciones no factibles de la población. Produce buenos resultados si el espacio de búsqueda es convexo y constituye una parte razonable del espacio de búsqueda total.
2. **Estrategias de Reparación:** Consiste en tomar una solución no factible y generar una factible a través de algún procedimiento de modificación o enmienda. Esta técnica es muy dependiente del problema particular y se han reportado desde casos de muy exitosa aplicación, hasta ejemplos en los cuales reparar un individuo no factible se transforma en un problema tan complejo como el original (Michalewicz, 1996).
3. **Estrategias de penalización:** Es un enfoque muy común, principalmente en AG. Transforma el problema *restringido* en un problema no restringido penalizando las soluciones no factibles. Esto se realiza adicionando un término a la función de adaptabilidad, que depende del grado de violación de las restricciones (Goldberg, 1989):

$$F_i = f(x_i) + RQ(x_i) \quad (4.2)$$

Donde f representa la función objetivo, Q tiene en cuenta las violaciones al conjunto de restricciones y R es un coeficiente de penalización. Q vale 0 si no hay violación de ninguna restricción y toma un valor positivo en el caso contrario (considerando minimización). En general Q es una función compuesta por un conjunto de funciones de penalización que miden el grado de violación de cada una de las restricciones impuestas al problema.

Otra alternativa es por ejemplo, la propuesta por Deb (2000), donde se utiliza la siguiente expresión para la función de *fitness*:

$$F(x_i) = \begin{cases} f(x_i) & \text{si } x_i \text{ es factible} \\ f_{max} + Q(x_i) & \text{si } x_i \text{ es no factible} \end{cases} \quad (4.3)$$

Donde f_{max} es el valor máximo de la función objetivo sobre todos los individuos factibles de la población.

Se han propuesto diversas funciones de penalización: Homaifar (1994) supone que para todas las restricciones se establecen una familia de intervalos en los cuales se determina un coeficiente de penalización apropiado asignando para cada restricción un nivel de violación. Para mayores grados de violación se asigna un coeficiente mayor. La función de aptitud de un individuo está dada por la siguiente expresión.

$$F(x) = f(x) + \sum_{j=1}^m R_{ij} f_j^2(x) \quad (4.4)$$

Donde R_{ij} es el coeficiente de penalización en el nivel i de la restricción j (evaluada a través de la función f_j) y m es el número de restricciones al problema. El principal problema de esta propuesta es el exceso de parametrización que involucra.

En esta tesis se utiliza una función de penalización constante e igual al costo de conectar todos los enlaces posibles en la red. La expresión de la función de aptitud esta dada por

$$F(x) = f(x) + \sum_{j=1}^m c_i \quad (4.5)$$

Donde c_i es el costo del enlace i y m es la cantidad de enlaces posibles.

4. **Modificación de los operadores genéticos:** En este enfoque los operadores de mutación y cruzamiento se diseñan en forma *ad-hoc* para un problema particular con el fin de preservar la factibilidad de cada solución. El problema asociado con esta estrategia es que en algunos casos resulta muy complejo el diseño de los operadores de forma tal que se preserve la factibilidad de la solución. Entre los ejemplos de operadores genéticos *ad-hoc* se pueden encontrar los operadores de

cruzamiento para resolver el problema del viajante o TSP. Para una representación de rutas mediante vector de enteros, Goldberg y Lingle (1985) propusieron el cruzamiento PMX (Partially Mapped Crossover), Davis (1985) utilizó un cruzamiento llamado OX (Order Crossover) y Oliver (1987) cruzó individuos con CX (Cyclic Crossover). Todos ellos permiten la obtención de una ruta válida como descendencia de dos individuos candidatos a solución para el TSP.

4.6 Mecanismos de selección de padres

Como se puede apreciar en la figura 4.1, el algoritmo hace evolucionar sucesivas generaciones de individuos hasta obtener la solución óptima. La creación de una nueva generación a partir de los operadores genéticos es un punto muy importante e involucra al proceso de selección y los operadores de cruzamiento y mutación.

El proceso de selección intenta simular la selección natural de los individuos vivos y es el encargado de seleccionar los padres de la próxima generación (los que sobrevivirán) en función de su *fitness*. De esta forma, los mejores individuos pasarán a la próxima generación sus características genéticas. La clave de la selección es que los más aptos tengan mayor probabilidad de recombinarse y contribuir con sus hijos a la próxima generación.

Una parte fundamental del funcionamiento de un algoritmo genético es, sin lugar a dudas, el proceso de selección de candidatos a reproducirse. En un algoritmo genético este proceso de selección suele realizarse de forma probabilística (es decir, aún los individuos menos aptos tienen una cierta oportunidad de sobrevivir).

De forma ideal, la selección debería asegurar una amplia exploración del espacio de búsqueda al comienzo del AE (presión selectiva baja), mientras que en las etapas finales es deseable una explotación marcada de las regiones más prometedoras (presión selectiva alta). Este conflicto entre exploración y explotación ha dado origen a varios métodos de selección de individuos de una población para su supervivencia y reproducción.

Entre las técnicas de selección más usadas en algoritmos genéticos pueden

mencionarse las siguientes:

1. Métodos de Selección Proporcional
2. Método de Selección por Jerarquía (Ranking)
3. Método de Selección por Torneo

4.6.1 Métodos proporcionales

La selección proporcional es una estrategia donde la probabilidad de ser padres es proporcional al *fitness* de cada individuo. Hay dos estrategias propuestas para este tipo de selección.

4.6.1.1 Método de la Ruleta

Esta técnica de selección proporcional fue propuesta por De Jong (1975), y ha sido el método más comúnmente usado en los orígenes de los algoritmos genéticos. El algoritmo, simple aunque ineficiente, tiene una complejidad $O(n^2)$.

Una ruleta honesta, consta de ranuras en una de las cuales caerá la bolita. Eso asegura que todos los números tienen la misma probabilidad de ser elegidos. Pero si el ancho de cada ranura fuera diferente, los números que identifican a las ranuras más anchas tendrán mayor probabilidad de ser elegidos que los que pertenecen a ranuras estrechas. Entonces si lo que se busca es que la probabilidad de que un cromosoma sea elegido como padre sea proporcional a su *fitness*, el ancho de cada ranura será justamente proporcional a la medida de aptitud.

En esta particular ruleta el ancho de cada ranura es el aporte del *fitness* de un cromosoma i , (F_i) con relación a la sumatoria de los *fitness* de todos los individuos de la población (FT). Si la población tiene N individuos entonces

$$FT = \sum_{i=1}^N F_i \quad (4.6)$$

y la probabilidad de que un cromosoma i sea seleccionado estará dada por

$$P_i = \frac{F_i}{FT} \quad (4.7)$$

Entonces, la simulación del lanzamiento de la bolita se consigue generando un número aleatorio entre 0 y FT y utilizando los valores de *fitness* acumulados se determina en que ranura cae y por lo tanto cual individuo es seleccionado para procrear.

El inconveniente de la selección proporcional es que en las primeras generaciones del AE los valores de adaptabilidad promedio son pobres. La presencia de individuos con un valor de adaptabilidad bueno provoca que el mecanismo de selección proporcional asigne un número grande de copias a estos superindividuos provocando así la convergencia temprana.

4.6.1.2 Selección Universal Estocástica.

La selección universal estocástica, propuesta por Baker (1985) tiene como objetivo minimizar la mala distribución de los individuos en la población en función de los valores esperados. Se puede pensar como un peine de dientes equidistantes en el cual el primero de dichos dientes tiene un valor aleatorio, mientras que los demás quedan determinados por aquél. De esa manera se evita el copamiento rápido de un individuo con *fitness* muy alto en relación al resto de la población. El algoritmo se muestra en la figura 4.5.

```

Función de Selección Universal Estocástica
Entrada: población P(t)=(a1.. aN)
Salida: Población modificada por la selección
Prt=Rand() /*regresa un número aleatorio entre 0 y 1 */
for(sum=0,i=1;i<=n;i++)
    for(sum+=Valesp(i,t);sum>ptr,ptr++)
        seleccionar(i)

```

Figura 4.5 - Algoritmo de selección Universal Estocástica

La complejidad del algoritmo es $O(n)$. Dado sus características puede tener los siguientes problemas: puede aún ocasionar convergencia prematura y hacer que los individuos más aptos se multipliquen muy rápidamente. Por otra parte no resuelve el problema más serio de la selección proporcional, o sea, la imprecisión entre los valores esperados y los números de copias de cada individuo que son realmente seleccionados. (Coello Coello, 2003).

4.6.2 Método de Selección por Jerarquías

Este método fue propuesto por Baker (1985) para evitar la convergencia prematura en las técnicas de selección proporcional. El objetivo de esta técnica es disminuir la presión de selección. En este caso, se presenta la técnica de jerarquías lineales, pero es posible también usar jerarquías no lineales, aunque la presión de selección sufre cambios más abruptos al usarse esta última.

Los individuos se clasifican en base a su aptitud (*fitness*) en orden decreciente, y se le asigna una probabilidad de selección en función de su rango (o jerarquía) sin tener en cuenta su *fitness* original. El uso de jerarquías hace que no se requiera escalar la aptitud, puesto que las diferencias entre las aptitudes absolutas se diluyen. Asimismo, las jerarquías previenen la convergencia prematura.

En el ranking lineal, la probabilidad de selección asignada a los individuos es proporcional al rango k de los mismos: el mejor individuo tiene rango 1 y el peor, rango N . La probabilidad de que un individuo i de rango k sea seleccionado está dado por la ecuación 4.8.

$$p_s(i) = \frac{1}{N} \left[\eta_{\max} - 2(\eta_{\max} - 1) \frac{k-1}{N-1} \right] \quad (4.8)$$

El parámetro η_{\max} representa el número esperado de veces que el mejor individuo será seleccionado. Teniendo en cuenta que $\sum_{i=1}^N p_s(i) = 1$, puede demostrarse que $1 \leq \eta_{\max} \leq 2$. Por lo tanto la elección de un valor particular de η_{\max} sirve para controlar

la presión selectiva. Baker recomendó $\eta_{m\acute{a}x} = 1.1$.

4.6.2.1 Selección Mediante Torneo

Los métodos de selección proporcional antes descritos requieren de dos pasos a través de toda la población en cada generación: 1) Calcular la aptitud media y 2) Calcular el valor esperado de cada individuo. Por otra parte, el uso de jerarquías requiere que se ordene toda la población (una operación cuyo costo puede volverse significativo en poblaciones grandes).

Una alternativa a los métodos anteriores es la selección por torneo (Bickle y co., 1996), donde un individuo necesita ganar una competencia entre un conjunto de individuos seleccionados aleatoriamente. En un k -torneo el mejor de k individuos es seleccionado en forma determinística, aunque ocasionalmente puede ser utilizada una selección estocástica (Bickle y co., 1996). La figura 4.6 muestra el algoritmo de selección por torneo.

```
* Función de Selección por torneo (k individuos)
  Entrada: población P(t)=(a1.. aN) (k∈{1.. N })
  Salida: Población modificada por la selección
  para i=1 to N hacer
    ai' ← mejor entre k individuos de P(t)
    seleccionados aleatoriamente .
  fin
  retornar(P'(t))
```

Figura 4.6 - Algoritmo de selección por torneo

La selección mediante torneo es similar a la de jerarquías en términos de la presión de selección, pero es computacionalmente más adecuada para implementarse en paralelo. Esta técnica fue propuesta por Wetzel (1989), la idea básica del método es seleccionar p candidatos de la población que competirán entre sí sobre la base de sus valores de aptitud. Con una probabilidad determinada de antemano (que puede ser 1, en cuyo caso se dice que el torneo es determinístico), el ganador de la competencia es seleccionado como padre.

La complejidad del algoritmo es $O(n)$. Es una técnica eficiente y fácil de implementar, ya que no requiere escalamiento de la función de aptitud (usa comparaciones directas).

Puede introducir una presión de selección muy alta porque a los individuos menos aptos no se les da oportunidad de sobrevivir. No tiene sentido aplicar jerarquías lineales a la selección mediante torneo porque la selección mediante torneo realiza comparaciones directas entre individuos y el uso de jerarquías lineales no cambiaría en nada la presión de selección.

4.7 Selección local

En la selección local cada individuo reside dentro de un entorno restringido, llamado vecindad. Los individuos interactúan sólo con otros dentro de esta región (en los otros métodos de selección la vecindad es la población completa).

La vecindad queda definida por la estructura de la distribución de la población, y puede verse como un grupo de potenciales compañeros de apareamiento de un individuo dado. El primer paso consiste en la selección de la primer mitad del conjunto de apareamiento en forma aleatoria o utilizando alguno de los métodos mencionados anteriormente. Luego se define una vecindad local para cada individuo seleccionado. Dentro de esta vecindad se selecciona el compañero de apareamiento mediante algún criterio.

La estructura de la vecindad puede ser lineal, bi o tridimensional y a su vez puede ser simétrica o asimétrica, con una distancia que está definida por la cantidad de individuos hacia uno u otro lado. Esta estructura crea una “distancia de aislamiento” entre individuos y la propagación de información se realiza por solapamiento de las vecindades.

El tamaño de la vecindad determina la velocidad de propagación de la información entre los individuos de la población. Por lo tanto éste debe ser elegido de manera tal de balancear la velocidad de convergencia y el mantenimiento de diversidad de los individuos de la población. Lo que se desea es mantener una alta diversidad para

evitar convergencias prematuras a un mínimo local. En general una selección local en una vecindad pequeña tiene mejor comportamiento que en una vecindad de gran tamaño.

4.8 Operadores Genéticos

Una vez seleccionado el conjunto de padres que generarán descendencia es necesario continuar con una fase importante en el proceso de evolución en la cual los individuos seleccionados intercambian material genético o alteran algunos de sus genes. Esta función es llevada a cabo por los operadores genéticos cruzamiento y mutación los cuales se describen a continuación.

4.8.1 Operador de cruzamiento

Los individuos de una población $P(t)$ deben generar descendencia transmitiendo sus características genéticas a sus hijos. Para ello deberá contarse con operadores que permitan dicha transformación y que necesariamente estarán influidos por el tipo de representación y de problema a resolver. Dos operadores utilizados en algoritmos genéticos son los operadores de cruzamiento y mutación. En forma concisa puede considerarse que el cruzamiento permite generar hijos a partir de padres seleccionados.

Formalmente el cruzamiento podría definirse como una función

$$C: S^q \rightarrow S^r \quad C(x_1^t, \dots, x_q^t) \rightarrow C(x_1^t, \dots, x_r^t)$$

Donde:

q es la cantidad de padres

r es la cantidad de hijos

x_i^t es el cromosoma i de la población P^t antes del cruzamiento

x_i^t es el cromosoma i de la población P^t después del cruzamiento

El cruzamiento o recombinación produce nuevos individuos adoptando la información contenida en dos o más padres. Esto se hace combinando los valores variables de los padres. Dependiendo de la forma de representación de las variables se deben utilizar diferentes métodos.

Hay diversos métodos de recombinación, que dependen del tipo de variable utilizado. A continuación se describen únicamente los métodos de cruzamiento para representación binaria, por ser estos de interés para este estudio. Durante la recombinación de variables binarias se intercambian sólo partes del individuo y dependiendo del número de partes de intercambio se definen las técnicas de cruzamiento.

Hay tres técnicas básicas de cruzamiento, comúnmente utilizadas con representación de cromosomas mediante cadenas binarias: Cruzamiento de un punto, Cruzamiento de dos o más puntos y Cruzamiento uniforme

4.8.1.1 Cruzamiento de un punto

Esta técnica fue propuesta por Holland (1975), y fue muy popular durante muchos años. Hoy, sin embargo, no se utiliza mucho en la práctica debido a sus inconvenientes, los cuales se analizan en esta sección.

En esta técnica se selecciona aleatoriamente un punto de cruce.

$$k \in [1, 2, \dots, N], \quad N = \text{número de variables de un individuo}$$

Luego las variables se intercambian entre los individuos a partir de este punto, de lo cual se obtienen dos descendientes. La figura 4.8 muestra el proceso.



Figura 4.8 - Cruzamiento con un punto

Como ejemplo se puede considerar dos individuos de 11 variables binarias cada uno:

individuo 1 0 1 1 1 0 0 1 1 0 1 0

individuo 2 1 0 1 0 1 1 0 0 1 0 1

El punto elegido de cruzamiento es 5, entonces los hijos generados son:

descendiente 1: 0 1 1 1 0 | 1 0 0 1 0 1

descendiente 2: 1 0 1 0 1 | 0 1 1 0 1 0

Puede demostrarse, por ejemplo, que hay varios esquemas que no pueden formarse bajo esta técnica de cruzamiento. Michalewicz define un esquema como la representación de todos los *strings* (un hiperplano o un subconjunto del espacio de búsqueda) que se ajusta con todas las otras posiciones distintas de “*”. Es decir, es una plantilla que permite la exploración de las similitudes entre los cromosomas, esto se construye introduciendo el símbolo “*” en representación de un gen en la cadena de bits de los individuos. Así por ejemplo para individuos que tienen cadenas binarias de longitud 3, el esquema (***) representaría a todos los individuos de la población. De igual forma el esquema (10*) representaría al individuo (100) y al (101). De esta manera, lo que procesa un AG son los esquemas y la población inicial evoluciona generación tras generación, donde los esquemas más aptos irán sobreviviendo, mientras que los menos aptos irán desapareciendo.

Por ejemplo, en los esquemas siguientes:

H1 = 11*****1

H2 = *****11**

Si se aplica cruzamiento de un punto a estos 2 esquemas, no hay manera de formar una instancia del esquema:

$$H = 11^{**}11^*1$$

El cruzamiento de un punto destruye esquemas en los que la longitud de definición es alta. Esto produce lo que se denomina sesgo posicional: los esquemas que pueden crearse o destruirse por el cruzamiento dependen fuertemente de la localización de los bits en el cromosoma (Eshelman, 1989). El problema fundamental del cruzamiento de un punto es que presupone que los bloques constructores son esquemas cortos y de bajo orden, y cuando esto no sucede (p. e., con cadenas largas), puede proporcionar resultados no apropiados.

Obviamente, las aplicaciones del mundo real requieren cadenas largas lo que hace que este cruzamiento no se use comúnmente en dichos problemas. El cruzamiento de un punto trata también preferencialmente algunas posiciones del cromosoma, como por ejemplo los extremos de una cadena.

El cruzamiento de un punto suele preservar también los *hitchhikers*, que son bits que no forman parte del esquema deseado, pero que debido a su similitud con ellos gozan de los beneficios del cruzamiento.

4.8.1.2 Cruzamiento de dos o más puntos

En esta técnica se seleccionan dos puntos de cruce en forma aleatoria y los intercambios de las variables suceden alrededor de esos dos puntos y se obtienen también dos nacimientos.

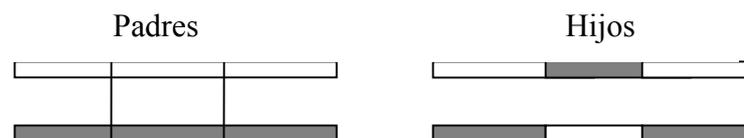


Figura 4.9 - Cruzamiento con dos puntos

De Jong (1975) fue el primero en implementar un cruzamiento de puntos, como una extensión del cruzamiento de un punto.

El cruzamiento de dos puntos reduce los efectos disruptivos (o destructivos) del cruzamiento razón por la cual es usado con frecuencia. En este sentido, no existe consenso en torno al uso de cruzamiento de más puntos de cruce. Los estudios empíricos al respecto proporcionan resultados que no resultan concluyentes respecto a las ventajas o desventajas de usar dichos valores. En general, sin embargo, es aceptado que el cruzamiento de dos puntos es mejor que el cruzamiento de un punto.

4.8.1.3 Cruzamiento uniforme

El cruzamiento de uno o dos puntos define los puntos de cruce donde cada individuo puede ser cortado. El cruzamiento uniforme generaliza este esquema haciendo cada "locus" un potencial punto de cruce. En la figura 4.10 puede verse el proceso.



Figura 4.10 - Cruzamiento uniforme

Esto puede lograrse mediante una expresión lógica, generando una máscara en forma aleatoria del mismo largo que los cromosomas y procediendo de la siguiente manera: si el bit j de la máscara es 1 en el hijo se copia el bit del padre en uno de los hijos y el de la madre en el otro hijo. En caso contrario la operación se invierte.

$$\mathbf{H} = \mathbf{P}\mathbf{M}_k + \overline{\mathbf{M}\mathbf{M}_k}$$

donde:

\mathbf{H} es el descendiente

\mathbf{P} es el padre

\mathbf{M} es la madre

\mathbf{M}_k es la máscara

Como ejemplo se puede considerar dos individuos de 11 variables binarias cada uno:

$$\mathbf{P}=[0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0]$$

$$\mathbf{M}=[1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1]$$

Las variables del padre que contribuye con su variable para el hijo se eligen aleatoriamente con igual probabilidad. Entonces el descendiente 1 se produce tomando el bit del padre 1 si el bit de la máscara correspondiente es 1 o el bit del padre 2 si el bit de la máscara es 0, el descendiente 2 se crea usando la máscara inversa. Siguiendo con el ejemplo, si la máscara esta dada por:

$$\mathbf{M}_k=[0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0]$$

Después del cruce se los nuevos individuos son:

$$\mathbf{H}_1=[1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

$$\mathbf{H}_2=[0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

El cruzamiento uniforme y el cruzamiento multipunto han sido desarrollados para reducir el sesgo asociado con la longitud de la representación binaria usada y la codificación particular para un conjunto de parámetros dado.

4.8.1.4 Sesgos del cruzamiento

El “sesgo” del cruzamiento se refiere a las tendencias de este operador a favorecer o no un cierto tipo de búsqueda. Realmente, la búsqueda aleatoria es la única que no presenta ningún tipo de sesgo, por lo que es de esperarse que cualquier operador heurístico como el cruzamiento sí lo tenga presente.

De hecho, se ha determinado que se requiere de algún tipo de sesgo para que una técnica de búsqueda sea efectiva (Mitchell, 1980). En algoritmos genéticos, se suelen considerar 2 tipos de sesgo para el cruzamiento: Distribucional y Posicional.

El sesgo distribucional se refiere al número de símbolos transmitidos durante una recombinación. Asimismo, se refiere a la medida en la que algunas cantidades tienen más tendencia a ocurrir que otras. El sesgo distribucional es importante porque está correlacionado con el número potencial de esquemas de cada padre que pueden ser recombinados por el operador de cruzamiento.

El cruzamiento de un punto y el de dos puntos no tienen sesgo distribucional, para más de dos puntos si lo tiene pero moderado, por otra parte, un cruzamiento uniforme tiene un sesgo distribucional muy fuerte.

El sesgo posicional caracteriza en qué medida la probabilidad de que un conjunto de símbolos se transmitan intactos durante la recombinación dependiendo de las posiciones relativas de los mismos en el cromosoma.

El sesgo posicional es importante porque muestra los esquemas que más probablemente se hereden de padres a hijos. Asimismo, también indica la medida en la que estos esquemas aparecerán en nuevos contextos. En este caso es posible decir que el cruzamiento de un punto tiene un fuerte sesgo posicional, además, todo parece indicar, que el cruzamiento de 2 puntos tiene también un sesgo posicional fuerte y, finalmente, el cruzamiento uniforme no tiene sesgo posicional.

4.8.1.5 Variantes del cruzamiento

Es posible modificar diversos aspectos del cruzamiento para mejorar su desempeño. Una variante, por ejemplo, consiste en retener sólo a uno de los dos hijos producidos por un cruzamiento. Holland (1975) describió una técnica de este tipo.

Estudios empíricos han mostrado, además, que retener a los 2 hijos producidos por un cruzamiento reduce sustancialmente la pérdida de diversidad en la población (Lashon, 1982).

Otra variante muy común es la de restringir los puntos de cruzamiento a aquellas posiciones en las que los padres difieran. A esta técnica se le conoce como sustitución reducida (Lashon, 1987), y su objetivo es mejorar la capacidad de el cruzamiento para

producir hijos que sean distintos a sus padres.

Una alternativa interesante es la llamada cruzamiento con mezcla (Eshelman, 1989). En este caso, se aplica un operador de permutación a una parte de las cadenas de los padres antes de efectuar el cruzamiento. Después del cruzamiento, se aplica la permutación inversa a fin de restaurar el orden original de los bits.

El cruzamiento con mezcla tiene como objeto contrarrestar la tendencia del cruzamiento de puntos ($n \geq 1$) a causar con más frecuencia disrupción en los conjuntos de bits que están dispersos que en los que están juntos.

4.8.1.6 Comportamiento Deseable del cruzamiento

Todos los operadores descritos anteriormente, siguen el principio Mendeliano de la herencia: cada gen de un hijo, es una copia de un gen heredado de alguno de sus padres.

Algunos investigadores han destacado que el énfasis del cruzamiento debe ser el poder generar todas las posibles combinaciones de bits (de longitud L) que hayan en el espacio de búsqueda del problema (Radcliffe, 1991). Dada una cierta representación binaria, ni el cruzamiento de un punto, ni el de $n < L$ puntos son capaces de lograr esto. El cruzamiento uniforme, sin embargo, sí puede hacerlo.

Algunos investigadores han propuesto otras variantes del cruzamiento motivados por este problema. Radcliffe propuso una técnica denominada recombinación respetuosa aleatoria. Según esta técnica, se genera un hijo copiando los bits en los que sus padres son idénticos, y eligiendo luego, valores al azar para llenar las posiciones restantes.

4.8.2 Mutación

El operador de mutación también dependerá, como ya se expresó, de la representación de los cromosomas. En el caso de representación por cadenas de bits, la mutación simplemente implica alterar bits de un cromosoma con cierta probabilidad.

Esta operación implica decidir, en cada posición de bit y acorde con la probabilidad de mutación, si el bit debe mutar o no.

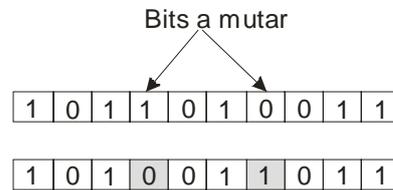


Figura 4.11 - Operador de mutación

La mutación es un operador que permite la exploración de otras regiones del espacio de búsqueda y evita que el AE quede atrapado en un óptimo local. No obstante, aplicar una alta probabilidad de mutación al algoritmo puede traer como consecuencia transformarlo en uno de búsqueda aleatoria por cuanto es factible destruir buenas soluciones (que quizás produzcan como descendencia, individuos aún mejores). De hecho una probabilidad de mutación igual a la unidad implica una búsqueda totalmente aleatoria. Por esta razón, en general, se utilizan bajas probabilidades de mutación (del orden de 10^{-3} a 10^{-2}). Algunos investigadores, sin embargo, han sugerido que el usar porcentajes altos de mutación al inicio de la búsqueda, y luego decrementarlos exponencialmente, favorece el desempeño de un AG (Fogarty, 1989).

Otros autores sugieren que $p_m = \frac{1}{L}$ (donde L es la longitud de la cadena cromosómica) es un límite inferior para el porcentaje óptimo de mutación (Bäck, 1993)

Finalmente se menciona la mutación heurística, propuesta por Gen y Cheng (Gen, 2000) que realiza un procedimiento más elaborado que es similar a algunos algoritmos básicos de búsqueda tabú (Glover, 2002): se selecciona un conjunto de genes en forma aleatoria a partir de los cuales se generan todas las permutaciones posibles de dicho conjunto (cada una de estas permutaciones son soluciones vecinas a la primera). Finalmente se evalúan todos los vecinos y se toma el mejor.

4.8.3 Cruzamiento vs. Mutación

El cruzamiento uniforme es más “explorativo” que el cruzamiento de un punto. Si, por ejemplo se consideran los cromosomas:

$$P1 = 1 * * * * 1$$

y

$$P2 = 0 * * * * 0$$

en los cuales los asteriscos simbolizan cualquier valor para el gen, mientras que el primero y el último tienen los valores asignados y fijos, el cruzamiento uniforme producirá individuos del esquema *******, mientras que el cruzamiento de un punto producirá individuos de los esquemas *1****0* y *0****1*.

Por otra parte si se analiza el poder exploratorio de la mutación, es posible distinguir tres situaciones:

- Si el porcentaje de mutación es cero, no hay alteración alguna.
- Si es uno, la mutación crea siempre complementos del individuo original.
- Si es 0.5, hay una alta probabilidad de alterar fuertemente el esquema de un individuo.

En otras palabras, es posible controlar el poder de alteración de la mutación y su capacidad de exploración puede hacerse equivalente a la del cruzamiento. El tipo de exploración efectuada por la mutación es, sin embargo, diferente a la del cruzamiento.

Por ejemplo, dados:

$$P1 = 10****$$

$$P2 = 11****$$

El cruzamiento producirá sólo individuos del esquema *1******.

El primer “1” en el esquema está garantizado (sin importar qué tipo de cruzamiento se use), porque es común en los esquemas de ambos padres. La mutación, sin embargo, no respeta necesariamente este valor.

El cruzamiento “preserva” los alelos (se denomina así a cada lugar del *string* que representa un cromosoma, por ejemplo un cromosoma de longitud 10 tiene 10 alelos) que son comunes en los 2 padres. Esta preservación limita el tipo de exploración que el cruzamiento puede realizar. Esta limitación se agudiza conforme la población pierde diversidad, puesto que el número de alelos comunes se incrementaría.

Cuando se busca localizar el óptimo global de un problema, la mutación puede ser más útil que el cruzamiento. Si lo que interesa es ganancia acumulada (el objetivo original del AG), el cruzamiento es entonces preferible.

El cruzamiento parece trabajar bien con funciones que están altamente correlacionadas o tienen epístasis moderada¹.

4.9 Elitismo

Se ha explicado que a partir de una generación dada se obtiene la siguiente por medio de los operadores descriptos. Sin embargo, los mejores individuos, si bien son los padres de los que forman la próxima generación, pueden quedar excluidos. El elitismo tiende a evitar dicha situación. Especificando la cantidad de individuos de elite, se puede permitir el paso de algunos de ellos a la siguiente generación. De esta manera se asegura que la siguiente generación tendrá siempre al menos un individuo de igual o mejor ajuste que en la anterior. El procedimiento es copiar, antes de comenzar a aplicar los operadores genéticos, los mejores k individuos de una población a la siguiente. k es un parámetro del AE.

4.10 Modelos de población

Un modelo de población puede ser distinguido de otro observando la estrategia de selección de padres y la definición del conjunto de apareamiento. Se pueden definir tres modelos de población.

¹ "El problema de la epístasis" consiste en la existencia de estas interacciones entre los genes de forma que el efecto de unos genes inhibe o potencia a otros

Modelo global o panmítico. En este modelo la selección tiene lugar en la población completa. Esto significa que cualquier individuo de la población es candidato y puede ser seleccionado para la producción de la descendencia.

Modelo regional o de islas. Este modelo obliga a que la selección de padres se realice en partes de la población aisladas unas de otras, las cuales son denominadas subpoblaciones. Dentro de cada subpoblación la selección es irrestricta (similar al modelo global). La interacción entre islas se realiza mediante migración de individuos.

Modelo de vecindades. En este modelo la selección de padres se realiza en una vecindad local, cuyo tamaño esta dada por la cantidad de vecinos de cada individuo. La interacción entre vecindades se garantiza a través de la pertenencia de cada individuo a más de una vecindad.

4.10.1 Modelo Global (Panmítico)

El modelo global no subdivide la población sino que la considera en forma completa. Durante el proceso de selección, un individuo seleccionado tiene la posibilidad de aparearse con cualquier otro individuo de la población. Aunque estos esquemas, tales como selección estocástica con reemplazo, son ampliamente utilizados, no tienen buen desempeño en algunos problemas con búsquedas de genotipos “exitosos” en paisajes de gran tamaño y/o multimodales (Collins, 1992). Sucede que la población es incapaz de mantener genotipos radicalmente diferentes con buen *fitness* debido al problema de convergencia prematura.

4.10.2 Modelo regional

El modelo regional (o modelo de migración) divide la población en múltiples subpoblaciones. Estas se desarrollan durante varias generaciones independientemente de las demás (tiempo de aislamiento). Después de este tiempo, algunos de los individuos se distribuyen entre las subpoblaciones (migración). La cantidad de intercambios de individuos (tasa de migración) y el método de selección de los individuos que migrarán determinarán cuanto puede variar la diversidad genética en la subpoblación y el intercambio de información entre subpoblaciones.

Es posible la implementación de algoritmos paralelos aplicados sobre un modelo regional de población. Más aún, en algunos casos, para una computadora de una única CPU, la implementación de un algoritmo paralelo en forma serial (pseudo paralelo) puede obtener mejores resultados (el algoritmo encuentra el óptimo global con menor cantidad de evaluaciones de la función de aptitud) que utilizando el modelo panmítico.

En cuanto a los criterios que pueden adoptarse para seleccionar los candidatos a ser migrados, puede utilizarse una selección aleatoria de los mismos o bien hacer uso de una función que dependa del *fitness* para favorecer la migración de los mejores individuos. Existen muchas posibilidades para la estructura de migración de individuos entre subpoblaciones. A continuación se esbozan algunas alternativas:

- Entre todas las subpoblaciones (formando una red con topología de malla completa, sin restricciones). En este caso cualquier subpoblación puede migrar individuos a cualquier otra. La figura 4.12 muestra el esquema de este tipo de migración.

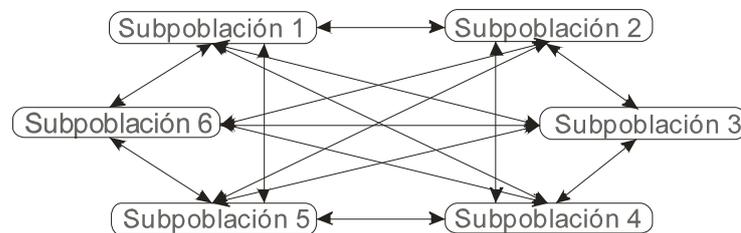


Figura 4.12 - Topología de migración irrestricta

- Topología en anillo. Las subpoblaciones se organizan formando un anillo, como se muestra en la figura 4.13. Los individuos migran desde cada subpoblación a la siguiente. Por ejemplo, en el esquema más básico de migración en anillo, en un conjunto de 6 subpoblaciones, los individuos de la subpoblación 6 migran sólo a la subpoblación 1, los de la subpoblación 1 sólo lo hacen a la 2 y así sucesivamente.

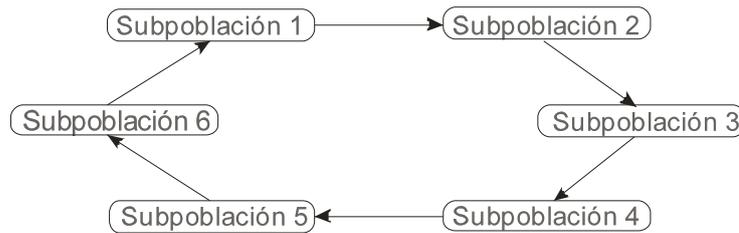


Figura 4.13 - Topología de migración en anillo – distancia 1

- Topología de vecindad. Es una estrategia similar a la topología en anillo. Las subpoblaciones son organizadas en grupos llamados vecindades. La migración está permitida sólo entre subpoblaciones vecinas aunque en este caso la migración puede ocurrir en cualquier dirección. Para cada subpoblación el posible inmigrante está determinado de acuerdo al método de selección elegido desde una subpoblación adyacente. Un ejemplo se muestra en la figura 4.14.

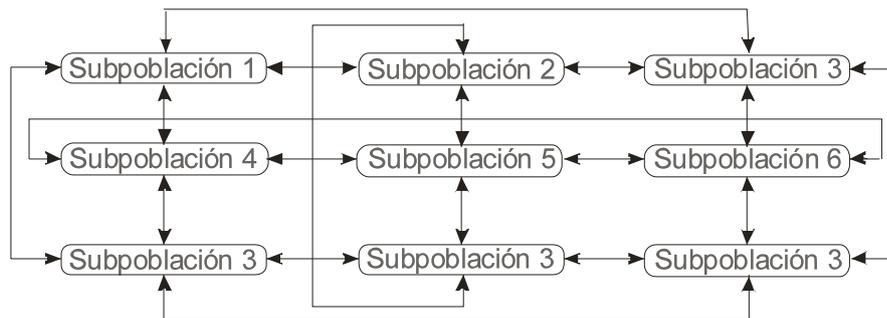


Figura 4.14 - Topología de migración entre vecindades
(en este caso: grilla de 2 dimensiones)

Los resultados reportados por Lohman (1991) indican que los algoritmos evolutivos con multipoblación obtienen mejores resultados que con una población única.

4.10.3 Modelo de difusión local

Este modelo maneja cada individuo separadamente y selecciona su compañero de apareamiento en una vecindad por selección local, En consecuencia, con este proceso la información se difunde a través de la población durante la búsqueda gracias a la pertenencia de un individuo a más de una vecindad.

Para trabajar con poblaciones estructuradas según un modelo de difusión local se utiliza la técnica denominada selección local. En esta técnica, cada individuo reside en un ambiente reducido o vecindad, y éste interactúa sólo con individuos dentro de esa región. Las vecindades están definidas por la estructura en la cual la población está distribuida y puede verse como grupos de potenciales compañeros de apareamiento.

El conjunto de apareamiento se conforma de la siguiente manera: a) Utilizando algún algoritmo de selección se incorporan los “padres” al *pool*. Ahora la vecindad está definida en el entorno de cada padre seleccionado. b) El compañero de apareamiento (“madre”) se selecciona dentro de cada vecindad con algún criterio predefinido (por ejemplo el mejor individuo dentro de la vecindad, con una probabilidad proporcional a su función de aptitud, etc.).

La estructura de la vecindad puede ser:

- Lineal
- Bidimensional
- Tridimensional
- Combinada: cualquier combinación de las estructuras anteriores.

La distancia entre los posibles vecinos junto con la estructura determina el tamaño de la vecindad, el cual a su vez determina la velocidad de propagación de información entre los individuos de la población. Un tamaño de vecindad grande significa un incremento en la velocidad de propagación de la información y una disminución de la diversidad. De hecho, si el tamaño de la vecindad coincide con el tamaño de la población se está en presencia del modelo panmítico.

4.10.4 Reinserción

Una vez que los descendientes han sido producidos por selección, cruzamiento y mutación de individuos de la población original, es posible determinar el *fitness* correspondiente. Para ello se debe tener en cuenta las siguientes consideraciones: Si se

han generado menos descendientes que el tamaño de la población original, éstos deben ser reinsertados de tal forma que este último valor no se altere. Por otra parte, si no se utilizan todos los descendientes de la nueva generación, o se han generados más hijos que el tamaño de la población original, es necesario utilizar un esquema de re inserción.

El esquema de re inserción queda determinado por el tipo de selección utilizado. Si la selección es local entonces es necesario aplicar un método de re inserción igualmente local. Para los otros tipos de selección la re inserción de descendientes sigue un esquema llamado re inserción global. A continuación se realiza una descripción de ambos esquemas de re inserción.

4.10.4.1 Re inserción Global

Existen diferentes variantes de re inserción global. Estas pueden ser:

- Producir tantos descendientes como cantidad de padres, reemplazándolos a todos estos por sus hijos (re inserción pura)
- Producir menos descendientes que padres, reemplazándolos aleatoriamente (re inserción uniforme).
- Producir menos descendientes que padres, reemplazando los peores de estos últimos (re inserción elitista).
- Producir más descendientes que los necesarios y reinsertar los mejores (re inserción basada en *fitness*).

La re inserción pura es el esquema de re inserción más simple, cada individuo vive sólo una generación. Este algoritmo se utiliza en los AGs simples, pero tiene la desventaja de que un muy buen individuo puede ser reemplazado sin dejar descendencia y su información se pierda.

4.10.4.2 Reinserción Local

Para la selección de padres a ser reemplazados y de descendiente a reinsertarse se pueden seguir las siguientes variantes:

- Insertar todos los descendientes y reemplazar los individuos en la vecindad en forma aleatoria.
- Insertar todos los descendientes y reemplazar los individuos más débiles de la vecindad.
- Insertar un descendiente que sea más apto que el individuo más débil en la vecindad, eliminando a este último.
- Insertar un descendiente que sea más apto que el individuo más débil en la vecindad, y eliminar aleatoriamente a uno de los padres.
- Insertar la descendencia en el lugar de los padres sólo si aquella es más apta que estos.

El elitismo combinado con reinserción basado en *fitness* es el método que se recomienda en Pohlheim (1999). A su vez se pueden diferenciar dos esquemas de selección: en uno de ellos, en cada generación se reemplaza una cierta cantidad de padres (los menos adecuados) por los descendientes más aptos. Este esquema implementa una selección entre descendientes antes de insertarlos en la población (es decir antes que puedan participar en el proceso de reproducción). Por un lado, el mejor individuo puede vivir por muchas generaciones, pero por otra parte no hay un control acerca de que los padres sean reemplazados por mejores o peores hijos. El otro esquema es realizar una competencia entre los padres y los descendientes para generar una población con los más aptos de los dos conjuntos. Este último es el esquema de reinserción utilizado en esta tesis compitiendo padres e hijos dentro de una vecindad.

4.11 Parámetros de los Algoritmos Evolutivos

Para que un AE pueda funcionar, es necesario definir un conjunto de valores

llamados parámetros. Los más importantes son: el tamaño de la población, el número máximo de iteraciones, la longitud de los individuos, la probabilidad de cruzamiento, la probabilidad de mutación, más los que son característicos de cada método en particular, como por ejemplo el coeficiente de la función de penalización, el tamaño de la vecindad en poblaciones estructuradas, etc.

Si bien no existen criterios definitivos para la selección de los valores a asignar al conjunto de parámetros, pueden encontrarse en la literatura ciertas recomendaciones de carácter general y otras específicas asociadas con la probabilidad de cruzamiento (p_c) y la probabilidad de mutación (p_m).

La probabilidad de cruzamiento se define como la relación entre el número esperado de hijos producidos en cada generación y el tamaño de la población; por lo tanto controla el número promedio de cromosomas que serán sometidos al operador de cruzamiento. Se han reportado valores de $p_c = 0.6$ y valores de $p_c \in [0.75 - 0.9]$ (Bäck, 1997).

La probabilidad de mutación está relacionada con el número esperado de bits a mutar en cada generación y en contraposición al cruzamiento, la mutación es introducida con una probabilidad de ocurrencia no superior a 0.05; aunque también se han reportado valores de $p_m = 1/l$ donde l corresponde a la longitud del cromosoma binario que representa la solución (Bäck, 1997).

El tamaño de la población (N) constituye un parámetro de gran importancia en el diseño de un AE. Puede decirse que en general valores pequeños de N conducen a una convergencia temprana del algoritmo, en tanto que valores muy grandes exigen un esfuerzo computacional alto sin lograr un mejor desempeño.

El número máximo de generaciones T , es un criterio de detención posible del algoritmo y los valores utilizados en la literatura varían en un rango muy amplio. Otras condiciones de terminación utilizadas son por ejemplo: 1) medir el progreso realizado por el algoritmo en un número predefinido de generaciones y detener la búsqueda cuando tal progreso es menor que algún valor dado; 2) correr el algoritmo hasta que la

población se haya estabilizado, es decir, cuando todos o la mayoría de los individuos tengan la misma aptitud (Michalewicz, 1996), etc.

4.12 Conclusiones

En este capítulo se han desarrollado algunos tópicos importantes que involucran los algoritmos evolutivos, se destacaron sus características y ventajas sobre otros métodos para la resolución de problemas como el planteado en esta tesis. Estos algoritmos tienen las ventajas de su robustez, su tiempo de ejecución acotado, la posibilidad de explorar varias regiones del espacio de búsqueda al mismo tiempo y una paralelización intrínseca. Como desventaja de los AE estriba en el hecho de no garantizar la obtención de un óptimo global. Sin embargo los problemas combinatorios, cuando se consideran instancias de tamaño considerable, tiene asociados espacios de búsquedas muy grandes donde los métodos determinísticos colapsan y los AE permiten encarar estos espacios de búsqueda completos constituido por todas las posibles combinaciones topológicas desde un enfoque exhaustivo, mucho más apropiado para abordar este tipo de problemas, lo cual justifica la utilización de los AE a pesar de la desventaja mencionada.

El mismo problema está relacionado con la convergencia prematura del algoritmo a un óptimo local. Se han realizado muchas propuestas para mantener un equilibrio entre la exploración y la explotación del espacio de búsqueda, proponiéndose distintos métodos de selección, estrategias de diversificación y estructuras de población para favorecer el equilibrio mencionado.

Los ingredientes de un AE son: la representación de los cromosomas, las estrategias de selección de individuos y los operadores genéticos como el cruzamiento y la mutación. Estos ingredientes pueden adaptarse al problema que se pretende resolver para lograr así mejores resultados.

CAPITULO 5

5 Diseño de un Algoritmo Evolutivo para la Optimización de Redes Confiables

5.1 Introducción

En los capítulos anteriores se desarrollaron conceptos relativos tanto a problemas de diseño de redes en general como a técnicas heurísticas haciendo hincapié en algoritmos evolutivos. En este capítulo se presenta el diseño de un AE para resolver un problema de diseño de redes en particular: obtener la topología de una red de mínimo costo sujeto a restricciones sobre la confiabilidad.

Se formula el problema completo, se presentan los criterios de diseño de cada una de las partes que componen un AE y se muestran los resultados del mismo sobre una red ejemplo tomada de la literatura. Sobre esta base, en el próximo capítulo se muestran los resultados comparativos para una veintena de instancias del problema.

El AE implementado es de tipo híbrido, con población inicial sembrada, y con incorporación de conocimiento específico del problema lo que permite una exploración más inteligente del espacio de búsqueda. Se ha considerado, además, un modelo de población estructurada, aprovechando las ventajas del mismo sobre el modelo de población panmítico.

5.2 Formulación del problema

Se considera un conjunto de nodos, su ubicación y un posible conjunto de enlaces bidireccionales que conectan los nodos. Se supone que los nodos son perfectamente confiables y los enlaces tienen dos estados posibles: operacional o no operacional (cortado o fuera de servicio); además, si fallan, lo hacen en forma

independiente de los demás y no se consideran reparaciones. El problema a resolver es encontrar el esquema de interconexión de los nodos, geográficamente distribuidos, mediante una topología de red de mínimo costo con una confiabilidad igual o superior a una especificada.

Cada enlace posible tiene una confiabilidad conocida y un costo asociado al mismo.

Este problema, tal como se mencionó en capítulos anteriores, es un problema de optimización combinatoria, donde el espacio de búsqueda Ω , para una red completamente conectada con n nodos y k posibles estados para cada uno de los enlaces, tiene una dimensión dada por:

$$|\Omega| = k^{\frac{n \cdot (n-1)}{2}} \quad (5.1)$$

En los casos que se tratan en este trabajo son posibles dos estados: 0 para un enlace no presente y 1 para un enlace presente entre dos nodos, i y j , cualesquiera. Si además se considera que los enlaces no tienen redundancia, entonces k es igual a 2. Por ejemplo, una red de 10 nodos ($n=10$) tiene $3.5 \cdot 10^{13}$ diseños posibles conformando el espacio de búsqueda total.

La red se modela a través de un grafo probabilístico $G = (V, E, p_e)$ donde V es el conjunto de vértices del grafo que representan los nodos de la red, E el conjunto de enlaces que unen a los nodos y p_e es la confiabilidad de los enlaces (si q_e es la no confiabilidad o probabilidad de falla, para todos los enlaces se cumple que $p_e + q_e = 1$). La confiabilidad se estima utilizando el método de reducción recursiva de la varianza (RVR).

El problema de optimización planteado puede formularse como sigue (Dengiz, 1997):

$$\begin{aligned} \min Z &= \sum_{i=1}^{N-1} \sum_{j=i+1}^N c_{ij} x_{ij} \\ s.a. \\ R(x) &\geq R_0 \end{aligned} \quad (5.2)$$

Donde N representa la cantidad de nodos, c_{ij} el costo del enlace entre los nodos i y j , $R(x)$ la confiabilidad de la red, R_0 la confiabilidad admisible y $x_{ij} = \{0,1\}$ es la variable de decisión.

En el capítulo 2 se presentó una formulación del problema general sobre la base de una representación vectorial para una red determinada la cual, en realidad, es equivalente a la presentada en la ecuación 5.2. Puede observarse que en la representación matricial para costos y enlaces, sólo se exploran los elementos por encima de la diagonal principal de la matriz. La ubicación de dichos elementos dentro de un vector es directa, como se mostró en el capítulo 2 y la ecuación 5.2 puede escribirse como sigue:

$$\begin{aligned} \min Z &= \sum_{i=1}^{N_e} c_i x_i \\ s.a. \\ R(x) &\geq R_0 \end{aligned} \quad (5.3)$$

Donde N_e es la cantidad de enlaces posibles entre los nodos. \mathbf{c} es el vector de costos donde cada elemento c_i es el costo del i -ésimo enlace, \mathbf{x} es el vector de enlaces en el cual cada componente x_i toma un valor 1 si el enlace correspondiente existe y 0 en caso contrario.

Esta formulación es elegida por cuanto resulta en una representación de los cromosomas más eficiente al momento de diseñar el AE.

5.3 Enfoque evolutivo

Para resolver el problema planteado se utiliza un algoritmo evolutivo enriquecido con conocimiento específico del problema. La fase de este problema que

insume la mayor cantidad de tiempo de cómputo es la evaluación de la función de aptitud, por cuanto es en dicha fase donde se realiza el cálculo de la confiabilidad. A continuación se describen las etapas del algoritmo diseñado y el conjunto de estrategias para proporcionar conocimiento específico del problema con el objeto de favorecer la búsqueda.

5.4 Representación de las soluciones

La variable de decisión se representa a través de un vector de bits \mathbf{x} donde cada componente x_i determina la existencia o ausencia del enlace i . El tamaño del vector \mathbf{x} expresado en función del número de nodos N de la red es $N \cdot (N - 1) / 2$.

La figura 5.1 muestra una red con seis nodos y con algunos enlaces activos. El conjunto de enlaces posibles es $\mathbf{E} = \{(1,2) (1,3) (1,4) (1,5) (1,6) (2,3) (2,4) (2,5) (2,6) (3,4) (3,5) (3,6) (4,5) (4,6) (5,6)\}$. El cromosoma que representa la red es $\mathbf{x} = [1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1]$, que determina el conjunto de enlaces activos $\mathbf{E} = \{(1,2) (1,3) (1,6) (2,3) (2,4) (2,6) (3,4) (4,5) (4,6) (5,6)\}$.

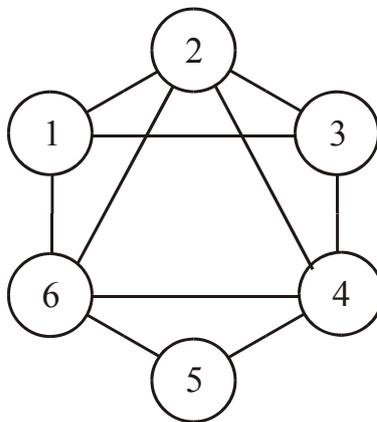


Figura 5.1 - Ejemplo de una red de 6 nodos no totalmente conectada

5.5 Generación de la población inicial

Si bien una de las características de un AE es su robustez, resulta conveniente incorporar conocimiento específico del problema, cuando se cuenta con él, siempre que sea posible. La generación de una población inicial (PI) sembrada permite un mejor punto de partida para el AE si se considera esta población conformada con soluciones

que, a priori, tengan buenas características.

Así, en primer lugar se evita que en la población inicial se incorporen cromosomas que representen topologías que contengan menos de $n-1$ enlaces activos dado que se sabe de antemano que no serán factibles porque no es posible siquiera alcanzar el más vulnerable de los casos: el *spanning tree*.

Habiéndose ejecutado el programa en repetidas oportunidades, se observó una mejora interesante con la incorporación de mayor conocimiento específico del problema, esta vez teniendo en cuenta las características recomendadas para el diseño de topologías confiables (Jan, 1993).

En ese sentido una PI que sea generada asignando mayor probabilidad de que la matriz de conexión que representa cada red tenga mayormente unos (1's) en el entorno de su diagonal principal, promueve topologías que tienden al anillo. Los enlaces restantes, si los hubiera, se distribuyen libremente.

Una de las últimas modificaciones permite que el usuario disponga de un coeficiente p_{ex} (raleo) que puede ajustarse entre 0 y 1 y que controla probabilísticamente la cantidad de enlaces activos en estos primeros individuos. A medida que el coeficiente se incrementa, las redes tienden a ser más ralas.

Sin embargo, es necesario tomar recaudos para no restringir demasiado las características de las soluciones sembradas a los efectos de evitar conducir al algoritmo a una convergencia temprana. Por ello en ningún caso, salvo por la cantidad de enlaces, hay imposiciones. Sólo se aumentan o disminuyen las probabilidades sin proscribir a aquellos cromosomas que, si bien individualmente resultan de mala calidad, pueden proveer de material genético valioso para generaciones futuras. Por las mismas razones tampoco se exige factibilidad inicial.

Respecto al número de individuos de cada población, se debe lograr una vez más un equilibrio, en este caso entre velocidad de convergencia y esfuerzo computacional. El tamaño de la población se fijó sobre la base del análisis de los reportes de diferentes trabajos y a partir de los resultados de las experiencias realizadas.

5.6 Función Objetivo y función de aptitud

La función objetivo, como se desprende de la formulación del problema, involucra la suma de todos los costos asociados a cada enlace existente. Para considerar las restricciones se le suma una función de penalización que depende del grado de violación de dichas restricciones. La función de aptitud, entonces, queda definida por:

$$\phi(x) = \sum_{i=1}^{N_e} c_i \cdot x_i + \psi \quad (5.4)$$

Donde ψ es la función de penalización. En este caso se determinó la expresión para establecer un valor para ψ dado por la ecuación 5.5

$$\psi = \begin{cases} 0 & \text{si la solución es factible} \\ \sum_{i=1}^{N_e} c_i & \text{en caso contrario} \end{cases} \quad (5.5)$$

De esta manera se penalizan las soluciones no factibles con un costo mayor al máximo costo posible.

5.7 Conocimiento específico del problema en el cálculo de la confiabilidad

El aporte del conocimiento específico del problema que se intenta resolver puede reducir de manera importante los tiempos de cálculo. En este caso la tarea más crítica en tiempo es la evaluación de la confiabilidad, razón por la cual las estrategias están orientadas a evitar el cálculo innecesario de este valor. Se ha tomado en cuenta la propuesta de Jan de definir un límite superior sobre la confiabilidad. Definiendo \mathbf{d} como el vector de grados de los nodos, la expresión que fija un límite superior para la confiabilidad de una red esta dada por la ecuación 5.6.

$$R(G) \leq H(\mathbf{d}) \quad (5.6)$$

donde

$$H(\mathbf{d}) = \sum_{i=1}^N q^{d_i} \cdot \prod_{k=1}^{m_i} (1 - q^{d_k - 1}) \cdot \prod_{k=m_i+1}^{i-1} (1 - q^{d_k}) \quad (5.7)$$

con

$$m_i = \min(d_i, i - 1) \text{ para todo } i$$

De esta manera, el procedimiento utilizado para evaluar una solución potencial es el siguiente: se verifica que la red candidata cumpla con la condición de que sus nodos tengan al menos dos enlaces incidentes (biconexión). Luego, se calcula un límite superior para la confiabilidad $H(\mathbf{d})$ y, sólo si esta confiabilidad es mayor que la admisible, se procede a estimar la confiabilidad utilizando el método RVR. Así, no todas las redes son sometidas al proceso de simulación y pueden mejorarse los tiempos de cálculo.

5.8 Operadores Genéticos

En este trabajo se centra la atención en el método de selección. Los operadores de cruzamiento y mutación son los estándares: se seleccionó cruzamiento uniforme a través de una máscara binaria de la misma longitud que los cromosomas heredando la descendencia el bit correspondiente al padre si el valor de la componente homóloga en la máscara es 1, y de la madre en caso contrario. La mutación se realiza complementando cada bit aleatoriamente con una cierta probabilidad.

5.8.1 Mecanismo de selección

Los AG tradicionales utilizan esquemas de selección y apareamiento *panmíticos*, esto significa que cada individuo compite globalmente con todos los otros presentes en la población. Durante el proceso de selección, un individuo seleccionado puede potencialmente aparearse con cualquier otro individuo en la población. Aunque estos esquemas, tales como selección estocástica con reemplazo, son ampliamente utilizados, ellos no están equipados para realizar búsquedas de genotipos “exitosos” en paisajes de gran tamaño y multimodales, ya que la población es incapaz de mantener radicalmente diferentes genotipos con buen *fitness* debido a la convergencia. El problema de la

convergencia prematura, o mejor aún el balance entre explotación y exploración, ha sido un problema constante en los AG tradicionales. En la literatura pueden encontrarse muchas variantes sobre el AG clásico que intentan tratar con este problema (De Jong, 1975, Deb, 1989)

Collins, en su trabajo “Estudios sobre Evolución Artificial enuncia: *“en los procesos naturales no aparecen problemas de convergencia muy difundidos, aún después de billones de años de evolución, la necesidad de métodos especiales para evitar la convergencia en AG sugiere que existe un problema básico en la forma en que la evolución está siendo modelada”* (Collins, 1992). El método propuesto por Collins, es utilizar poblaciones estructuradas. Los elementos de un AE son conectados de forma tal de restringir el intercambio de información dentro de una vecindad. La estructura espacial tiene el efecto de dividir a la población en subpoblaciones genéticamente independientes. Aún cuando el proceso de evolución conduzca cada subpoblación local a la fijación sobre un sub-óptimo, cada una podría fijarse en un genotipo diferente, manteniendo diversidad en la población como un todo. En otras palabras descentralizar una población simple en varias subpoblaciones, permite un mejor muestreo del espacio de búsqueda y mejora el comportamiento del algoritmo básico tanto en calidad de soluciones como en tiempos de ejecución (Alba y co., 2002).

Varios modelos de subpoblaciones han sido propuestos en la bibliografía, usualmente categorizados como modelos de islas o modelos de vecindades.

En el primero la población es dividida en varias subpoblaciones sobre las que se ejecutan AEs independientes, que cooperan entre si a través de un mecanismo denominado migración. El rol de la migración es intercambiar información (por ejemplo por cruzamiento) entre una isla y otra; la cantidad de individuos migrados (velocidad) y la frecuencia de migración determinan la cantidad de información intercambiada.

En el segundo modelo, conocido como AE celulares, el intercambio de información entre los individuos está restringido a una estructura de vecindad local. Esto es, un individuo dado tiene su propio conjunto de compañeros potenciales de apareamiento definido por la vecindad del individuo; al mismo tiempo un mismo

individuo pertenece a varios conjuntos de apareamiento. Con el objetivo de simular una topología del espacio de búsqueda determinada, los individuos se distribuyen en una grilla unidimensional (anillo) o bidimensional (toroide) (Sarma y co., 1997); el solapamiento de las vecindades permite una difusión uniforme de las buenas soluciones a través de la grilla.

La figura 5.2 ilustra esquemáticamente el modelo panmítico junto con los modelos de islas y de vecindades.

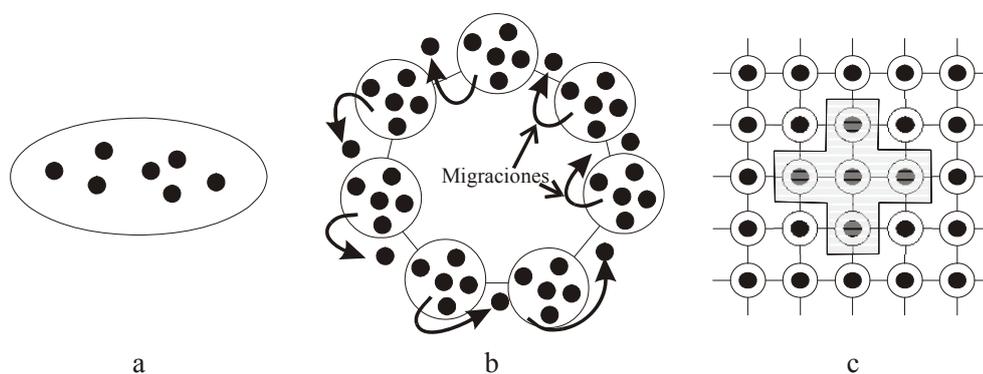


Figura 5.2 - Modelos de población: a) modelo panmítico; b) modelo de islas; c) modelo de vecindades

En conclusión, las características y diferencias de esos tres modelos pueden resumirse como sigue:

- En el modelo panmítico todos los individuos están en la misma población, la cual, generalmente es de mayor tamaño y todos los individuos interactúan con cualquiera de los otros.
- El modelo de islas mantiene más de una población de menor cantidad de individuos que en el modelo anterior, los individuos interactúan dentro de cada isla y la interacción entre islas se realiza a través del mecanismo de migración.
- El modelo de vecindades locales puede pensarse como micro poblaciones cuyo tamaño está dada por la cantidad de vecinos de cada individuo y la

interacción entre las vecindades se permite por la pertenencia de cada individuo a varias vecindades.

- Una vez que ha sido diseñado un modelo de población estructurada, éste puede ser implementado tanto en una máquina con un solo procesador, como en un hardware en paralelo. Un AE es intrínsecamente paralelizable y existe una fuerte tendencia a utilizar este modelo en las implementaciones en paralelo.

En este trabajo se utilizó selección local sobre un modelo de vecindad lineal: cada individuo tiene una vecindad dada por los w individuos que lo preceden o suceden. Así un individuo k tiene $k-w$ precedentes y $k+w$ sucesores. En los extremos, algunos vecinos corresponderán al extremo opuesto.

El método define, en primer lugar, un conjunto de individuos que son seleccionados de alguna manera para ser "padres". La selección de las "madres" se realiza, para cada padre entre los individuos de su vecindad. Para evitar diferencias desproporcionadas en los valores de *fitness*, se somete a la población a un *ranking* lineal antes de comenzar el proceso. De esta manera los cromosomas son seleccionados proporcionalmente a su rango más que a los valores reales de la función de aptitud, de forma de evitar que superindividuos eliminen material genético deseable y hagan que la evolución converja prematuramente (Michalewicz, 1996).

5.8.2 Operador de Cruzamiento

En este trabajo se ha optado por el método de cruzamiento uniforme, descrito anteriormente, donde cada bit del nuevo individuo es copiado desde el padre o la madre de acuerdo con una máscara de bits generados en forma aleatoria. Por lo tanto, si ambos progenitores tienen un gen idéntico, este será indefectiblemente igual en el descendiente a la salida de este operador. Los progenitores, por su parte, son tomados del vector de salida del procedimiento de selección.

La elección de los valores de los parámetros se basa en hipótesis teóricas y trabajos previos. En base a lo expuesto y basado también en experiencias propias con el

problema específico tratado, se fijó la probabilidad de mutación en un valor de 0.7, lo que significa que, en promedio, se somete al 70% de la población al operador de cruzamiento.

5.8.3 Operador de Mutación

El operador de mutación permite la inclusión de elementos diferentes a los existentes en la población actual. En este trabajo se utilizó un operador de mutación estándar, descrito en la sección 4.7.2.

Como se mencionó anteriormente, este operador añade una cuota de aleatoriedad a la búsqueda, por esa razón, y a diferencia del cruzamiento, la probabilidad de mutación no supera el 5%; evitándose de esta forma afectar desfavorablemente el balance entre explotación y exploración del espacio de búsqueda, que constituye una de las ventajas más importante de los AE respecto a otras técnicas estocásticas. Se utilizó un valor de probabilidad de mutación de 0,2% (dos por mil) con buenos resultados.

5.8.4 Búsqueda local

La búsqueda local combina el AE con una técnica que pretende potenciar la explotación de algunas de las soluciones que se van encontrando. Se implementó un mecanismo de búsqueda local que intenta mejorar las soluciones que el AE encuentra en pasos sucesivos. El método se basa en balancear los n grados de los nodos de una red.

Si en una red cualquiera $G = (V, E, p)$ con una secuencia ordenada de los grados de sus nodos, d , y si $d_i + 1 < d_j$ donde $i < j$, siempre existe un nodo k con un enlace $(k, j) \in E$ y uno $(k, i) \notin E$. La remoción del enlace (k, j) de G y su reemplazo por (k, i) da como resultado una red $G' = (V, E', p)$ con un mayor límite superior para la confiabilidad (Jan, 1993).

En cada iteración, se busca la mejor solución obtenida y se determinan los nodos j e i de mayor y menor grado respectivamente. Si la diferencia entre ambos es mayor o igual a 2 se determina el enlace (k, j) más caro conectado al nodo de mayor grado (este

enlace es el candidato a ser eliminado del grafo). Este enlace es reemplazado por (k, i) que une el nodo de menor grado con el nodo i que quedó con un enlace menos por la remoción de (k, j)

El lema de Jan, asegura un mayor límite superior pero no asegura que la confiabilidad de la red lo sea. Por lo tanto, una vez realizados los cambios mencionados la red podría ser menos confiable que la primera (aunque siempre será de menor costo). La estrategia entonces es reemplazar con el resultado de la búsqueda local al peor individuo de la población (menor *fitness*) con el resultado de la búsqueda local.

Notar que esta estrategia aún mantiene la posibilidad de reemplazar un individuo de la población por otro menos confiable. Sin embargo, la probabilidad que esto ocurra es baja no sólo porque no es común que el peor individuo posea una buena aptitud, sino porque además, y al menos en principio, se espera que el nuevo cromosoma sea el mejor de todos. La figura 5.3 describe el algoritmo de búsqueda implementado.

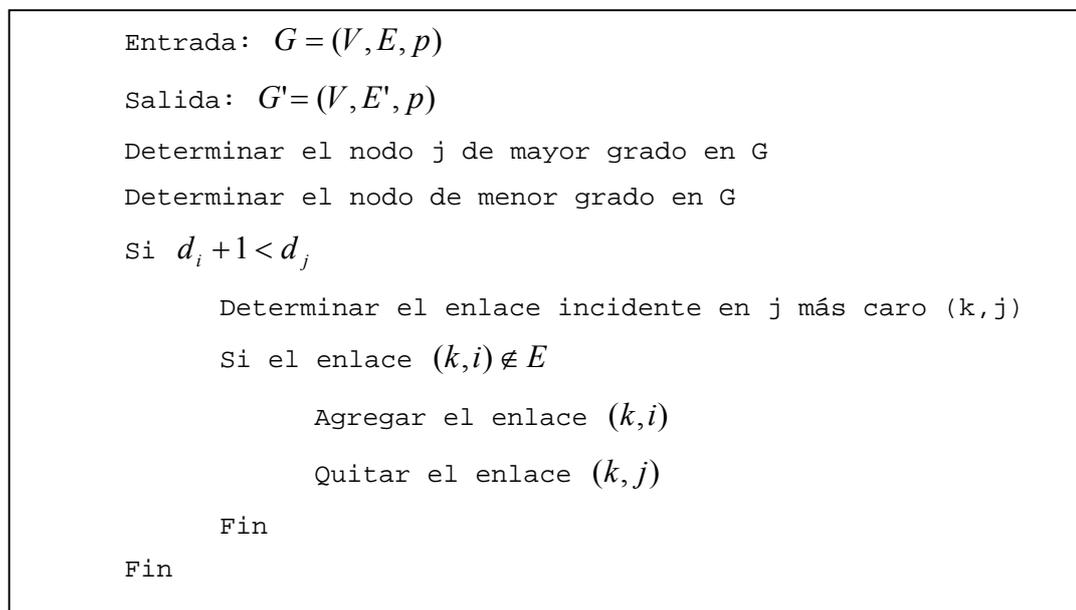


Figura 5.3 - Algoritmo de mejora de calidad de soluciones

Para verificar el rendimiento de este algoritmo evolutivo híbrido, se procedió a ejecutarlo con búsqueda local y sin ella. La figura 5.4 muestra las dos curvas de convergencia, observándose una ventaja a favor del caso híbrido.

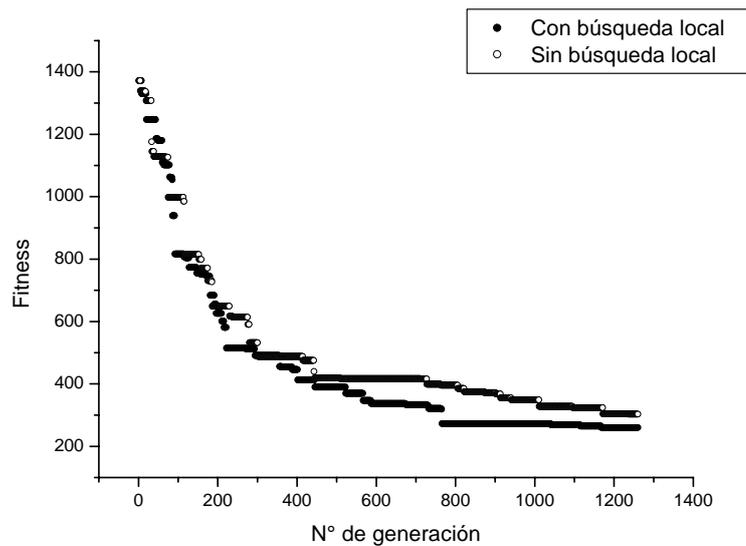


Figura 5.4 - Efecto de la búsqueda local sobre la convergencia del AE

5.8.5 Almacenamiento de Soluciones Evaluadas

La inclusión de esta modificación, si bien conceptualmente simple, implica un punto clave, que confirma la superioridad de este algoritmo en términos de eficiencia en el procesamiento computacional, por sobre algunos de los algoritmos encontrados en las bibliografías de referencia.

El cálculo de las restricciones, es decir el cálculo de la confiabilidad exacta supone una gran demanda de tiempo de cálculo, por otra parte, el problema NP-Hard sigue siendo una limitante, aún después de haber abandonado las técnicas determinísticas para la exploración del espacio de búsqueda.

Establecer el valor de *fitness* para cada uno de los individuos de la población en cada iteración, requiere saber si la solución es factible o no. Cada vez que la prueba de factibilidad no pueda determinarlo mediante verificación de conectividad y el límite superior, es necesario utilizar el método de simulación (RVR).

Como los individuos van evolucionando, es esperable que la aptitud (*fitness*) de un buen porcentaje de la población inicial pueda establecerse en tiempos acotados. Sin embargo, a medida que crece el número de iteraciones, los candidatos se hacen más y más confiables y en ese caso ya no es posible evitar el proceso de simulación. De hecho,

casi el 80% del total de los *fitness* recurre al cálculo, por lo que se estima que es necesario en más del 95% de aquellos casos pertenecientes a generaciones relativamente avanzadas.

En el caso que se trata, esta técnica para estimar la confiabilidad puede necesitarse hasta medio centenar de veces por iteración. Esto se traduce en tiempos muy importantes y a veces, prohibitivos. Una alternativa consiste en reducir el número de iteraciones, pero esto puede derivar en la detención del algoritmo antes de la convergencia.

Sin embargo, varios individuos de una población pueden mantenerse imperturbados de una generación a otra. Posiblemente el *fitness* de un porcentaje de individuos ya haya sido calculado. Se consideró como una buena solución a los problemas planteados anteriormente almacenar los *fitness* de cada individuo y luego antes de hacer la prueba de factibilidad de algún individuo recuperar el valor sin repetir el cálculo. De esta manera se logra un factor de reducción en el esfuerzo computacional de alrededor de 10 en promedio.

5.8.6 Post Optimización

Durante las experiencias realizadas se observó un buen desempeño de los programas, pero con ciertos problemas se encontraron resultados promedios algo elevados; es decir, si bien se hallaba el óptimo global en varias de las experiencias realizadas, el promedio era mayor que el reportado en la bibliografía de referencia debido a las ejecuciones en las que no se lo hallaba.

El análisis de estos resultados puso en evidencia que muchos de los subóptimos encontrados diferían del óptimo global en tan solo un enlace. La comparación de ambos individuos daba una distancia de Hamming igual a 2 ($dH=2$).

Una explicación consistente con la situación radica en que en la mayoría de estos casos existían dos enlaces disponibles de costo muy similar y que, estando activo tanto uno como otro en una determinada topología, brindaban una confiabilidad muy parecida o incluso, idéntica. La gran calidad de la solución hallada tornaba improbable que el AE

lograra escapar de ese óptimo local.

En consecuencia se decidió implementar un mecanismo de mejora de una solución mediante una exploración de su vecindad, definida como el conjunto de individuos cuya distancia de hamming respecto a la candidata es 2 ($dH=2$). Si existe una solución mejor, entonces esta última reemplaza a la primera (Martí, 2006)

En el peor de los casos, la cantidad máxima de verificaciones será de $(l/2)^2$, que resultó una cantidad manejable en todos los problemas ejecutados. Si bien puede intuirse que esta técnica no es aconsejable en redes relativamente grandes, esto no es cierto necesariamente, pues la mayoría de las combinaciones se descartan inmediatamente por mayor costo. De no ser así, antes de recurrir a la simulación todavía hay que verificar conectividad y el límite superior, lo que elimina otras tantas en tiempos despreciables. Finalmente, todavía hay posibilidades de tener guardado el *fitness* de un cálculo previo. Más aún, es posible que este procedimiento no se ejecute siempre, sino sólo en las situaciones que se consideran convenientes.

Según lo explicado en la sección anterior, también es posible que la salida de este procedimiento sea una solución más económica, pero por error en la evaluación de la confiabilidad, no factible. A pesar de esto, no supone mayores problemas si se atienden las advertencias anteriores.

5.8.7 Experimentos y resultados

Para las pruebas de ajuste del algoritmo implementado se tomó un ejemplo de la literatura de una red de siete nodos que se muestra en la Figura 5.3 (Jan, 1993) resuelto con un método de *branch and bound* y se compararon con los resultados obtenidos mediante un AE con selección local y un AE con esquema de selección panmítico.

La matriz de costos asociada se indica en la Figura 5.4. Para este experimento se solicitó una confiabilidad total del 90%, todos los enlaces son permitidos y tienen una probabilidad de falla de 0.10. La probabilidad de cruzamiento utilizada fue de 0.7 y la de mutación 0.002. Se seleccionó un 90% de la población total para apareamiento y la presión selectiva fue de 1.3.

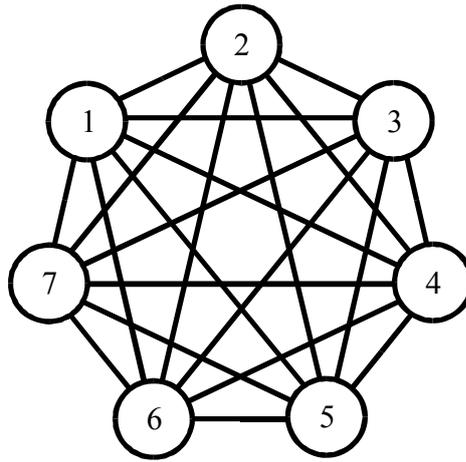


Figura 5.5 - Red de ejemplo

$$C = \begin{bmatrix} - & 125 & 150 & 125 & 150 & 150 & 130 \\ 125 & - & 75 & 100 & 150 & 200 & 250 \\ 150 & 75 & - & 75 & 90 & 250 & 200 \\ 125 & 100 & 75 & - & 75 & 100 & 150 \\ 150 & 150 & 90 & 75 & - & 75 & 100 \\ 150 & 200 & 250 & 100 & 75 & - & 75 \\ 130 & 250 & 200 & 150 & 100 & 75 & - \end{bmatrix}$$

Figura 5.6 - Matriz de costo para la red de la Figura 5.5

La red solución encontrada mediante el AE con selección local propuesto se muestra en la figura 5.7 y corresponde al conjunto de enlaces $E = \{(1,2) (1,7) (2,3) (3,4) (3,5) (4,5) (5,6) (6,7)\}$, con un costo de 720 y una confiabilidad de 0.91. Esta solución coincide con la hallada por métodos exactos.

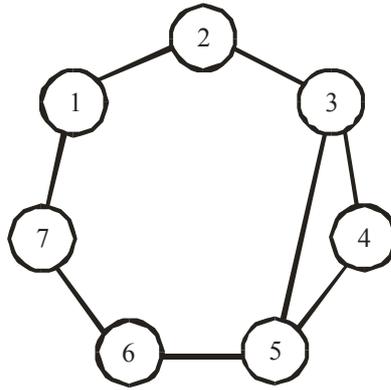


Figura 5.7 - Topología óptima encontrada para el ejemplo de siete nodos.

La figura 5.8 muestra las curvas de convergencia para ambos tipos de AEs, donde se observa una clara ventaja del esquema de población estructurada frente a un esquema panmítico.

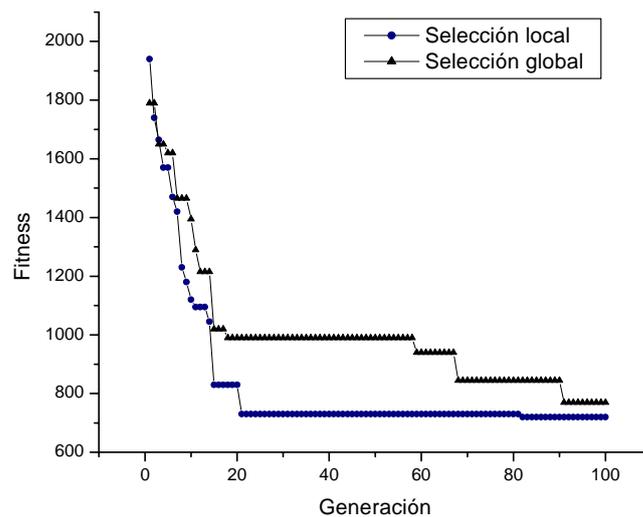


Figura 5.8 - Comparación de convergencia del AE con selección local y selección global

La calidad de las soluciones encontradas por ambos algoritmos puede apreciarse en la distribución de las mismas en la generación 200, mostrada en la Figura 5.9. Para el AE con selección local el valor mínimo de *fitness* encontrado corresponde a 720, con un promedio de 777.3 y una dispersión de 71.9. En tanto que para un AE simple los valores respectivos son 770, 2294.1 y 1212.6.

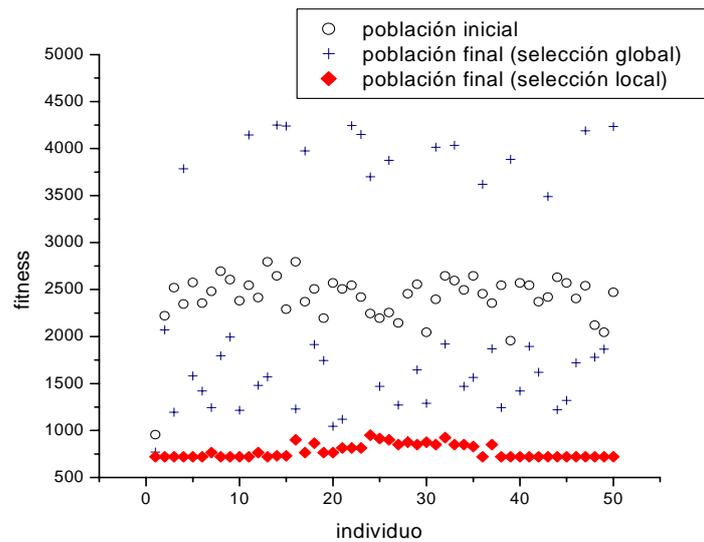


Figura 5.9 - Comparación de la distribución de *fitness* con selección local y global

5.9 Conclusiones

En este capítulo se ha presentado la aplicación de técnicas evolutivas basadas en un método de selección local por vecindades a la resolución del problema de optimización topológica de una red con restricciones de confiabilidad. El cálculo de la confiabilidad asociada a cada topología se realizó a través del método de reducción recursiva de la varianza, junto con la determinación de límites superiores sobre la misma que permitan identificar soluciones no factibles, reduciendo por lo tanto el costo computacional asociado. Las bondades de la utilización de un AE equipado con una selección local fueron mostradas al comparar su desempeño frente a un AE tradicional como así también el aporte de la búsqueda local empleada para mejorar la calidad de las soluciones y la convergencia.

La flexibilidad que presentan las técnicas de optimización basadas en algoritmos evolutivos permite adaptar el AE propuesto a la resolución de otros problemas de diseño de redes de comunicación con diferentes criterios de desempeño y conjunto de restricciones; e incluso proveer herramientas para la toma de decisiones en diseños con objetivos múltiples.

CAPÍTULO 6

6 EJEMPLOS DE APLICACIÓN

6.1 Introducción

En este capítulo se presentan ejemplos de ejecución del algoritmo evolutivo implementado, aplicados a 20 problemas de diseño de redes confiables presentes en la literatura, (Dengiz, 1997), con el objetivo de medir cuantitativamente la calidad del mismo. En muchos de los casos que se analizan se conocen los óptimos globales obtenidos a través de métodos exactos mientras que en otros sólo puede realizarse una comparación con los mejores valores obtenidos hasta el momento dado que no se conoce el valor óptimo. Esto resulta de espacios de búsqueda muy grandes: uno de los ejemplos resueltos aquí tiene un espacio de búsqueda de $2 \cdot 10^{90}$. Se presentan a continuación el análisis detallado de algunos de los experimentos y se resume más adelante la totalidad de los casos en una tabla de resultados comparativos.

6.2 Descripción de los experimentos

6.2.1 Caso 1

En primer lugar se considera una red de pequeñas dimensiones con 5 nodos y 10 enlaces posibles (todos los enlaces son candidatos a conformar la solución), cuya matriz de costo se muestra en la figura 6.1, y con óptimo global conocido y determinable por simple enumeración. Se presentan dos casos utilizando esta red, los cuales se han denominado respectivamente problema 1 y problema 2. En el primer caso se considera una confiabilidad de los enlaces de 0.80 y se impone una confiabilidad admisible de 0.90. En el segundo caso los valores mencionados se han fijado en 0.90 y 0.95. Es de esperar, dado que se pide una confiabilidad mayor a la que poseen los enlaces, que la red solución tenga redundancia, dado que un *spanning tree* no puede tener una

confiabilidad mayor a la de los enlaces.

$$\mathbf{A} = \begin{bmatrix} 0 & 32 & 54 & 62 & 25 \\ 32 & 0 & 34 & 58 & 45 \\ 54 & 34 & 0 & 36 & 52 \\ 62 & 58 & 36 & 0 & 29 \\ 25 & 45 & 52 & 29 & 0 \end{bmatrix}$$

Figura 6.1. Matriz de costo problemas 1 y 2

El AE diseñado se utilizó con 51 individuos en una población sembrada, un coeficiente de raleo de 0.01, probabilidad de cruzamiento de 0.7 y probabilidad de mutación de 0.001. Respecto del cálculo de la confiabilidad de cada topología, el método de reducción recursiva de la varianza realizó 6000 ensayos para determinar su valor.

Se realizaron 100 ejecuciones del algoritmo obteniendo un costo de 255 con una confiabilidad de 0.9177 en el mejor de los casos con buena repetitividad: en el 79% de los casos en el problema 1 se obtuvo el óptimo global. El promedio de los resultados arrojados es de 257.28 y el desvío estándar de 4.45. La figura 6.2 muestra el resultado de las distintas ejecuciones con los costos obtenidos.

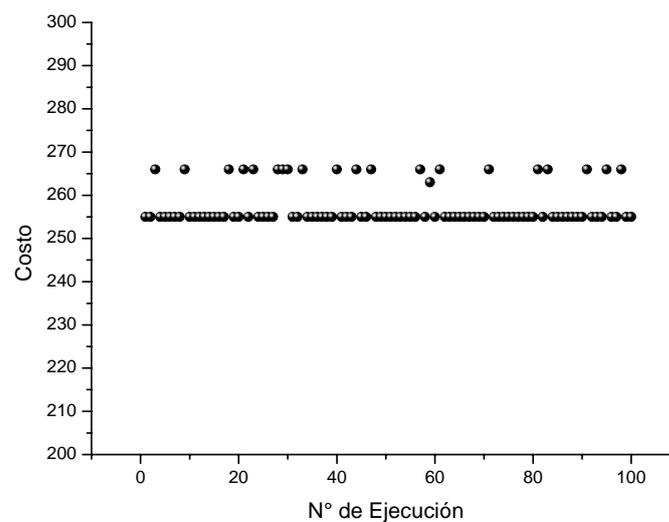


Figura 6.2. Resultado de 100 ejecuciones para el problema 1

La red óptima obtenida a través del AE se muestra en la figura 6.3 para el caso del problema 1, que corresponden al conjunto de enlaces $E^* = \{(1,2) (1,3) (1,5) (2,3) (2,5) (3,4) (4,5)\}$.

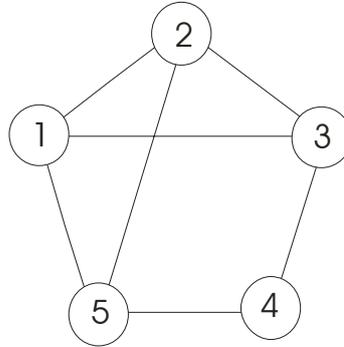


Figura 6.3. Topología óptima obtenida para el problema 1

Para el segundo problema del caso 1 se obtuvieron igualmente buenos resultados los cuales se detallan en la tabla de resumen. En este problema se fijó una confiabilidad admisible de 0.95 y se le asignó una confiabilidad a los enlaces de 0.90.

6.2.2 Caso 2

El segundo caso que se presenta consiste de una red de 7 nodos, con 21 enlaces, cuya matriz de costo se muestra en la figura 6.4, y un espacio de búsqueda de $2 \cdot 10^6$. La confiabilidad de los enlaces fue fijada en 0.90 y la confiabilidad admisible para la red fue $R_0=0.95$. En este caso, el óptimo global fue encontrado en la gran mayoría de los casos con un promedio de 845.2 y un desvío estándar de 30.32. Estos resultados son mejores que los reportados en la bibliografía de referencia (aunque esta situación no se mantiene en todos los casos analizados).

$$A = \begin{bmatrix} 0 & 125 & 150 & 125 & 150 & 150 & 130 \\ 125 & 0 & 75 & 100 & 150 & 200 & 250 \\ 150 & 75 & 0 & 75 & 90 & 250 & 200 \\ 125 & 100 & 75 & 0 & 75 & 100 & 150 \\ 150 & 150 & 90 & 75 & 0 & 75 & 100 \\ 150 & 200 & 250 & 100 & 75 & 0 & 75 \\ 130 & 250 & 200 & 150 & 100 & 75 & 0 \end{bmatrix}$$

Figura 6.4. Matriz de costo problemas 3 4 y 5

En este caso se ha elegido mostrar la evolución poblacional al cabo de 100 iteraciones. En la figura 6.5 se puede observar, para una de las ejecuciones, la población en la generación 0 (población inicial) y la distribución de fitness al cabo de 100 generaciones. Se puede observar una muy buena convergencia sobre el valor óptimo de una gran cantidad de individuos. La figura 6.6 muestra la topología óptima con una confiabilidad de 0.9511 y formada por el conjunto de enlaces $E = \{(1,2) (1,7) (2,3) (2,4) (3,4) (3,5) (4,6) (5,6) (6,7)\}$ para el problema 4.

Se experimentó con otros problemas correspondientes al mismo caso cuyos resultados se muestran en la tabla resumen.

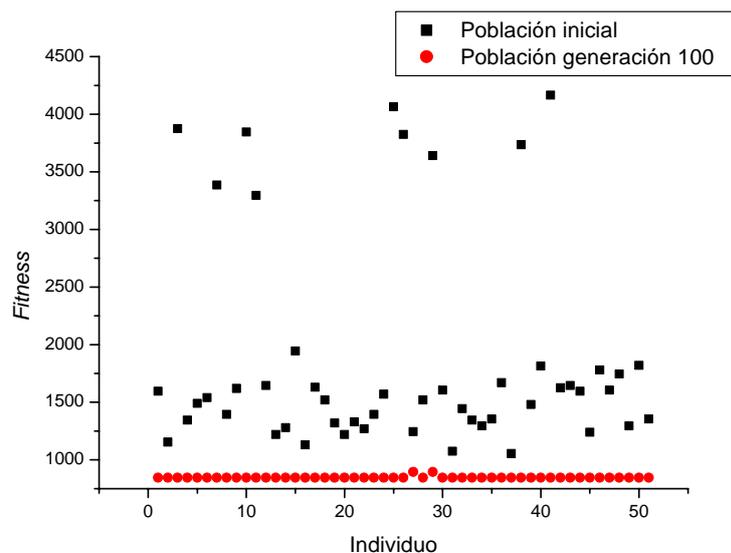


Figura 6.5. Evolución del fitness de los individuos. Problema 4

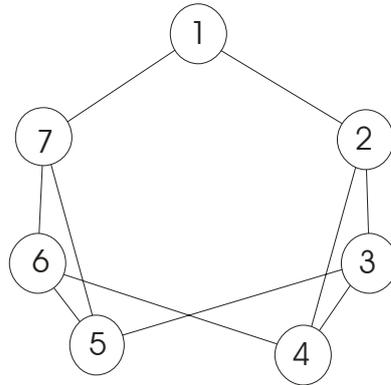


Figura 6.6. Topología óptima para el problema 4

6.2.3 Caso 3

Se experimentó en este caso con redes de mayor cantidad de nodos y de enlaces posibles. Dentro de este caso se han incorporado los problemas 12, 13 y 14 que consideran una red cuya matriz de costo se muestra en la figura 6.7.

$$A = \begin{bmatrix} 0 & 24 & 26 & 69 & 25 & 48 & 03 & 82 & 45 & 98 \\ 24 & 0 & 12 & 75 & 22 & 33 & 82 & 54 & 04 & 82 \\ 26 & 12 & 0 & 30 & 08 & 75 & 38 & 21 & 79 & 23 \\ 69 & 75 & 30 & 0 & 67 & 18 & 64 & 50 & 78 & 12 \\ 25 & 22 & 08 & 67 & 0 & 72 & 92 & 94 & 21 & 96 \\ 48 & 33 & 75 & 18 & 72 & 0 & 05 & 81 & 18 & 84 \\ 03 & 82 & 38 & 64 & 92 & 05 & 0 & 19 & 37 & 34 \\ 82 & 54 & 21 & 50 & 94 & 81 & 19 & 0 & 07 & 98 \\ 45 & 04 & 79 & 78 & 21 & 18 & 37 & 07 & 0 & 50 \\ 98 & 82 & 23 & 12 & 96 & 84 & 34 & 98 & 50 & 0 \end{bmatrix}$$

Figura 6.7. Matriz de costo problemas 12,13 y 14

Dentro de este caso se analiza a continuación el problema 13: 45 enlaces posibles cada uno de los cuales tiene una confiabilidad de 0.90 y se solicita una confiabilidad mínima admisible $R_0=0.95$. En este problema, comienza a aparecer un mayor desvío en los resultados obtenidos.

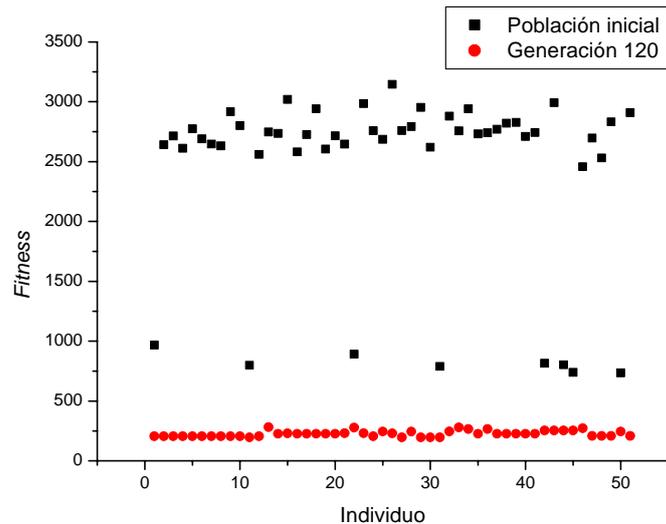


Figura 6.8. Evolución del fitness de los individuos. Problema 13

El resultado óptimo es de 197 y el AE lo alcanza. Esto constituye una mejora respecto de los algoritmos planteados en la referencia que arroja un valor mínimo de 205. El valor medio obtenido por el AE es de 229.74 mientras que en la referencia dicho valor es menor (206.6). El desvío estándar que se obtuvo es de 32.34. La figura 6.8 muestra la evolución del fitness de los individuos en la población inicial y la generación 120.

Comparando la figura 6.8 con la figura 6.5 se puede observar que al crecer el tamaño de la instancia del problema, se agranda la brecha entre los individuos pertenecientes a la población inicial y los de la última generación. Aunque el valor promedio ha sido mayor que en la referencia se ha logrado obtener un mejor resultado dado que se alcanzó el óptimo global. La figura 6.9 muestra la topología óptima para este problema. Se puede observar que todos los nodos tienen al menos dos enlaces incidentes sobre ellos.

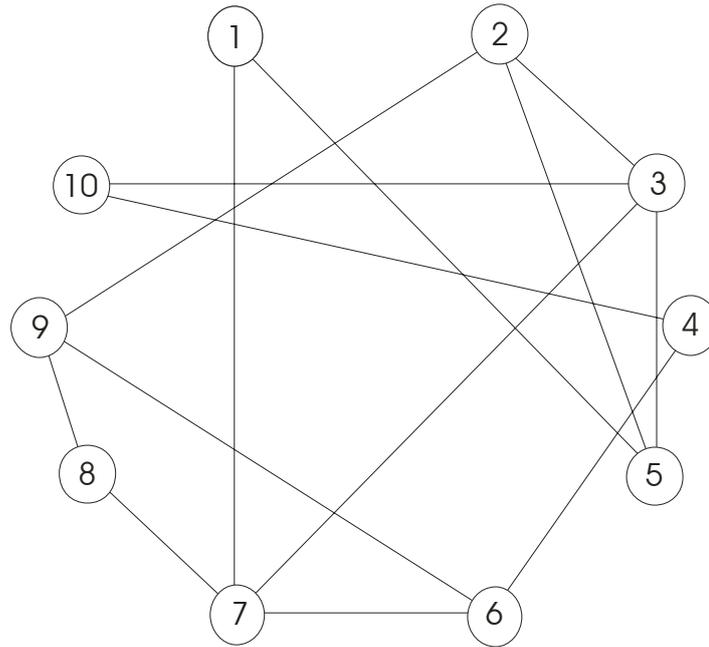


Figura 6.9. Topología óptima para el problema 13

6.2.4 Caso 4

En este caso se analizaron redes no totalmente conectadas. Esto significa que no todos los enlaces están disponibles y debe buscarse la solución respetando esta restricción. Los problemas que se analizan en este caso están numerados como 18, 19 y 20, de los cuales se muestra en detalle, a continuación, el problema 19. La matriz de costos de dicho problema se muestra en la figura 6.10 (sólo se muestran los elementos por encima de la diagonal principal de dicha matriz simétrica). Esta red tiene 16 nodos y de los 120 enlaces posibles sólo se permiten 24. En primer lugar se ensayó el AE asignando costo muy alto a aquellos enlaces que estaban prohibidos, pero los resultados fueron malos. Se realizó una modificación para que se respetara la restricción de nodos prohibidos con lo cual se mejoró mucho el desempeño del algoritmo sin pérdida de generalidad.

El problema 19 está planteado en la bibliografía de referencia con enlaces de confiabilidad 0.90 y se impone una confiabilidad global mínima de 0.95. Por otra parte reporta como solución un enlace que no está permitido (1,13).

Al realizar los primeros cálculos se comprobó que los 24 enlaces permitidos con una confiabilidad de 0.90 no alcanzan a la confiabilidad admisible en ningún caso. En

consecuencia, se tomó la decisión de modificar las condiciones iniciales del problema y mantener la solución propuesta, para ello se incorporó el enlace (1,13) con un costo de 16 y se consideró una confiabilidad en los enlaces de 0.95. Con estos valores se obtienen los valores reportados como solución en la bibliografía.

$$\begin{bmatrix} 0 & 13 & 99 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 0 & 0 \\ & 0 & 12 & 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 68 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & 0 & 39 & 38 & 79 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & 0 & 61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & 0 & 92 & 0 & 87 & 0 & 0 & 0 & 0 \\ & & & & & & & & & 0 & 80 & 81 & 0 & 0 & 0 & 0 \\ & & & & & & & & & & 0 & 55 & 0 & 89 & 0 & 0 \\ & & & & & & & & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 50 & 36 \\ & & & & & & & & & & & & & 0 & 86 & 51 \\ & & & & & & & & & & & & & & 0 & 98 \\ & & & & & & & & & & & & & & & 0 \end{bmatrix}$$

Figura 6.10. Matriz de costo para el problema 19

La solución óptima tiene un costo de 1022 (coincidente con el valor hallado por métodos exactos) con un promedio sobre 100 ejecuciones de 1081.82 (contra 1032 de la referencia) y un desvío estándar de 71.67. La evolución de las soluciones se puede ver en la figura 6.11 para las poblaciones inicial y final. En dicha figura puede observarse que al cabo de 250 iteraciones la población ha sido copada por el mejor individuo. Sin embargo, en la iteración 83 el AE ya obtuvo el valor óptimo. Las siguientes generaciones, por lo tanto, ya no pueden provocar una mejora sino la convergencia de los individuos a dicho valor óptimo.

Este caso, la cantidad de nodos se ha incrementado respecto de los anteriores. Sin embargo el hecho de restringir los enlaces disponibles a sólo un subconjunto del total de enlaces que podrían conectar los nodos, produce una disminución en los requerimientos de recursos computacionales dado que el espacio de búsqueda se reduce.

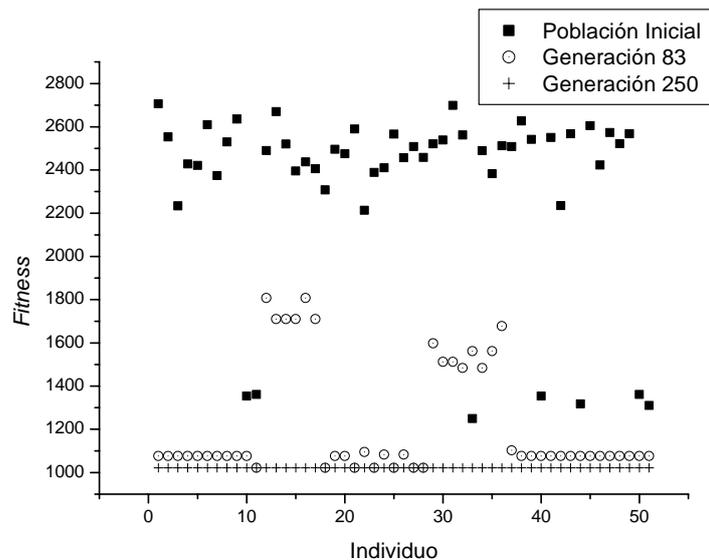


Figura 6.11. Evolución del fitness de los individuos. Problema 19

En el apéndice de este capítulo se encuentra la totalidad de los experimentos realizados en este trabajo.

6.3 Conclusiones

Se ha aplicado el AE desarrollado a 20 problemas tomados de la literatura con el fin de analizar su desempeño y se han comparado resultados con métodos exactos y otros algoritmos basados en la misma heurística. Se han logrado mejoras respecto de la capacidad del AE para encontrar soluciones de mejor calidad que las reportadas en la literatura, aunque con mayor desvío. En algunos casos el AE presentado en este trabajo ha encontrado valores óptimos, coincidentes con los encontrados por métodos exactos y en otros, si bien no se ha obtenido el óptimo se ha logrado una solución aceptable.

6.4 Apéndice

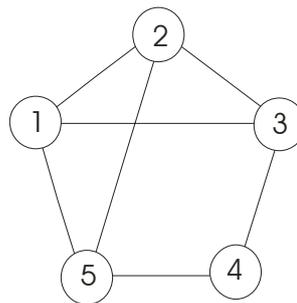
6.4.1 Redes totalmente conectadas

Caso 1: problemas 1 y 2. Matriz de 5 nodos y 10 enlaces

$$A = \begin{bmatrix} 0 & 32 & 54 & 62 & 25 \\ 32 & 0 & 34 & 58 & 45 \\ 54 & 34 & 0 & 36 & 52 \\ 62 & 58 & 36 & 0 & 29 \\ 25 & 45 & 52 & 29 & 0 \end{bmatrix}$$

Solución óptima del Problema 1: $E^* = \{(1,2) (1,3) (1,5) (2,3) (2,5) (3,4) (4,5)\}$

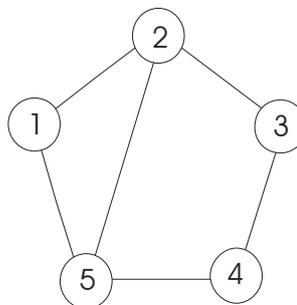
Costo=255, confiabilidad=0.9177



Topología óptima para el problema 1

Solución óptima del Problema 2: $E^* = \{(1,2) (1,5) (2,3) (2,5) (3,4) (4,5)\}$

Costo=201, confiabilidad=0.9546



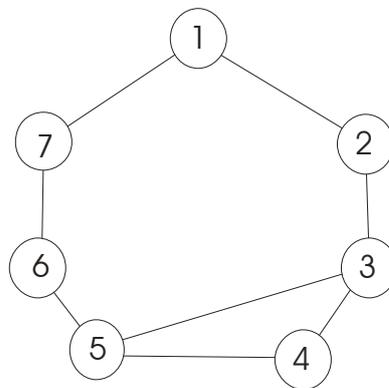
Topología óptima para el problema 2

Caso 2: problemas 3, 4 y 5. Matriz de 7 nodos y 21 enlaces

$$A = \begin{bmatrix} 0 & 125 & 150 & 125 & 150 & 150 & 130 \\ 125 & 0 & 75 & 100 & 150 & 200 & 250 \\ 150 & 75 & 0 & 75 & 90 & 250 & 200 \\ 125 & 100 & 75 & 0 & 75 & 100 & 150 \\ 150 & 150 & 90 & 75 & 0 & 75 & 100 \\ 150 & 200 & 250 & 100 & 75 & 0 & 75 \\ 130 & 250 & 200 & 150 & 100 & 75 & 0 \end{bmatrix}$$

Solución óptima del Problema 3: $E^* = \{(1,2) (1,7) (2,3) (3,4) (3,5) (4,5)(5,6)(6,7)\}$

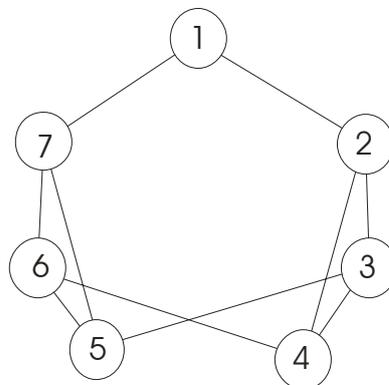
Costo=720, confiabilidad=0.9000



Topología óptima para el problema 3

Solución óptima del Problema 4: $E^* = \{(1,2) (1,7) (2,3) (2,4) (3,4) (3,5) (4,6)(5,6)(6,7)\}$

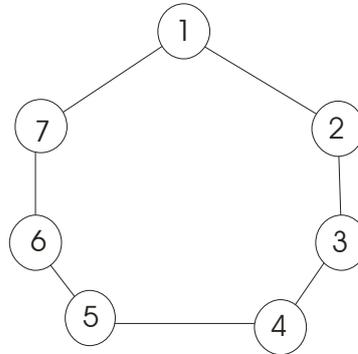
Costo=845, confiabilidad=0.9511



Topología óptima para el problema 4

Solución óptima del Problema 5: $E^* = \{(1,2) (1,7) (2,3) (3,4) (4,5) (5,6)(6,7)\}$

Costo=630, confiabilidad=0.9500



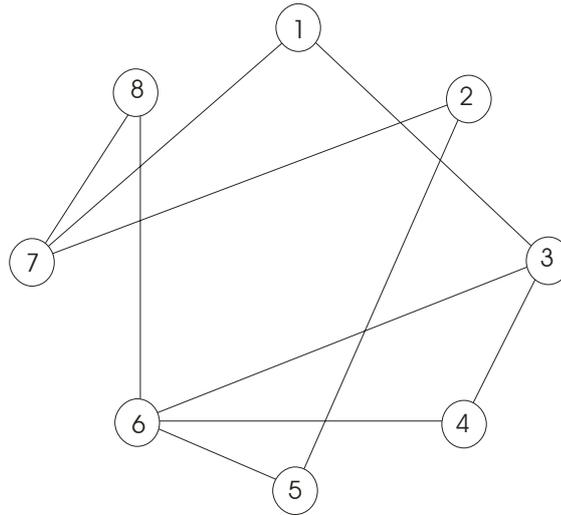
Topología óptima para el problema 5

Caso 3: problemas 6, 7 y 8. Matriz de 8 nodos y 28 enlaces

$$A = \begin{bmatrix} 0 & 59 & 19 & 98 & 77 & 35 & 40 & 93 \\ 59 & 0 & 68 & 39 & 16 & 48 & 12 & 81 \\ 19 & 68 & 0 & 17 & 41 & 24 & 89 & 41 \\ 98 & 39 & 17 & 0 & 60 & 23 & 72 & 45 \\ 77 & 16 & 41 & 60 & 0 & 23 & 51 & 84 \\ 35 & 48 & 24 & 23 & 23 & 0 & 54 & 1 \\ 40 & 12 & 89 & 72 & 51 & 54 & 0 & 33 \\ 93 & 81 & 41 & 45 & 84 & 1 & 33 & 0 \end{bmatrix}$$

Solución óptima del Problema 6: $E^* = \{(1,3) (1,7) (2,5) (2,7) (3,4) (3,6) (4,6) (5,6) (6,8) (7,8)\}$

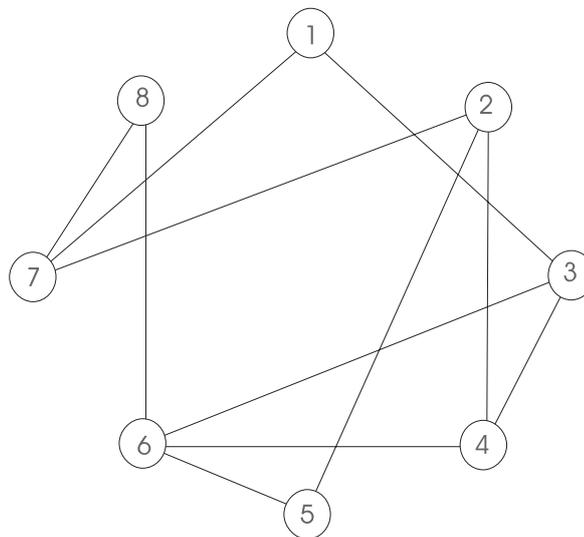
Costo=208, confiabilidad=0.9257



Topología óptima para el problema 6

Solución óptima del Problema 7: $E^* = \{(1,3) (1,7) (2,4) (2,5) (2,7) (3,4) (3,6) (4,6) (5,6) (6,8) (7,8)\}$

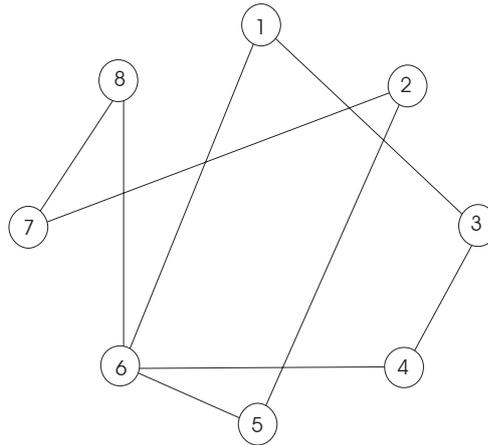
Costo=247.00 confiabilidad=0.9607



Topología óptima para el problema 7

Solución óptima del Problema 8: $E^* = \{(1,3) (1,6) (2,5) (2,7) (3,4) (4,6) (5,6) (6,8) (7,8)\}$

Costo=179.00 confiabilidad=0.9585



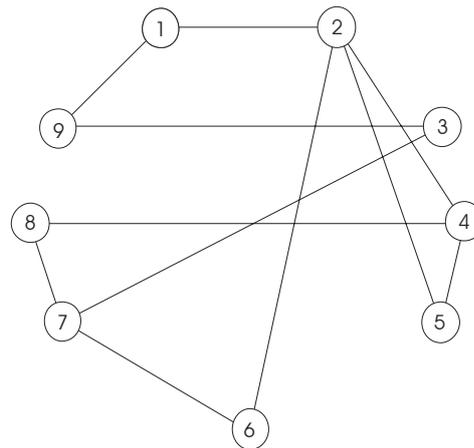
Topología óptima para el problema 8

Caso 4: problemas 9, 10 y 11: Matriz de 9 nodos y 36 enlaces

$$A = \begin{bmatrix} 0 & 37 & 77 & 61 & 97 & 58 & 41 & 63 & 03 \\ 37 & 0 & 40 & 30 & 04 & 53 & 61 & 37 & 63 \\ 77 & 40 & 0 & 56 & 63 & 71 & 13 & 90 & 34 \\ 61 & 30 & 56 & 0 & 33 & 70 & 39 & 07 & 35 \\ 97 & 04 & 63 & 33 & 0 & 89 & 55 & 97 & 65 \\ 58 & 53 & 71 & 70 & 89 & 0 & 23 & 57 & 88 \\ 41 & 61 & 13 & 39 & 55 & 23 & 0 & 02 & 70 \\ 63 & 37 & 90 & 07 & 97 & 57 & 02 & 0 & 77 \\ 03 & 63 & 34 & 35 & 65 & 88 & 70 & 77 & 0 \end{bmatrix}$$

Solución óptima Problema 9: $E^* = \{(1,2) (1,9) (2,4) (2,5) (2,6) (3,7) (3,9) (4,5) (4,8) (6,7) (7,8)\}$

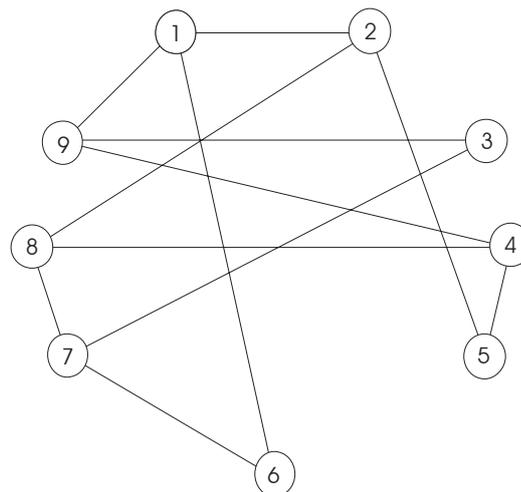
Costo = 239.00 Confiabilidad = 0.9036



Topología óptima para el problema 9

Solución Óptima Problema 10: $E^* = \{(1,2) (1,6) (1,9) (2,5) (2,8) (3,7) (3,9) (4,5) (4,8) (4,9) (6,7) (7,8)\}$

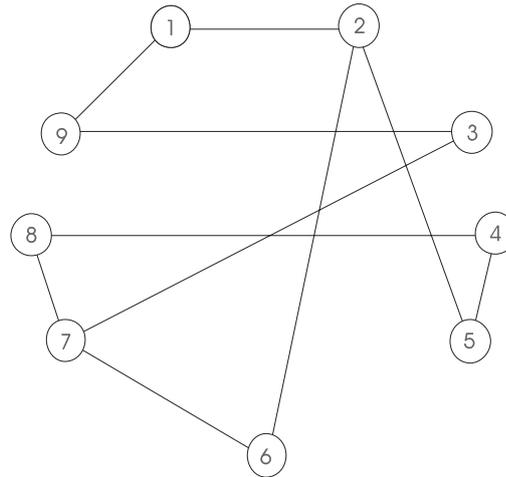
Costo = 286.00 Confiabilidad = 0.9563



Topología óptima para el problema 10

Solución Óptima Problema 11: $E^* = \{(1,2) (1,9) (2,5) (2,6) (3,7) (3,9) (4,5) (4,8) (6,7) (7,8)\}$

Costo = 209.00 Confiabilidad = 0.9625



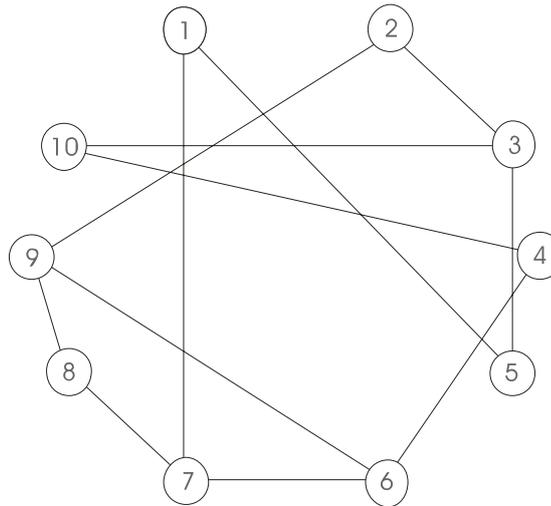
Topología óptima para el problema 11

Caso 5: problema 12, 13 y 14. Matriz de 10 nodos y 45 enlaces

$$A = \begin{bmatrix} 0 & 24 & 26 & 69 & 25 & 48 & 03 & 82 & 45 & 98 \\ 24 & 0 & 12 & 75 & 22 & 33 & 82 & 54 & 04 & 82 \\ 26 & 12 & 0 & 30 & 08 & 75 & 38 & 21 & 79 & 23 \\ 69 & 75 & 30 & 0 & 67 & 18 & 64 & 50 & 78 & 12 \\ 25 & 22 & 08 & 67 & 0 & 72 & 92 & 94 & 21 & 96 \\ 48 & 33 & 75 & 18 & 72 & 0 & 05 & 81 & 18 & 84 \\ 03 & 82 & 38 & 64 & 92 & 05 & 0 & 19 & 37 & 34 \\ 82 & 54 & 21 & 50 & 94 & 81 & 19 & 0 & 07 & 98 \\ 45 & 04 & 79 & 78 & 21 & 18 & 37 & 07 & 0 & 50 \\ 98 & 82 & 23 & 12 & 96 & 84 & 34 & 98 & 50 & 0 \end{bmatrix}$$

Solución óptima del Problema 12: $E^* = \{(1,5) (1,7) (2,3) (2,9) (3,5) (3,10) (4,6) (4,10) (6,7) (6,9) (7,8) (8,9)\}$

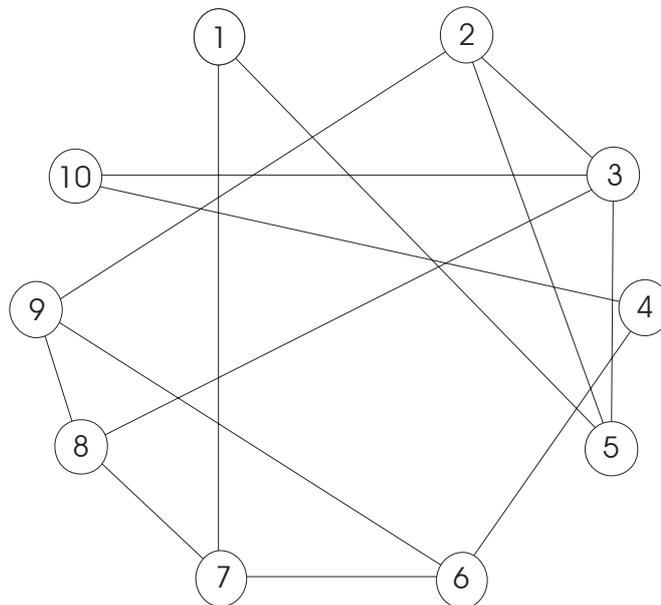
Costo = 154 confiabilidad = 0.9005



Topología óptima para el problema 12

Solución óptima del Problema 13: $E^* = \{(1,5) (1,7) (2,3) (2,5) (2,9) (3,5) (3,8) (3,10) (4,6) (4,10) (6,7) (6,9) (7,8) (8,9)\}$

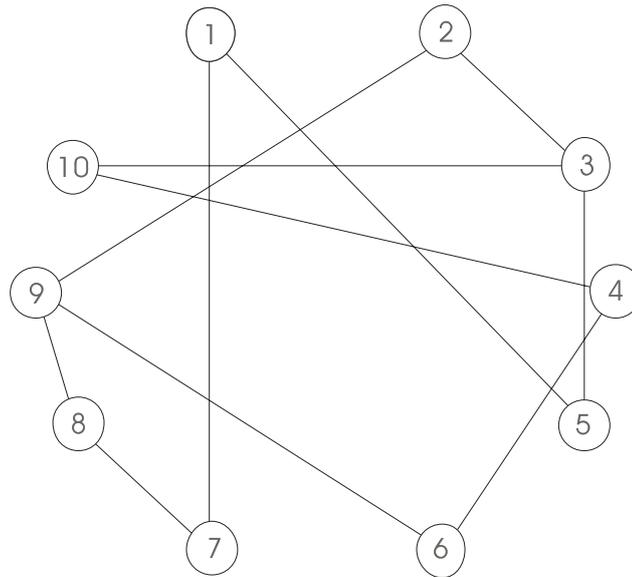
Costo = 197.00 Confiabilidad = 0.9524



Topología óptima para el problema 13

Solución óptima del Problema 14: $E^* = \{(1,5) (1,7) (2,3) (2,9) (3,5) (3,10) (4,6) (4,10) (6,9) (7,8) (8,9)\}$

Costo = 136.00 Confiabilidad = 0.9602



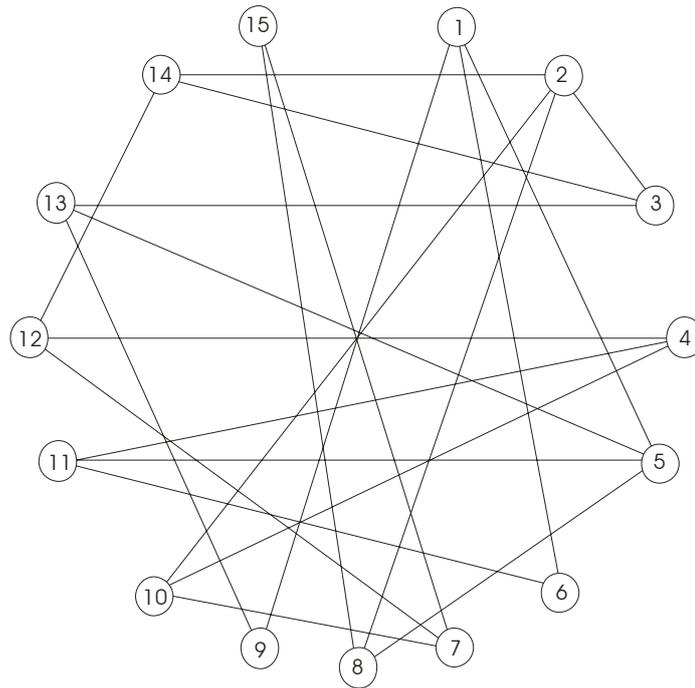
Topología óptima para el problema 14

Caso 6: problema 15. Matriz 15 nodos y 105 enlaces

$$A = \begin{bmatrix} 0 & 70 & 97 & 75 & 2 & 12 & 66 & 66 & 2 & 70 & 20 & 36 & 53 & 17 & 48 \\ 70 & 0 & 14 & 75 & 27 & 39 & 93 & 24 & 59 & 3 & 41 & 43 & 20 & 9 & 33 \\ 97 & 14 & 0 & 34 & 52 & 99 & 90 & 39 & 50 & 66 & 44 & 29 & 27 & 20 & 31 \\ 75 & 75 & 34 & 0 & 38 & 38 & 78 & 38 & 87 & 38 & 13 & 7 & 95 & 60 & 66 \\ 2 & 27 & 52 & 38 & 0 & 84 & 57 & 10 & 38 & 93 & 24 & 55 & 11 & 35 & 32 \\ 12 & 39 & 99 & 38 & 84 & 0 & 91 & 38 & 61 & 66 & 20 & 44 & 73 & 49 & 21 \\ 66 & 93 & 90 & 78 & 57 & 91 & 0 & 90 & 43 & 2 & 25 & 16 & 55 & 23 & 6 \\ 66 & 24 & 39 & 38 & 10 & 38 & 90 & 0 & 95 & 11 & 61 & 81 & 44 & 63 & 14 \\ 2 & 59 & 50 & 87 & 38 & 61 & 43 & 95 & 0 & 62 & 40 & 53 & 16 & 72 & 51 \\ 70 & 3 & 66 & 38 & 93 & 66 & 2 & 11 & 62 & 0 & 69 & 17 & 90 & 96 & 65 \\ 20 & 41 & 44 & 13 & 24 & 20 & 25 & 61 & 40 & 69 & 0 & 51 & 29 & 69 & 94 \\ 36 & 43 & 29 & 7 & 55 & 44 & 16 & 81 & 53 & 17 & 51 & 0 & 98 & 13 & 100 \\ 53 & 20 & 27 & 95 & 11 & 73 & 55 & 44 & 16 & 90 & 29 & 98 & 0 & 43 & 88 \\ 17 & 9 & 20 & 60 & 35 & 49 & 23 & 63 & 72 & 96 & 69 & 13 & 43 & 0 & 35 \\ 48 & 33 & 31 & 66 & 32 & 21 & 6 & 14 & 51 & 65 & 94 & 100 & 88 & 35 & 0 \end{bmatrix}$$

Solución óptima del Problema 15: $E^* = \{(1,5) (1,6) (1,9) (2,3) (2,8) (2,10) (2,14) (3,13) (3,14) (4,10) (4,11) (4,12) (5,8) (5,11) (5,13) (6,11) (7,10) (7,12) (7,15) (8,15) (9,13) (12,14)\}$

Costo = 303.00 Confiabilidad = 0.9515



Topología óptima para el problema 15

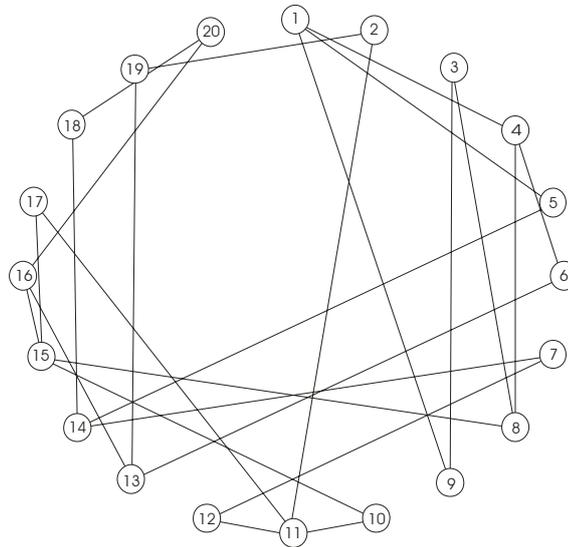
Caso 7: problema 16. Matriz de 20 nodos y 190 enlaces

$$A = \begin{bmatrix} 0 & 67 & 95 & 11 & 13 & 62 & 67 & 69 & 2 & 93 & 49 & 28 & 48 & 63 & 42 & 78 & 82 & 86 & 46 & 77 \\ 67 & 0 & 74 & 83 & 84 & 51 & 79 & 24 & 41 & 42 & 12 & 14 & 68 & 32 & 27 & 10 & 70 & 13 & 7 & 95 \\ 95 & 74 & 0 & 71 & 86 & 97 & 61 & 2 & 2 & 36 & 62 & 94 & 61 & 50 & 56 & 55 & 75 & 49 & 39 & 46 \\ 11 & 83 & 71 & 0 & 31 & 10 & 33 & 13 & 89 & 39 & 96 & 20 & 66 & 76 & 50 & 5 & 25 & 42 & 59 & 43 \\ 13 & 84 & 86 & 31 & 0 & 100 & 72 & 96 & 82 & 38 & 75 & 78 & 15 & 5 & 14 & 64 & 65 & 13 & 95 & 47 \\ 62 & 51 & 97 & 10 & 100 & 0 & 24 & 67 & 17 & 74 & 71 & 19 & 3 & 87 & 41 & 84 & 33 & 60 & 37 & 79 \\ 67 & 79 & 61 & 33 & 72 & 24 & 0 & 57 & 81 & 3 & 52 & 18 & 6 & 4 & 39 & 50 & 66 & 68 & 62 & 30 \\ 69 & 24 & 2 & 13 & 96 & 67 & 57 & 0 & 38 & 98 & 45 & 81 & 77 & 61 & 16 & 61 & 84 & 90 & 31 & 89 \\ 2 & 41 & 2 & 89 & 82 & 17 & 81 & 38 & 0 & 61 & 93 & 31 & 97 & 74 & 25 & 7 & 98 & 27 & 67 & 59 \\ 93 & 42 & 36 & 39 & 38 & 74 & 3 & 98 & 61 & 0 & 13 & 7 & 47 & 53 & 2 & 70 & 61 & 16 & 91 & 69 \\ 49 & 12 & 62 & 96 & 75 & 71 & 52 & 45 & 93 & 13 & 0 & 1 & 29 & 81 & 42 & 95 & 9 & 85 & 14 & 12 \\ 28 & 14 & 94 & 20 & 78 & 19 & 18 & 81 & 31 & 7 & 1 & 0 & 77 & 46 & 82 & 81 & 72 & 2 & 90 & 48 \\ 48 & 68 & 61 & 66 & 15 & 3 & 6 & 77 & 97 & 47 & 29 & 77 & 0 & 40 & 18 & 1 & 47 & 16 & 14 & 34 \\ 63 & 32 & 50 & 76 & 5 & 87 & 4 & 61 & 74 & 53 & 81 & 46 & 40 & 0 & 65 & 84 & 30 & 24 & 48 & 72 \\ 42 & 27 & 56 & 50 & 14 & 41 & 39 & 16 & 25 & 2 & 42 & 82 & 18 & 65 & 0 & 20 & 14 & 36 & 24 & 76 \\ 78 & 10 & 55 & 5 & 64 & 84 & 50 & 61 & 7 & 70 & 95 & 81 & 1 & 84 & 20 & 0 & 89 & 63 & 51 & 16 \\ 82 & 70 & 75 & 25 & 65 & 33 & 66 & 84 & 98 & 61 & 9 & 72 & 47 & 30 & 14 & 89 & 0 & 28 & 49 & 60 \\ 86 & 13 & 49 & 42 & 13 & 60 & 68 & 90 & 27 & 16 & 85 & 2 & 16 & 24 & 36 & 63 & 28 & 0 & 75 & 3 \\ 46 & 7 & 39 & 59 & 95 & 37 & 62 & 31 & 67 & 91 & 14 & 90 & 14 & 48 & 24 & 51 & 49 & 75 & 0 & 58 \\ 77 & 95 & 46 & 43 & 47 & 79 & 30 & 89 & 59 & 69 & 12 & 48 & 34 & 72 & 76 & 16 & 60 & 3 & 58 & 0 \end{bmatrix}$$

Solución óptima del Problema 16: $E^* = \{(1,4) (1,5) (1,9) (2,8) (2,19) (3,8) (3,9) (4,6)$

(4,8) (5,14) (6,13) (7,12) (7,14) (8,15) (10,11) (10,15) (11,12) (11,17) (13,16) (13,19)
 (14,18) (15,16) (15,17) (16,20) (18,20)}

Costo = 235.00 Confiabilidad = 0.9540



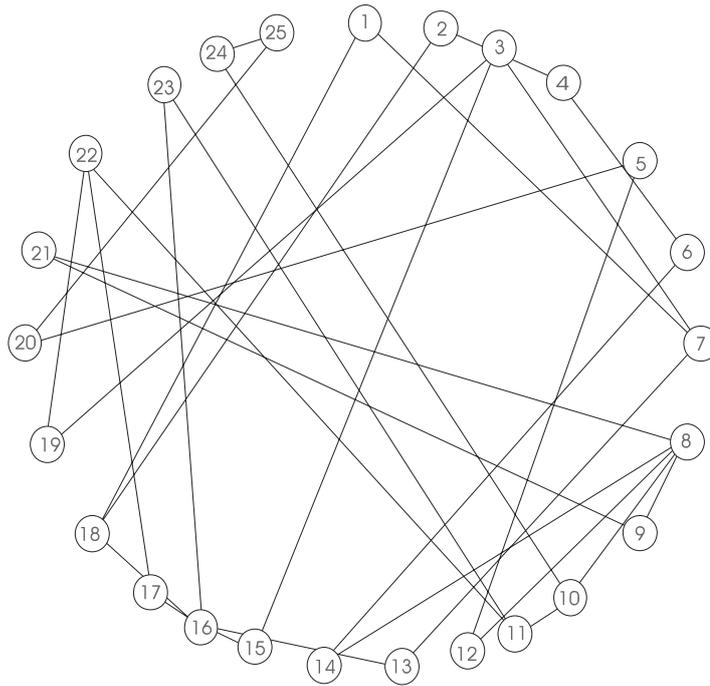
Topología óptima para el problema 16

Caso 8: problema 17. Matriz de 25 nodos y 300 enlaces

	0	34	62	7	46	26	19	31	29	26	15	16	68	37	15	100	58	10	49	86	100	36	31	49	78
	34	0	100	22	98	80	8	98	15	7	80	35	54	85	22	71	81	5	29	46	37	29	79	17	20
	62	100	0	39	25	84	5	36	22	12	86	96	9	79	15	54	42	27	25	39	52	90	80	35	58
	7	22	39	0	19	2	34	4	43	51	64	19	36	26	36	16	71	41	51	52	50	13	34	58	73
	46	98	25	19	0	66	76	84	81	20	45	10	61	34	86	50	18	21	94	25	27	50	61	81	33
	26	80	84	2	66	0	54	18	93	7	62	18	75	28	12	37	73	62	34	89	44	85	96	78	7
	19	8	5	34	76	54	0	57	50	43	48	78	25	53	16	45	55	71	11	69	50	93	86	62	18
	31	98	36	4	84	18	57	0	23	9	73	22	44	24	32	31	3	50	47	76	11	92	63	44	24
	29	15	22	43	81	93	50	23	0	30	63	46	66	44	70	23	10	72	7	63	9	17	87	41	64
	26	7	12	51	20	7	43	9	30	0	42	70	42	78	23	92	6	89	53	55	91	34	12	42	16
	15	80	86	64	45	62	48	73	63	42	0	59	45	83	19	79	21	5	41	56	50	11	6	50	2
	16	35	96	19	10	18	78	22	46	70	59	0	7	46	59	34	51	26	55	32	83	38	85	24	99
A =	68	54	9	36	61	75	25	44	66	42	45	7	0	80	78	6	32	45	97	13	73	69	25	5	72
	37	85	79	26	34	28	53	24	44	78	83	46	80	0	31	55	52	47	93	42	54	24	32	34	16
	15	22	15	36	86	12	16	32	70	23	19	59	78	31	0	1	88	79	57	71	40	92	53	73	89
	100	71	54	16	50	37	45	31	23	92	79	34	6	55	1	0	13	10	48	98	98	23	21	27	28
	58	81	42	71	18	73	55	3	10	6	21	51	32	52	88	13	0	98	29	46	14	3	83	52	50
	10	5	27	41	21	62	71	50	72	89	5	26	45	47	79	10	98	0	82	17	80	33	85	57	99
	49	29	25	51	94	34	11	47	7	53	41	55	97	93	57	48	29	82	0	42	48	45	57	74	52
	86	46	39	52	25	89	69	76	63	55	56	32	13	42	71	98	46	17	42	0	67	72	47	54	42
	100	37	52	50	27	44	50	11	9	91	50	83	73	54	40	98	14	80	48	67	0	23	36	35	61
	36	29	90	13	50	85	93	92	17	34	11	38	69	24	92	23	3	33	45	72	23	0	64	98	51
	31	79	80	34	61	96	86	63	87	12	6	85	25	32	53	21	83	85	57	47	36	64	0	53	74
	49	17	35	58	81	78	62	44	41	42	50	24	5	34	73	27	52	57	74	54	35	98	53	0	4
	78	20	58	73	33	7	18	24	64	16	2	99	72	16	89	28	50	99	52	42	61	51	74	4	0

Solución óptima del Problema 17: $E^* = \{(1,7) (1,18) (2,4) (2,18) (3,7) (3,15) (3,19) (4,6) (5,12) (5,20) (6,14) (7,13) (8,9) (8,10) (8,12) (8,14) (8,21) (9,21) (10,11) (10,24) (11,22) (11,23) (13,16) (15,16) (16,17) (16,18) (16,23) (17,22) (19,22) (20,25) (24,25)\}$

Costo = 523,00 confiabilidad = 0.9067



Topología óptima para el problema 17

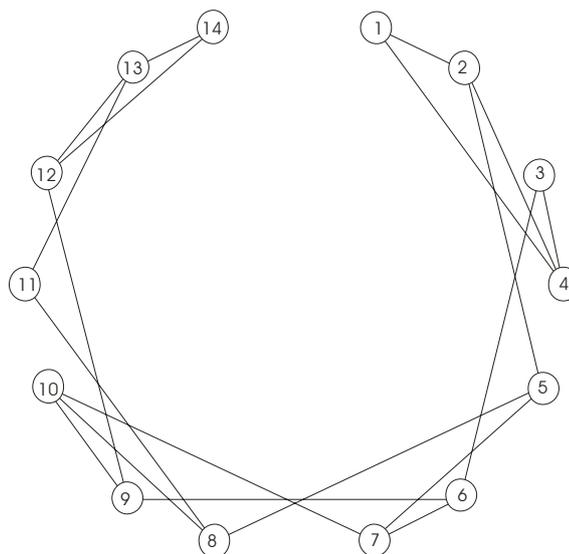
6.4.2 Redes no totalmente conectadas

Caso 9: problema 18. Matriz de 14 nodos y 21 enlaces

$$A = \begin{bmatrix} 0 & 47 & 61 & 20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 46 & 95 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 25 & 0 & 41 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 98 & 45 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & 0 & 42 & 0 & 98 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & 0 & 0 & 0 & 50 & 0 & 0 & 0 & 0 \\ & & & & & & & 0 & 0 & 78 & 46 & 0 & 0 & 0 \\ & & & & & & & & 0 & 22 & 0 & 87 & 0 & 0 \\ & & & & & & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & & 0 & 0 & 60 & 95 \\ & & & & & & & & & & & 0 & 77 & 66 \\ & & & & & & & & & & & & 0 & 10 \\ & & & & & & & & & & & & & 0 \end{bmatrix}$$

Solución óptima del Problema 18: E= {(1, 2) (1,4) (2,4) (2,5) (3,4) (3,6) (5,8) (6,7) (6,9) (7,10) (8,10) (8,11) (9,10) (9,12) (11,13) (12,13) (12,14) (13,14)}

Costo = 1063.00 Confiabilidad = 0.9003

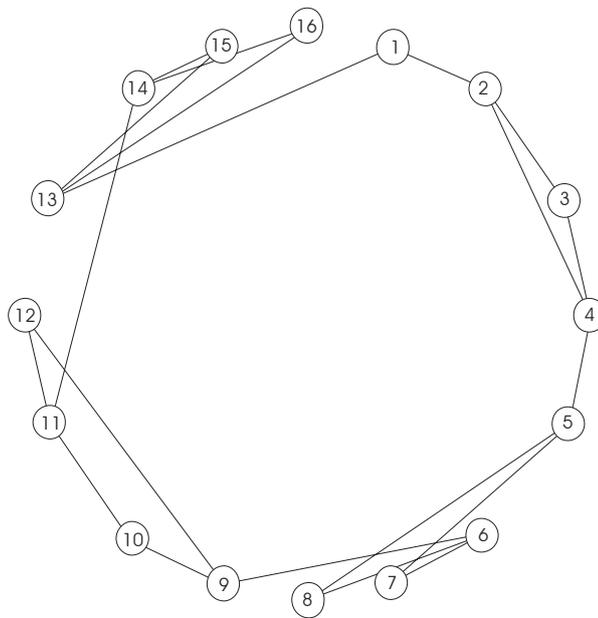


Topología óptima para el problema 18

Caso 10: problema 19. Matriz de 16 nodos y 24 enlaces

$$A = \begin{bmatrix} 0 & 13 & 99 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 0 & 0 \\ 0 & 12 & 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 68 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 39 & 38 & 79 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Solución óptima del Problema 19: $E = \{(1,2) (1,13) (2,3) (2,4) (3,4) (4,5) (5,7) (5,8) (6,7) (6,8) (6,9) (9,10) (9,12) (10,11) (11,12) (11,14) (13,15) (13,16) (14,15) (14,16)\}$
 Costo = 1022.00 Confiabilidad = 0.9502



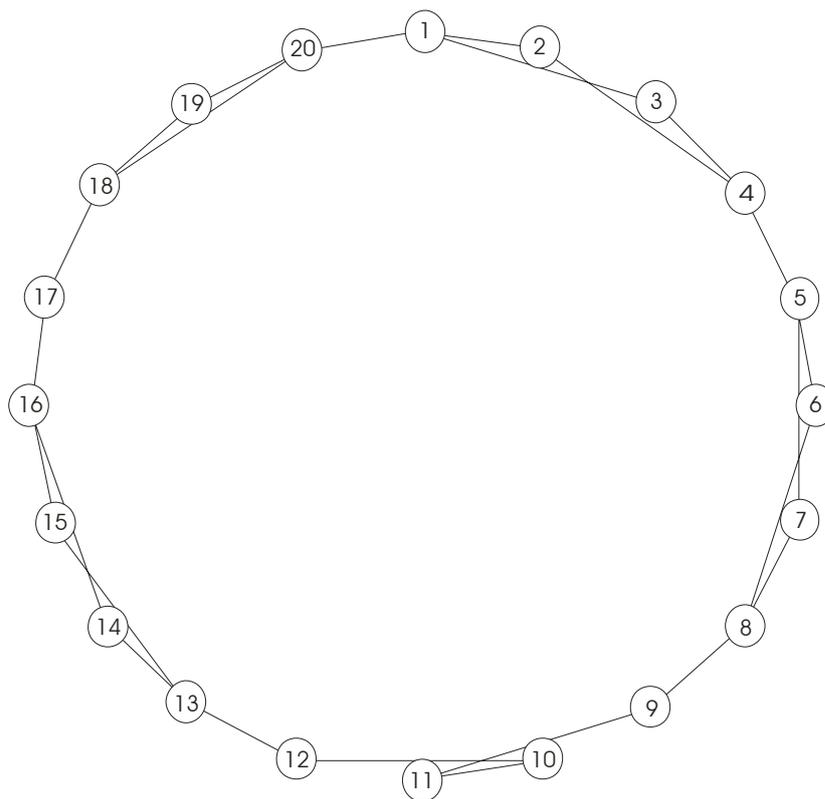
Topología óptima para el problema 19

Caso 11: problema 20. Matriz de 20 nodos y 30 enlaces

$$A = \begin{bmatrix} 0 & 7 & 40 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & 46 & 51 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 34 & 19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 84 & 24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 66 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 60 & 25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 11 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 76 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 21 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 33 & 65 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 59 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 44 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Solución óptima del Problema 20: E = {(1,2) (1,3) (1,20) (2,4) (3,4) (4,5) (5,6) (5,7) (6,8) (7,8) (8,9) (9,11) (10,11) (10,12) (12,13) (13,14) (13,15) (14,16) (15,16) (16,17) (17,18) (18,19) (18,20) (19,20)}

Costo = 596.00 confiabilidad = 0.9013



Topología óptima para el problema 20

6.4.3 Tabla de comparación de resultados

Redes completamente conectadas										
Problema	N	L	Ro	p	Costo óptimo	Resultados del AG				
						Mejor costo Referencia	Costo promedio Referencia	Mejor costo	Costo promedio	Conf.
Problema 1	5	10	0.90	0.80	255	255	255.0	255	257.2	0.9177
Problema 2	5	10	0.95	0.90	201	201	201.0	201	201.0	0.9535
Problema 3	7	21	0.90	0.90	720	720	720.0	720	721.0	0.9000
Problema 4	7	21	0.95	0.90	845	845	857.0	845	845.2	0.9511
Problema 5	7	21	0.95	0.95	630	630	656.0	630	630.0	0.9500
Problema 6	8	28	0.90	0.90	208	208	210.1	208	223.14	0.9257
Problema 7	8	28	0.95	0.90	247	247	249.5	247	254.1	0.9607
Problema 8	8	28	0.95	0.95	179	179	180.3	179	187.4	0.9585
Problema 9	9	36	0.90	0.90	239	239	245.1	239	264.8	0.9036
Problema 10	9	36	0.95	0.90	286	286	298.2	286	305.3	0.9563
Problema 11	9	36	0.95	0.95	209	209	227.2	209	237.02	0.9625
Problema 12	10	45	0.90	0.90	154	156	169.8	154	185.05	0.9005
Problema 13	10	45	0.95	0.90	197	205	206.6	197	229.74	0.9524
Problema 14	10	45	0.95	0.95	136	136	150.4	136	174.51	0.9602
Problema 15	15	105	0.95	0.90	?	317	344.6	303	437.85	0.9515
Problema 16	20	190	0.95	0.95	?	926	956.0	235	309	0.954
Problema 17	25	300	0.90	0.95	?	1606	1651.3	523	553.33	0.9067

Redes no completamente conectadas										
Problema	N	L	Ro	P	Costo óptimo	Resultados del AG				
						Mejor costo referencia	Costo promedio referencia	Mejor costo	Costo promedio	Conf
Problema 18	14	21	0.90	0.90	1063	1063	1076.1	1063	1070.18	0.9003
Problema 19	16	24	0.95(*)	0.95	1022	1022	1032.0	1022	1081.82	0.9502
Problema 20	20	30	0.90	0.95	596	596	598.6	596	606.89	0.9013

(*) La referencia publica erróneamente un valor de 0.90
 ? Óptimo global no encontrado por el método exacto.

CAPÍTULO 7

7 CONCLUSIONES

Los algoritmos evolutivos se basan en la idea de que la evolución es un proceso de optimización que puede ser simulado en un computador, y en que mediante la emulación de sus mecanismos (principalmente cruzamiento, mutación, competencia y reproducción) pueden resolverse problemas de optimización.

Es muy natural, por consiguiente, buscar describir evolución en términos de un algoritmo que puede usarse para solucionar problemas difíciles de optimización en ingeniería. Las técnicas clásicas del gradiente descendiente, el escalado de la colina determinista, y la búsqueda aleatoria pura (sin herencia) han sido generalmente poco satisfactorias cuando han sido aplicadas a la resolución de problemas no lineales de optimización, en particular a problemas combinatorios. Sin embargo, estos problemas han sido resueltos por la naturaleza. La evolución, entonces, provee inspiración para buscar soluciones a problemas que anteriormente aparecían como intratables.

Los AEs brindan una alternativa robusta para la atacar este y otros tipos de problemas combinatorios, la mayoría de los cuales son de muy difícil resolución por cuanto el tiempo de cómputo asociado crece exponencialmente con el tamaño de la instancia del problema.

La incorporación de conocimiento específico del problema, así como la adaptación de estrategias que provienen de otras heurísticas mejoran el desempeño de manera muy importante. Tanto es así que puede afirmarse que, dada la necesidad de encontrar una solución a un problema real de tipo combinatorio es buena práctica la adaptación y articulación de diferentes métodos cualquiera sea su procedencia.

En el desarrollo de esta tesis se ha abordado la resolución del problema de optimización de redes confiables donde la variable de decisión es el costo y con restricciones sobre la confiabilidad. Para ello se han utilizado heurísticas basadas en Algoritmos Evolutivos los cuales han sido hibridizados y enriquecidos para favorecer su desempeño.

Las características del AE desarrollado fueron las siguientes:

- Población inicial sembrada en forma tal que se garantice que todos los individuos presentes verifiquen ciertas características como biconexión, y distribución de enlaces.
- Tratamiento de restricciones utilizando el método de penalización
- Población estructurada con modelo de vecindades lineales
- Selección local por ranking
- Hibridización

El desempeño del algoritmo se analizó para instancias de diferentes tamaños desde una pequeña red de 5 nodos con un espacio de búsqueda de tamaño $1.02 \cdot 10^3$ hasta una red con 25 nodos y 300 enlaces con un tamaño de espacio de búsqueda del orden de 10^{90} . Como puede observarse en los resultados plasmados en el capítulo 6, se consiguen buenas soluciones con un espacio explorado reducido. De hecho la cantidad de evaluaciones de la función objetivo utilizada para medir el desempeño del algoritmo diseñado es baja si se la compara con el espacio de búsqueda que corresponde a cada caso analizado.

Las principales ventajas observadas en la utilización de AE para la resolución de diseños de redes de instrumentos son las siguientes:

- 1- No requieren de una formulación matemática especial de la función objetivo, más aún, ellos están diseñados para utilizar las funciones objetivo como cajas

negras, por lo tanto pueden adaptarse a la resolución de diferentes diseños de instrumentación, incluyendo diseños con criterios múltiples de desempeño.

- 2- En problemas combinatorios complejos, donde la formulación particular no admite la obtención de soluciones óptimas mediante un algoritmo de tiempo polinomial, los AE pueden proveer buenas soluciones en tiempos de ejecución razonables.
- 3- A diferencia de otras metaheurísticas utilizadas para la resolución de problemas combinatorios, ellos exploran en forma simultánea diferentes regiones del espacio de búsqueda. Esto significa que en cualquier estadio de la evolución del algoritmo se tiene un conjunto de candidatos o potenciales soluciones, en contraste con los métodos secuenciales o constructivos que consideran una única solución o una parte de ella respectivamente.
- 4- Resulta relativamente fácil ejecutarlos en las modernas arquitecturas masivas en paralelo. Esta implementación puede ser deseable en los casos en que estén involucrados diseños con un gran número de variables de proceso y por lo tanto la demanda de recursos computacionales se incrementa considerablemente.

Finalmente a continuación se detallan futuras tareas de investigación:

- Investigar diferentes modelos de AE desde el punto de vista de la estructura de la población. Concretamente experimentar con otras topologías para el modelo de vecindades y con el modelo de islas.
- Implementar los modelos anteriores sobre arquitecturas en paralelo.
- Abordar el problema como uno de optimización multiobjetivo. Se ha observado en algunos casos que el algoritmo encuentra soluciones de muy buena calidad desde el punto de vista del costo que son penalizadas por no cumplir las restricciones de confiabilidad. Sin embargo, desde un punto de vista ingenieril, el diseñador puede tomar la decisión de flexibilizar la restricción (sobre todo cuando la diferencia es pequeña) en función del beneficio económico. Un enfoque multiobjetivo permite brindar la información necesaria para tomar dicha decisión.

- Comparar el desempeño del AE frente a otras técnicas heurísticas, tales como búsqueda tabú o búsqueda dispersa.

8 Referencias y bibliografía

- Aggarwal K. K, S. Rai, “Reliability evaluation in computer-communication networks,” IEEE Transactions on Reliability, vol R-30, 1981 Apr, pp 32-35.
- Aggarwal K. K, Y. C. Chopra, J. S. Bajwa, “Reliability evaluation by network decomposition”, IEEE Transactions on Reliability, vol R-31, 1982 Oct, pp 355-358.
- Alba, E. and M. Tomassini, “Parallelism and Evolutionary Algorithms”, IEEE Transactions of Evolutionary Computation, 6, 443-462 (2002).
- Bäck, Thomas. Optimal Mutation Rates in Genetic Search. In Stephanie Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pages 2–8. Morgan Kaufmann Publishers, San Mateo, California, July 1993.
- Bäck, T., Hammel, U. and Schwefel, P. “Evolution Computation: Comments on the History and Current State”. IEEE Transactions of Evolutionary Computation, 1, 3-17 (1997).
- Bagajewicz, M. and Cabrera E., “A New MILP Formulation for Instrumentation Network Design and Upgrade,” Proc. of IFAC Workshop on On-line Fault Detection and Supervision in the Chemical Industries, Seoul, Korea, 303-308 (2001).
- Baker, James Edward. Adaptive Selection Methods for Genetic Algorithms. In John J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms*, pages 101–111. Lawrence Erlbaum Associates, Hillsdale, New Jersey, July 1985.
- Ball, M., Colbourn C. y Provan J., “Network Reliability” (Reporte Técnico 92-94) University of Maryland, University of Waterloo, University of North Carolina. (1992).
- Bickle, T. and Thiele, L. “A Comparison of Selection Schemes used in Evolutionary Algorithms”, Evolutionary Computation, 4, 361-394 (1996).

- Brecht, Timoty, “Lower Bounds for Two Terminal Network Reliabilit”, Thesis 1995.
- Cancela Bosi, M.E. Urquhart, G. Rubino, Network reliability evaluation by the Ahmad Method H. en Proceedings de XXIII Conferencia Latinoamericana de Informática CLEI, Valparaiso, Chile (1997).
- Cancela H y El Khadiri M. A Recursive variance-reduction algorithm for estimating communication network reliability. IEEE Transations on Reliability, 44(4):595-602, 1995.
- Cancela, H., Rubino, G. Urquhart, M. “Evaluation and design of communication networks” Proceedings of the ICIL'95, Ouro Preto, Brazil, December 1995. University of Southampton, UK, and Naval Monterrey School, USA.
- Cancela, H., Urquhart, M, y Rubino, G., “HEIDI – Una herramienta de apoyo para el diseño de redes de comunicaciones, InCo – Facultad de Ingeniería, Uruguay, IRISA- INRIA, Remes, Francia. (1993).
- Cavers J. K., “Cutset manipulations for communication network reliability estimation,” IEEE Transactions on Communications, vol Com-23, 1975 Jun, pp 569-575.
- Coello Coello, Carlos A. Introducción a la Computación Evolutiva (Notas de Curso), Departamento de Ingeniería Eléctrica Sección de Computación, mayo 2003.
- Colbourn, C., “Combinatorial Aspects of Networks Reliability”. Annals of Operation Research, Vol 33, pp 3-15, 1991.
- Collins, J. “Studies in Artificial Evolution”. PhD Thesis, University of California, 1992.
- Davis, L., Applying Adaptive Algorithms to Epistatic Domains, Proceedings of the International Joint Conference on Artificial Intelligence, pp.162-164, 1985.
- De Jong, A. K. An Analysis of the Behavior of a Class of Genetic Adapative Systems. PDH Thesis, University of Michigan, 1975.

- Deb, K., “An efficient Constraint Handling Method for Genetic Algorithms”, *Comp. Meth. Appl. Mech.Eng.*, 186, 311-338 (2000).
- Dengiz Berna, Fulya Altipamak, “Efficient optimization of all-terminal reliable network using evolutionary approach”. *IEEE Transation on Reliability*, march 1997.
- Dengiz, B. “Efficient Optimization of All Reliable networks, using an Evolutionary Approach”. (*IEEE transaction on reliability vol 46 nro 1 1997*).
- Deo, N., *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall, Englewood Cliffs, N.J. (1974).
- Eshelman, Larry J., Richard A. Caruana, and J. David Schaffer. Biases in the Crossover Landscape. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 10–19, San Mateo, California, 1989. Morgan Kaufmann Publishers.
- Fard, N.; Lee, T. Spanning tree approach in all-terminal network reliability expansion. *Computer Communications*, N° 24. pp 1348-1353. 2001
- Fishman G.S. (1997) *Monte Carlo: Concepts, algorithms and applications*. Springer
- Fogarty, Terence C., Varying the Probability of Mutation in the Genetic Algorithm. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 104–109, San Mateo, California, 1989. Morgan Kaufmann Publishers.
- Garey, M. and D.Johnson, *Computer and Intractability. A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York (1999).
- Garey, M. and D.Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co, San Francisco (1979).
- Gen, M and Cheng, R., “Genetic Algorithms & Engineering Optimisation”. *Wiley Serires in Engineering Design and Automation*. John Wiley & Sons, New York, 2000.

- Gen, M., *Genetic Algorithms and Engineering Design*, Wiley & Sons, Inc. New York (1996).
- Glover, F; Laguna, M. *Tabu Search*. 6th Ed. Kluwer Academic Publishers. 2002.
- Goldberg, D. E., Lingle, R. Alleles, Loci, and the TSP. En *Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Hillsdale, NJ. 154-159, 1985.
- Golberg, David E., *Genetic Algorithms in Search, Optimizations and Machine Learning*. Addison Wesley Publishing Co., Reading Massachusetts 1989.
- Holland, John H. *Adaptation in Natural and Artificial Systems*. Ann Harbor: University of Michigan Press, 1975.
- Homaifar A, S. H. Y. Lai, and X. Qi. *Constrained Optimization via Genetic Algorithms*. *Simulation*, 62(4):242–254, 1994.
- Jan, R. H., “Design of Reliable Networks”, *Computers and Operations Research*, vol. 20, January 1993, pp 25-34 1993.
- Jan, R. H., Hwang, F.J., Cheng, S.T., “Topological Optimization of a Communication Network Subject to a Reliability Constraint”, *IEEE Transactions on Reliability*, vol. 42, No. 1, March 1993 pp 63-70.
- Joines, J. and C. Houck. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs. In David Fogel, editor, *Proceedings of the first IEEE Conference on Evolutionary Computation*, pages 579–584, Orlando, Florida, 1994. IEEE Press.
- Koide, T., Shinmori S., Ishii H., “Topological Optimization with a Network Reliability Constraint”, *Discrete Applied Mathematics* 115 (March, 2001), Japan 135-149.
- Lashon B. Booker. *Improving Search in Genetic Algorithms*. In Lawrence Davis, editor, *Genetic Algorithms and Simulated Annealing*. Morgan Kaufmann Publishers, San Mateo, California, 1987.

- Lashon B. Booker. Intelligent Behavior as an Adaptation to the Task Environment. PhD thesis, Logic of Computers Group, University of Michigan, Ann Arbor, Michigan, 1982.
- Lohman, R., Applications of Evolution Strategy in Parallel Population. In Parallel Problem Solving from Nature – PPSN I, pp. 198-208, 1991.
- Martí, R; Laguna, M.; Glover, F. Principles of scatter search. European Journal of Operations Research. pp 359-372. 2006
- Michalewicz Zbigniew. A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. In J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, editors, Proceedings of the 4th Annual Conference on Evolutionary Programming, pages 135–155. The MIT Press, Cambridge, Massachusetts, 1995.
- Michalewicz Zbigniew. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, New York, third edition, 1996.
- Mitchell, Tom M. The Need for Biases in Learning Generalizations. Technical Report CBM-TR-117, Department of Computer Science, Rutgers University, 1980.
- Nakawaza H., “Decomposition method for computing the reliability of complex networks,” IEEE Transactions on Reliability, vol R-30, 1981 Aug, pp 289-292.
- Nemhauser, G. and Wolsey, L. Integer and Combinatorial Optimization, Wiley Interscience, New York, (1999).
- Oliver, I.M., Smith, D.J. y Holland, J.R.C. A Study of Permutation Crossover Operators on the Traveling Salesperson Problem, Proceedings of the Second International Conference on Genetic Algorithms and their Applications (pp. 224-230). Grefenstette, J.J. (ed.), Lawrence Erlbaum Associates, Hillsdale, 1987.
- Papadimitrou, C., and K. Steiglitz, Combinatorial Optimization Algorithms and Complexity, Prentice Hall, New Jersey (1982).

- Pohlheim, Harmut, “Evolutionary algorithms: Overview, Methods and Operators”. Genetic and Evolutionary Algorithms toolbox for use with Matlab (august 1999)
- Radcliffe, Nicholas J. Forma Analysis and Random Respectful Recombination. In Richard K. Belew and Lashon B. Booker, editors, Proceedings of the Fourth International Conference on Genetic Algorithms, pages 222–229. Morgan Kaufmann Publishers, San Mateo, California, July 1991.
- Reichelt, D.; Rothlauf, F. “Reliable Communication Network Design with Evolutionary Algorithms”. ResearchPaper Fakultät für Betriebswirtschaftslehre. 2004.
- Reichelt, D.; Gmilkowsky, P.; Linser, S. “A Study of an Iterated Local Search on the Reliable Communication Networks Design Problem”. EvoWorkshops 2005. Pp156-165. 2005
- Rai S., “A cutset approach to reliability evaluation in communication networks,” IEEE Transactions on Reliability, vol R-31, 1982 Dec, pp 428-431.
- Sarma, J. De Jong, K., “An Analysis of Local Selection Algorithms in a Spatially Structured Evolutionary Algorithm”, Proceedings of ICGA-97, the 7th International Conference on Genetic Algorithms, Lansing, MI, July 1997.
- Schoenauer Marc, Evolutionary Computation and Applications at Centre de Mathématiques Appliquées de L’Ecole Polytechnique, nov. 1995, revised march. 1997.
- Srivaree-Ratana, C.; Konak, A.; Smith, A. “Estimation of all-terminal Network Reliability using an Artificial Neural Network”. Computers and Operations Research, V. 29, pp 849- 868. 2002.
- Wetzel A. Evaluation of the effectiveness of genetic algorithms in combinational optimization. University of Pittsburgh, Pittsburgh (unpublished), 1983