

# Diseño Económico de Redes confiables empleando A-Teams

Ing. Fabián Laufer  
*flaufer@cnc.una.py*

D.Sc. Benjamín Barán  
*bbaran@cnc.una.py*

Centro Nacional de Computación  
Universidad Nacional de Asunción  
Casilla de Correos 1439  
Campus Universitario de San Lorenzo  
Paraguay

## Resumen

El presente trabajo propone una combinación de algoritmos asíncronos, o *A-Team*, basado en la combinación de algoritmos genéticos con operadores de búsqueda local, para el diseño económico de redes confiables de computadoras. Resultados experimentales de la propuesta demuestran las ventajas obtenidas al paralelizar el proceso de diseño utilizando como plataforma una red de computadoras personales que cooperan en un contexto asíncrono, logrando con esto resolver problemas de dimensiones considerables o con menos restricciones de diseño. El trabajo presenta resultados experimentales que avalan las ventajas de la propuesta frente a otras publicaciones anteriores.

**Palabras claves:** *Sistemas distribuidos y paralelismo, Inteligencia artificial, Optimización combinatoria, Redes de computadoras, Informática en ingeniería.*

## 1. Introducción

La optimización topológica de redes es un problema clásico de investigación de operaciones con aplicabilidad práctica en diversos campos como las telecomunicaciones [1,2,3], planificación de redes de computadoras [4,5,6], desagües cloacales, gasoductos y oleoductos [7] entre otros. En especial, el presente trabajo enfoca el diseño económico de redes de telecomunicaciones de alta confiabilidad. El problema de diseño se plantea como la interconexión de un conjunto de nodos, geográficamente distribuidos, mediante una topología de red de bajo costo con capacidad de proveer una confiabilidad superior a la mínima especificada en los parámetros de diseño.

Como medida de confiabilidad, se considerará una métrica de conectividad conocida como confiabilidad uniforme, definida como la probabilidad de que exista comunicación entre cada par de nodos durante un periodo determinado de tiempo, esto es, que la topología sea al menos un *spanning tree*. Además, al igual que en [8], se exigirá conectividad dos, esto es, cada nodo tiene cuando menos dos arcos incidentes al mismo, evitando que la caída de un único enlace deje fuera de funcionamiento a un nodo. Siguiendo los delineamientos de problemas similares presentados en otras publicaciones anteriores [1 - 10], se asume que la ubicación de los nodos, los costos, y las confiabilidades de los enlaces disponibles son conocidos. Los nodos son perfectamente confiables, los enlaces son bidireccionales y pueden encontrarse en uno de dos estados: operacionales o en falla. Además, se considera que estos presentan mal funcionamiento en forma independiente unos de otros y no se analizan sus reparaciones.

Al igual que otros problemas de análisis combinatorio, la complejidad de la tarea de diseño esta directamente relacionada con el tamaño del espacio de búsqueda, el cual crece en forma exponencial con el número de nodos a interconectar. De hecho, este es un problema NP-Hard [11] que a su vez tiene la complejidad adicional de evaluar para cada solución candidata la confiabilidad cuyo cálculo es a su vez un problema NP-Hard [8]. Por esta razón, la estimación de confiabilidad es realizada utilizando una simulación basada en el método Monte Carlo, con considerable costo computacional.

Problemas similares han sido largamente estudiados utilizando tanto métodos enumerativos [2,6,12], como heurísticos. Entre estos se pueden citar: *Tabu Search* [13,14], *Simulated Annealing* [1,3,5] y *Genetic Algorithms* [8,9,15], siendo estos últimos los que han demostrado los mejores desempeños. En [9] se presenta un algoritmo genético cuya función objetivo se basa en una minimización de costo bajo la restricción de confiabilidad uniforme mínima. Dicho trabajo no considera redundancia y se asume que todos los enlaces son de igual confiabilidad. El algoritmo genético propuesto emplea operadores de cruzamiento *single point* y mutación *flip bit* [9]. Para la estimación de confiabilidad, se utilizan dos métodos distintos, una cota superior de confiabilidad y un método especializado de Monte Carlo. Más tarde, los mismos autores presentan en [8] una versión mejorada, introduciendo operadores genéticos especializados con búsqueda local y reparación. Posteriormente, en Deeter et al. [15] se extiende el problema paradigma, permitiendo el uso de distintos tipos de enlaces, además de demostrar la versatilidad de los AGs para tratar problemas relacionados, como la maximización de confiabilidad bajo restricción de coste máximo, así como el uso de distintas métricas de confiabilidad.

Si bien los trabajos arriba citados han conseguido importantes avances en problemas reales de diseño, las mismas tropiezan con las problemáticas típicas de los algoritmos genéticos, como ser: el elevado costo computacional y la falta de un método de parada universal. Esto se ve agravado por una función objetivo extremadamente pesada, hecho que limita el uso práctico de los mismos. Ante esta dificultad y la existencia en la literatura de diversos métodos ya publicados, surge la posibilidad de combinar distintos métodos con distintas características para la resolución de problemas tan complejos. Esta combinación de algoritmos diferentes interactuando en la solución de un mismo problema global se conoce como *Team Algorithm* [16]. La solución de problemas utilizando *Team Algorithm* puede ser implementada en paralelo, asignando diversos procesos a los procesadores disponibles de un sistema distribuido asíncrono, como una red de computadoras, en cuyo caso se lo conoce como *A-Teams* (*Asynchronous Team*) [17]. En efecto, un trabajo anterior realizado por Talukdar et al. [17] presenta un *A-Team* combinando algoritmos genéticos con reconocidos métodos numéricos, como el Método de Newton, aplicados a la resolución de sistemas algebraicos de ecuaciones no lineales. A la fecha, se puede observar un creciente interés en la utilización de *A-Teams* para diversas aplicaciones prácticas y en especial, de ingeniería [16,17].

El presente trabajo propone un *A-Team* basado en la combinación de algoritmos genéticos con operadores de búsqueda local como método eficiente de diseño para el tratamiento de problemas de dimensiones reales. A su vez, se extiende el paradigma planteado en los trabajos arriba citados de manera a considerar distintos tipos de enlaces así como redundancia en los enlaces. Con la idea de simplificar la lectura del presente trabajo, se presenta a continuación un resumen de la notación utilizada.

## Notación

- $G(N, L)$  Grafo probabilístico con un conjunto  $N$  de nodos y un conjunto  $L$  de enlaces;
- $K$  Conjunto de tecnologías de enlaces disponibles;
- $L = \{l_{ijk}\}$  Conjunto de enlaces, representando  $l_{ijk}$  un enlace del tipo  $k$  que une los nodos  $i$ - $j$ ;
- $c(l_{ijk})$  Costo del enlace  $l_{ijk}$ ;
- $p(l_{ijk})$  Confiabilidad del enlace  $l_{ijk}$ ;
- $x_{ijk}$  Variable binaria de decisión que indica la existencia (1) o no (0) de un enlace  $l_{ijk}$ ;
- $X = \{x_{ijk}\}$  Topología de red candidata;
- $R(X)$  Confiabilidad uniforme de una red  $X$ ;
- $R_0$  Requerimiento de confiabilidad mínima;
- $Z(x)$  Función objetivo;
- $c_{\max}$  El máximo valor de  $c(l_{ijk})$ ;
- $\delta$  Variable binaria igual a 0 si  $R_0 \leq R(X)$ ; o a 1 si  $R(x) \leq R_0$ .

## 2. Formulación matemática del problema

Una red de comunicaciones puede ser modelada mediante un grafo probabilístico  $G = \{N, L\}$ , donde  $N$  representa un conjunto de nodos y  $L$  un conjunto de enlaces. Entre cada par de nodos  $i, j \in N$  pueden existir distintos tipos de conexiones  $l_{ijk} \in L$ , donde  $k \in K$  representa la tecnología utilizada. Cada enlace  $l_{ijk}$  tiene un costo  $c(l_{ijk})$  y una confiabilidad  $p(l_{ijk})$  que pueden diferir de un enlace a otro. El problema de optimización consiste básicamente en la minimización de la expresión:

$$\sum_{i=1}^{|N|-1} \sum_{j=i+1}^{|N|} \sum_{k=1}^K c(l_{ijk}) x_{ijk} \quad (1)$$

Sujeto a  $R(X) \geq R_0$

donde  $x_{ijk} \in \{0, 1\}$  es una variable de decisión,  $R(X)$  la confiabilidad uniforme de la topología candidata  $X$  y  $R_0$  es la confiabilidad mínima requerida para el diseño. Se asume además que:

- los nodos son bidireccionales, esto es  $l_{ijk} = l_{jik}$ ;
- los nodos son perfectamente confiables;
- los enlaces fallan en forma independiente unos de otros;
- no se consideran reparaciones; y
- no se considera la replicación de un mismo enlace; esto es, entre cada par de nodos  $i$ - $j$  a lo más existe un enlace del tipo  $k$ . Se adopta esta estrategia debido a que normalmente, cuando más de un enlace de un mismo tipo une dos puntos, los mismos se extienden juntos compartiendo una misma infraestructura. Por ejemplo, el costo de tender una fibra óptica a través de un ducto puede ser de 10\$/m, mientras que pasar dos fibras costaría menos de 14\$/m, ya que las mismas compartirían el costo del ducto. En todo caso, se puede representar a cada número de replicaciones de enlaces como un tipo de enlaces distinto ( $k=1$ , una fibra;  $k=2$ , cinco fibras; etc.).

### 3. Descripción del A-Team propuesto

El *A-Team* propuesto se compone básicamente de un coordinador y varios procesos esclavos. La misión del coordinador es distribuir la carga computacional entre los procesadores disponibles en la red, iniciar los procesos esclavos, llevar las estadísticas globales y terminar la corrida de acuerdo al criterio de parada descrito más adelante, conforme se muestra a continuación en el pseudocódigo del proceso Master.

Distribuir el número total de individuos entre los procesadores Iniciar un proceso esclavo en cada procesador DO WHILE (Queden procesos sin homogeneización) Recolectar datos sobre las estadísticas END DO Parar todos los procesos esclavos
--

**Pseudocódigo 1:** Proceso Coordinador.

Por su parte, los procesos esclavos son los que realizan el trabajo efectivo de optimización, conforme se muestra en el Pseudocódigo 2. Los mismos actúan básicamente como un algoritmo genético paralelo, pero con la peculiaridad de combinar las optimizaciones propias de los AGs con búsquedas locales inmersas dentro de los mismos operadores genéticos de cruzamiento y mutación. Los resultados parciales obtenidos son intercambiados en forma asíncrona entre las diversas poblaciones residentes en distintos procesadores. Como en cualquier algoritmo genético paralelo (AGP), la política de migración juega un rol preponderante en la eficiencia y eficacia del mismo. Para explicar la política aquí empleada se presenta el siguiente análisis intuitivo.

El tiempo de corrida del algoritmo genético secuencial puede ser expresado esquemáticamente como:

$$T_s = g * n * (t_c + t_s + t_m + t_{CR}) \quad (2)$$

donde:

- $T_s$  ... tiempo de proceso del algoritmo secuencial;
- $g$  ..... número de generaciones;
- $n$  ..... tamaño de la población;
- $t_c$  .... tiempo de cálculo de la función objetivo (cálculo de costo + cálculo de confiabilidad);
- $t_s$  .... tiempo del operador de selección;
- $t_m$  .... tiempo del operador de mutación;
- $t_{CR}$  ... tiempo del operador de cruzamiento.

El algoritmo propuesto distribuye a los individuos de la población original en  $p$  procesadores distintos proporcionalmente a la capacidad computacional relativa de cada procesador de forma tal que:

$$\sum_{i=1}^p n_i'' = n \quad (3)$$

donde  $n_i''$  representa la población local del procesador  $p_i$ .

De esta forma, en cada procesador  $p_i$  y a cada generación, se reemplaza la población anterior por una nueva población con  $n_i''$  nuevos individuos y estos son enviados a todos los procesadores restantes. Así, al momento de la selección, cada procesador tiene una población de  $n_i'$  individuos constituida por la población local más los individuos recién llegados. En consecuencia, el tiempo total de proceso de una población local en igual número de generaciones  $g$  esta dado por:

$$T_p = g * (n_i' * t_s + n_i'' * (t_c + t_{CR} + t_M)) \quad (4)$$

Como el tiempo por generación es relativamente similar en cada población local,  $n_i'$  es aproximadamente igual a  $n$ , por lo que cuantitativamente la diversidad genética al momento de la selección es semejante a la versión secuencial. Dado que  $t_c \gg t_s + t_M + t_{CR}$ , las ecuaciones (2) y (4) se pueden denotar aproximadamente como:

$$T_s = g * n * t_c \quad (5)$$

$$T_p = g * n_i'' * t_c \quad (6)$$

De esta forma la aceleración esperada, si  $n_i'' = n/p$  en todos los procesadores, sería aproximadamente de

$$\frac{T_s}{T_p} \cong \frac{g * n * t_c}{g * n_i'' * t_c} = \frac{n}{n_i''} = \frac{n}{n/p} = p \quad (7)$$

Esto es, el tiempo de computo disminuye en forma aproximadamente lineal con el número de procesadores, por lo que es de esperar una buena escalabilidad del problema, inclusive en entornos asíncronos (como las redes de computadoras), dado que los procesadores pueden continuar su computación independientemente de la llegada o no de la información procedente de otros procesadores. Con esto, se espera aprovechar la capacidad de cómputo ociosa, generalmente disponible, sin que exista una penalidad importante debida al asincronismo propio de las redes de computadoras de uso compartido con otras aplicaciones.

```

Generar población Inicial
DO WHILE (el proceso Master no ordene parar)
    Importar Soluciones enviadas por otros procesos
    Aplicar, selección sobre la población local más la importada
    Generar la nueva población local (Mutación y Cruzamiento)
    Informar al Master sobre el índice de homogeneización
    Enviar la población generada a los demás procesos
END DO
    
```

Pseudocódigo 2: Proceso esclavo.

## 4. Algoritmo Genético Paralelo

El algoritmo propuesto se basa en el AG secuencial publicado en [8], el cual se describe a continuación.

### 4.1 Función Objetivo

La función objetivo es la suma total del costo de todos los enlaces más una función de penalización cuadrática para aquellas redes que no satisfagan el requerimiento de confiabilidad mínima. El objetivo de la función de penalidad es el de permitir la existencia de redes candidatas económicas, aunque no factibles, dentro de la población. Esto es importante debido a que se pueden llegar a buenas soluciones a partir del cruzamiento de redes factibles con redes económicas no factibles. Además, el cruzamiento de dos soluciones factibles no asegura que los descendientes también lo sean, especialmente en problemas con restricciones muy duras. La penalidad se basa en el grado de incumplimiento de la restricción de confiabilidad  $(R(X)-R_0)$ , con una penalización cuadrática (no lineal). La función objetivo está dada entonces por:

$$\sum_{i=1}^{|N|-1} \sum_{j=i+1}^{|N|} \sum_{k=1}^K c(l_{ijk})x_{ijk} + \delta(c_{\max}) * (R(X)-R_0)^2 \quad (8)$$

donde:

$$\delta \quad \begin{array}{l} 0, \text{ si } R_0 \leq R(X); \\ 1, \text{ si } R(x) \leq R_0 \end{array}$$

$c_{\max}$  costo del enlace más costoso.

### 4.2 Cálculo de confiabilidad

Para el cálculo de  $R(X)$ , Dengiz et al. [8] propone el uso de dos métodos diferentes: Primero una cota superior de confiabilidad Jan [10] es aplicada a la población completa para asignar rápidamente la confiabilidad máxima que puede alcanzar cada red. Luego es aplicado el método Monte Carlo especializado de Yeh [18] solo sobre aquellas redes que puedan alcanzar la confiabilidad requerida. Sin embargo, la cota superior de Jan limita la generalidad del problema, ya que la misma requiere que todos los enlaces sean de igual confiabilidad. Por otro lado el método de Yeh [18], si bien tiene una convergencia matemática más acelerada, tiene también un costo computacional en general mayor que el Monte Carlo tradicional. Por lo tanto, el presente trabajo sigue el mismo esquema arriba presentado en lo referente a la utilización de 2 niveles de estimadores de confiabilidad, pero utilizando el método Monte Carlo tradicional con distintos números de iteraciones. Con esto, el método implementado permite el estudio de redes con enlaces de características diferenciadas.

### 4.3 Codificación

Las redes son representadas por medio de una cadena continua de bits, donde cada bit representa el estado de un enlace  $l_{ijk}$ . El orden de los enlaces dentro de la cadena es decreciente respecto al costo, de forma a optimizar la eficiencia de los operadores genéticos especializados.

## 4.4 Operadores Especiliados de Cruzamiento y Mutación

### **Cruzamiento**

El operador de cruzamiento tiene la forma de un cruzamiento uniforme con reparación para asegurar que cada hijo sea al menos un *spanning tree* con conectividad dos. Para su implementación, se utilizan operadores lógicos (AND, OR) a nivel de bits en la ejecución de las operaciones de conjunto.

```
Seleccionar dos redes candidatas llamadas T1 y T2.
Determinar los links comunes =  $T1 \cap T2$ .
Asignar los enlaces comunes a los hijos T1' y T2'.  $T1' = T2' = T1 \cap T2$ .
Asignar los enlaces restantes de manera que  $t1 = T1 - (T1 \cap T2)$ ;  $t2 = T2 - (T1 \cap T2)$ .
IF (T1 no es un spanning trees) THEN
    Agregar los enlaces de t1 ordenados por costo a T1' hasta que T1' sea un spanning
IF (T2 no es un spanning trees) THEN
    Agregar los enlaces de t2 ordenados por costo a T2' hasta que T2' sea un spanning
Determinar que enlaces de  $T1 \cup T2$  no existen en T1' y T2': CT1 y CT2 correspondientemente.
Asignar  $T1' = T1 \cup CT2$ ;  $T2' = T2 \cup CT1$ .
```

Pseudocódigo 3: Operador especializado de cruzamiento.

### **Mutación**

La mutación toma la forma de una búsqueda local avariciosa aleatoria. El operador es aplicado en forma distinta, de acuerdo al grado de los nodos de la red, donde el grado de un nodo se define como el número de enlaces incidentes al mismo. Esto se hace para asegurar la conectividad en por lo menos grado 2, para cada nodo.

```
Determinar el grado de los nodos  $grad(j)$  de la red para  $j=1,2,\dots,|N|$ .
DO CASE
    CASE  $grad(j)=2$  para todo j
        Seleccionar aleatoriamente un enlace no presente en la red y agregarlo
    CASE  $grad(j)>2$  para todo j
        DO WHILE(queden enlaces sin revisar y no se haya removido ningún enlace)
            Ordenar los enlaces en orden decreciente de costo
            Eliminar el enlace de costo máximo de la red
            IF(la red no tiene conectividad dos) THEN
                Cancelar la eliminación del enlace
    OTHERWISE .
        Generar un variable aleatoria u entre 0 y 1
        IF( $u < r_d$ ) THEN /* $r_d$  es el coeficiente de eliminación/
            Tratar como si  $grad(j)>2$ 
        ELSE
            Tratar como si  $grad(j)=2$ 
```

Pseudocódigo 3: Operador especializado de mutación.

#### 4.5 Criterio de parada

Tradicionalmente, los AGs utilizan la varianza de la función objetivo de una población con respecto a la media para medir el grado de convergencia de la misma. Sin embargo, para el caso del diseño de redes, este criterio de parada no es aplicable debido a que numerosas redes de topologías totalmente diferentes pueden tener *fitness* muy similares. Atendiendo a lo expuesto, el presente trabajo propone el uso de un índice de homogeneización basado en la información genética de la población.

La definición del índice de homogeneización se presenta a continuación mediante un ejemplo. Supóngase que se cuenta con una población de tres individuos, cada uno de ellos representados genéticamente por una cadena de 6 bits, donde cada bit representa un enlace  $l_{ijk}$  (ver Tabla 1).

Individuos	Enlaces					
	1	2	3	4	5	6
Red 1	1	0	1	1	0	1
Red 2	0	0	1	0	0	1
Red 3	0	0	1	1	1	1
<b>AND</b>	0	0	1	0	0	1
<b>OR</b>	1	0	1	1	1	1

Tabla 1: Ejemplo para el cálculo del índice de homogeneización.

Al aplicar el operador lógico AND entre toda la población, se obtienen los enlaces que se encuentran presentes en todos los individuos de la población. Siguiendo el ejemplo anterior, se encuentra que los enlaces 3 y 6 son los únicos enlaces comunes entre los tres individuos. Por otro lado, al aplicar el operador lógico OR, se obtiene el número total de enlaces presentes dentro de la población. De esta manera, se puede observar que en la población anterior se encuentran presentes 5 de los 6 enlaces posibles. Dividiendo el número de enlaces comunes en la población por el número total de enlaces existentes en la misma, se obtiene como resultado un índice buscado, que indica cuan similares son los individuos dentro de la población. Para el caso de la población ejemplo, el índice indica una homogeneización del 40%.

Al aumentar el número de enlaces comunes y mantenerse el número total de enlaces en la población relativamente constante, el índice crece reflejando la homogeneización de la misma. Debe notarse que el valor máximo (100%) equivale a que todos los individuos son idénticos. De esta forma, el criterio de parada consiste en observar el comportamiento del índice y una vez que este haya alcanzado un umbral de parada dado, es signo que la población no contiene suficiente diversidad genética para seguir justificando una optimización sistemática. En consecuencia, a partir de ese punto el algoritmo tendría un comportamiento similar a una búsqueda aleatoria y no justificaría continuar la corrida.

Como el *A-Team* propuesto trabaja con numerosos AGs paralelos y existe un intercambio de individuos que podría heterogeneizar una población ya homogeneizada, se considera como criterio de parada al momento en el cuál todas las poblaciones participantes han superado el umbral de parada. La Figura 1 muestra el comportamiento del índice de homogeneización en una corrida típica. El cuadro superior representa la función objetivo de la mejor solución alcanzada con respecto al número de generaciones mientras que la gráfica inferior describe el comportamiento del índice de homogeneización.



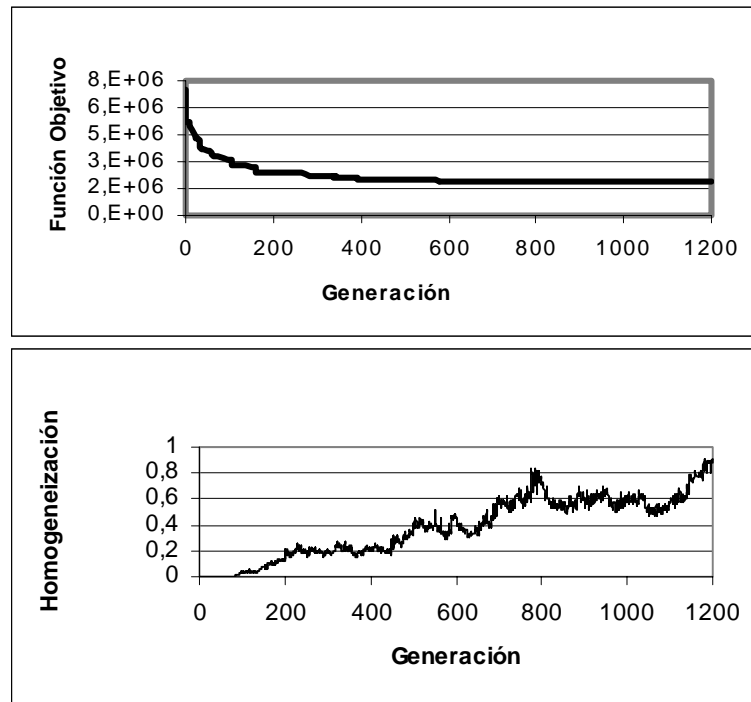


Figura 1:Comportamiento del índice de homogeneización en una corrida típica.

## 5. Resultados Experimentales

### 5.1 Ambiente computacional

El sistema distribuido utilizado en la implementación consiste en una red de seis computadoras personales con procesadores AMD K6-2 de 350 MHz y 128 MB de memoria RAM. Estas computadoras están conectadas en topología estrella mediante un *switch* Ethernet IBM 524, formando una red de área local (LAN). Como plataforma de *software* se utilizó PVM (*Parallel Virtual Machine*) sobre el sistema operativo LINUX RED HAT 6.0. La implementación fue desarrollada utilizando el lenguaje de programación ANSI C, en combinación con las librerías proveídas por PVM.

### 5.2 Problema de Referencia

Como problema de referencia se adopta el ejemplo 3.5 publicado en [15], el cual es escogido por ser el de mayor dimensión encontrado durante el desarrollo de este trabajo. El mismo se basa en una versión simplificada de un problema real de diseño emprendido por el gobierno federal de Turquía, la red ULAK-NET. El referido proyecto pretende interconectar 19 centros académicos localizados en 9 ciudades distintas. Para fines de diseño, se consideran tres tipos de cables de fibra óptica de costos y confiabilidades distintas, (333 \$/Km, 96%), (433\$/Km,97.5%) y (583 \$/Km, 99%). El costo por unidad de cable incluye material, costos de empalmes, repetidores e instalación. Por último, el objetivo del diseño es una red de por lo menos 99% de confiabilidad uniforme. El planteamiento del problema en [15] tiene como restricción el no considerar redundancia. Sin embargo, en el presente trabajo esta restricción es relajada para permitir uno o más enlaces de distintas tecnologías de comunicación incidentes a un mismo par de nodos. De esta forma, el espacio de búsqueda crece de  $3.8 \cdot 10^{81}$  ( $K^{N(N-1)/2}$ ) a  $2.6 \cdot 10^{154}$  ( $2^{K \cdot N \cdot (N-1)/2}$ ).

### 5.3 Implementación del A-Team

Para establecer el rendimiento y la aceleración o *speedup* del *A-Team* (inherentemente paralelo), se compararán los resultados obtenidos utilizando distintos números de procesadores, frente a los

conseguidos con una implementación secuencial idéntica al algoritmo genético empleado en los procesos esclavos. Para ello se efectuaron diez corridas tanto con el AG secuencial como con el *A-Team* utilizando cuatro y seis procesadores. En todas las corridas se utilizaron los mismos coeficientes de probabilidad para los operadores genéticos:  $r_c=0.7$ ,  $r_m=0.3$ . Para que la comparación entre las corridas empleando distintos números de procesadores sea equitativa, se mantuvo constante el tamaño de la población total del sistema, de forma que en todas las corridas se dispongan inicialmente de la misma riqueza genética. En el gráfico 2 (correspondiente a los datos de la tabla 2), se presenta el tiempo promedio de las corridas empleando uno (secuencial) cuatro y seis procesadores, utilizando el método de parada propuesto. En la misma se puede apreciar que el tiempo de procesamiento disminuye en forma casi lineal con el número de PCs empleadas.

	<i>Secuencial</i>			<i>A-Team con 4 procesadores</i>			<i>A-Team con 6 procesadores</i>		
	Costo	Conf.	Tiempo[h]	Costo	Conf.	Tiempo[h]	Costo	Conf.	Tiempo[h]
<b>Prom</b>	2.129.712	0.992	6.58	1.970.685	0.992	2.38	2.046.347	0.992	1.58
<b>Desv.</b>	163.816	0.001	2.04	55.8440	0.001	0.83	156.140	0.001	0.57

Tabla 2: Comparación entre corridas con distintos números de procesadores.

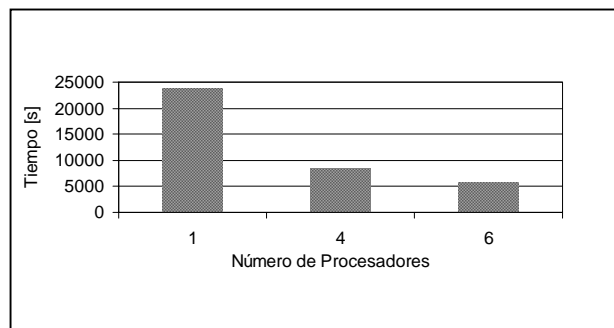


Gráfico 2: Tiempo promedio de Cálculo por número de procesadores (Parada por Homogeneización).

Al mismo tiempo, es importante notar que el costo promedio de las redes obtenidas utilizando el *A-Team* es ligeramente inferior al obtenido con la versión secuencial; en otras palabras, el *A-Team* mejora la calidad de la solución obtenida secuencialmente. Como los experimentos se realizaron con dos métodos distintos de parada: (i) homogeneización y (ii) generación máxima; los mismos son analizados por separado. Como se puede observar en la tabla 3, el rendimiento alcanzado respecto a la generación máxima refleja un buen desempeño de la implementación paralela. Debe también notarse que el *speedup* y rendimiento conseguido con el criterio de homogeneización es inferior al alcanzado considerando el otro método de parada. Este hecho se debe principalmente a que la homogeneización del *A-Team* al esperar que todas las poblaciones homogeneicen es más estricta y por lo tanto más lenta que la versión secuencial, lo que de alguna forma explica la mejor calidad de las soluciones obtenidas con un *A-Team*. En consecuencia, queda por estudiar el uso de criterios más relajados como por ejemplo, parar con la homogeneización de solo un porcentaje de las poblaciones, o si el promedio de homogeneización de las poblaciones alcanza cierto umbral.

<i>Nro. De Procesadores</i>	<i>Homogeneización</i>		<i>Gen. Máxima</i>	
	<i>Speedup</i>	<i>Rendimiento</i>	<i>Speedup</i>	<i>Rendimiento</i>
4	2,77	0,69	3,93	0,98
6	4,17	0,69	5,74	0,95

Tabla 3: Análisis de aceleración y rendimiento.

#### 5.4 Comparación con trabajos anteriores

Los resultados experimentales obtenidos en el presente trabajo son comparados con los publicados en el ejemplo 3.5 de Deeter et. al. [15] en la tabla 4. Como la mejor solución encontrada con el A-Team contiene redundancia, también se presenta una solución que cumple con todas las restricciones impuestas en [15].

	<b>Deeter et al. [15]</b>	<b>Sin Redundancia</b>	<b>Con Redundancia</b>
<b>Costo</b>	7.694.708	1.987.805	1.761.236
<b>Confiabilidad</b>	0.999	0.991	0.991
<b>Tiempo</b>	15 hs.	1.6 hs.	52 min.
<b>Plataforma</b>	Sun Enterprise II dual	6 PCs	6 PCs
<b>Costo aprox.</b>	25.000 US\$	5.500 \$	5.500 \$

Tabla 4: Comparación con los resultados obtenidos en [15].

Finalmente es interesante enmarcar el presente trabajo dentro de la secuencia de trabajos publicados en esta línea de investigación, para lo cual se presenta la Tabla 5.

<b>Año</b>	<b>Autor</b>	<b>Trabajo</b>	<b>Publicación</b>	<b>Espacio de búsqueda</b>	<b>Enlaces Distintos</b>	<b>Redundancia</b>	<b>Paralelismo</b>
1997	Dengiz et al.	Local Search Algorithm for Design of reliable Networks	IEEE Transactions on evolutionary computation	$10^{16}$	No	No	No
1998	Deeter et al.	Economic Design of reliable networks	IIE Transaction Special Issue on Economics of Reliability Engineering	$10^{81}$	Si	No	No
1999	Baran et al.	Topological Optimization of Reliable Networks using A-Teams	SCI/ISAS Proceedings Vol 5 Computer Science and Engineering	$10^{16}$	No	No	Si
2000	Laufer et. al.	Presente Trabajo	CLEI 2000	$10^{154}$	SI	Si	Si

Tabla 5: Cronología de trabajos en está línea de investigación.

## 6. Conclusiones

Los resultados experimentales demuestran que el *A-Team* propuesto es superior tanto en eficiencia como en eficacia frente a su correspondiente AG secuencial. Si bien el número de nodos y el número de tecnologías de comunicación planteados en el problema de referencia es básicamente el presentado en [15], al extender el paradigma y considerar redundancia, también se expande el espacio de búsqueda del orden de  $10^{81}$  a  $10^{154}$ . El *A-Team* presentado demostró su capacidad no solo de resolver el problema en este nuevo espacio de búsqueda extendido, sino que, además encontró mejores soluciones aun con el problema original. Finalmente, es importante señalar que el tiempo promedio de computación se redujo de unas 15 horas a menos de una hora utilizando 6 procesadores, lo que augura una buena escalabilidad con la consecuente posibilidad de tratar problemas de dimensiones y restricciones reales de diseño.

## Bibliografía

- [1] M. Atiqullah, S. Rao, "Reliability optimization of communication networks using simulated annealing", *Microelectronics and Reliability*, Vol. 33, No. 9, 1993.
- [2] R. Jan, F. Hwang, S. Chen, "Topological optimization of a communication network subject to a reliability constraint", *IEEE Transaction on Reliability*, Vol. 42, No. 1, 1993.
- [3] S. Pierre, M. Hyppolite, J. Bourjolly, O. Dioume, "Topological design of computer communication networks using simulated annealing", *Engineering Applications of Artificial Intelligence*, Vol 8, No 1, 1995.
- [4] K. Aggarwal, Y. Chopra, J. Bajwa, "Topological layout of links for optimizing the overall reliability in a computer communication system", *Microelectronics and Reliability*, Vol 22, No. 3, 1982.
- [5] P. Fetterolf, G. Anandalingman, "Optimal design of LAN-WAN internetworks: an approach using simulated annealing", *Annals of Operation Research*, Vol. 36, 1992.
- [6] R. Wilkov, "Design of computer networks based on a new reliability measure", *Symposium on Computer-Communications Networks and Teletraffic*, Polytechnic Institute of Brooklyn, 1972.
- [7] G. Walters, D. Smith, "Evolutionary design algorithm for optimal layout of tree networks". *Engineering Optimization*, Vol 24, 1995.
- [8] B. Dengiz, F. Altiparmak, A. Smith, "Local Search Genetic Algorithm for Optimal Design of Reliable Networks", *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 3, 1997.
- [9] B. Dengiz, F. Altiparmak, A. Smith, "Efficient optimization of all-terminal reliable networks using an evolutionary approach", *IEEE Transaction on Reliability*, Vol. 46, 1997.
- [10] R. Jan, "Design of reliable networks", *Computer and Operation Research*, Vol. 20, No.1, 1993.
- [11] M. Garey, D. Johnson, *Computers and Intractability: A guide to the Theory of NP-Completeness*, W.H. Freeman and Co., 1979.
- [12] K. Aggarwal, S. Rai, "Reliability evaluation in computer-communication networks", *IEEE Transactions on Reliability*, Vol R-30, No.1, 1981.
- [13] F. Glover, M. Lee, J. Ryan, "Least-cost network topology design for a new service: an application of a tabu search", *Annals of Operations Research*, Vol. 33, 1991.
- [14] S. Koh, C. Lee, "A tabu search for the survivable fiber optic communication network design" *Computers and Industrial Engineering*, Vol. 28, No. 4, 1995.
- [15] D. Deeter, A. Smith, "Economic Design of Reliable Networks", *IEEE Transactions, Special Issue on Economics of Reliability Engineering*, 1998.
- [16] B. Barán, B. Kaskurewics, "Parallel Asynchronous Team Algorithms: Convergence and Performance Analysis". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, No. 7, 1996.
- [17] P. Souza, S. Talukdar, "Genetics Algorithms in Asynchronous Teams", *ICGA-91*, 1991.
- [18] M. Yeh, J. Lin, W. Yeh, "New Monte Carlo method for estimating network reliability", *Proc. 16<sup>th</sup> Int. Conf. Computers & Industrial Engineering*, 1994.