

Resolución del Problema de Steiner Generalizado utilizando un Algoritmo Genético Paralelo

Miguel Aroztegui

Santiago Árraga

Sergio Nesmachnow

Instituto de Computación

Facultad de Ingeniería, Universidad de la República

Herrera y Reissig 565, Montevideo, Uruguay.

Resumen

Dado un grafo con un conjunto distinguido de nodos llamados terminales sobre los cuales se definen requerimientos de conexión, el Problema Generalizado de Steiner consiste en encontrar un subgrafo de costo mínimo tal que para todo par de nodos terminales del grafo exista un número de caminos disjuntos mayor o igual al requerimiento de conexión correspondiente. Este problema ha sido utilizado para modelar el diseño de redes de comunicaciones confiables, donde es necesario garantizar con alta probabilidad la conexión entre diversos puntos de la red. Se trata de un problema NP difícil, para el que existen pocos algoritmos propuestos. Este trabajo presenta un algoritmo genético para resolver el Problema Generalizado de Steiner. Para poder atacar instancias medianas del problema en un tiempo razonable, además del modelo panmítico serial se estudia un modelo paralelo. Los resultados obtenidos indican que el modelo paralelo obtiene un speedup casi lineal en tiempo de ejecución sobre el modelo serial, sin pérdida de calidad de las soluciones obtenidas.

Palabras clave: Problema generalizado de Steiner, Algoritmos Genéticos, Paralelismo.

1. Introducción

El continuo crecimiento en el tamaño de los problemas de diseño de redes de comunicaciones ha conducido a los investigadores a proponer alternativas a los enfoques exactos tradicionales para la resolución de problemas complejos. En este contexto, varias heurísticas se han aplicado al problema de diseño de redes de comunicaciones confiables y las técnicas de programación evolutiva, y los algoritmos genéticos (AG) en particular, se han manifestado como métodos flexibles y robustos para la solución de los complejos problemas de optimización subyacentes.

Considerando una red de comunicaciones con nodos distinguidos denominados terminales, el Problema de Steiner Generalizado (Generalized Steiner Problem – GSP) refiere al diseño de una subred de mínimo costo, que verifique ciertos requerimientos prefijados de conexión entre pares de nodos terminales. Con respecto a otros problemas de diseño de redes de comunicaciones, el GSP incorpora requisitos adicionales de conectividad para garantizar la alta confiabilidad, haciendo al funcionamiento de la red más resistente a fallas en sus componentes.

Las técnicas de procesamiento paralelo y distribuido se aplican al modelo clásico de AG con el objetivo de obtener mejoras de eficiencia y perfeccionar la calidad de la búsqueda [19]. Desde la perspectiva de la eficiencia, paralelizar un AG permite afrontar la lentitud de convergencia para problemas cuya dimensión motiva el uso de poblaciones numerosas, o múltiples evaluaciones de funciones de fitness complejas. Desde el punto de vista algorítmico, los Algoritmos Genéticos Paralelos (AGP) pueden explotar el paralelismo intrínseco del mecanismo evolutivo, trabajando simultáneamente sobre varias poblaciones semi independientes para resolver un problema. Eventuales intercambios de soluciones (migraciones) introducen diversidad para evitar problemas de convergencia en óptimos locales.

Este trabajo describe los primeros resultados en la resolución del Problema de Steiner Generalizado utilizando un Algoritmo Genético, desarrollado en el marco del proyecto "Algoritmos Genéticos Paralelos y su Aplicación al Diseño de Redes de Comunicaciones Confiables"¹. El resto del artículo se organiza del modo que se describe a continuación: la sección 2 brinda una presentación del Problema de Steiner Generalizado y sus variantes, la sección 3 explica los detalles de diseño y configuración del AGP, la sección 4 resume los resultados obtenidos en la validación y calibración del AGP y en la resolución del GSP y la sección 5 ofrece las conclusiones y líneas de trabajo actual y futuro.

2 – El problema de Steiner Generalizado

2.1 – Formulación del GSP

La siguiente formulación del GSP se basa en la definición original de Krarup [18].

Sean los siguientes elementos :

- Un grafo no dirigido $G=(V,E)$, con una matriz de costos c asociados a sus aristas.
- Un conjunto distinguido de nodos terminales T , de cardinalidad $n_T=|T|$, tal que $2 \leq n_T \leq n$, siendo $n=|V|$ la cardinalidad del conjunto de vértices del grafo G .
- Una matriz $R=\{r_{ij}\}$ con $i,j \in T$, de dimensión $n_T \times n_T$, cuyos elementos son enteros positivos que indican los requerimientos de conectividad –cantidad de caminos– entre todo par de nodos terminales.

¹ Financiado por CSIC (Comisión Sectorial de Investigación Científica, Univ. de la República, Uruguay)

El problema GSP propone encontrar un subgrafo G_T de G , de costo mínimo, tal que para todo par de nodos $i, j \in T$, $i \neq j$ estos sean r_{ij} arista conexos (es decir, que existan r_{ij} caminos disjuntos, que no comparten aristas, entre los nodos terminales i y j).

Sobre los nodos no pertenecientes al conjunto de nodos terminales no se plantean requisitos de conectividad. Estos nodos, llamados *nodos de Steiner*, pueden formar parte o no de la solución óptima, de acuerdo a la conveniencia de utilizarlos.

2.2 – Modelo Matemático del GSP

El modelo del problema GSP arista conexo define para cada arista $(i, j) \in E$ una variable binaria x_{ij} , y una variable real y_{ij}^{kl} que denota la utilidad de la arista (i, j) en la dirección de i hacia j en un camino que une a los nodos terminales k y l .

El cuadro 1 presenta el modelo matemático del GSP como problema de Programación Lineal Entera.

$$\begin{aligned} & \text{Min} \sum_{(i,j) \in E} c_{ij} \cdot x_{ij} \quad \text{sujeto a} \\ & x_{ij} \geq y_{ij}^{kl} + y_{ji}^{kl} \quad \forall (i, j) \in E, \forall k, l \in T, k \neq l \\ & \sum_{(k,j) \in E} y_{kj}^{kl} \geq r_{kl} \quad \forall k, l \in T, k \neq l \\ & \sum_{(i,l) \in E} y_{il}^{kl} \geq r_{kl} \quad \forall k, l \in T, k \neq l \\ & \sum_{(p,j) \in E} y_{pj}^{kl} - \sum_{(i,p) \in E} y_{ip}^{kl} \geq 0 \quad \forall k, l \in T, \forall p \in V \setminus \{k, l\} \\ & x_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \\ & y_{ij}^{kl} \geq 0 \quad \forall i, j : (i, j) \in E, \forall k, l \in T, k \neq l \end{aligned}$$

Cuadro 1: Modelo matemático del GSP como problema de programación lineal entera

Llamando $U = \{u_{ij}\}$ a la solución óptima del problema, el conjunto de aristas determinado por $\{(i, j) \in E / u_{ij} = 1\}$ define el subgrafo solución G_{SOL} .

2.3 – Casos particulares del problema de Steiner

La complejidad del problema de Steiner obedece a la generalidad de su planteo, al exigir requerimientos variables de conectividad entre pares de nodos terminales que implican la existencia de un número variable de caminos disjuntos entre nodos en la solución.

Los *problemas de k conexión* simplifican los requerimientos, exigiendo una cantidad k (fija) de caminos disjuntos entre pares de nodos terminales, mientras que el caso más simple de problema de Steiner solo exige un camino entre pares de nodos ($k=1$) y se conoce como *problema del árbol de Steiner*.

2.4 – Complejidad del problema de Steiner

El Problema Generalizado de Steiner es NP-Completo, tal cual fue demostrado por Krarup [18].

El propio problema del árbol de Steiner, que plantea la restricción menos general de caminos, es NP completo, de acuerdo a la prueba de Karp [17].

La complejidad del problema de Steiner hace que su resolución utilizando algoritmos exactos sea cada vez menos tratable al aumentar el tamaño del problema. Por tal motivo, se buscan soluciones alternativas utilizando heurísticas que permitan encontrar buenas soluciones en tiempos razonables.

2.5 – Técnicas evolutivas aplicadas a la resolución del problema de Steiner

El estudio del estado del arte realizado [19] [20] permitió comprobar que no existen antecedentes de aplicación de los Algoritmos Genéticos al Problema Generalizado de Steiner.

Existen varias referencias a aplicaciones de técnicas evolutivas al problema del árbol de Steiner y sus variantes. Hesser et al [11] proponen un algoritmo genético para la optimización de árboles de Steiner a partir árboles de cubrimiento.

El Problema de Steiner Rectilíneo ha sido abordado mediante AG. Este problema, que tiene aplicación directa en el diseño de redes, sistemas mecánicos, tableros de circuitos y paquetes VLSI consiste en hallar el árbol de Steiner de costo mínimo sobre un grafo planar. Los nodos se asumen ubicados en el plano euclídeo y las conexiones entre nodos deben ser horizontales o verticales. Entre los trabajos importantes pueden citarse los de Kapsalis et al. [16], Julstrom [13] [14] y [15] y Wakabayashi [26].

Esbensen [5] propuso alternativas a la propuesta de Kapsalis. Impuso el trabajo con soluciones factibles y evitando términos de penalización en la función de fitness. Sobre esta idea, Esbensen y Mazumder [6] diseñaron un AG para el problema del árbol de Steiner y su aplicación al diseño de circuitos VLSI, obteniendo soluciones de calidad superior respecto a dos heurísticas bien conocidas para el problema, siendo competitivo en términos de performance.

Diversos problemas sobre redes de comunicaciones tienen como solución el problema del árbol de Steiner, y sobre ellos se han aplicado AG. Zhu et al. [28] utilizaron un híbrido para hallar árboles de Steiner, combinando la idea de Esbensen y un AG propio para optimizar requerimientos de conexiones multipunto sobre la red de comunicaciones. Voß y Gutenschwager [25] investigaron sobre la hibridación entre AG y técnicas de inteligencia artificial aplicadas al problema del árbol de Steiner. Hwang et al. [12] y Zhou et al. [27] propusieron AG aplicados al problema de ruteo multicasting.

Galiasso y Wainwright [9] implementaron un AG híbrido para el problema de ruteo multipunto, que extiende las ideas de Zhu et al. [28] y Esbensen [5] determinando el orden óptimo de procesamiento de requerimientos y porcentajes de ancho de banda asignados a los diversos caminos de comunicación.

3 – El Algoritmo Genético Paralelo

Investigar la aplicabilidad de los Algoritmos Genéticos para resolver problemas de confiabilidad en redes de comunicaciones es uno de los objetivos del proyecto "Algoritmos Genéticos Paralelos y su Aplicación al Diseño de Redes de Comunicaciones Confiables". Tomando en cuenta la dimensión de los problemas, se propuso implementar modelos paralelos de Algoritmos Genéticos para manejar eficientemente poblaciones numerosas.

El Algoritmo Genético Paralelo diseñado en esta primera instancia corresponde a un modelo de población distribuida, organizada en subpoblaciones semi independientes que evolucionan en paralelo. De acuerdo a la taxonomía de Nowostawski y Poli [22] se clasifica dentro de la categoría de modelos distribuidos de subpoblaciones con migración.

3.1 – Codificación y operadores

La elección de una codificación adecuada es un factor influyente sobre el funcionamiento del algoritmo genético, y su importancia es destacada en la literatura especializada [8] [5].

En el modelo implementado se trabajó con una codificación binaria simple, donde cada cromosoma se representa como un array de n bits, siendo n el número de aristas del grafo. Las aristas se numeran de 0 a $n-1$ y la presencia de una arista en el grafo queda determinada por el valor 1 en la posición correspondiente en el cromosoma.

1	0	1	1	1	0	0	...	1
---	---	---	---	---	---	---	-----	---

Figura 1 – Codificación binaria simple

Los operadores de evolución son sencillos al utilizar esta representación, pero se cuenta con el inconveniente de que luego de un cruzamiento simple los cromosomas resultantes pueden no ser solución factible del problema de Steiner.

Luego de cada cruzamiento o mutación, es necesario verificar la factibilidad del nuevo cromosoma. Por simplicidad, si el cromosoma no es factible se lo descarta, evitando la tarea de cuantificar cuan lejos de una solución factible se encuentra y determinar un valor de fitness adecuado. Esta idea sigue la propuesta de Esbensen en sus trabajos sobre el problema del árbol de Steiner [5] [6].

Los operadores utilizados por el AG corresponden a

- Selección proporcional al fitness escalado.

- Cruzamiento de un punto, aleatorio en el rango $[0, n-1]$, que intercambia los trozos del cromosoma.
- Mutación de inversión de alelo probabilísticamente seleccionado.

3.2 – Función de fitness

La función de fitness utilizada evalúa el costo del grafo representado por el cromosoma. Corresponde

$$a \text{ la función } F = C - \sum_{i=0}^{n-1} [BIT(i) * COSTO(i)]$$

donde C representa el costo del grafo original, $COSTO : N \rightarrow R$ devuelve el costo de una arista y $BIT : N \rightarrow [0,1]$ retorna el valor correspondiente a la arista i -ésima en el cromosoma.

Para evitar problemas de convergencia prematura y de indiferencia en la diversidad que pueden afectar el mecanismo evolutivo, se implementó un escalado lineal del fitness, de acuerdo a Goldberg [8].

Tomando en cuenta el promedio de fitness de la población f_{AVG} , el máximo fitness f_{MAX} , y un factor de escalado C , se define una función F mediante:

- (1) $F(f_{AVG}) = f_{AVG}$
- (2) $F(f_{MAX}) = C f_{AVG}$

La ecuación (1) asegura que los cromosomas de fitness medios tengan una copia en las nuevas generaciones. La ecuación (2) regula el fitness de los cromosomas con fitness máximo. La constante C corresponde al número de copias esperadas para el individuo con máximo fitness. El escalado tiende a separar el fitness de los cromosomas de f_{AVG} .

3.3 – Cálculo de factibilidad

Para determinar si un cromosoma es factible, se verifica que el grafo representado cumpla las restricciones del GSP.

Una sencilla heurística de orden $O(t^2)$, siendo t la cardinalidad del conjunto de nodos terminales, se utiliza para verificar los grados de los vértices y las restricciones de caminos impuestas.

Si el grafo cumple la condición anterior, se utiliza el algoritmo de Ford–Fulkerson [7] para hallar los caminos entre cada par de vértices terminales, asignando a un vértice el rol de fuente y al otro el rol de pozo. Como el algoritmo de Ford–Fulkerson trabaja sobre grafos dirigidos, cada arista del grafo se considera como un par de aristas de sentidos opuestos. La capacidad de las aristas se asume unitaria, de modo que el flujo máximo entre fuente y pozo coincide con el máximo número de caminos disjuntos entre los mismos. Si éste es menor que el valor del requerimiento, el grafo es no factible.

El algoritmo descrito utiliza una función de capacidad de flujo unitaria (integral) para cada arista, y tiene orden $O(|E| \cdot r_{MAX} \cdot t^2)$, siendo E el conjunto de aristas, y r_{MAX} el máximo de los requerimientos, de acuerdo a Baratz [1].

3.4 – Modelo de paralelismo

El modelo de paralelismo implementado utiliza subpoblaciones que evolucionan en forma semi independiente. Luego de un número determinado de generaciones, cada población selecciona un conjunto de individuos para migrar a poblaciones vecinas. La topología de migración entre poblaciones es estática y se encuentra definida en el archivo de configuración del sistema.

El cuadro 2 ofrece el seudocódigo del AGP diseñado

```
Generar población inicial pop con sizepop cromosomas.
Mientras (no convergencia)
    escalar fitness de población
    Crear nueva población vacía newpop
    Mientras (tamaño de newpop < sizepop)
        Seleccionar 2 cromosomas padre de pop.
        Cruzar y mutar, se obtienen hijo1 e hijo2
        Si hijo1 es factible
            Insertar cromosoma1 en newpop.
        Si hijo2 es factible
            Insertar cromosoma2 en newpop.
    Recibir y enviar inmigrantes.
    pop = newpop
```

Cuadro 2 : Seudocódigo del AGP diseñado

Cada proceso del sistema manipula una subpoblación del AGP independientemente. Un proceso actúa de controlador del sistema, manteniendo la información de la mejor solución encontrada hasta la generación actual y su valor de fitness, de acuerdo a las comunicaciones en cuales se envían emigrantes por parte de las poblaciones.

La migración utiliza selección proporcional para elegir los individuos emigrantes y selección proporcional inversa para determinar los individuos reemplazados en las poblaciones destino, seleccionando con mayor probabilidad individuos de peor fitness. Este modelo constituye una variante probabilística del mecanismo elitista descrito por Cantú-Paz [4] donde los mejores individuos de cada población emigran y reemplazan a los peores individuos en la población destino.

La comunicación de cromosomas emigrantes se lleva a cabo de forma asincrónica, mediante envío y recepción de mensajes no bloqueantes. Esta característica permite al algoritmo continuar su procesamiento sin necesidad de confirmación del arribo de sus mensajes a destino, o de bloquearse aguardando por mensajes dirigidos a él.

Varias condiciones fueron implementadas como criterios de parada independientes. El AGP se detiene cuando el proceso controlador alcanza el número máximo de generaciones especificado en la configuración, cuando el tiempo de ejecución (en segundos) del proceso controlador supera el fijado en la configuración, o cuando el AGP ha convergido

($\frac{3}{4}$ de sus subpoblaciones cumplen que la relación entre fitness promedio en la generación actual y en la generación anterior es menor que un valor indicado en la configuración).

Cuando el AGP finaliza su ejecución, se presenta la mejor solución encontrada hasta el momento.

3.5 – Implementación y plataforma de ejecución

Para codificar el AGP se utilizó el lenguaje C++, con el compilador GNU G++ y la implementación MPICH de la biblioteca de desarrollo de programas paralelos y distribuidos MPI [10].

La ejecución distribuida se realizó sobre un cluster de cuatro equipos Intel monoprocesador (Pentium III, 400 MHz) con sistema operativo SuSE Linux 8.0 conectados mediante una LAN ethernet a 100 Mbps

4 - Resultados obtenidos

4.1 – Validación del AGP

Para validar el funcionamiento del AG diseñado, se utilizó su versión serial para encontrar soluciones al GSP sobre grafos para los cuales se dispone de una solución conocida. Los grafos de prueba utilizados se obtuvieron de Robledo [23], posibilitando la comparación de las soluciones obtenidas con las presentadas en el referido trabajo.

Para la validación el AGP fue configurado con los parámetros que se detallan en la tabla 1.

Número de procesos	1
Condición de parada	100 generaciones
Tamaño de la población	30 individuos.
Probabilidad de mutación	0.00333
Probabilidad de cruzamiento	0.95
Factor de escalado de fitness	2

Tabla 1 : configuración del AGP para validación

Los valores de probabilidad de mutación y cruzamiento y escalado de fitness utilizados corresponden a sugerencias de Goldberg [8].

Las características de los grafos utilizados se resumen en la tabla 2, conjuntamente con los resultados obtenidos en cada caso.

Nodos	Aristas	Terminales	Costo [23]	Costo AG
5	9	2	5	5
6	9	3	7	7
6	14	6	9	9
5	16	4	22	18
10	15	6	20	20
16	25	6	41	39
6	3	11	9	9

Tabla 2 : resultados de la prueba de validación

Las soluciones encontradas por el algoritmo coincidieron con las soluciones publicadas en [21] para 5 de los 7 problemas de prueba. Para dos de los problemas el AG encontró mejores soluciones que las publicadas, obtenidas con heurísticas basadas en Ant Systems.

4.2 – Configuración del modelo de migración

Para obtener una buena configuración del modelo paralelo, se investigó la sensibilidad de los resultados en función de los parámetros que caracterizan al modelo de migración.

El objetivo de los experimentos de configuración consistió en comprender los efectos de la migración en la intensidad de selección de individuos y sus implicancias en el mecanismo evolutivo.

Los parámetros que caracterizan al modelo de migración se describen a continuación.

- Frecuencia : Número de generaciones ocurridas entre dos migraciones consecutivas.
- Tasa de migración: Porcentaje de individuos de la población local que emigra.
- Topología de migración: Grafo dirigido cuyos nodos representan poblaciones y sus arcos indican la migración de individuos desde población origen a población destino.
- Tamaño y número de subpoblaciones en las cuales se organiza el modelo.
- Política de migración: determina que individuos son elegidos para migrar y cuáles individuos se reemplazan en la población destino.

Tomando en cuenta la limitación del equipamiento disponible, se decidió trabajar con un número fijo de subpoblaciones (4 demes, uno por equipo).

Se trabajó con la estrategia elitista probabilística de selección y reemplazo para la migración ya comentada, que provee un aumento en la presión de selección [4]. El resto de los valores fueron analizados en las pruebas de calibración, los valores candidatos se ofrecen en la tabla 3.

Frecuencia	1,2,4,6,8,10
Tasa migración	0.01,0.05,0.1,0.2,0.3,0.4,0.5
Topología migración	grafo completo, anillo, anillo+1

Tabla 3 : valores candidatos para configurar el AGP

Los valores candidatos intentaron abarcar una gama representativa de configuraciones del AGP. De todos modos, el estudio se realizó sobre la base del conocimiento de que la escasa disponibilidad de recursos impidió un análisis más exhaustivo de las topologías de migración.

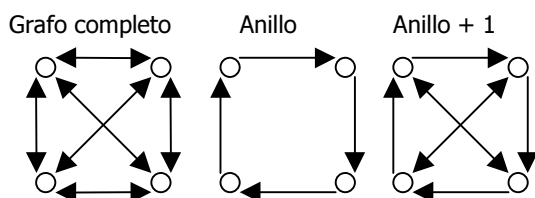


Figura 2 : Topologías de migración consideradas

El estudio de los parámetros de configuración involucró experimentos sobre 5 grafos considerados representativos de los problemas a resolver (100 nodos, 500 aristas, 20 terminales y entre 0 y 4

restricciones por par de nodos), generados aleatoriamente utilizando una variante del AG.

Ciertos parámetros que no caracterizan al modelo de migración fueron asumidos constantes. Estos parámetros se presentan en la tabla 4.

Número de demes	4
Condición de parada	300 generaciones
Tamaño de la población	120 individuos.
Probabilidad de mutación	0.00333
Probabilidad de cruzamiento	0.95
Factor de escalado de fitness	2

Tabla 4 : constantes en la configuración del AGP

En las pruebas de calibración se utilizó un criterio de parada por generaciones, con el objetivo de concentrarse exclusivamente en la influencia de los parámetros de migración en la calidad de los resultados obtenidos. La eficiencia de los modelos se evaluaron secundariamente, solamente para decidir entre configuraciones que ofrezcan similar calidad en los resultados obtenidos.

4.3 – Topologías

Todos los experimentos se realizaron tomando en cuenta las tres topologías de migración propuestas. Los resultados obtenidos no permiten apreciar diferencias significativas en la calidad de los resultados finales al utilizar diferentes topologías. El resultado es coherente con las especulaciones a priori, formuladas sobre la base de que el escaso número de máquinas hace que las topologías sean muy similares entre sí, sin permitir a una destacarse sobre las restantes.

4.4 – Frecuencia

El operador de migración introduce complejidad adicional en el funcionamiento del AGP, que puede afectar su eficiencia. Por este motivo es pertinente considerar los efectos de retrasar la migración, aumentando su frecuencia. El gráfico 1 resume el análisis de los resultados al utilizar diferentes frecuencias sobre la calidad del resultado obtenido.

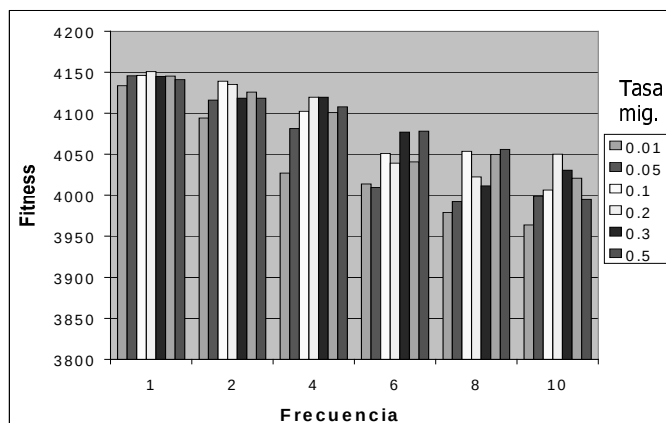


Gráfico 1 : estudio de la frecuencia de migración

En las tres topologías estudiadas los mejores resultados ocurren con frecuencia de migración igual a uno, correspondiendo al caso de mayor cantidad de comunicaciones. Al no detectarse mejoras significativas de performance al retrasar las migraciones, se adoptó el criterio de utilizar la cota superior para la frecuencia de migración, conduciendo a un modelo cercano al modelo panmítico de población distribuida [2].

4.5 – Tasa de migración

La cantidad de individuos participantes en cada migración influye sobre la presión de selección y por ello sobre la calidad de los resultados.

El gráfico 2 ofrece los resultados obtenidos en función de la tasa de migración utilizada para las frecuencias y topologías consideradas.

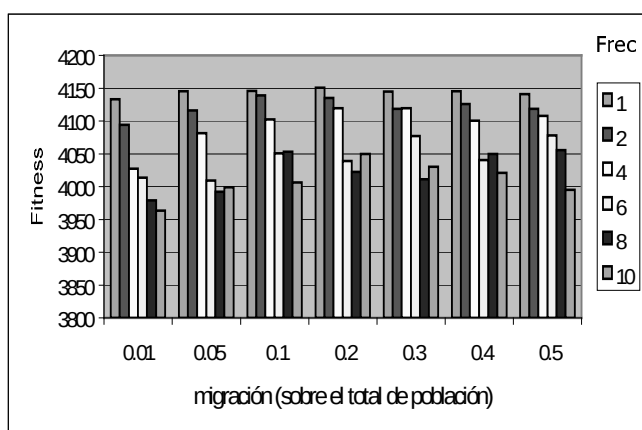


Gráfico 2 : estudio de la tasa de migración

El análisis no refleja diferencias significativas en la calidad de los resultados finales dependiendo de la tasa de migración utilizada. De acuerdo a Tanese [24], tasas “moderadas” de migración son necesarias para alcanzar una calidad de resultados que mejore la del AG serial. Tanese sugiere el valor 0.2 en su trabajo como aceptable.

4.6 – Conclusiones sobre la calibración

En función de cada parámetro considerado, es posible extraer algunas conclusiones respecto a los parámetros que configuran el AGP.

- Topología

La escasa disponibilidad de equipos no permite extraer resultados concluyentes sobre la topología de migración óptima. En los experimentos realizados no se aprecian diferencias significativas en la calidad de los resultados obtenidos. Siguiendo la idea de Cantú-Paz quien indica que “aunque la topología de conexión completa no integra eficientemente a varios demes, su configuración óptima es una buena elección para reducir el tiempo de ejecución” [4 pp. 87], se decidió utilizar

la topología completa como configuración del AGP, aún teniendo en cuenta que es poco escalable.

- Frecuencia

Los mejores resultados se obtuvieron consistentemente con frecuencia uno. La calidad de los resultados empeora al retrasar la migración.

- Tasa de migración

Tomando en cuenta que la cantidad de cromosomas migrados no afectó significativamente la calidad del resultado en los experimentos realizados, se optó por utilizar un valor intermedio de 0.3, considerado un valor de equilibrio entre un modelo fuertemente distribuido y un modelo de población panmítica [2].

4.7 – Comparación entre modelo paralelo y panmítico

La comparación entre el modelo paralelo y el modelo panmítico serial toma en cuenta la calidad del resultado, estudiando la evolución del fitness para cada generación y la eficiencia del modelo, cuantificando el tiempo de procesamiento.

Los parámetros utilizados en la comparación se indican en la tabla 5.

Modelo	migración	panmítico
Número de demes	4	1
Tamaño de la población	30	120
Tasa de migración	0.3	ignorada
Frecuencia de migración	1	ignorada
Topología	completa	ignorada
Parámetros con iguales valores en ambos modelos		
Grafos (vértices, aristas, terminales, restricciones)	cinco grafos (100,500,20,0..4)	
Condición de parada	300 generaciones	
Probabilidad de mutación	0.00333	
Probabilidad de cruzamiento	0.95	
Factor de escalado de fitness	2	

Tabla 5 : Configuración para la comparación entre modelos

Los resultados obtenidos se resumen en el gráfico 3 que muestra la evolución de la función de fitness para cada modelo, respecto al tiempo de ejecución.

En cuanto a la calidad de los resultados, no es posible apreciar una superioridad significativa del modelo paralelo sobre el serial. La función de fitness evoluciona parejamente a través de las generaciones en ambos modelos.

Respecto a la eficiencia en términos de tiempo, el modelo de subpoblaciones resultó del orden de 3.5 veces más rápido que el modelo panmítico, utilizando 4 procesadores para resolver el problema. El speedup casi lineal obtenido alienta a investigar la escalabilidad del modelo para la resolución de problemas más complejos.

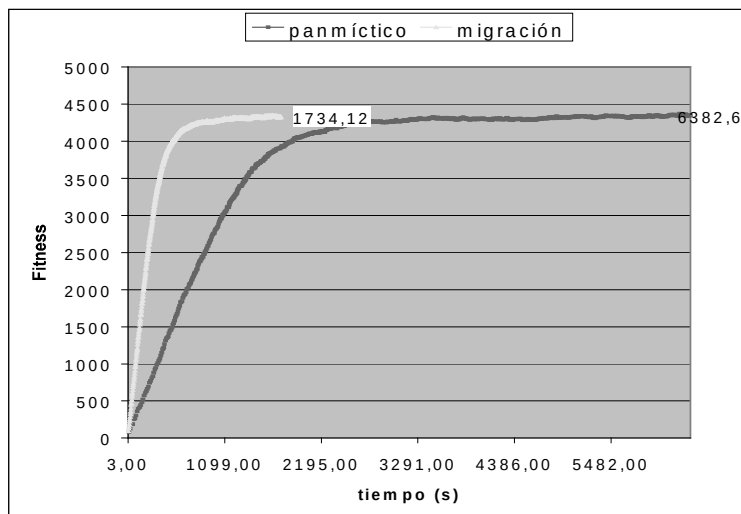


Gráfico 3: Comparación de eficiencia entre modelos

5 – Conclusiones e investigación futura

5.1 – Conclusiones

Los resultados obtenidos ofrecen perspectivas promisorias sobre la aplicabilidad de las técnicas de programación evolutiva para la resolución de problemas de optimización vinculados con el diseño de redes de comunicaciones confiables.

El enfoque evolutivo serial se mostró preciso para las pruebas de validación, obteniendo resultados de calidad superior a los obtenidos con otras heurísticas de optimización.

Los resultados del algoritmo genético paralelo demostraron que escogiendo una adecuada configuración de los parámetros de migración, es posible alcanzar soluciones de calidad comparable a las obtenidas con el algoritmo genético serial.

Respecto a la performance del modelo de poblaciones distribuidas, evaluada mediante el tiempo de ejecución, el análisis mostró que es posible obtener significativas mejoras de eficiencia respecto al modelo serial, utilizando un número reducido de poblaciones.

Estos resultados alientan a continuar la investigación en el área, estudiando tópicos de diseño no considerados en el estudio preliminar, ampliando el horizonte de aplicabilidad y analizando aquellos factores para los cuales no fue posible extraer afirmaciones concluyentes.

5.2 – Investigación actual y futura

Las líneas de investigación actual y futura se orientan hacia el estudio del funcionamiento del AGP, tratando de deducir las condiciones óptimas de configuración y ejecución para la resolución del problema de Steiner generalizado.

La escasa disponibilidad de recursos computacionales limitó los estudios iniciales de configuración y eficiencia del modelo. Actualmente se analizan alternativas para aumentar el número de equipos del cluster, y en tal sentido las investigaciones se desarrollarán en torno a los aspectos cuya influencia no pudo ser aclarada.

Los puntos a abordar en el futuro comprenden:

- Investigar la escalabilidad del modelo distribuido al aumentar los recursos computacionales.

Los experimentos primarios se realizaron con un número fijo de demes de población baja, para utilizar balanceadamente la capacidad de cómputo. Al disponer de mayor poder de cálculo será posible estudiar las mejoras de performance al agregar nuevas subpoblaciones, orientando la investigación a verificar la aplicabilidad de los estudios de Cantú-Paz [4] sobre cantidad óptima de demes y tamaño de las subpoblaciones.

- Estudio de las topologías de migración, orientado a determinar una topología de migración óptima.

Al incrementarse la cantidad de poblaciones, el overhead de comunicación de las topologías fuertemente conectadas puede afectar la eficiencia del modelo. La topología de migración óptima deberá garantizar un equilibrio que no disminuya excesivamente la presión de selección atribuida a la migración, y que no genere comunicaciones excesivas que afecten la performance del modelo.

- Investigar otras representaciones para codificar las soluciones al problema.

La codificación binaria utilizada es sencilla, pero requiere de un proceso de corrección luego del cruzamiento para trabajar con soluciones factibles. Una opción es utilizar representaciones complejas que garanticen la factibilidad de las soluciones. En este sentido se está trabajando en un AG que utiliza una representación basada en lista de caminos, de modo de comparar su calidad y eficiencia con el modelo con representación binaria. Trabajar con soluciones no factibles constituye otra alternativa, penalizando la no factibilidad en el valor de la función de fitness de modo de conservar características de soluciones no factibles pero próximas a los óptimos, en un intento por mejorar la calidad y eficiencia del AG.

- Estudio de modelos alternativos para AGP de población distribuida.

Ciertos avances se han realizado sobre un modelo de difusión, donde la migración explícita se reemplaza por un mecanismo de distribución de material genético a través de individuos compartidos entre los demes [4]. El modelo se encuentra en proceso de evaluación, y una vez estudiados sus parámetros de configuración se contrastará con el modelo de migración diseñado.

- Modelos de población dinámica

Al potenciar el cluster se incorporarán equipos de características heterogéneas, y por ello se ha trabajado en un modelo de poblaciones dinámicas [21], en las cuales el tamaño de los demes se adapta a la potencia computacional del equipo donde ejecuta. Esta solución permite mantener la homogeneidad de utilización de los recursos computacionales de modo de no producirse desfases evolutivos entre las poblaciones.

6 – Referencias bibliográficas

- [1] A. Baratz. "The complexity of the network flow problem" MIT Lab. for Comp. Sci. 1980. En línea en la URL <http://citeseer.nj.nec.com/524085.html>, consultado julio 2002.
- [2] E. Cantú-Paz, D. Goldberg "Parallel genetic algorithms with distributed panmictic populations" IlliGAL Technical Report 99006, 1999.
- [3] E. Cantú-Paz, "Topologies, migration rates, and multi-population parallel genetic algorithms", Proceedings of the GECCO 99, pp. 91-98, 1999.
- [4] E. Cantú-Paz, Efficient and Accurate Parallel Genetic Algorithms, Kluwer (2001).
- [5] H. Esbensen, "Computing Near-Optimal Solutions to the Steiner Problem in a Graph Using a Genetic Algorithm", Networks 26, pp. 173-185 1995
- [6] H. Esbensen, P. Mazumder "A Genetic Algorithm for the Steiner Problem in a Graph" EDAC-ETC-EUROASIC, pp. 402-406, 1994
- [7] L. Ford, D. Fulkerson. Flows in Networks. Princeton University Press, Princeton, 1962
- [8] D. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, 1989.
- [9] P. Galiasso, R. Wainwright, "A hybrid genetic algorithm for the point to multipoint routing problem with single split paths", SAC 2001, pp. 327-332.
- [10] W. Gropp, E. Lusk, A. Skjellum. Using MPI, Portable Parallel Programming with the Message Passing Interface, The MIT Press, 1999.
- [11] J. Hesser, R. Männer, O. Stucky, "On Steiner Trees and Genetic Algorithms", Lecture Notes in Computer Science 565, pp. 509-525, 1991.
- [12] R. Hwang, W. Do, S. Yang, "Multicast Routing Based on Genetic Algorithms", Journal of Information Science and Engineering, Vol. 16, 6, pp. 885-901, 2000.
- [13] B. Julstrom, "A genetic algorithm for the rectilinear Steiner problem", Proc. of the Fifth Int. Conf. on Genetic Algorithms, pp. 474-480, 1993.
- [14] B. Julstrom, "Seeding the population: improved performance in a genetic algorithm for the rectilinear Steiner problem", ACM Symposium on Applied Computing (SAC), pp. 222-226, 1994.
- [15] B. Julstrom, "Encoding rectilinear Steiner trees as lists of edges", ACM Symposium on Applied Computing (SAC), pp. 356-360, 2001.
- [16] A. Kapsalis, V. Rayward-Smith, G. Smith, "Solving the graphical Steiner tree problem using genetic algorithms" Journal of the Operational Research Society 44, pp. 397-406, 1993.
- [17] R. Karp, "Reducibility among combinatorial problems", Complexity of Computer Communications, 85-103, Plenum Press, 1972.
- [18] J. Krarup, "Generalized Steiner problem", Nota no publicada, 1978. Citado en [23].
- [19] S. Nesmachnow, "Aplicación de las técnicas de procesamiento de alta performance a los algoritmos genéticos", Reporte Técnico INCO 06-02, Universidad de la República, Uruguay, 2002.
- [20] S. Nesmachnow, "Evolución en el diseño y clasificación de algoritmos genéticos paralelos", Conferencia Latinoamericana de Informática, CLEI 2002.
- [21] M. Nowostawski, R. Poli. "Dynamic Demes Parallel Genetic Algorithm" Proc. of 3rd IEEE KES'99, pp. 93-98, 1999.
- [22] M. Nowostawski, R. Poli. "Parallel Genetic Algorithms Taxonomy", Proc. of 3rd IEEE KES'99, pp. 88-92, 1999.
- [23] F. Robledo, "Diseño Topológico de Redes, Casos de Estudio: The Generalized Steiner Problem y The Steiner 2-Edge-Connected Subgraph Problem", Tesis PEDECIBA Informática, 2001.
- [24] R. Tanese. "Distributed Genetic Algorithms", Proceedings of the Third International Conference on Genetic Algorithms, pp. 434-439, 1989.
- [25] S. Voß, K. Gutenschwager, "A chunking based genetic algorithm for the Steiner tree problem in graphs", DIMACS Series in Disc. Math. and Theor. Comp. Science 40, pp. 335-355, 1998
- [26] S. Wakabayashi, "A Genetic Algorithm for the Rectilinear Steiner Tree Problem in VLSI Interconnect Layout", IPSJ Vol 43, 05, 019, 2002.
- [27] X. Zhou, C. Chen, G. Zhu, "A Genetic Algorithm for Multicasting Routing Problem" Proceedings of the ICCT2000, S07.3, 2000
- [28] L. Zhu, D. Schoenefeld, R. Wainwright, "A Genetic Algorithm for the Point to Multipoint Routing Problem with Varying Number of Requests" Proceedings of the ICEC'98, pp. 171-176, 1998.