

# Una Versión Paralela del Algoritmo Evolutivo para Optimización Multiobjetivo NSGA-II y su Aplicación al Diseño de Redes de Comunicaciones Confiables

Sergio Nesmachnow

sergion@fing.edu.uy

Centro de Cálculo, Instituto de Computación

Facultad de Ingeniería.

Universidad de la República, Uruguay.

**Resumen** — Este trabajo presenta una versión Paralela del Algoritmo Evolutivo para Optimización Multiobjetivo NSGA-II, original de Deb et al. (2000). Se introducen los detalles de diseño e implementación de una versión paralela basada en subpoblaciones semi-independientes y se analiza la calidad de resultados y la eficiencia computacional, comparando con los resultados y tiempos de ejecución de la versión secuencial del algoritmo NSGA-II sobre un conjunto de problemas de prueba estándar. Adicionalmente, se estudia la aplicación de la versión paralela propuesta a la resolución de un problema de diseño de redes de comunicaciones confiables.

**Palabras clave** — Algoritmos Evolutivos, Optimización Multiobjetivo, Paralelismo, NSGA-II.

## I. INTRODUCCIÓN

Los *Algoritmos Evolutivos* (EAs) se han popularizado como métodos robustos y efectivos para la resolución de problemas de optimización. Tradicionalmente, los problemas abordados consideraban la optimización de una única función objetivo, pero en la última década se han desarrollado una amplia gama de EAs para afrontar problemas con objetivos múltiples. Estos problemas cuentan con complejidades propias que los distinguen de los problemas monoobjetivo, y por ello los EAs para Optimización Multiobjetivo tienen características que los diferencian de los EAs tradicionales.

Las técnicas de procesamiento paralelo y distribuido se aplican a los modelos clásicos de EAs con el objetivo de obtener mejoras en la eficiencia computacional y perfeccionar la calidad del mecanismo evolutivo (Cantú-Paz, 2001). Desde la perspectiva de la eficiencia, paralelizar un EA permite afrontar la lentitud de convergencia para problemas cuya dimensión motiva el uso de poblaciones numerosas o múltiples evaluaciones de complejas funciones objetivo. Desde el punto de vista algorítmico, los modelos paralelo-distribuidos de algoritmos evolutivos pueden explotar el paralelismo intrínseco del mecanismo evolutivo trabajando simultáneamente sobre varias poblaciones semi independientes para resolver un problema.

En el área de la optimización multiobjetivo, pocos trabajos han abordado el estudio del paralelismo aplicado y sus influencias en la eficiencia computacional y calidad de soluciones de los algoritmos. Este trabajo propone el estudio de un modelo de algoritmo evolutivo paralelo para optimización multiobjetivo, que corresponde a una versión paralela del conocido algoritmo NSGA-II original de Deb et al. (2000) sobre el cual se ha aplicado un esquema de paralelismo basado en poblaciones semi-independientes y migración.

El resto del documento se organiza del modo que se describe a continuación: la sección 2 brinda una breve introducción a los conceptos relacionados con los problemas de optimización multiobjetivo. La sección 3 presenta los conceptos básicos sobre los algoritmos evolutivos y su aplicación a los problemas de optimización multiobjetivo, examinando la versión secuencial del algoritmo NSGA-II. Asimismo, se explican los mecanismos de aplicación de las técnicas de procesamiento paralelo a los algoritmos evolutivos en general y se reseñan las propuestas existentes de aplicación de paralelismo al algoritmo NSGA-II. Los detalles de diseño e implementación de la versión paralela del algoritmo se presentan en la sección 4. El conjunto de problemas de prueba y las métricas utilizadas para evaluar la eficiencia y la calidad de las soluciones obtenidas se presentan en la sección 5. Los resultados experimentales son presentados y analizados en la sección 6. La sección 7 presenta la aplicación del algoritmo diseñado a la resolución de un problema de diseño de redes de comunicaciones confiables. Por último, la sección 8 ofrece las conclusiones del trabajo y posibles líneas de trabajo futuro.

## II. PROBLEMAS DE OPTIMIZACIÓN MULTIOBJETIVO

Esta sección presenta una breve introducción a los problemas de optimización multiobjetivo y sus conceptos relacionados.

Un problema de optimización multiobjetivo plantea la optimización (minimización o maximización) de un conjunto de funciones, habitualmente en conflicto entre sí. La existencia de múltiples funciones objetivo plantea una diferencia fundamental con un problema monoobjetivo: no existirá una *única* solución al problema, sino un conjunto de soluciones que plantearán diferentes compromisos entre los valores de las funciones a optimizar.

La Figura 1 presenta la formulación general de un problema de optimización multiobjetivo. Cabe mencionar que la mayor parte de los problemas de optimización subyacentes a problemas del mundo real tienen una formulación de este tipo, aunque en muchos casos son abordados siguiendo un enfoque monoobjetivo.

Minimizar / Maximizar	$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))$
sujeto a	$\mathbf{G}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_S(\mathbf{x})) \geq \mathbf{0}$
	$\mathbf{H}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_R(\mathbf{x})) = \mathbf{0}$
	$x_i^{(L)} \leq x_i \leq x_i^{(U)} \quad 1 \leq i \leq N$

*Figura 1: Problema de Optimización Multiobjetivo*

La solución al problema de optimización multiobjetivo corresponderá a un vector de variables de decisión  $\mathbf{x} = (x_1, x_2, \dots, x_N)$  que satisfaga las restricciones impuestas por las funciones  $\mathbf{G}$  y  $\mathbf{H}$ , ofreciendo valores que representen un compromiso adecuado para las funciones  $f_1, f_2, \dots, f_M$ .

Considerando el caso de un problema de minimización de funciones, un punto  $\mathbf{x}^*$  es "óptimo de Pareto" si para todo  $\mathbf{x}$  en  $\Omega$ , la región factible del problema, se cumple que  $f_i(\mathbf{x}) = f_i(\mathbf{x}^*) \quad \forall i \in \{1, \dots, M\}$ , o para al menos un valor de  $i$  se cumple que  $f_i(\mathbf{x}) > f_i(\mathbf{x}^*)$ . Esto significa que no existe un vector factible que sea "mejor" que el óptimo de Pareto en alguna función objetivo sin que empeore los valores de alguna de las restantes funciones objetivo.

Asociada con la definición anterior, se introduce una relación de orden parcial denominada *dominancia* entre vectores solución del problema de optimización multiobjetivo. Un vector  $\mathbf{w} = (w_1, w_2, \dots, w_N)$  domina a otro  $\mathbf{v} = (v_1, v_2, \dots, v_N)$  si  $w_i \leq v_i \quad \forall i \in \{1, \dots, M\} \wedge \exists i \in \{1, \dots, M\} / w_i < v_i$ . En este caso se nota  $w \preceq v$ .

Dado que diferentes valores de las variables de decisión representan diferentes compromisos, la resolución de un problema de optimización multiobjetivo no se concentra en hallar un único valor solución, sino que se plantea hallar un conjunto de soluciones *no dominadas*, de acuerdo a la definición presentada anteriormente.

El conjunto de soluciones óptimas al problema de optimización multiobjetivo se compone de los vectores factibles *no dominados*. Este conjunto se denomina *conjunto óptimo de Pareto* y está definido por  $P^* = \{x \in \Omega / \neg \exists x' \in \Omega f(x') \preceq f(x)\}$ . La región de puntos definida por el conjunto óptimo de Pareto en el espacio de valores de las funciones objetivo se conoce como *frente de Pareto*. Formalmente, el frente de Pareto está definido por  $FP^* = \{\mathbf{u} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) / x \in P^*\}$ .

## III. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN MULTIOBJETIVO

La complejidad inherente a los problemas de optimización multiobjetivo plantea un difícil reto para su resolución mediante algoritmos exactos determinísticos a medida que crece la dimensión del espacio de soluciones. De este modo, las técnicas clásicas como los algoritmos enumerativos o los métodos exactos de búsqueda local, basados en gradientes o que utilizan las técnicas estándar de programación determinística –métodos *greedy*, técnicas de *branch & bound*, etc.–, si bien pueden ser aplicables para problemas de complejidad reducida, exigen un costo computacional excesivo para la resolución de problemas multiobjetivo complejos con aplicación en el mundo real.

En este contexto, las *técnicas heurísticas estocásticas* han sido propuestas como alternativas para la resolución de problemas de optimización multiobjetivo, para alcanzar soluciones aproximadas de buena calidad en tiempos de cómputo razonables. Enmarcados en esta categoría, las técnicas de computación

evolutiva se han manifestado como métodos robustos y efectivos para resolución de los problemas de optimización multiobjetivo y se han popularizado en la última década como consecuencia de su éxito.

Los *Algoritmos Evolutivos para Optimización Multiobjetivo* (MOEAs) surgen como una extensión de los EAs para problemas monoobjetivo, utilizando fundamentalmente varios conceptos relacionados con el tratamiento de funciones multimodales por parte de los EAs monoobjetivo.

Esta sección ofrece una breve introducción a los algoritmos evolutivos y a los conceptos relevantes en sus variantes aplicadas a la resolución de problemas multiobjetivo, examinando en detalle la versión secuencial del algoritmo NSGA-II. Complementariamente, se presentan las ideas fundamentales sobre la aplicación de técnicas de procesamiento paralelo a los algoritmos evolutivos en general, y se resumen brevemente las propuestas de paralelismo aplicadas al algoritmo NSGA-II encontradas en la revisión bibliográfica realizada.

#### A. Algoritmos Evolutivos

Los EAs basan su funcionamiento en la simulación del proceso de evolución natural (Goldberg, 1989; Davis, 1991). Consisten en una técnica iterativa que aplica operadores estocásticos sobre un conjunto de individuos –la población– con el propósito de mejorar su *fitness*, una medida relacionada con la función objetivo del problema en cuestión. Cada individuo de la población representa una solución potencial del problema, codificada de acuerdo a un esquema de representación, generalmente basado en números binarios o reales.

Inicialmente la población se genera de forma aleatoria, y luego evoluciona mediante la aplicación iterativa de interacciones denominadas *operadores de reproducción*, que incluyen recombinaciones de individuos –los *cruzamientos*– y modificaciones aleatorias –las *mutaciones*–. Esta evolución es guiada por una estrategia de selección de los individuos más adaptados a la resolución del problema, de acuerdo a sus valores de *fitness*.

La Figura 2 presenta un esquema genérico de un EA, donde puede identificarse el mecanismo evolutivo descrito y los operadores comentados.

```
Inicializar(P(0))
generacion = 0
Evaluar(P(0))
mientras (no CriterioParada) hacer
    Padres = Seleccion(P(generacion))
    Hijos = Operadores de Reproduccion(Padres)
    NuevaPop = Reemplazar(Hijos, P(generacion))
    generacion ++
    P(generacion) = NuevaPop
retornar Mejor Solucion Hallada
```

Figura 2: Esquema de un Algoritmo Evolutivo.

#### B. Algoritmos Evolutivos para Optimización Multiobjetivo

De acuerdo a Coello et al. (2002), la capacidad de los EAs para resolver problemas con múltiples objetivos fue sugerida en la década de 1960 por Rosenberg, pero hasta mediados de la década de 1980 no se presentó la implementación de un algoritmo evolutivo para optimización multiobjetivo (Schaffer, 1984). A partir de la década de 1990 fueron realizadas una gran cantidad de propuestas de MOEAs, formándose una comunidad de investigadores en el área que trabaja activamente en la actualidad.

Dado que trabajan en paralelo sobre un conjunto de soluciones, los EAs tienen la potencialidad de tratar problemas con objetivos múltiples, hallando en cada ejecución un conjunto de soluciones aproximadas al frente de Pareto. Esto representa una importante ventaja respecto a los algoritmos tradicionales, que solamente generan una solución por ejecución. Complementariamente, los EAs tienen otras ventajas respecto a los algoritmos tradicionales, como ser menos sensibles a la forma o a la continuidad del frente de Pareto o permitir abordar problemas con espacio de soluciones de gran dimensión.

Un MOEA debe diseñarse para lograr dos propósitos en forma simultánea: lograr buenas aproximaciones al frente de Pareto y mantener la diversidad de las soluciones, de modo de muestrear adecuadamente el espacio de soluciones y no converger a una solución única o a una sección acotada del frente. La Figura 3 presenta gráficamente los propósitos de un MOEA, donde se ha demarcado con la región celeste (gris) el espacio de búsqueda de funciones objetivo de un problema hipotético, mientras que la línea roja (oscura) gruesa representa al frente de Pareto. El mecanismo evolutivo de los EAs permite lograr el primer propósito, mientras que para preservar la diversidad los MOEAs utilizan las técnicas de *nichos*, *sharing*, *crowding* o similares, utilizadas tradicionalmente por los EAs en la optimización de funciones multimodales.

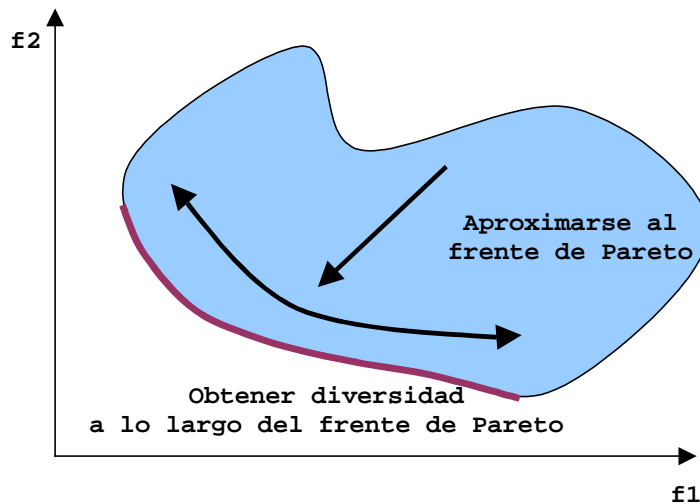


Figura 3: Propósitos de un Algoritmo Evolutivo para Optimización Multiobjetivo.

La figura T presenta el esquema genérico de un MOEA, basado en Veldhuizen et al. (2003). Este esquema cubre la mayoría de los MOEAs más comunes encontrados en la literatura, incluyendo el algoritmo NSGA-II considerado en este estudio.

En la Figura 4 es posible observar dos operadores característicos de un MOEA, que no aparecen en la estructura genérica de un EA. El Operador de diversidad aplica la técnica utilizada para evitar la convergencia a un sector del frente de Pareto (*nichos, fitness sharing, crowding*, etc.). Asimismo, se incluye un procedimiento de Asignación de fitness, orientado a brindar mayor chance de perpetuarse a aquellos individuos con mejores características, considerando los valores de las funciones objetivo y los resultados de la métrica utilizada para evaluar la diversidad.

```

Inicializar(P(0))
generacion = 0
Evaluar(P(0))
mientras (no CriterioParada) hacer
  Operador de diversidad(P(generacion))
  Asignar fitness(P(generacion))
  Padres = Selección(P(generacion))
  Hijos = Operadores de Reproducción(Padres)
  NuevaPop = Reemplazar(Hijos,P(generacion))
  generacion ++
  P(generacion) = NuevaPop
  Evaluar(P(0))
retornar Mejor Solucion Hallada

```

Figura 4: Esquema de un Algoritmo Evolutivo para Optimización Multiobjetivo.

### C. El Algoritmo NSGA-II

El algoritmo NSGA-II (*Non-dominated Sorting Genetic Algorithm*, versión II) fue presentado por K. Deb y sus colegas del Laboratorio de Algoritmos Genéticos del Instituto Tecnológico Kanpur en India en el año 2000 (Deb et al., 2000). Surgió como una versión mejorada del algoritmo NSGA (Srinivas y Deb, 1994), de quién heredó su estructura principal, pero incluyendo características distintivas para resolver tres aspectos fuertemente criticados en la comunidad de investigadores sobre el NSGA: el ordenamiento no dominado, la ausencia de elitismo y la dependencia del parámetro  $\sigma$  para aplicar la técnica de *sharing*.

Las características principales del algoritmo NSGA-II abarcan:

- El ordenamiento no-dominado elitista mediante una técnica de comparación que utiliza una subpoblación auxiliar, que le permite disminuir la complejidad de los chequeos de dominancia de  $O(MP^3)$  a  $O(MP^2)$ , siendo  $M$  el número de funciones objetivo y  $P$  el tamaño de la población utilizada.
- La utilización de una técnica de *crowding* que no requiere especificar parámetros adicionales para la preservación de diversidad en la población, eliminando la dependencia de parámetros como el  $\sigma$  de *sharing* utilizado por el NSGA original.

- La asignación de valores de fitness en base a los niveles o rangos de no dominancia, se hereda del NSGA-II original, aunque se considera en el procedimiento de asignación los valores de distancia de crowding utilizados para evaluar la diversidad de las soluciones.

La figura 5 presenta un esquema del algoritmo NSGA-II, basado en la descripción de Deb et al. (2000). Pueden apreciarse los operadores mencionados, utilizados para el ordenamiento no dominado, evaluación de la diversidad mediante la técnica de crowding y asignación de fitness.

```

Inicializar(P(0))
generacion = 0
Evaluar(P(0))
mientras (no CriterioParada) hacer
    R = Padres  $\cup$  Hijos
    Frentes = Sorting No Dominado(R)
    NuevaPop =  $\emptyset$ 
    i=1
    mientras |NuevaPop| + |Frentes(i)|  $\leq$  sizepop
        Calcular Distancia de Crowding (Frentes(i))
        NuevaPop = NuevaPop  $\cup$  Frentes(i)
        i++
    Sorting por Distancia (Frentes(i))
    NuevaPop = NuevaPop  $\cup$  Frentes(i)[1:(sizepop - |NuevaPop|)]
    Hijos = Selección y Reproducción(NuevaPop)
    generacion ++
    P(generacion) = NuevaPop
retornar Mejor Solucion Hallada

```

Figura 5: Esquema del Algoritmo NSGA-II.

#### D. Algoritmos Evolutivos Paralelos

Las técnicas de procesamiento paralelo y distribuido se aplican a los modelos clásicos de EAs con el propósito de obtener mejoras en la eficiencia computacional y proporcionar un mecanismo diferente de exploración del espacio de búsqueda (Cantú-Paz, 2001). Dividiendo la población en varios elementos de procesamiento, los Algoritmos Evolutivos Paralelos (pEAs) permiten afrontar la lentitud de convergencia para problemas complejos que motivan el uso de poblaciones numerosas o múltiples evaluaciones de funciones objetivo que exigen un costo computacional elevado. Conjuntamente, los pEAs introducen un modelo de evolución diferente al modelo panmítico secuencial, que posibilita explotar el trabajo simultáneo sobre poblaciones geográficamente distribuidas o localizadas de acuerdo a una estructura de organización espacial subyacente.

La organización de la población constituye el principal criterio utilizado por los investigadores para clasificar los modelos de pEAs (Alba y Tomassini, 2002). De este modo, se destacan tres grandes familias de pEAs:

- pEAs basados en el *modelo maestro-esclavo*, donde se distribuye la evaluación de la función de fitness y se mantiene el mecanismo de evolución con interacción panmítica que es característico de los modelos secuenciales. La Figura 6 presenta gráficamente el modelo maestro-esclavo.

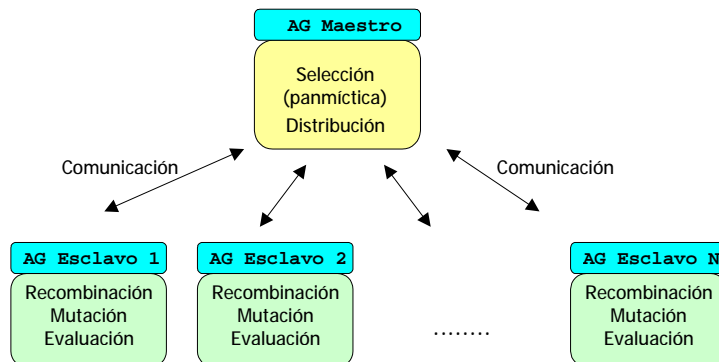


Figura 6: Algoritmo Evolutivo Paralelo Modelo Maestro-Esclavo.

- pEAs de población distribuida, que trabajan con un conjunto de subpoblaciones independientes (islas) con la limitación de que las interacciones solamente son posibles entre individuos de la misma isla. Un operador adicional llamado *migración* posibilita intercambios ocasionales de individuos entre islas, introduciendo una nueva fuente de diversidad. La Figura 7 presenta gráficamente el modelo de población distribuida.

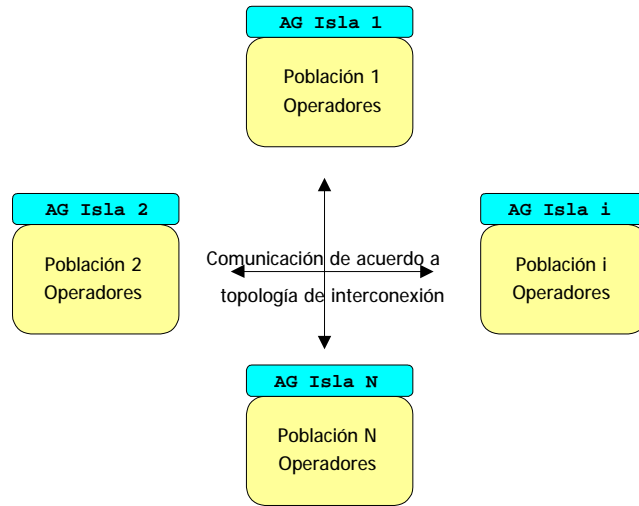


Figura 7: Algoritmo Evolutivo Paralelo Modelo de Población Distribuida.

- pEAs celulares, caracterizados por poseer una estructura espacial subyacente a la población y por su modelo especial de propagación de características de individuos, denominado *difusión*, que sigue las direcciones definidas por la topología de interconexión de elementos de procesamiento. La Figura 8 presenta gráficamente el modelo celular.

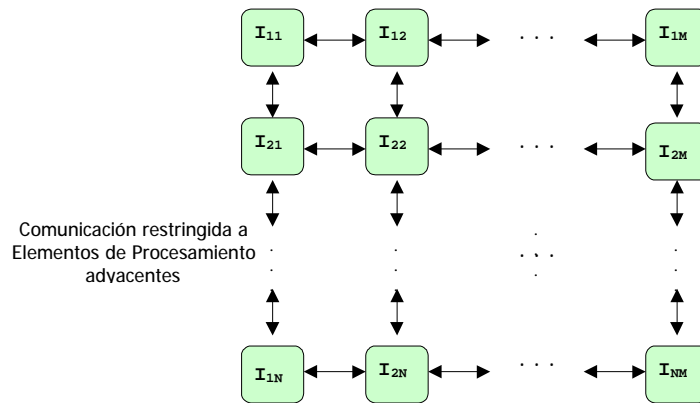


Figura 8: Algoritmo Evolutivo Paralelo Modelo Celular.

Estos modelos tienen diferentes variantes, y existen modelos *híbridos* que combinan las características de dos o más de los modelos de la categorización general presentada.

En la última década, los modelos paralelos de EAs se han popularizado por su eficiencia computacional y aplicabilidad para la resolución de variados problemas en diversas áreas como industria, economía, telecomunicaciones, bioinformática y otras (Alba y Tomassini, 2002).

#### E. Algoritmos Evolutivos Paralelos para Optimización Multiobjetivo

De acuerdo al texto referencia (Coello et al., 2002) el número de propuestas de aplicación de paralelismo en el campo de MOEAs es muy bajo, y el alcance de las propuestas es bastante restringido. La referencia inmediata sobre paralelismo aplicado a MOEAs la constituye el artículo de Veldhuizen et al. (2003) donde los autores exponen las principales consideraciones de diseño, implementación y testeo de algoritmos evolutivos paralelos para optimización multiObjetivo.

Existen únicamente tres referencias a propuestas de aplicación de paralelismo a variantes del algoritmo NSGA. Dos de ellas se concentraron en diseñar versiones paralelas dentro del modelo maestro-esclavo en el área de fluidodinámica, concretamente en problemas de diseño de perfiles aerodinámicos. La tercera propuesta corresponde al propio creador del algoritmo quien propuso un modelo de poblaciones distribuidas.

La primera referencia corresponde a Mäkinen et al. (1996) quienes propusieron un modelo paralelo maestro-esclavo que les permitiera evaluar eficientemente costosas funciones de fitness en un problema de diseño de perfiles aerodinámicos en dos dimensiones. Los autores modificaron el algoritmo NSGA original, sustituyendo la selección mediante ruleta por selección mediante torneo y modificando el mecanismo tradicional de *sharing* por una variante denominada *tournament slot sharing*, que fija al parámetro  $\sigma$  un valor relativo a un slot del torneo. Adicionalmente, introdujeron un mecanismo de evolución elitista, perpetuando los individuos no dominados en cada generación. El algoritmo fue implementado utilizando la biblioteca MPICH de pasaje de mensajes. Las ejecuciones reportadas sobre un IBM SP2 de 8 procesadores modelo 390 presentan buenos valores de eficiencia, aunque el costo total de ejecución de una optimización resulta, aún para el modelo paralelo, excesivamente alto.

Por otra parte, Marco et al. (1999), propusieron aplicar una estrategia de paralelismo en dos niveles para abordar un problema similar, de diseño de perfiles aerodinámicos. La estrategia de paralelismo aplicada al algoritmo NSGA corresponde a la distribución de la evaluación de funciones de fitness –cálculo de flujos Eulerianos–. Los autores reportan haber utilizado un equipo SGI Origin 2000 con procesadores R10000 a 195 Mhz, y trabajando con 8 procesadores obtener un tiempo razonable para la resolución del problema de diseño abordado, pero no se realizan comparaciones de eficiencia contra modelos seriales.

Un último enfoque lo constituye la propuesta de *dominancia guiada por distribución* de Deb et al. (2002), que plantea una búsqueda concurrente mediante división de dominio. Los autores proponen métodos para guiar la búsqueda a diferentes secciones del frente de Pareto introduciendo una transformación de los objetivos a través de una suma ponderada, que modifica el concepto de dominancia posibilitando que diferentes procesos se enfoquen en diferentes regiones de búsqueda. Para posibilitar la cooperación entre procesos, se plantea la necesidad de introducir un operador de migración. Se reportan buenos resultados en lo referente a cobertura del frente de Pareto, y muy buenos resultados respecto a la eficiencia, logrando *speedup* superlineal en el conjunto de problemas de prueba estudiados.

Las técnicas de procesamiento paralelo han sido aplicadas a otros MOEAs, pero hasta el momento de la redacción de este artículo (diciembre de 2003) los artículos referentes al tema no superan los 40 (Coello, 2003), poniendo de manifiesto la escasa atención que ha suscitado esta área a los investigadores. Este hecho presenta como muy promisorio la investigación y el desarrollo de versiones paralelas de MOEAs y su uso para resolver problemas con aplicación en la vida real.

#### IV. UNA VERSIÓN PARALELA DEL ALGORITMO NSGA-II

La idea principal detrás de este trabajo consistió en aplicar un modelo de paralelismo al algoritmo NSGA-II, basado en la propuesta de simplificar al máximo posible el diseño del algoritmo paralelo. El modelo diseñado aplica el paralelismo trabajando con subpoblaciones semi-independientes e introduciendo un operador de migración, que consistió en la única modificación significativa realizada sobre el mecanismo evolutivo de la versión serial del algoritmo NSGA-II. Este capítulo presenta los detalles de diseño e implementación de la versión paralela propuesta, desarrollada sobre el propio código del algoritmo NSGA-II disponible públicamente en la página web del Kanpur Genetic Algorithm Laboratory (<http://www.iitk.ac.in/kangal/soft.htm>)

##### A. Modelo de paralelismo

Se propuso trabajar sobre el modelo de subpoblaciones semi-independientes, también conocido como *modelo de islas*, cuyo esquema aplicado a un algoritmo evolutivo multiobjetivo se presenta en la Figura 9. *Emigrantes* denota al conjunto de individuos a intercambiar con otra isla, seleccionados de acuerdo a una política determinada por *SeleccionMigracion*, eventualmente diferente de la utilizada para seleccionar individuos para el proceso de reproducción. El operador *Migración* intercambia los individuos entre islas de acuerdo a un grafo de conectividad definido entre ellos, generalmente un anillo unidireccional. La *CondicionMigracion* determina cuándo se lleva a cabo el intercambio de individuos. Los individuos *Inmigrantes* son insertados en la población destino, reemplazando a individuos locales de acuerdo a una política de reemplazo determinada.

```

Inicializar(P(0))
generacion = 0
Evaluar(P(0))
mientras (no CriterioParada) hacer
  Operador de diversidad(P(generacion))
  Asignar fitness(P(generacion))
  Padres = Seleccion(P(generacion))
  Hijos = Operadores de Reproduccion(Padres)
  NuevaPop = Reemplazar(Hijos, P(generacion))
  generacion ++
  P(gener) = NuevaPop
  Si (CondicionMigracion)
    Emigrantes = SeleccionMigracion(P(generacion))
    Inmigrantes = Migracion(Emigrantes)
    Insertar(Inmigrantes, P(generacion))
retornar Mejor Solucion Hallada

```

Figura 9: Esquema de un Algoritmo Evolutivo Paralelo Multiobjetivo de Población Distribuida.

Siguiendo el esquema presentado se definió un operador de migración para comunicar asincrónicamente los procesos que evolucionan en paralelo. Se adoptó una topología de migración considerando a los procesos conectados en un anillo unidireccional.

La evolución de cada subpoblación finaliza al alcanzar la condición determinada por el criterio de parada. En ese momento, cada isla envía la totalidad del frente de individuos no dominados de su población a una isla distinguida que actúa como receptora de los individuos no dominados del resto de las islas. De ser necesario, la isla receptora aumenta dinámicamente el tamaño de su población y aplica el proceso evolutivo durante un pequeño número de generaciones extra, potenciando al algoritmo distribuido con una interacción panmíctica que permite mejorar la convergencia –aumentando el número de puntos no dominados globales– y también mejorar la distribución, al aplicar el mecanismo de selección guiada mediante la distancia de crowding sobre la población global del algoritmo. El número de generaciones de interacción panmíctica es un parámetro del modelo paralelo implementado, sugiriéndose utilizar en general un valor reducido para mantener las características del modelo distribuido. Como se comentará en el Capítulo VI, se realizaron experimentos no formalizados para estudiar la conveniencia del uso de la interacción panmíctica y la mejora obtenida al variar el número de generaciones durante la cual se aplica, concluyéndose que con un número reducido de generaciones es posible lograr significativas mejoras en el número de puntos no dominados.

La implementación del modelo de paralelismo se realizó sobre la versión 1.12 de la implementación MPICH de la biblioteca de desarrollo de programas paralelos y distribuidos MPI (MPI Forum, 2003).

### B. Operadores

Para el diseño de los operadores involucrados en el proceso de migración, se siguieron las sugerencias de Veldhuizen et al. (2003). Se adoptó una estrategia de selección elitista aleatoria para la elección de individuos emigrantes, lo que implica que se seleccionen al azar un conjunto de individuos no dominados, de cardinalidad igual a un cierto porcentaje del tamaño de la población. El tamaño del conjunto de emigrantes se especifica como parámetro del algoritmo, pero la idea es utilizar valores pequeños para este parámetro. Mediante el uso de esta estrategia de elección de emigrantes se intenta obtener una presión selectiva "relativamente alta" tratando de lograr un compromiso que permita a las subpoblaciones evolucionar de modo semi-independiente, evitando convergencia hacia el mismo conjunto de puntos no dominados, pero permitiendo la cooperación entre subpoblaciones, determinada por el intercambio ocasional de un conjunto reducidos de individuos bien adaptados para la resolución del problema.

Respecto a la política de reemplazo, se siguió nuevamente una estrategia elitista, que consiste en reemplazar los individuos dominados de la población destino por los inmigrantes arribados. De este modo se trata de mantener los individuos no dominados hallados en el proceso de búsqueda de la subpoblación destino, tratando de obtener una mejor convergencia hacia el frente de Pareto. Tomando en cuenta que para problemas continuos "sencillos" las subpoblaciones obtienen rápidamente un conjunto de puntos no dominados que abarca el total de la población, se implementó una segunda estrategia de reemplazo a aplicar en estos casos. Esta política de reemplazo utiliza como criterio la distancia de *crowding* calculada por el algoritmo, eliminando individuos con valores de distancia de crowding pequeños, considerando que existirán otros individuos representativos de esa sección del espacio fenotípico.



Los mecanismos de migración se ejecutan asincrónicamente, ya que las operaciones de envío y recepción de individuos no bloquean la ejecución del algoritmo, posibilitando que continúe la evolución en cada subpoblación. De acuerdo a la capacidad de procesamiento de los procesos, el intercambio podrá realizarse en diferentes generaciones en las distintas subpoblaciones.

### C. Otras modificaciones

No se realizaron otras modificaciones significativas a la estructura algorítmica del NSGA-II, más allá de las comentadas en la sección anterior. Fue necesario modificar la operativa de acceso a memoria secundaria, en especial la lectura y escritura de archivos de parámetros y de reportes. Se constató que los múltiples accesos a archivos de reportes influyen significativamente en la eficiencia del algoritmo serial y afectan negativamente los tiempos de ejecución de la versión paralela, en especial cuando se utiliza una infraestructura distribuida compuesta por equipos conectados mediante una red de área local, donde el acceso a archivos sea vía NFS. Para evitar este inconveniente y obtener una comparación justa de tiempos de ejecución de los algoritmos, se eliminaron los reportes intermedios del algoritmo, manteniendo únicamente el reporte final. Si fuera necesario obtener datos intermedios de la evolución, es posible reportarlos a archivos locales a los equipos donde se ejecuta cada uno de los procesos.

## V. PROBLEMAS DE PRUEBA

Un amplio número de problemas han sido propuestos para evaluar el desempeño de las versiones secuenciales de MOEAs. Estos conjuntos de problemas se han aplicado también a la evaluación de modelos paralelos, extendiéndose al análisis de la eficiencia computacional. Esta sección presenta el conjunto de problemas de prueba y las métricas utilizadas para evaluar la calidad de las soluciones obtenidas y la eficiencia computacional de la versión paralela del algoritmo NSGA-II diseñada.

### A. Problemas de Prueba

El conjunto de problemas de prueba intentó abarcar una amplia gama de características. En general se trabajó con problemas de dos objetivos que corresponden a funciones continuas aunque se incluyeron casos con más objetivos y casos con funciones discretas. Se trabajó con problemas con frentes de Pareto convexos y cóncavos, conectados y desconectados, problemas con seudofrentes y diferentes distribuciones de puntos en el frente de Pareto real. Asimismo, se consideraron problemas con restricciones y problemas con más de dos funciones objetivo. Los problemas considerados se describen a continuación.

#### Problemas de Zitzler-Deb-Thiele

Zitzler, Deb y Thiele (2000) diseñaron un conjunto de problemas siguiendo los lineamientos presentados por Deb (1999) para el desarrollo sistemático de instancias de problemas de diferente complejidad para evaluar MOEAs. Estos problemas tienen variadas características, pero la principal motivación de su uso la constituye el hecho de poseer frentes de Pareto calculables analíticamente, lo cual permite evaluar de forma precisa la calidad de soluciones obtenidas por un algoritmo.

Los seis problemas del estudio mencionado se generan a partir de la formulación genérica presentada en la Figura 10. Esta formulación se liga con un proceso constructivo que permite afectar diferentes características de los problemas diseñados mediante la manipulación de las funciones involucradas ( $f_1$ ,  $h$  y  $g$ ).

$$\begin{cases} \text{Minimizar} & f_1(\mathbf{x}) = f_1(x_1, x_2, \dots, x_k) \\ \text{Minimizar} & f_2(\mathbf{x}) = g(x_{k+1}, x_{k+2}, \dots, x_n) \cdot h(f_1, g) \end{cases}$$

Figura 10: Formulación genérica de los problemas ZDT

La tabla 1 presenta los seis problemas del conjunto de prueba, indicando su denominación, número de variables y su dominio y especificando las diferentes instancias de las funciones  $f_1$ ,  $f_2$  y  $g$  correspondientes a la formulación general que definen cada problema.

Nombre	N	Dominio	Funciones $f_1, f_2$	Función $g(x)$
ZDT1	30	$x_i \in [0,1]$	$f_1 = x_1$ $f_2 = g(x)\left(1 - \left(\sqrt{x_1/g(x)}\right)\right)$	$1 + \frac{9}{N-1} \sum_{i=2}^N x_i$
ZDT2	30	$x_i \in [0,1]$	$f_1 = x_1$ $f_2 = g(x) \cdot (1 - (x_1/g)^2)$	$1 + \frac{9}{N-1} \sum_{i=2}^N x_i$
ZDT3	30	$x_i \in [0,1]$	$f_1 = x_1$ $f_2 = g(x) \cdot (1 - \sqrt{x_1/g} - (x_1/g) \sin(10\pi x_1))$	$1 + \frac{9}{N-1} \sum_{i=2}^N x_i$
ZDT4	10	$x_1 \in [0,1]$ $x_i \in [-5,5]$ $i = 2, \dots, n$	$f_1 = x_1$ $f_2 = g(x)\left(1 - \left(\sqrt{x_1/g(x)}\right)\right)$	$1 + 10(N-1) + \sum_{i=2}^N (x_i^2 - 10 \cos(4\pi x_i))$
ZDT5	11	$x_i$ booleana $x_1$ 30 bits $x_i$ 5 bits para $i > 1$	$f_1 = u(x_1)$ $f_2 = g(x) / f_1$ $u(x_i) = \text{número de "1s" en } x_i$ $v(u(x_i)) = \begin{cases} 2 + u(x_i) & \text{si } u(x_i) < 5 \\ 1 & \text{si } u(x_i) = 5 \end{cases}$	$\sum_{i=2}^N v(u(x_i))$
ZDT6	10	$x_i \in [0,1]$	$f_1 = 1 - e^{-4x_1} \sin^6(4\pi x_1)$ $f_2 = g(x) \cdot (1 - (f_1(x)/g(x))^2)$	$1 + 9 \left( \left( \sum_{i=2}^N x_i \right) / (N-1) \right)^{0.25}$

Tabla 1: Problemas ZDT

El problema ZDT1 tiene 30 variables en el rango  $[0,1]$ . Su frente de Pareto es convexo, y está determinado por  $0 \leq x_1^* \leq 1$  y  $x_i^* = 0$ , para  $i = 2, \dots, 30$ . Es el problema más sencillo del conjunto, tiene un frente de Pareto continuo y una distribución uniforme de soluciones a lo largo del frente.

El problema ZDT2 tiene 30 variables en el rango  $[0,1]$ . Su frente de Pareto es no convexo, y está determinado por  $0 \leq x_1^* \leq 1$  y  $x_i^* = 0$ , para  $i = 2, \dots, 30$ . La distribución de soluciones a lo largo del frente de Pareto es uniforme.

El problema ZDT3 tiene 30 variables en el rango  $[0,1]$ . Su frente de Pareto es discontinuo, y está determinado por  $x_i^* = 0$  para  $i = 2, \dots, 30$  (no todos los puntos que verifican  $0 \leq x_1^* \leq 1$  pertenecen al frente). La distribución de soluciones a lo largo del frente es uniforme.

El problema ZDT4 tiene 10 variables en el rango  $[0,1]$ . Su frente de Pareto es convexo, y está determinado por  $0 \leq x_1^* \leq 1$  y  $x_i^* = 0$ , para  $i = 2, \dots, 10$ . La complejidad de este problema está dada por la existencia de 100 frentes de Pareto locales en el espacio de soluciones.

El problema ZDT5 utiliza funciones booleanas definidas sobre strings. El frente de Pareto corresponde a los  $x^*$  que hacen mínimo el valor de  $g$ , dado por  $g(x^*) = 10$ . Constituye un caso de problema *deceptivo* ya que la forma de la función auxiliar  $v$  determina que la mayor parte del espacio de búsqueda se concentra cerca de óptimos locales, mientras que el óptimo global se halla relativamente aislado.

El problema ZDT6 tiene 10 variables en el rango  $[0,1]$ . Su frente de Pareto es no convexo, y está determinado por  $0 \leq x_1^* \leq 1$  y  $x_i^* = 0$ , para  $i = 2, \dots, 10$ . La complejidad de este problema está dada por la combinación de la forma no convexa del frente de Pareto y la distribución no uniforme de soluciones a lo largo de él.

### Problemas de Schaffer

Aunque de complejidad muy reducida, los problemas de Schaffer han sido abordados ampliamente por la comunidad de MOEAs. Su importancia es más que nada histórica, ya que fueron introducidos en el primer artículo donde se presentó un MOEA (Schaffer, 1984). Hemos incluido estos simples problemas de prueba, aún conociendo que no proponen retos significativos para los algoritmos actuales. La Tabla 2 presenta la formulación de los problemas de Schaffer.

Nombre	N	Dominio	Planteo del problema
SCH1	1	$x_i \in [-A, A]$	Minimizar $f_1(x) = x^2$ Minimizar $f_2(x) = (x-2)^2$
SCH2	1	$x_i \in [-5, 10]$	Minimizar $f_1(x) = \begin{cases} -x & \text{si } x \leq 1 \\ x-2 & \text{si } 1 \leq x \leq 3 \\ 4-x & \text{si } 3 \leq x \leq 4 \\ x-4 & \text{si } x > 4 \end{cases}$ Minimizar $f_2(x) = (x-2)^2$

Tabla 2: Problemas de Schaffer

Ambos problemas tienen una formulación sencilla, utilizando una única variable de decisión. El problema SCH1 tiene un frente de Pareto convexo dado por la ecuación  $(\sqrt{x^*} - 2)^2$ , para  $0 \leq x^* \leq 4$ . El problema SCH2 tiene un frente de Pareto discontinuo, compuesto por dos secciones correspondientes a los intervalos  $1 \leq x^* \leq 2$  y  $4 \leq x^* \leq 5$  para la variable de decisión  $x$ .

### Problema de Kursawe

Kursawe (1991) presentó un problema de dos objetivos con frente de Pareto discontinuo, formada por tres regiones no convexas, cuya formulación se ofrece en la Tabla 3.

Nombre	N	Dominio	Planteo del problema
KUR	3	$x_i \in [-5, 5]$	Minimizar $f_1(\mathbf{x}) = \sum_{i=1}^2 \left[ -10 \cdot \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2}) \right]$ Minimizar $f_2(\mathbf{x}) = \sum_{i=1}^3 \left[  x_i ^{0.8} + 5 \sin(x_i^3) \right]$

Tabla 3: Problema de Kursawe

Este problema no tiene una expresión analítica simple para su frente de Pareto.

### Problema simple con restricciones de Deb

Deb (2001) presenta en su texto un problema de dos objetivos para ejemplificar el tratamiento de restricciones por parte de los MOEAs, cuya formulación se ofrece en la Tabla 4.

Nombre	N	Dominio	Planteo del problema
CEX	2	$x_1 \in [0, 1]$ $x_2 \in [0, 5]$	Minimizar $f_1(\mathbf{x}) = x_1$ Minimizar $f_2(\mathbf{x}) = \frac{1 + x_2}{x_1}$ sujeto a $g_1(\mathbf{x}) \equiv x_2 + 9x_1 \geq 6$ $g_2(\mathbf{x}) \equiv -x_2 + 9x_1 \geq 1$

Tabla 4: Problema con restricciones de Deb

Parte del frente de Pareto del problema sin restricciones queda dentro de la región de soluciones no factibles al introducir las restricciones. Adicionalmente, una nueva región de óptimos de Pareto surge como consecuencia de las restricciones.

### Problema de Binh y Korn

El problema utilizado por Binh y Korn (1997) corresponde también a una optimización con restricciones, cuya formulación se presenta en la Tabla 5.

Nombre	N	Dominio	Planteo del problema
BNH	2	$x_1 \in [0,5]$ $x_2 \in [0,3]$	Minimizar $f_1(\mathbf{x}) = 4x_1^2 + 4x_2^2$ Minimizar $f_2(\mathbf{x}) = (x_1 - 5)^2 + (x_2 - 5)^2$ sujeto a $g_1(\mathbf{x}) \equiv (x_1 - 5)^2 + x_2^2 \leq 25$ $g_2(\mathbf{x}) \equiv (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7$

Tabla 5: Problema de Binh y Korn

La restricción  $g_2(x)$  es redundante ya que no restringe el dominio de soluciones, mientras que la restricción  $g_1(x)$  reduce la densidad de soluciones en una sección del espacio de funciones objetivo. La forma y la continuidad del frente de Pareto no son afectadas por la introducción de las restricciones.

### Problemas de Viennet

Viennet et al. (1996) presentaron un conjunto de problemas con tres funciones objetivo, de las cuales en este trabajo se consideran los dos problemas cuya formulación se ofrece en la Tabla 6.

Nombre	N	Dominio	Planteo del problema
VNT2	2	$x_i \in [-4,4]$	Minimizar $f_1(\mathbf{x}) = \frac{(x_1 - 2)^2}{2} + \frac{(x_2 + 1)^2}{13} + 3$ Minimizar $f_2(\mathbf{x}) = \frac{(x_1 + x_2 - 3)^2}{36} - \frac{(-x_1 + x_2 + 2)^2}{8} - 17$ Minimizar $f_3(\mathbf{x}) = \frac{(x_1 + 2x_2 - 1)^2}{175} - \frac{(-x_1 + 2x_2)^2}{17} - 13$ sujeto a $g_1(\mathbf{x}) \equiv x_1^2 + x_2^2 \leq 0$ $g_2(\mathbf{x}) \equiv x_1 - 3x_2 + 10 \leq 0$
VNT3	2	$x_i \in [-3,3]$	Minimizar $f_1(\mathbf{x}) = \frac{(x_1^2 + x_2^2)}{2} + \sin(x_1^2 + x_2^2)$ Minimizar $f_2(\mathbf{x}) = \frac{(3x_1 - 2x_2 + 4)^2}{8} - \frac{(x_1 - x_2 + 1)^2}{27} + 15$ Minimizar $f_3(\mathbf{x}) = \frac{1}{(x_1^2 + x_2^2 + 1)} - 1.1e^{(-x_1^2 - x_2^2)}$ sujeto a $g_1(\mathbf{x}) \equiv x_1^2 + x_2^2 \leq 0$ $g_2(\mathbf{x}) \equiv x_1 - 3x_2 + 10 \leq 0$

Tabla 6: Problemas de Viennet

### B. Métricas de desempeño y eficiencia

Una variada gama de métricas han sido propuestas para evaluar la calidad de los resultados por los MOEAs en los textos de referencia (Coello et al, 2002; Deb, 2001). En general las métricas evalúan el desempeño de los algoritmos en el espacio de fenotipos, como consecuencia de que la mayoría de los algoritmos se enfocan en hallar el verdadero frente de Pareto en la resolución de problemas.

En este trabajo hemos utilizado cuatro métricas para evaluar la calidad de los resultados de la versión paralela del algoritmo NSGA-II, comparativamente con los resultados de la variante serial. Se han tomado en cuenta los dos propósitos generales de los MOEAs, la convergencia al frente de Pareto y la distribución de puntos no dominados en el frente de Pareto calculado por el algoritmo.

Debe tenerse en mente que la evaluación comparativa de resultados entre MOEAs es un tema complejo. Para enfocar este asunto se presenta el siguiente ejemplo, tomado de Deb (2002). En un escenario como el presentado en la Figura 11 es posible concluir que el algoritmo A supera en calidad de resultados al algoritmo B, al mostrar en sus resultados finales mejor convergencia al frente de Pareto y aproximadamente la misma diversidad. Pero en una situación más confusa como la que se presenta en la Figura 12, las conclusiones no pueden ser formuladas ligeramente. Algunos puntos del algoritmo A dominan a los resultados finales del algoritmo B, mientras que otros son dominados. La determinación de cual algoritmo es "mejor" que el otro dependerá de la definición exacta de las métricas utilizadas. Asimismo, la naturaleza no determinística de los EAs hace necesario realizar estudios estadísticos sobre un número considerable de ejecuciones independientes para tomar decisiones sobre la superioridad de un algoritmo sobre otro en términos de calidad de soluciones obtenidas.

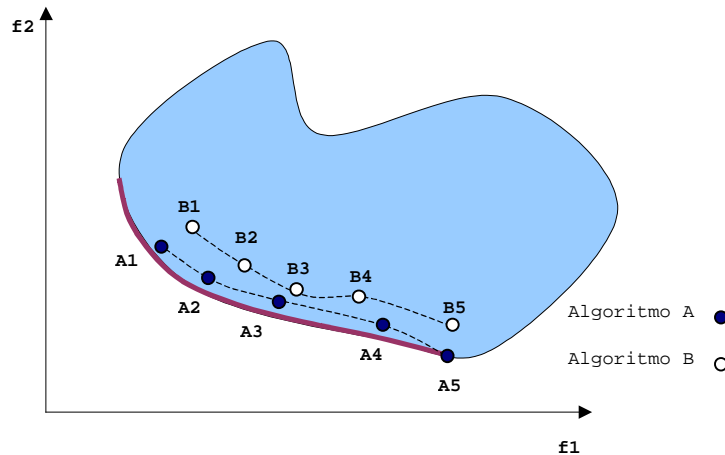


Figura 11: El algoritmo A obtiene mejores resultados que el algoritmo B.

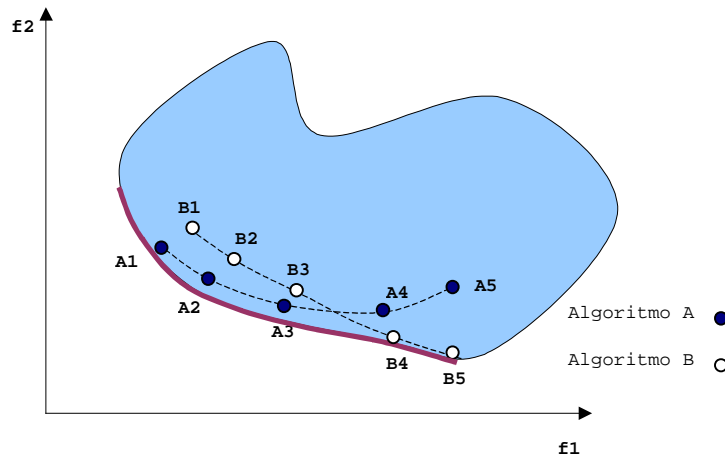


Figura 12: Los resultados de los algoritmos A y B son difíciles de comparar.

Adicionalmente, para evaluar la eficiencia computacional, se decidió utilizar un criterio habitual en el área de la computación paralela: medir los tiempos de ejecución del algoritmo paralelo y calcular el *speedup* obtenido al utilizar un determinado número de procesadores en la resolución de los problemas presentados.

A continuación se presentan las métricas utilizadas para evaluar la calidad de las soluciones obtenidas y la eficiencia computacional de los algoritmos.

#### Número de puntos no dominados

Esta métrica evalúa la cantidad efectiva de puntos no dominados *diferentes* que halla el algoritmo NSGA-II. Dado que no se introducen restricciones para la cantidad de individuos idénticos en la población, es necesario tomar en cuenta esta medida para evaluar adecuadamente el tamaño de puntos no dominados obtenidos en la población final. En la presentación de las restantes métricas ser referenciará al número de puntos no dominados *diferentes* al cual notaremos por  $q$ .

### Distancia generacional

Esta métrica calcula la distancia promedio entre los puntos del frente de Pareto calculado por el algoritmo y el frente de Pareto verdadero. La Ecuación 1 presenta la formulación de esta métrica, introducida por Veldhuizen y Lamont (1999).

$$GD = \frac{1}{q} \left( \sum_{i=1}^q d_i^p \right)^{1/p}$$

*Ecuación 1: Definición de la métrica distancia generacional.*

En la ecuación 1, el término  $d_i^p$  denota a la distancia entre la solución  $i$ -ésima del frente de Pareto calculado por el algoritmo y el punto del frente de Pareto verdadero más próximo. Por ejemplo, para el caso  $p = 2$ ,

corresponde a la distancia Euclideana  $d_i^p = \min_j \sqrt{\sum_{k=1}^M (f_k^i - f_k^{*j})^2}$ , donde  $f_k^i$  y  $f_k^{*j}$  indican el valor

de la  $k$ -ésima función objetivo evaluada en la solución  $i$ -ésima del frente de Pareto calculado por el algoritmo y en el punto  $j$ -ésimo (siendo éste el el más cercano a la solución  $i$ -ésima) del frente de Pareto real.

### Spacing

La métrica *spacing* evalúa la distribución de puntos no dominados en el frente de Pareto calculado por el algoritmo. La Ecuación 2 presenta la definición de la métrica utilizada, original de Schott (1995) y reproducida en los textos de referencia (Coello et al, 2002) (Deb, 2001).

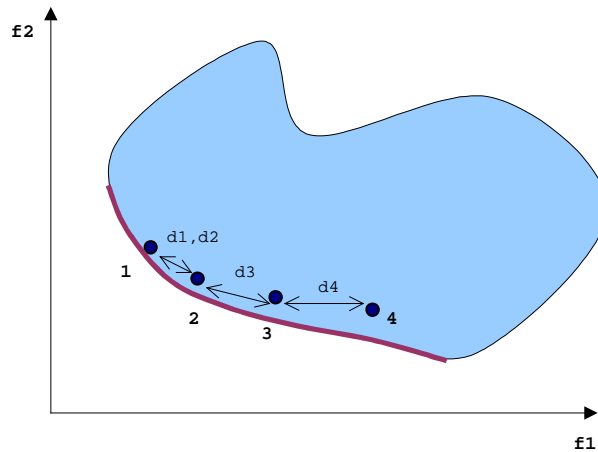
$$Spacing = \sqrt{\frac{1}{q-1} \sum_{i=1}^q (\bar{d} - d_i)^2}$$

*Ecuación 2: Definición de la métrica spacing.*

En la Ecuación 2, el término  $d_i$  mide la distancia en el espacio de las funciones objetivo entre la solución  $i$ -ésima y su vecino más próximo –la solución  $j$ -ésima– en el frente de Pareto calculado por el algoritmo:

$d_i = \min_j (|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)| + \dots + |f_M^i(x) - f_M^j(x)|)$ ,  $i = 1, \dots, q$ . El valor  $\bar{d}$

corresponde al promedio de los  $d_i$ . Un valor ideal de cero en esta métrica indicaría que todos los puntos obtenidos en el frente de Pareto calculado por el algoritmo están igualmente espaciados. La figura 13 presenta una descripción gráfica de las distancias involucradas en el cálculo de la métrica *spacing*.



*Figura 13: Distancias involucradas en el cálculo de la métrica spacing*

## Spread

La métrica *spacing* evalúa la distribución de puntos en el frente de Pareto calculado, pero puede no dar una idea certera de la cobertura de las soluciones, ya que no utiliza información adicional sobre el frente de Pareto real. La métrica *spread*, presentada por Deb et al. (2000), propone utilizar como información adicional la distancia a los "extremos" (puntos con menores valores en cada una de las funciones objetivo) del frente de Pareto real para tener una medida más precisa de la cobertura del frente.

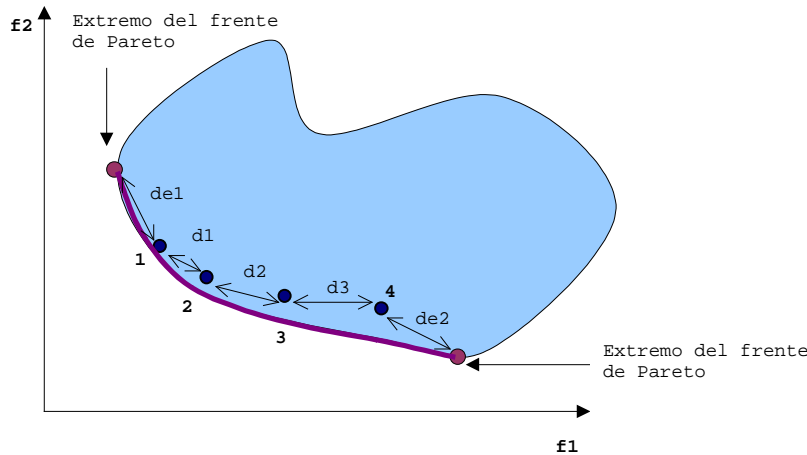
Tomando en cuenta los "extremos" del frente de Pareto real se evita el problema comentado de la métrica *spacing*, que podría dar buenos valores de distribución, aún cuando en realidad no se estuviera muestreando adecuadamente el frente real. La Ecuación 3 ofrece la definición de la métrica *spread*.

$$Spread = \frac{\sum_{k=1}^M d_k^e + \sum_{i=1}^q |\bar{d} - d_i|}{\sum_{k=1}^M d_k^e + q \cdot \bar{d}}$$

*Ecuación 3 : Definición de la métrica spread.*

En la ecuación 3, el término  $d_k^e$  mide la distancia entre el punto "extremo" del frente de Pareto real, tomando en cuenta la función objetivo  $k$ -ésima, y el punto del frente de Pareto calculado más cercano. El término  $d_i$  corresponde a una medida de la distancia, en el espacio de las funciones objetivo, entre la solución  $i$ -ésima del frente de Pareto calculado y su vecino más cercano. El valor  $\bar{d}$  corresponde al promedio de los  $d_i$ .

La figura 14 presenta una descripción gráfica de las distancias involucradas en el cálculo de la métrica *spread*. Los extremos del frente de Pareto se han marcado con círculos para su mejor visualización.



*Figura 14: Distancias involucradas en el cálculo de la métrica spread.*

## Speedup y eficiencia

La siguiente definición de *speedup* en el contexto de Algoritmos Evolutivos Paralelos puede encontrarse en el artículo de Alba (2002) que se ha tomado como referencia. El *speedup* mide la relación entre el tiempo medio de ejecución del algoritmo ejecutando sobre un procesador y el tiempo medio de ejecución del algoritmo ejecutando sobre  $m$  procesadores. Corresponde a la expresión de la Ecuación 4, donde se ha notado por  $T_m$  el tiempo de ejecución de un algoritmo ejecutando sobre  $m$  procesadores. La introducción de los valores medios obedece a la naturaleza no determinística de los tiempos de ejecución de programas paralelos.

$$Speedup_m = \frac{E[T_1]}{E[T_m]}$$

*Ecuación 4: Definición de speedup.*

Esta medida evalúa que tan eficiente resulta el algoritmo paralelo cuando se dispone de  $m$  recursos computacionales, referente a la eficiencia del algoritmo serial y tomando como medida el tiempo de ejecución de los procesos. Utilizando esta definición, se distingue entre los casos de *speedup sublineal* ( $Speedup_m < m$ ) y *speedup superlineal* ( $Speedup_m > m$ ).

En problemas vinculados a otras áreas de investigación, uno de los objetivos principales al aplicar técnicas de procesamiento paralelo lo constituye el alcanzar valores cercanos al speedup lineal, propuesto generalmente como una cota superior para la eficiencia. En el área de los algoritmos evolutivos, la naturaleza de los modelos y las características de las plataforma de ejecución hacen frecuente el alcanzar valores de speedup superlineal, tal como se expone en Alba (2002).

Considerando la clasificación presentada en el artículo de referencia mencionado anteriormente, en este trabajo se utiliza una versión de speedup algorítmico débil, tomando en cuenta que se comparan los tiempos de ejecución del algoritmo paralelo contra los de su versión serial.

La medida de *eficiencia* normaliza el valor del speedup a un porcentaje, correspondiendo el valor 100% al speedup lineal, de acuerdo a la expresión de la Ecuación 5.

$$Eficiencia = \frac{Speedup_m}{m} \cdot 100\%$$

Ecuación 5 : Definición de eficiencia.

## VI. RESULTADOS

Esta sección presenta los detalles de los experimentos comparativos realizados para evaluar la versión paralela diseñada, los resultados obtenidos y su análisis. Para cada uno de los problemas presentados en la sección precedente se realizaron 30 ejecuciones independientes de las versiones paralela y serial del algoritmo NSGA-II y se evaluaron comparativamente los resultados obtenidos considerando las métricas de proximidad al frente de Pareto, diversidad de soluciones y eficiencia computacional presentadas.

### A. Plataforma de ejecución

Siguiendo las sugerencias de Veldhuizen et al. (2003), la Tabla 7 presenta los parámetros relevantes de la plataforma de ejecución utilizada, en un intento por ofrecer una descripción formal del equipamiento que permita una comparación adecuada con futuros experimentos de aplicación de pMOEAs a algunos de los problemas estudiados en este trabajo.

Característica	Descripción
Computador paralelo	Cluster Intel
CPU	Pentium IV a 2.4 GHz
Nodos	Disponibles : 16, utilizados : 4 y 8
Memoria por nodo	512 Mb RAM
Sistema Operativo	SuSE Linux 8.0
Red de Comunicación	LAN Fast Ethernet a 100Mb/sec.
Biblioteca de desarrollo	MPI, implementación MPICH v 1.12

Tabla 7: Características de la plataforma de ejecución.

### B. Ajuste de operadores y parámetros

No se realizaron experimentos con el objetivo de hallar una configuración óptima de valores de los parámetros del algoritmo serial para cada problema. El alto número de problemas de prueba y la gran cantidad de instancias requeridas para estimar los mejores valores impidieron realizar el estudio paramétrico. Tampoco se realizaron pruebas para determinar la influencia de utilizar un determinado operador en aquellos casos en que existen alternativas (como en el caso de cruzamiento simple contra cruzamiento uniforme, por ejemplo), utilizándose cruzamiento y mutación simple en cada caso.

Los resultados que se presentan fueron obtenidos con la configuración de parámetros que se ofrecen en la Tabla 8. Los valores de probabilidad fueron tomados de Deb et al. (2000) quien los utiliza para la resolución de 5 problemas ZDT y problemas con restricciones, entre otros. Aún tomando en cuenta que la configuración paramétrica escogida evidentemente no es la adecuada para obtener los mejores resultados para todos los problemas considerados, en caso de que los parámetros influyan en el mecanismo de búsqueda lo harán tanto en el modelo secuencial como en el paralelo, sin afectar de sobremanera el estudio comparativo.



Se definió un criterio de parada basado en especificar un esfuerzo prefijado, determinado por un número de generaciones. Se trabajó con dos valores elevados para el tope, fijados en 200 y 500 generaciones, tratando de dar al algoritmo una potente capacidad de búsqueda.

Respecto al tamaño de las poblaciones utilizadas, inicialmente se propuso trabajar con un tamaño de 100 individuos, valor sugerido en Deb et al. (2000). Tomando en cuenta la escasa complejidad de las funciones objetivo de los problemas considerados, se realizaron luego ejecuciones con poblaciones de 400 individuos para poder apreciar la capacidad del modelo paralelo de mejorar la eficiencia computacional.

<i>Parámetro</i>	<i>Valor</i>
Probabilidad de cruzamiento	0.9
Probabilidad de mutación	$1/N$ ó $1/L$ siendo $N$ el número de variables en problema con codificación real y $L$ el largo de string en problema con codificación binaria
Índice de distribución para cruzamiento real	20
Índice de distribución para mutación real	20
Número de generaciones utilizadas como	200 y 500 generaciones
Tamaño de la población	100 y 400 individuos
<b><i>Parámetros Adicionales para la Versión Paralela</i></b>	
Número de subpoblaciones utilizadas	4
Frecuencia de migración	25 generaciones
Individuos participantes en la migración	5 individuos
Topología de migración	Anillo unidireccional
Número de generaciones de interacción	10 generaciones

*Tabla 8: Parámetros utilizados.*

Se realizaron experimentos no formalizados para determinar valores adecuados para los parámetros del modelo de migración, utilizando los problemas ZDT1 y ZDT2. Se observó que, tomando en cuenta la escasa complejidad de los problemas involucrados, la sensibilidad de los resultados respecto a variaciones en los parámetros no era significativa. Se decidió utilizar un valor alto para la frecuencia de migración (25 generaciones) y un valor pequeño para el número de individuos intercambiados en cada migración (5 individuos). Disminuyendo la frecuencia de migración o incrementando el número de individuos participantes en cada migración, la versión distribuida tendría un comportamiento cada vez más parecido al modelo panmítico serial, situación que se intentó evitar.

Por otra parte, se investigó la influencia del número de generaciones de interacción panmítica en los resultados, mostrando que un valor bajo de 10 generaciones era suficiente para lograr un número de puntos no dominados similar a la obtenida por el algoritmo serial.

### *C. Resultados*

La Tabla 9 resume los resultados obtenidos por los modelos secuencial y paralelo del algoritmo NSGA-II sobre los 14 problemas de prueba considerados. Se ofrecen los valores promedio y de desviación estándar de las medidas tiempo de ejecución (en segundos), puntos no dominados diferentes hallados por cada algoritmo, distancia generacional, spread y spacing.

Los resultados presentados en la Tabla 9 corresponden a la ejecución con una población de 400 individuos y un criterio de parada de 500 generaciones, valores de los parámetros que permiten apreciar las ventajas del algoritmo paralelo en lo referente al tiempo de ejecución.

Como fue presentado en el capítulo precedente, el cálculo de la distancia generacional asume conocido el frente de Pareto real para cada problema. En los casos de los problemas ZDT, que fueron diseñados para tener una expresión sencilla del frente de Pareto, y para los problemas SCH1, SCH2, CEX y BNH donde existe una expresión analítica para el frente de Pareto, se generó el mismo utilizando una discretización de paso  $1 \times 10^{-5}$ . Para el resto de los problemas, se utilizaron como valores ideales del frente de Pareto los calculados por Nebro et al. (2003) en el marco del proyecto ESaM (Enumerative Search applied to Multi-objective optimization), donde se utiliza una técnica enumerativa simple basada en evaluar los puntos de un espacio discretizado finito. Del mismo modo se calcularon los extremos del frente de Pareto, necesarios para el cálculo de la métrica spread.

En el Anexo I se presentan la totalidad de los resultados obtenidos en cada una de las ejecuciones realizadas.

Para el problema ZDT5 no se han calculado las métricas de diversidad propuestas, tomado en cuenta las características del espacio de búsqueda, que se encuentra discretizado en valores enteros de la función  $f_i$ . En este problema, la cantidad de puntos no dominados diferentes constituye una medida de que tan bien se muestrean los 31 puntos del frente de Pareto discreto.

Problema	Modelo	Medida	Tiempo (seg.)	Puntos ND	Distancia Generacional	Spread	Spacing
SCH1	Serial	Promedio	21.99	378.0	0.0001	0.393	0.128
		Desv. Est.	0.30	5.23	0.0000	0.015	0.007
	Paralelo	Promedio	3.54	375.5	0.0001	0.385	0.123
		Desv. Est.	0.20	5.22	0.0000	0.015	0.002
SCH2	Serial	Prom	36.10	379.0	0.0001	0.767	0.137
		Desv. Est.	0.14	4.44	0.0000	0.011	0.002
	Paralelo	Prom	3.52	370.0	0.0001	0.776	0.138
		Desv. Est.	0.08	4.54	0.0000	0.011	0.002
ZDT1	Serial	Promedio	21.67	368.0	0.000007	0.406	0.0611
		Desv. Est.	1.74	5.4	0.000004	0.013	0.0007
	Paralelo	Promedio	5.43	366.5	0.000008	0.406	0.0611
		Desv. Est.	0.17	6.2	2.17E-06	0.014	0.0010
ZDT2	Serial	Promedio	26.00	366.5	0.000004	0.401	0.061
		Desv. Est.	0.52	5.49	1.1E-06	0.014	0.001
	Paralelo	Promedio	5.58	365.5	0.000004	0.398	0.062
		Desv. Est.	0.16	5.29	1.8E-06	0.020	0.001
ZDT3	Serial	Promedio	29.24	368.0	0.000024	0.401	0.0623
		Desv. Est.	2.42	4.6	1.9E-06	0.013	0.0010
	Paralelo	Promedio	6.32	370.0	0.000024	0.402	0.0622
		Desv. Est.	0.92	4.2	1.4E-06	0.018	0.0008
ZDT4	Serial	Promedio	26.41	390.0	0.000056	0.368	0.061
		Desv. Est.	2.86	3.69	5.2E-05	0.012	0.001
	Paralelo	Promedio	4.92	390.0	0.000180	0.359	0.061
		Desv. Est.	0.77	3.69	6.8E-05	0.020	0.001
ZDT5	Serial	Promedio	34.65	25.0	0.3445	-	-
		Desv. Est.	3.54	0.0	0.00	-	-
	Paralelo	Promedio	9.13	25.0	0.2283	-	-
		Desv. Est.	1.06	1.75	0.0724	-	-
ZDT6	Serial	Promedio	25.90	361.5	0.0715	0.466	0.110
		Desv. Est.	0.39	5.84	0.0065	0.026	0.003
	Paralelo	Promedio	3.06	362.0	0.0700	0.433	0.056
		Desv. Est.	0.14	6.23	0.0041	0.018	0.001
KUR	Serial	Promedio	20.57	387.5	0.0094	0.483	0.322
		Desv. Est.	0.10	3.73	0.0006	0.038	0.008
	Paralelo	Promedio	3.02	388.0	0.0096	0.452	0.163
		Desv. Est.	0.11	2.97	0.0005	0.021	0.002
CEX	Serial	Promedio	21.62	355.5	0.00043	0.734	0.125
		Desv. Est.	0.04	5.52	7.9E-05	0.015	0.002
	Paralelo	Promedio	2.90	352.0	0.00059	0.942	0.114
		Desv. Est.	0.07	6.41	8.4E-05	0.027	0.003
BNH	Serial	Promedio	27.31	395.5	0.0132	0.450	0.577
		Desv. Est.	0.30	2.85	0.0011	0.015	0.006
	Paralelo	Promedio	3.54	395.0	0.0152	0.445	0.579
		Desv. Est.	1.05	1.97	0.0013	0.014	0.004
VI2	Serial	Promedio	25.84	391.0	0.0015	0.6137	0.097
		Desv. Est.	0.03	2.72	0.0002	0.027	0.002
	Paralelo	Promedio	3.24	392.5	0.0015	0.6296	0.087
		Desv. Est.	0.22	3.20	0.0002	0.030	0.002
VI3	Serial	Promedio	23.28	374.5	0.0223	0.7519	0.154
		Desv. Est.	0.16	4.47	0.0006	0.028	0.003
	Paralelo	Promedio	3.95	375.5	0.0222	0.7638	0.155
		Desv. Est.	0.07	4.96	0.0004	0.029	0.002

Tabla 9: Resultados obtenidos sobre los problemas de prueba.

Como puede apreciarse en la Tabla 9, el algoritmo paralelo obtuvo resultados de similar calidad que el algoritmo serial, tanto al evaluar la aproximación al frente de Pareto como la diversidad de las soluciones finales. Para el problema ZDT5 el algoritmo paralelo alcanzó mejores valores de distancia generacional, mientras que para los problemas ZDT6 y KUR el algoritmo paralelo alcanzó mejores valores de distribución de puntos. Por otra parte, para el problema ZDT4 el algoritmo paralelo alcanza un peor valor de distancia generacional que el algoritmo serial.

La Tabla 10 presenta los resultados obtenidos en experimentos realizados con menores valores de tamaño de población (100 individuos) y de número de generaciones (200), para los problemas en que el algoritmo paralelo mejoró los resultados del serial.

Problema	Modelo	Medida	Tiempo (seg.)	Puntos ND	Distancia Generacional	Spread	Spacing
ZDT5	Serial	Promedio	2.08	23	0.5912	-	-
		Desv. Est.	0.04	0	0.0	-	-
	Paralelo	Promedio	2.51	30	0.0511	-	-
		Desv. Est.	0.21	1	0.011	-	-
ZDT6	Serial	Promedio	0.76	89	0.07418	0.466	0.110
		Desv. Est.	0.01	3.09	0.00463	0.026	0.003
	Paralelo	Promedio	1.12	89	0.00024	0.415	0.113
		Desv. Est.	0.03	2.41	0.02436	0.047	0.004
KUR	Serial	Promedio	2.25	98	0.0128	0.908	0.322
		Desv. Est.	0.24	0.97	0.0023	0.008	0.008
	Paralelo	Promedio	3.02	98	0.015	0.874	0.288
		Desv. Est.	0.11	1.34	0.003	0.011	0.020

Tabla 10: Resultados para los problemas donde la versión paralela obtuvo mejores resultados que la serial.

Analizando la Tabla 10 puede apreciarse que el algoritmo paralelo obtuvo una significativa mejora respecto al serial en cuanto a la distancia generacional para el problema ZDT5. El algoritmo serial obtiene para este problema el mismo conjunto de soluciones no dominadas en las diferentes ejecuciones, mostrando siempre el mismo valor de distancia generacional. El algoritmo paralelo muestra una diversidad mayor, alcanzando un mayor número de puntos no dominados, que le permite disminuir en un factor de 10 la distancia promedio al Frente de Pareto. Una situación diferente se presenta en el problema ZDT6, donde ambas versiones del algoritmo obtienen el mismo número de puntos no dominados, pero con la versión paralela muestreando casi de forma "perfecta" el frente de Pareto.

En lo referente al desempeño computacional, la Tabla 11 presenta los tiempos de ejecución (en segundos) y los valores de speedup y eficiencia obtenidos para cada uno de los problemas abordados.

		SCH1	SCH2	ZDT1	ZDT2	ZDT3	ZDT4	ZDT5	ZDT6	KUR	CEX	BNH	VI2	VI3
Modelo	Tiempo Serial	21.99	36.10	21.67	26.00	29.24	26.41	34.65	25.90	20.57	21.62	27.31	25.84	23.28
	Tiempo Paralelo	3.54	3.52	5.43	5.58	6.32	4.92	9.13	3.06	3.02	2.90	3.54	3.24	3.95
Medida	Speedup	6.21	10.26	3.99	4.66	4.63	5.37	3.80	8.47	6.82	7.45	7.73	7.98	5.89
	Eficiencia	1.55	2.56	0.99	1.16	1.15	1.34	0.95	2.12	1.71	1.86	1.93	1.99	1.47

Tabla 11: Evaluación del desempeño.

Analizando los valores de la Tabla 11, puede apreciarse la notoria mejora en el desempeño obtenido al utilizar el algoritmo paralelo para resolver los problemas utilizando 4 procesadores. Se obtuvo un speedup superlineal para todos los casos estudiados, salvo en los problemas ZDT1 y ZDT5 donde el comportamiento observado correspondió a un speedup lineal. Estos resultados confirman la argumentación de Deb et al. (2002) en su estudio sobre la estrategia de paralelismo por división de dominio aplicada sobre el algoritmo NSGA-II, donde se reporta haber alcanzado valores de speedup superlineal, aunque no se presentan resultados numéricos de la evaluación de eficiencia.

## VII. APLICACIÓN AL DISEÑO DE REDES DE COMUNICACIONES CONFIABLES

Al diseñar redes de comunicaciones, un problema importante consiste en hallar una topología de conexión de sus nodos cuyas propiedades aseguren la comunicación confiable de datos. En los últimos años, el rápido desarrollo de la infraestructura de redes, del software y de los servicios de Internet han renovado el interés por los problemas de diseño de redes de comunicaciones. Como consecuencia del continuo crecimiento en el tamaño de las redes, los problemas de optimización subyacentes sobrepasan con frecuencia la capacidad de los algoritmos exactos tradicionales. En este contexto, varias heurísticas se han aplicado al diseño de redes de comunicaciones confiables cuando se desea resolver problemas de dimensiones reales en tiempos razonables. Entre ellas, las técnicas de computación evolutiva se han manifestado como métodos flexibles y robustos para la solución de los complejos problemas de optimización relacionados con el diseño de redes confiables.

Este capítulo presenta la aplicación de la versión paralela del algoritmo NSGA-II diseñada a la resolución de una versión multiobjetivo del Problema de Steiner Generalizado, un problema de optimización combinatoria que modela el diseño de redes de comunicaciones de alta confiabilidad topológica.

### A. El Problema de Steiner Generalizado

Considerando una red de comunicaciones con nodos distinguidos denominados *terminales*, el Problema de Steiner Generalizado (GSP) consiste en diseñar una subred de mínimo costo que verifique requisitos prefijados de conexión entre pares de nodos terminales. Usualmente, la minimización del costo de las conexiones se contrapone con el objetivo de maximizar las propiedades de confiabilidad de la red. Como ejemplo, un modelo que no agregue un mínimo nivel de redundancia de caminos conducirá a una topología de árbol para la red, poco útil en escenarios reales, ya que no es capaz de soportar fallos en sus componentes. El GSP incorpora requisitos adicionales de conectividad para garantizar la alta confiabilidad en las comunicaciones que demandan los escenarios reales.

La siguiente formulación del GSP se basa en la presentada en el compendio de problemas de optimización NP de Kahn y Crescenzi (2003).

Considérense los siguientes elementos:

- Un grafo no dirigido  $G = (V, E)$ , siendo  $V$  el conjunto de nodos y  $E$  el conjunto de aristas que representan a los enlaces bidireccionales de comunicación entre nodos.
- Una matriz de costos  $C$  asociados a las aristas del grafo  $G$ .
- Un subconjunto fijo del conjunto de nodos  $T \subseteq V$  llamados nodos *terminales*, de cardinalidad  $n_T = |T|$ , tal que  $2 \leq n_T \leq n$ , siendo  $n = |V|$  la cardinalidad del conjunto de nodos  $V$ .
- Una matriz  $n_T \times n_T$  simétrica  $R = r_{ij}$  con  $i, j \in T$ , cuyos elementos son enteros no negativos que indican los requisitos de conectividad –cantidad de caminos disjuntos requeridos– entre todo par de nodos terminales  $i$  y  $j$ .

El GSP plantea encontrar un subgrafo  $G_T \subseteq G$  de costo mínimo, tal que todo par de nodos  $i, j \in T$ , sean  $r_{ij}$  arista-conexos en  $G_T$ , es decir que existan  $r_{ij}$  caminos disjuntos, que no comparten aristas, entre los nodos  $i$  y  $j$  en  $G_T$ . Sobre los nodos no terminales no se plantean requisitos de conectividad. Éstos, llamados *nodos de Steiner*, pueden formar parte o no de la solución óptima, de acuerdo a la conveniencia de utilizarlos.

La formulación presentada corresponde al modelo arista conexo del GSP, donde se asumen que los enlaces de comunicación pueden fallar, pero los nodos son perfectos. Una formulación análoga existe para la versión nodo conexa del GSP, donde los nodos son susceptibles a fallas y por ello los diversos caminos entre cada par de nodos terminales deben ser disjuntos respecto a los nodos que incluyen.

La complejidad del problema de Steiner obedece a la generalidad de su planteamiento, al exigir requisitos variables de conectividad entre pares de nodos terminales. Ciertas variantes simplifican estos requisitos: la subclase de *Problemas de k-conexión* exigen un número fijo  $k$  de caminos disjuntos entre pares de nodos terminales. El caso más simple de problema de Steiner exige sólo un camino entre pares de nodos; la solución a este problema tiene topología de árbol y por ello se conoce como *Problema del Árbol de Steiner*.

El GSP pertenece a la clase de problemas NP difíciles (Kahn y Crescenzi, 2003). El propio Problema del Árbol de Steiner, que plantea las restricciones más simples, es NP-completo (Karp, 1972; Garey y Johnson, 1979). La complejidad de los problemas de Steiner hace difícil su resolución mediante algoritmos exactos al aumentar el tamaño del problema. Por este motivo, se buscan alternativas utilizando heurísticas que permitan abordar instancias complejas y encontrar buenas soluciones en tiempos razonables.

Si bien las técnicas evolutivas han sido usadas para abordar variantes simples del problema de Steiner, fuera de nuestro entorno de trabajo no existen antecedentes de aplicación al problema generalizado. El problema GSP ha sido abordado utilizando algoritmos evolutivos en recientes trabajos propios (Árraga et al, 2003), (Nesmachnow et al., 2003), siempre considerando la versión con un único objetivo.

En la versión monoobjetivo del GSP los requerimientos de conexión son "duros", en el sentido de que una solución que no satisfaga los requerimientos se considera como no factible para la resolución del problema. El único objetivo del problema consiste en minimizar los costos de conexión de la red diseñada. En la versión multiobjetivo que se presenta en este trabajo se propone relajar las restricciones de caminos, asignando un valor de confiabilidad a cada red de acuerdo al modelo que se presenta en la subsección siguiente. De este modo se tienen dos objetivos contrapuestos, la minimización del costo de diseño y la maximización de una medida de la confiabilidad de la red.

#### B. Modelo multiobjetivo del problema GSP.

La primer función objetivo evalúa el costo del grafo representado y su formulación se presenta en la Ecuación 6. Esta función corresponde a la tradicional función de costo del problema monoobjetivo, donde la función  $Costo: E \rightarrow R$  devuelve el costo de un enlace de comunicación presente en una solución al GSP.

$$f_1 = \sum_{e \in E} Costo(e)$$

*Ecuación 6: Función objetivo que evalúa el costo de una red.*

El segundo objetivo del problema plantea evaluar la confiabilidad de la red de comunicaciones. Dado que la evaluación de una medida exacta de confiabilidad constituye un problema NP-difícil, en este trabajo se propone un modelo simple para aproximar una cota inferior de la confiabilidad. El modelo se basa en calcular las probabilidades de comunicación entre nodos terminales, asignando una probabilidad de falla constante  $p_F$  a cada enlace de comunicación (arista).

Para un camino de largo  $l$  la probabilidad de falla será  $1 - (1 - p_F)^l$  y para un par de nodos terminales  $(i, j)$  entre los que existan  $k(i, j)$  caminos disjuntos, una medida que aproxima a la *probabilidad de comunicación* estará dada por la expresión que se presenta en la Ecuación 7.

$$p_{COM}(i, j) = \begin{cases} 1 & \text{si } k(i, j) \geq r_{i, j} \\ 1 - \prod_{h=1}^{k(i, j)} (1 - (1 - p_F)^{l(c_h)}) & \text{si } 0 < k(i, j) < r_{i, j} \\ 0 & \text{si } k(i, j) = 0 \end{cases}$$

*Ecuación 7: Probabilidad de comunicación entre nodos terminales.*

En la Ecuación 7 se discriminan tres casos para determinar la probabilidad de comunicación. El primer caso asigna probabilidad 1 en caso de que se cumpla con la restricción de caminos planteada para los nodos terminales  $(i, j)$ . Se considera este caso como "modelo ideal", en el sentido que si se cumplen los requisitos, la probabilidad de falla puede despreciarse. Se decidió asignar el mismo valor unitario a todas aquellas soluciones que satisfagan (exactamente o superando) el número de caminos planteados en la formulación del problema. El segundo caso contempla la existencia de un número menor de caminos que los especificados como requisito, siempre que al menos exista uno. La probabilidad de comunicación considera el largo de estos caminos, denotado por  $c_h$  y sus probabilidades de falla. Por último, se asigna valor de probabilidad de comunicación nulo en caso de no existir al menos un camino entre los nodos terminales  $(i, j)$ .

El segundo objetivo del problema corresponde a maximizar la función que estima la confiabilidad de la red. Una cota inferior de la confiabilidad queda determinada por el producto de las probabilidades de comunicación para cada par de nodos terminales, de acuerdo a la expresión presentada en la Ecuación 8.

$$f_2 = \prod_{(i, j) \in T} p_{COM}(i, j)$$

*Ecuación 8: Función objetivo que estima una cota inferior para la confiabilidad de una red.*

Como las probabilidades de comunicación entre nodos terminales no son independientes, la expresión de la Ecuación 8 no proporciona un valor exacto de la probabilidad que todos los pares de nodos terminales se comuniquen. Como la dependencia que existe es de covarianza positiva (ya que de existir comunicación entre un par  $(i,j)$  es más probable que exista también entre otro par  $(k,l)$ ), el producto de las probabilidades es menor o igual a la probabilidad conjunta y por ello la expresión de la Ecuación 8 corresponde a una cota inferior de la probabilidad de comunicación entre pares de nodos terminales. Aplicada sobre una solución en la cual no exista al menos un camino entre algún par de nodos terminales, la función objetivo definida asignará confiabilidad nula al diseño de la red propuesto.

Para transformar el problema en una doble minimización, en la resolución del problema multiobjetivo se trabaja con un valor de *no-confiabilidad*  $1 - f_2$ .

Cabe mencionar que el hecho de que la probabilidad de falla de arista se asume constante ha sido solamente para simplificar el modelo y posibilitar el uso de instancias de prueba ya diseñadas sin necesidad de introducir una matriz de probabilidad de fallas de enlaces. Esta hipótesis no implica ninguna limitación del modelo propuesto, que puede extenderse para contemplar probabilidades variables para fallas en los enlaces.

### C. Codificación del problema

Se utilizó una codificación binaria simple basada en aristas para representar grafos que constituyen soluciones de la versión multiobjetivo del problema GSP. Una solución se representa como un arreglo de bits (indexado entre 0 y  $|E|-1$ ); cada bit en la representación indica la presencia o ausencia de una arista existente en el grafo original en la solución codificada. La Figura 15 presenta un grafo de ejemplo y su codificación en la representación binaria propuesta, donde las aristas presentes en el grafo solución se han marcado con líneas llenas y las aristas no presentes con líneas punteadas.

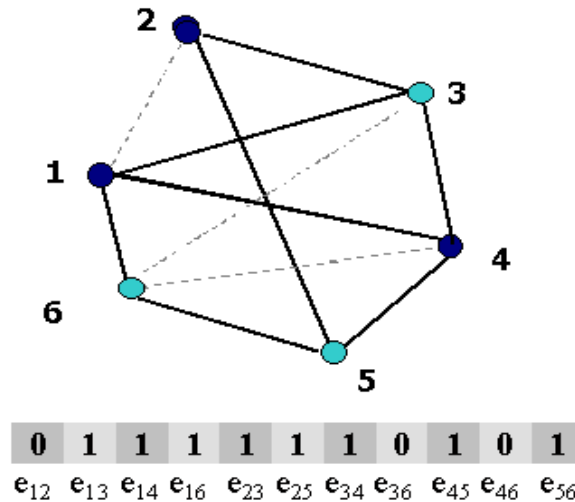


Figura 15: Codificación binaria basada en aristas.

Utilizando esta representación binaria es posible utilizar los operadores de cruzamiento y mutación simples sin necesidad de modificarlos para adaptarlos al problema en cuestión.

La estimación de la confiabilidad requiere el cálculo de caminos entre pares de nodos terminales. Se ha utilizado una variante del algoritmo de Ford y Fulkerson (1962) para hallar los caminos entre cada par de nodos terminales, considerando uno como fuente y el otro como pozo. Asumiendo la capacidad de las aristas unitaria, el flujo máximo entre fuente y pozo coincide con el número máximo de caminos disjuntos entre los nodos terminales cuando se considera cada enlace como un par de aristas de sentidos opuestos. Este valor es utilizado para estimar la confiabilidad de acuerdo al modelo presentado en la sección precedente.

El procedimiento de cálculo de caminos descrito anteriormente tiene un elevado costo computacional, que afecta el desempeño del algoritmo cuando se abordan problemas de grandes dimensiones. Es en este contexto donde se prevé que el modelo paralelo mostrará su ventaja mejorando la eficiencia del algoritmo serial.

#### D. Problemas de prueba

Dado la escasa literatura existente sobre la aplicación de técnicas heurísticas para la resolución del GSP, no existen conjuntos de problemas de prueba estándares.

Para evaluar los algoritmos se utilizaron las tres instancias presentadas en Nesmachnow et al. (2003), diseñadas seleccionando aleatoriamente topologías de conexión, utilizando costos proporcionales a las distancias euclidianas entre nodos y seleccionando los requerimientos de conexión de modo uniforme entre 1 y 4. La Tabla 12 resume las características de los grafos de prueba, indicando el número total de nodos, de terminales, de aristas y el grado de conectividad promedio (definido como el cociente entre el número de aristas del grafo diseñado y el número de aristas del grafo completo).

	<i>Nodos</i>	<i>Terminales</i>	<i>Aristas</i>	<i>Grado Conectividad Promedio</i>
grafo 100-10	100	10	500	0.1
grafo 75-25	75	25	360	0.13
grafo 50-15	50	15	249	0.2

Tabla 12: Características de los grafos de prueba para el GSP.

Los problemas diseñados pueden considerarse “representativos” para redes de comunicaciones de mediano tamaño con requisitos de conexión variables. Los grafos se designan por un nombre que define la cantidad total de nodos y la de terminales. Por ejemplo *grafo 50-15* es un grafo que tiene 50 nodos de los cuales 15 son terminales. Los problemas de prueba y el generador aleatorio de grafos utilizado se encuentran disponibles públicamente en <http://www.fing.edu.uy/inco/grupos/cecal/hpc/gsp>.

Se dispone de los resultados para la versión monoobjetivo del problema (Nesmachnow et al. 2003), que intentaremos utilizar como base para evaluar la calidad de los resultados obtenidos para el problema multiobjetivo, comparando los valores de costo para las soluciones de confiabilidad unitaria.

#### E. Configuración de parámetros

Los parámetros utilizados en la resolución de la versión multiobjetivo del GSP se presentan en la Tabla 13. Se decidió utilizar valores bajos de tamaño de población (100 individuos) y de generaciones como criterio de parada (200 generaciones), tomando en cuenta los tiempos de ejecución requeridos como consecuencia del cómputo de la función que evalúa la medida de confiabilidad de la red.

<i>Parámetro</i>	<i>Valor</i>
Probabilidad de cruzamiento	0.9
Probabilidad de mutación	0.01
Número de generaciones utilizadas como criterio de parada	200 generaciones
Tamaño de la población	100 individuos
<i>Versión Paralela</i>	
Número de subpoblaciones utilizadas	4
Frecuencia de migración	20 generaciones
Individuos participantes en la migración	5 individuos
Topología de migración	Anillo unidireccional
Número de generaciones de interacción panmictica	10 generaciones

Tabla 13: Parámetros utilizados en la resolución del GSP.

## F. Resultados

La Tabla 14 presenta los resultados obtenidos para las tres instancias del problema GSP estudiadas. En ella se presentan los tiempos de ejecución promedio (en segundos), el número de puntos de la población final y el número de puntos no dominados obtenidos por las versiones serial y paralela del algoritmo NSGA-II en las 30 ejecuciones independientes realizadas. No se calcularon valores de distancia generacional ni para las métricas de diversidad al no conocerse las características de los frentes de Pareto del problema.

En el Anexo II se presentan la totalidad de los resultados obtenidos en cada una de las ejecuciones realizadas para cada instancia abordada del problema de Steiner generalizado.

Problema	Modelo	Medida	Tiempo (seg.)	Puntos	Puntos ND
grafo 50_15	Serial	Promedio	149.73	200.0	15.5
		Desv. Est.	0.84	0.00	4.29
	Paralelo	Promedio	69.53	52.0	16.0
		Desv. Est.	1.08	7.19	3.46
grafo 75_25	Serial	Promedio	643.68	200.0	15.5
		Desv. Est.	7.29	0.00	3.30
	Paralelo	Promedio	180.10	50.0	15.5
		Desv. Est.	2.97	7.35	3.02
grafo 100_10	Serial	Promedio	144.94	200.0	10.5
		Desv. Est.	0.99	0.00	2.68
	Paralelo	Promedio	75.15	43.0	11.5
		Desv. Est.	0.90	7.84	2.65

Tabla 14: Resultados para la versión multiobjetivo del GSP.

En la Tabla 14 se aprecia que el algoritmo paralelo alcanza resultados de similar calidad que los obtenidos por el algoritmo serial, pese a que maneja un número de individuos cuatro veces menor en la población final.

La evaluación de la eficiencia computacional para las tres instancias del problema abordadas se presenta en la Tabla 15, ofreciendo los tiempos de ejecución (en segundos) de las versiones serial y paralela, y los valores de speedup y eficiencia obtenidos en cada caso. Puede constatar que el algoritmo paralelo muestra un comportamiento de speedup sublineal, con valores de eficiencia no muy altos, pero que crecen al aumentar la complejidad de la instancia considerada (la complejidad de la versión multiobjetivo del GSP presentada está dada por el número de nodos terminales, ya que la complejidad del algoritmo utilizado para hallar los caminos entre pares de terminales es de orden cuadrático en el número de nodos terminales). Para la instancia del problema especificada por el *grafo 75-25* el tiempo promedio de ejecución se reduce de casi 11 minutos en la versión secuencial a 3 minutos en la versión paralela.

	Medida (segundos)	grafo 50_15	grafo 75_25	grafo 100_10
Modelo	Tiempo Serial	149.73	643.68	144.94
	Tiempo Paralelo	69.53	180.10	75.15
Medida	Speedup	2.15	3.57	1.93
	Eficiencia	0.54	0.89	0.48

Tabla 15: Análisis de desempeño en la resolución de la versión multiobjetivo del GSP.

Las Figuras 16, 17, y 18 presentan gráficamente los puntos no dominados para tres ejecuciones representativas de las versiones serial y paralela del algoritmo NSGA-II sobre cada una de las instancias consideradas. Puede apreciarse una cierta forma característica de los frentes de individuos no dominados obtenidos por los algoritmos, que es bastante similar en las tres instancias que se resolvieron.



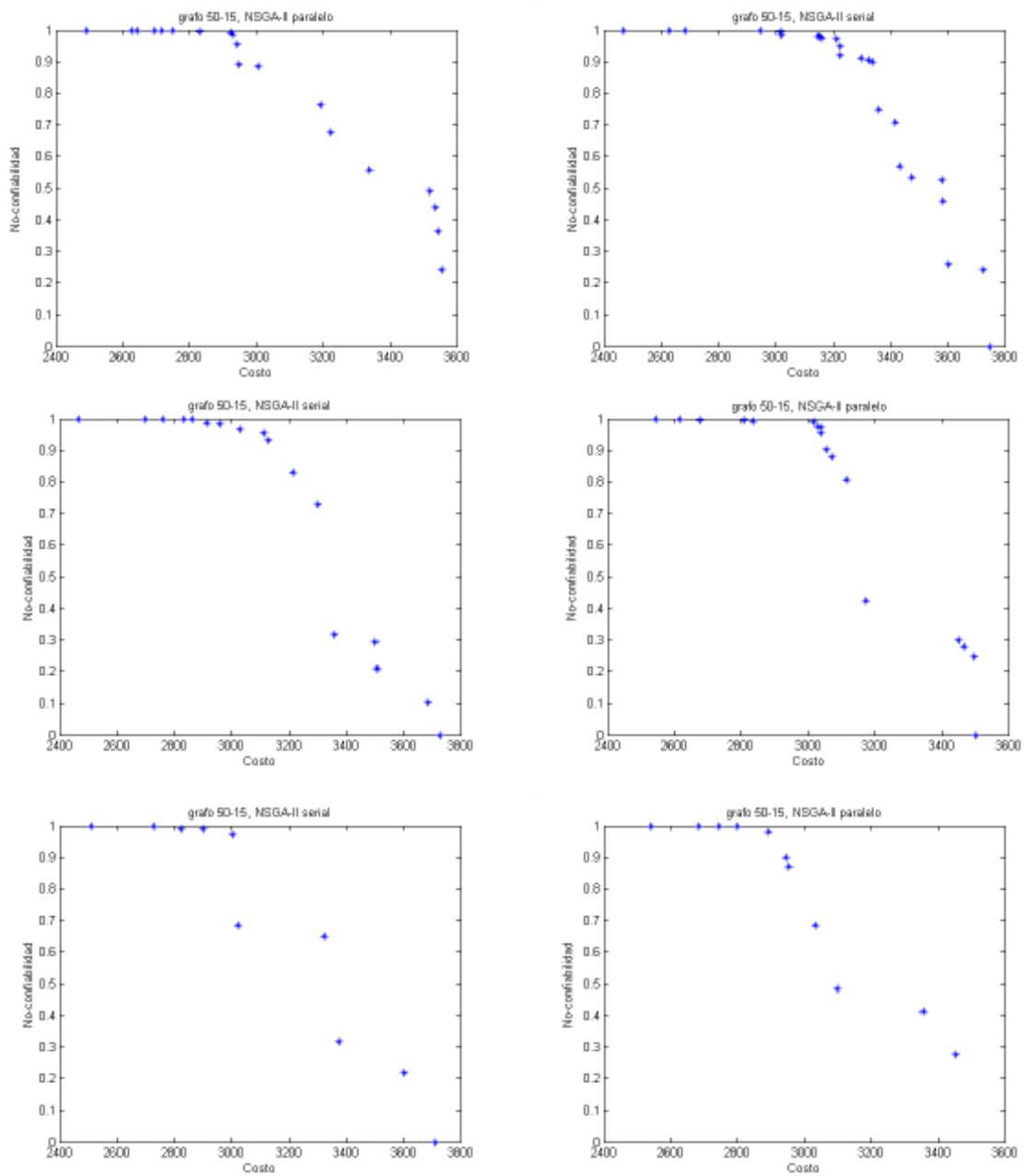


Figura 16: Frentes de Pareto obtenidos para el problema 50-15.

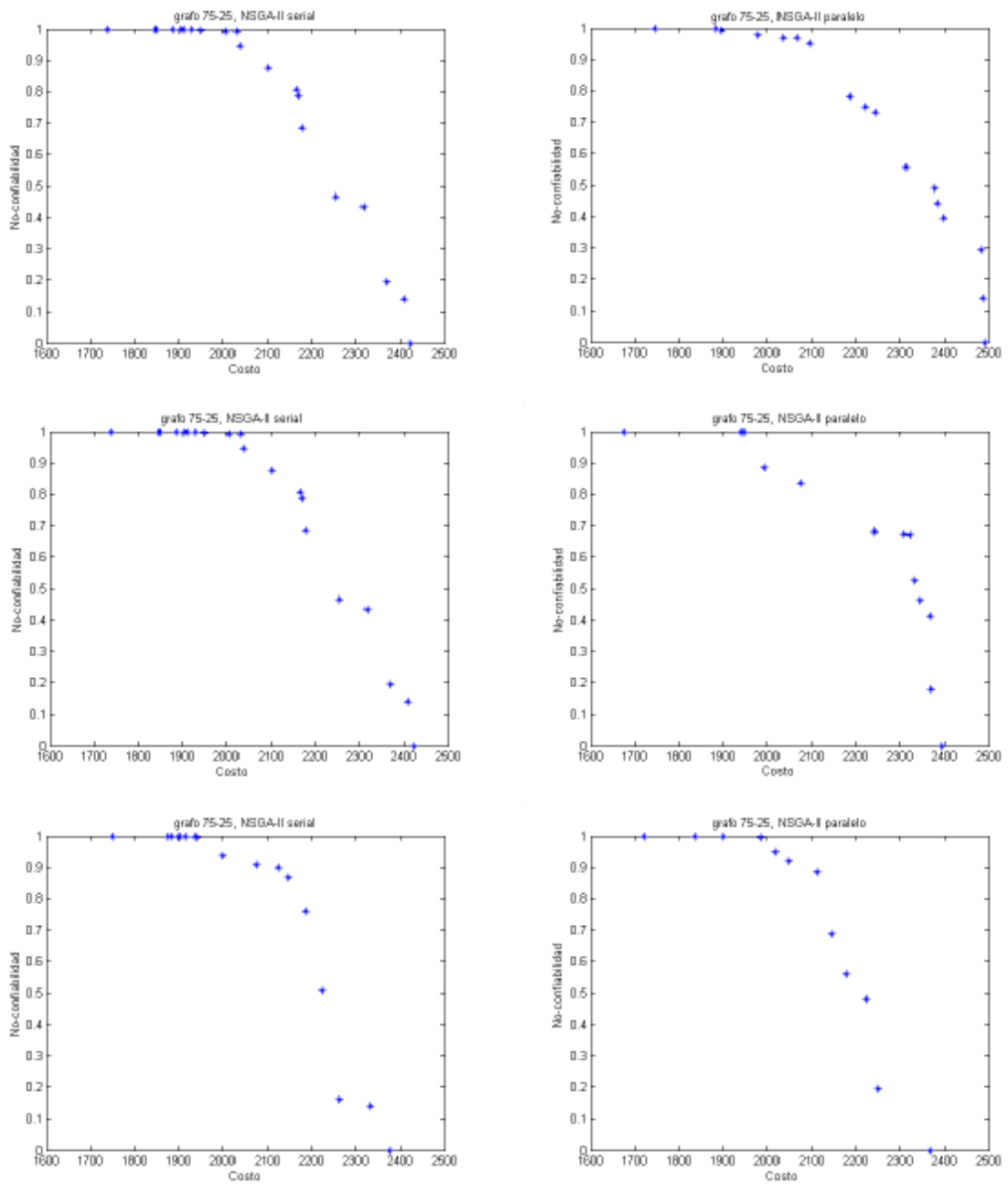


Figura 17: Frentes de Pareto obtenidos para el problema 75-25.

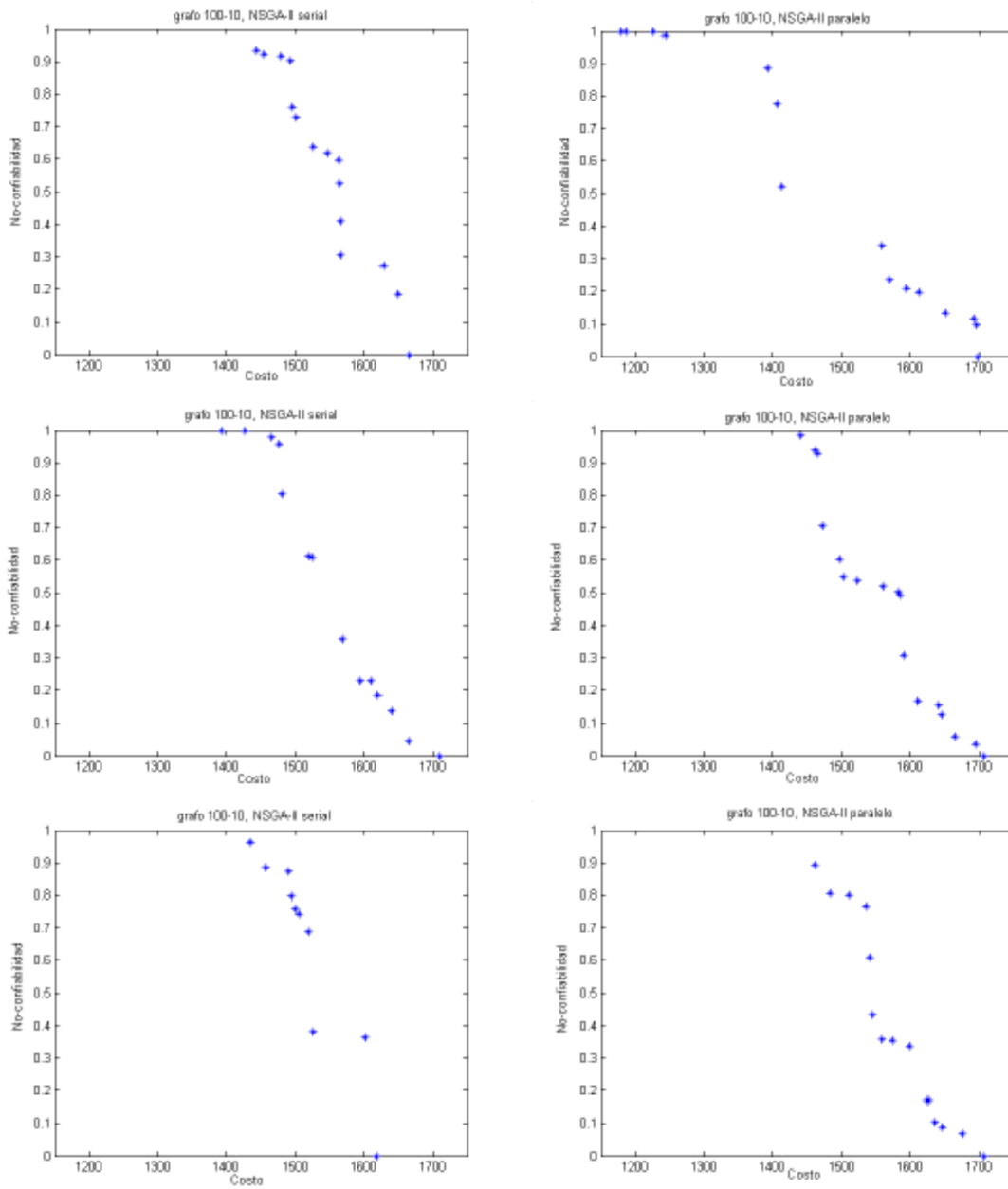


Figura 18: Frentes de Pareto obtenidos para el problema 100-10.

Es posible establecer algunas conclusiones respecto a la calidad de los resultados obtenidos, si se comparan con los resultados de la versión monoobjetivo del problema, correspondientes a diseños de confiabilidad unitaria. La Tabla 16 presenta el análisis comparativo de un subconjunto de resultados a los cuales hemos denominado diseños "*altamente confiables*" (correspondientes a diseños con valores estimados de confiabilidad mayor a 0.8) hallados en la versión multiobjetivo con los resultados de confiabilidad unitaria del modelo monoobjetivo propuesto en Nesmachnow et al (2003). Analizando la Tabla 16, puede observarse que los resultados del modelo multiobjetivo tienen mejores valores de costo que los resultados obtenidos por el algoritmo genético monoobjetivo en 200 generaciones para los tres problemas considerados.

<i>Modelo</i>	Multiobjetivo (confiabilidad > 0.8)	Monoobjetivo ( confiabilidad = 1 )
<i>grafo 50_15</i>	3800	4600
<i>grafo 75_25</i>	2500	3000
<i>grafo 100_10</i>	1700	1900

Tabla 16: Análisis de valores de costo para soluciones "*altamente confiables*" del GSP.

Sin embargo, experimentos realizados modificando el criterio de parada del algoritmo NSGA-II paralelo permitieron comprobar que los resultados de la versión multiobjetivo no mejoran sustancialmente al incrementar el número de generaciones, mostrándose el algoritmo incapaz de lograr una aproximación precisa del frente de Pareto del problema (impidiendo alcanzar los valores de costo mínimo para confiabilidad cercana al valor unitario). Este comportamiento es diferente al del algoritmo genético para el modelo monoobjetivo, que es capaz de mejorar sus resultados al continuar la evolución, superando notoriamente los resultados obtenidos para el modelo multiobjetivo a partir de las 500 generaciones en promedio. Aún permitiendo una evolución prolongada del algoritmo NSGA-II paralelo –se realizaron experimentos hasta un número de 1000 generaciones– los resultados obtenidos "*a largo plazo*" por el problema multiobjetivo son sensiblemente inferiores a los obtenidos para la versión monobjetivo.

La comparación de resultados al utilizar un número elevado de generaciones como criterio de parada indica que aún resta trabajo por realizar para ajustar adecuadamente el algoritmo, comprender las causas del fenómeno de convergencia prematura que impide una aproximación precisa del frente de Pareto del problema y de este modo mejorar la calidad de los resultados obtenidos para el modelo multiobjetivo del problema.

## VIII. CONCLUSIONES Y TRABAJO FUTURO

Este trabajo ha presentado una versión paralela del algoritmo evolutivo para optimización multiobjetivo NSGA-II y su aplicación a un problema de diseño de redes de comunicaciones confiables. La propuesta se basó en aplicar un paradigma de diseño sencillo para el algoritmo paralelo, tratando de modificar lo menos posible la estructura algorítmica del NSGA-II. Se escogió para el diseño del algoritmo paralelo el modelo de subpoblaciones con migración, tratando de sacar ventajas de infraestructuras disponibles para la resolución de complejos problemas de optimización que requirieran un considerable esfuerzo computacional.

La versión paralela fue evaluada sobre un total de 14 problemas estándar de prueba en el área de los MOEA y se cotejó su calidad de resultados y su eficiencia computacional con los obtenidos al aplicar la versión serial del algoritmo. Complementariamente, se utilizó el algoritmo diseñado para resolver una versión multiobjetivo del problema de Steiner generalizado, un problema que modela el diseño de redes de comunicaciones confiables.

Al trabajar sobre los problemas de prueba, la versión paralela del algoritmo NSGA-II diseñada mostró muy buenos resultados de eficiencia computacional. Se alcanzaron valores de speedup superlineal cuando se abordaron problemas con poblaciones numerosas y/o con funciones objetivo con elevados requerimientos de cómputo. Respecto a la calidad de resultados obtenidos, solamente se detectaron mejoras significativas respecto a los resultados del algoritmo serial sobre tres de los problemas de prueba, en especial al utilizar poblaciones de tamaño mediano y un número no elevado de generaciones como criterio de parada.

En la aplicación a la versión multiobjetivo del problema de Steiner generalizado se obtuvieron resultados de calidad competitiva al comparar los costos de los mejores individuos hallados con los resultados de la versión multiobjetivo del problema. Como contrapartida, se comprobó que los resultados no mejoran significativamente al aumentar el número de generaciones.

Dos líneas de trabajo se plantean para abordar en el futuro. La primera de ellas referente al diseño del algoritmo paralelo y la segunda en lo que refiere a la aplicación al problema de diseño de redes de comunicaciones presentado.

El modelo de paralelismo propuesto en este trabajo es muy sencillo, ya que la versión paralela fue diseñada con la idea de restringir al máximo las modificaciones del código del algoritmo NSGA-II. En este sentido se plantean como trabajo futuro modificaciones al diseño que permitan a la versión paralela abordar problemas "difíciles" tomando partido del paralelismo. Algunas de las líneas a considerar incluyen la división del espacio de búsqueda de modo de que diferentes islas puedan concentrarse en hallar soluciones en diferentes regiones y la posible incorporación de heterogeneidad al modelo, permitiendo que las islas evolucionen de acuerdo a diferentes operadores y/o configuraciones de parámetros.

El análisis de políticas de migración y reemplazo en los resultados obtenidos al resolver problemas de optimización complejos y profundizarse el estudio de la influencia de la estrategia de centralización aplicada luego de recibir los resultados de las diferentes islas. El breve análisis preliminar realizado permitió comprobar que un número bajo de interacciones panmícticas ayuda al algoritmo a mejorar sensiblemente sus resultados, pero el análisis empírico no ha sido adecuadamente formalizado.

Por otra parte, debe investigarse la causa por la cual al resolver la versión multiobjetivo del problema de Steiner generalizado los resultados no mejoran significativamente al aumentar el número de generaciones. Un detallado estudio de los operadores y sus parámetros se hace necesario para comprender este fenómeno de estancamiento prematuro y de ese modo estar en condiciones de proponer alternativas para mejorar la calidad de los resultados obtenidos.

## REFERENCIAS

- Alba E. (2002), *Parallel Evolutionary Algorithms Can Achieve Super-Linear Performance*, Information Processing Letters, Elsevier, 82(1):7-13, abril 2002
- Alba E., Tomassini M. (2002). *Parallelism and Genetic Algorithms*, IEEE Transactions on Evolutionary Computation 6, 5, pp. 443-462.
- Árraga S., Aroztegui M., Nesmachnow S. (2003). *Resolución del Problema de Steiner Generalizado utilizando un Algoritmo Genético Paralelo*, Actas del Segundo Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, pp 387-394, Gijón, España, 2003.
- Binh T., Korn U. (1997), *Multiobjective Evolution Strategy for Constrained Optimization Problems*, Proceedings of the IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, pp. 357-362, Berlin, Alemania.
- Cantú-Paz E. (2001). *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publisher.
- Coello C., Van Veldhuizen D., Lamont G. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publisher.
- Davis L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.
- Deb K. (1999), *Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems*. Evolutionary Computation 7(3): 205-230, 1999.
- Deb K., Agrawal S., Pratab A., Meyarivan T. (2000). *A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II*, Proceedings of the Parallel Problem Solving from Nature VI Conference, pages 849-858. Springer, 2000
- Deb K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, NY.
- Deb K., Zope P., Jain A. (2002). *Distributed Computing of Pareto-Optimal Solutions Using Multi-Objective Evolutionary Algorithms*. Report No. 2002008, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, Setiembre 2002. Disponible en <http://www.iitk.ac.in/kangal/pub.htm>.
- Ford L., Fulkerson D. (1962). *Flows in Networks*. Princeton University Press, Princeton.
- Garey M., Johnson D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.
- Goldberg D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
- Kahn V., Crescenzi P. (2003). *A compendium of NP optimization problems*, Disponible en línea <http://www.nada.kth.se/theory/problemlist.html>. Consultada diciembre 2003.
- Karp R. (1972). *Reducibility among combinatorial problems*. Complexity of Computer Communications, pp. 85-103, Plenum Press.
- Kursawe F. (1991). *A variant of evolution strategies for vector optimization*, Parallel Problem Solving from Nature. PPSN I, volume 496 of Lecture Notes in Computer Science, pp. 193-197, Berlin, Springer-Verlag.
- Mäkinen R., Neittaanmäki P., Periaux J., Sefrioui M., Toivanen J (1995). *Parallel genetic solution for multiobjective MDO*. Parallel CFD'96 Conference, pp. 352-359, Capri, 1996. Elsevier.
- Marco N., Lanteri S., Desideri J., Périaux J. (1999), *A Parallel Genetic Algorithm for Multi-Objective Optimization in Computational Fluid Dynamics*, Evolutionary Algorithms in Engineering and Computer Science, capítulo 22, pp. 445-456. John Wiley & Sons Ltd, Chichester, UK, 1999.
- MPI Forum (2003), *MPI (Message Passing Interface) Forum Home Page*. Disponible en línea <http://www.mpi-forum.org/>. Consultada diciembre 2003.
- Nebro A., Alba E., Luna F. (2003), Optimización Multiobjetivo y Computación Grid, A aparecer en Actas del Tercer Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, Córdoba, España.

- Nesmachnow S., Cancela H., Alba E. (2003). *Técnicas Evolutivas Aplicadas al Diseño de Redes de Comunicaciones Confiables*. A aparecer en Actas del Tercer Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados Córdoba, España.
- Schaffer D. (1984). *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University.
- Schott J. (1995). *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995.
- Srinivas N., Deb K. (1994), *Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms*. Evolutionary Computation 2(3): 221-248, 1994.
- Van Veldhuizen D., Lamont G. (1999), *Multiobjective Evolutionary Algorithm Test Suites*, Proceedings of the 1999 ACM Symposium on Applied Computing, pp. 351-357, San Antonio, Texas, 1999. ACM.
- Van Veldhuizen D., Zydallis J., Lamont G. (2003), *Considerations in engineering parallel multiobjective evolutionary algorithms*. IEEE Trans. Evolutionary Computation 7(2): 144-173, 2003.
- Viennet, R., Fontiex, C., Marc, I. (1996), *Multicriteria Optimization Using a Genetic Algorithm for Determining a Pareto Set*, Journal of Systems Science 27(2): 255-260, 1996.
- Zitzler E., Deb K., Thiele L. (2000), *Comparison of Multiobjective Evolutionary Algorithms: Empirical Results*. Evolutionary Computation 8(2): 173-195, 2000.

## Anexo I : Tablas de Resultados

Este anexo reúne los resultados obtenidos para cada una de las 30 ejecuciones independientes de las versiones serial y paralela del algoritmo NSGA-II realizadas para cada problema de prueba.

### Problema SCH1

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Spread	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	23.59	381	0.0001	0.4097	0.123	1	3.90	369	0.0001	0.3767	0.124
2	22.11	380	0.0001	0.3708	0.140	2	3.54	380	0.0001	0.3982	0.121
3	22.11	377	0.0001	0.4117	0.122	3	3.46	366	0.0001	0.3824	0.125
4	21.94	383	0.0001	0.4115	0.123	4	3.45	372	0.0001	0.4142	0.120
5	21.95	378	0.0001	0.3823	0.140	5	4.37	368	0.0001	0.3856	0.123
6	21.94	367	0.0001	0.3753	0.128	6	3.39	377	0.0001	0.3749	0.123
7	22.03	386	0.0001	0.3995	0.136	7	3.66	374	0.0001	0.3785	0.123
8	22.04	370	0.0001	0.3709	0.142	8	3.47	374	0.0001	0.4017	0.121
9	21.92	381	0.0001	0.4073	0.123	9	3.57	382	0.0001	0.3961	0.121
10	21.97	376	0.0001	0.3791	0.140	10	3.55	381	0.0001	0.4065	0.119
11	21.96	386	0.0001	0.3995	0.136	11	3.36	375	0.0001	0.3757	0.124
12	21.98	378	0.0001	0.4101	0.139	12	3.48	378	0.0001	0.4001	0.122
13	21.92	374	0.0001	0.4031	0.139	13	3.54	375	0.0001	0.3851	0.123
14	22.10	371	0.0001	0.3782	0.128	14	3.41	388	0.0001	0.3682	0.122
15	22.10	386	0.0001	0.3995	0.136	15	3.54	378	0.0001	0.3705	0.123
16	21.95	383	0.0001	0.4316	0.121	16	3.42	371	0.0001	0.3803	0.125
17	21.99	379	0.0001	0.3685	0.127	17	3.45	371	0.0001	0.3753	0.125
18	22.01	372	0.0001	0.3885	0.140	18	3.73	375	0.0001	0.3844	0.122
19	21.98	373	0.0001	0.3978	0.126	19	3.60	378	0.0001	0.4152	0.120
20	21.99	369	0.0001	0.3869	0.126	20	3.41	381	0.0001	0.4212	0.119
21	22.05	379	0.0001	0.3929	0.124	21	3.65	376	0.0001	0.3877	0.123
22	21.97	373	0.0001	0.3836	0.127	22	3.50	378	0.0001	0.4131	0.119
23	21.95	371	0.0001	0.3977	0.125	23	3.53	376	0.0001	0.3850	0.123
24	22.09	379	0.0001	0.3877	0.124	24	3.51	379	0.0001	0.3974	0.121
25	22.05	373	0.0001	0.3836	0.127	25	3.86	375	0.0001	0.3831	0.123
26	22.00	372	0.0001	0.3885	0.140	26	3.51	377	0.0001	0.3731	0.123
27	21.94	378	0.0001	0.4129	0.137	27	3.57	382	0.0001	0.4199	0.121
28	22.03	382	0.0001	0.3938	0.138	28	3.61	372	0.0001	0.3876	0.122
29	21.97	375	0.0001	0.4183	0.123	29	3.58	368	0.0001	0.3976	0.123
30	22.07	378	0.0001	0.3871	0.139	30	3.56	365	0.0001	0.3848	0.126
Promedio	21.99	378.0	0.0001	0.3933	0.128	Promedio	3.54	375.5	0.0001	0.3853	0.123
Desv. Est.	0.30	5.23	0.0000	0.0155	0.007	Desv. Est.	0.20	5.22	0.0000	0.0153	0.002



## Problema SCH2

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Spread	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	36.19	382	0.0001	0.759	0.137	1	3.64	377	0.000109	0.789	0.137
2	36.23	374	0.0001	0.775	0.136	2	3.56	366	0.000118	0.773	0.138
3	36.09	370	0.0001	0.759	0.138	3	3.53	366	0.000120	0.757	0.139
4	36.52	375	0.0001	0.779	0.138	4	3.42	372	0.000119	0.775	0.138
5	36.55	382	0.0001	0.786	0.132	5	3.63	368	0.000121	0.787	0.135
6	36.09	373	0.0001	0.770	0.138	6	3.61	371	0.000122	0.777	0.138
7	36.09	380	0.0001	0.760	0.137	7	3.55	367	0.000114	0.779	0.140
8	36.11	377	0.0001	0.769	0.135	8	3.54	370	0.000122	0.782	0.140
9	36.28	372	0.0001	0.773	0.138	9	3.54	378	0.000126	0.776	0.136
10	36.30	379	0.0001	0.764	0.136	10	3.51	372	0.000126	0.789	0.134
11	36.07	380	0.0001	0.760	0.137	11	3.62	373	0.000119	0.763	0.136
12	36.21	371	0.0001	0.793	0.135	12	3.47	364	0.000119	0.775	0.141
13	36.04	378	0.0001	0.759	0.138	13	3.45	370	0.000128	0.780	0.134
14	36.10	371	0.0001	0.763	0.138	14	3.37	374	0.000125	0.769	0.137
15	36.06	380	0.0001	0.760	0.137	15	3.47	373	0.000127	0.758	0.138
16	36.02	379	0.0001	0.772	0.136	16	3.38	368	0.000110	0.777	0.138
17	36.23	383	0.0001	0.774	0.137	17	3.56	370	0.000119	0.770	0.139
18	36.25	380	0.0001	0.758	0.137	18	3.48	374	0.000120	0.775	0.136
19	36.07	383	0.0001	0.779	0.131	19	3.58	370	0.000119	0.788	0.136
20	36.06	374	0.0001	0.778	0.136	20	3.49	365	0.000120	0.776	0.138
21	36.35	378	0.0001	0.784	0.136	21	3.68	370	0.000129	0.753	0.141
22	36.07	383	0.0001	0.762	0.135	22	3.49	365	0.000125	0.775	0.140
23	36.09	377	0.0001	0.763	0.136	23	3.59	363	0.000119	0.785	0.139
24	36.09	380	0.0001	0.775	0.136	24	3.47	360	0.000121	0.785	0.141
25	36.07	383	0.0001	0.762	0.135	25	3.43	369	0.000122	0.756	0.138
26	36.28	380	0.0001	0.758	0.137	26	3.49	375	0.000116	0.778	0.138
27	36.21	375	0.0001	0.758	0.138	27	3.43	363	0.000129	0.760	0.141
28	36.09	383	0.0001	0.771	0.136	28	3.59	370	0.000125	0.777	0.137
29	36.31	370	0.0001	0.776	0.138	29	3.52	364	0.000116	0.774	0.137
30	36.05	385	0.0001	0.739	0.137	30	3.43	362	0.000116	0.750	0.144
Prom	36.10	379.0	0.0001	0.767	0.137	Prom	3.52	370.0	0.0001	0.776	0.138
Desv. Est.	0.14	4.44	0.0000	0.011	0.002	Desv. Est.	0.08	4.54	0.0000	0.011	0.002

Problema ZDT1

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	21.47	365	0.000006	0.428	0.0604	1	5.17	362	0.00001	0.407	0.0622
2	21.66	369	0.000008	0.394	0.0605	2	5.53	365	0.000006	0.406	0.0620
3	21.66	368	0.000006	0.383	0.0618	3	5.50	375	0.000007	0.393	0.0617
4	21.68	383	0.000009	0.405	0.0601	4	5.25	370	0.00001	0.396	0.0615
5	21.66	368	0.000006	0.400	0.0620	5	5.35	367	0.000006	0.389	0.0614
6	21.64	366	0.000007	0.406	0.0614	6	5.44	374	0.000006	0.416	0.0599
7	21.68	362	0.000007	0.413	0.0610	7	5.49	368	0.000009	0.439	0.0596
8	21.63	361	0.000008	0.399	0.0620	8	5.41	377	0.000009	0.422	0.0592
9	21.69	364	0.000009	0.380	0.0624	9	5.32	381	0.000016	0.394	0.0602
10	21.68	367	0.000011	0.421	0.0608	10	5.30	357	0.000008	0.415	0.0617
11	21.64	362	0.000007	0.413	0.0610	11	5.31	371	0.000011	0.399	0.0608
12	21.63	373	0.000006	0.424	0.0604	12	5.36	371	0.000007	0.387	0.0610
13	23.36	370	0.000006	0.406	0.0614	13	5.40	360	0.000007	0.393	0.0625
14	21.28	374	0.000007	0.406	0.0604	14	5.80	370	0.000008	0.410	0.0607
15	21.21	362	0.000007	0.413	0.0610	15	5.70	362	0.000006	0.395	0.0622
16	21.49	366	0.000011	0.399	0.0617	16	5.30	374	0.000009	0.426	0.0599
17	21.63	361	0.000010	0.399	0.0626	17	5.16	374	0.00001	0.403	0.0603
18	21.36	370	0.000006	0.406	0.0613	18	5.08	368	0.000007	0.416	0.0606
19	21.20	368	0.000016	0.395	0.0607	19	5.56	359	0.000006	0.382	0.0634
20	21.46	376	0.000029	0.422	0.0604	20	5.29	366	0.000009	0.414	0.0605
21	24.90	370	0.000006	0.409	0.0606	21	5.58	366	0.000009	0.422	0.0609
22	25.45	371	0.000007	0.393	0.0614	22	5.58	365	0.000007	0.415	0.0612
23	25.19	358	0.000006	0.436	0.0611	23	5.34	366	0.000009	0.422	0.0610
24	25.20	365	0.000010	0.417	0.0607	24	5.59	369	0.000011	0.418	0.0604
25	24.94	371	0.000007	0.393	0.0614	25	5.66	354	0.000007	0.412	0.0626
26	25.41	370	0.000006	0.406	0.0613	26	5.69	359	0.000006	0.398	0.0621
27	25.12	373	0.000008	0.416	0.0601	27	5.64	364	0.000007	0.391	0.0615
28	25.18	363	0.000009	0.397	0.0619	28	5.50	371	0.00001	0.386	0.0616
29	25.01	372	0.000008	0.398	0.0611	29	5.43	364	0.000007	0.395	0.0620
30	25.36	361	0.000006	0.426	0.0613	30	5.43	365	0.00001	0.407	0.0611
Promedio	21.67	368.0	0.000007	0.406	0.0611	Promedio	5.43	366.5	0.000008	0.406	0.0611
Desv. Est.	1.74	5.4	0.000004	0.013	0.0007	Desv. Est.	0.17	6.2	2.17E-06	0.014	0.0010

Problema ZDT2

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	26.02	365	0.000004	0.398	0.062	1	5.77	373	6E-06	0.397	0.061
2	25.71	374	0.000004	0.394	0.061	2	5.34	367	5E-06	0.419	0.061
3	25.78	373	0.000006	0.382	0.061	3	5.49	361	7E-06	0.389	0.062
4	26.04	367	0.000004	0.412	0.061	4	5.40	365	9E-06	0.393	0.062
5	25.89	372	0.000004	0.394	0.061	5	5.87	370	4E-06	0.413	0.061
6	25.95	378	0.000004	0.418	0.060	6	5.48	370	4E-06	0.398	0.061
7	25.87	360	0.000004	0.421	0.061	7	5.54	370	4E-06	0.369	0.062
8	26.06	361	0.000004	0.384	0.063	8	5.50	363	4E-06	0.410	0.062
9	26.43	353	0.000004	0.396	0.063	9	5.61	354	4E-06	0.409	0.062
10	26.05	367	0.000004	0.384	0.062	10	5.83	357	4E-06	0.405	0.062
11	26.23	360	0.000004	0.421	0.061	11	5.50	368	4E-06	0.385	0.062
12	26.24	372	0.000006	0.403	0.061	12	5.68	359	4E-06	0.412	0.062
13	26.61	364	0.000004	0.405	0.061	13	5.48	373	4E-06	0.405	0.061
14	25.97	362	0.000004	0.387	0.062	14	5.66	362	4E-06	0.443	0.061
15	26.35	360	0.000004	0.421	0.061	15	5.75	375	4E-06	0.433	0.060
16	25.98	360	0.000004	0.418	0.062	16	5.74	371	4E-06	0.424	0.060
17	26.18	375	0.000004	0.400	0.061	17	5.65	359	1.2E-05	0.384	0.062
18	25.95	367	0.000008	0.383	0.062	18	5.56	368	4E-06	0.395	0.061
19	25.95	372	0.000004	0.404	0.061	19	5.50	362	4E-06	0.400	0.062
20	26.24	370	0.000004	0.405	0.061	20	5.68	362	4E-06	0.389	0.062
21	26.66	367	0.000004	0.404	0.062	21	5.61	368	4E-06	0.433	0.060
22	26.30	366	0.000004	0.398	0.062	22	5.60	361	4E-06	0.395	0.063
23	26.05	363	0.000004	0.402	0.061	23	5.48	364	5E-06	0.360	0.063
24	26.19	366	0.000004	0.398	0.062	24	5.69	373	4E-06	0.422	0.060
25	25.89	366	0.000004	0.398	0.062	25	5.38	366	4E-06	0.385	0.062
26	25.94	367	0.000008	0.383	0.062	26	5.42	368	4E-06	0.392	0.061
27	24.15	372	0.000004	0.387	0.062	27	5.61	364	4E-06	0.388	0.061
28	24.79	366	0.000004	0.415	0.060	28	5.69	364	4E-06	0.379	0.062
29	25.00	363	0.000004	0.429	0.061	29	5.14	364	4E-06	0.391	0.062
30	25.24	367	0.000004	0.421	0.060	30	5.47	373	4E-06	0.430	0.059
Promedio	26.00	366.5	0.000004	0.401	0.061	Promedio	5.58	365.5	4E-06	0.398	0.062
Desv. Est.	0.52	5.49	1.1E-06	0.014	0.001	Desv. Est.	0.16	5.29	1.8E-06	0.020	0.001

### Problema ZDT3

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	34.42	367	0.000024	0.413	0.0625	1	6.02	362	0.000024	0.383	0.0637
2	30.92	369	0.000024	0.426	0.0613	2	6.93	365	0.000027	0.389	0.0633
3	30.38	372	0.000023	0.396	0.0620	3	6.94	372	0.000025	0.394	0.0624
4	30.02	367	0.000023	0.391	0.0630	4	6.76	374	0.000026	0.425	0.0614
5	25.80	363	0.000023	0.406	0.0628	5	5.42	368	0.000025	0.394	0.0626
6	26.35	368	0.000024	0.407	0.0622	6	5.59	364	0.000025	0.414	0.0622
7	27.14	365	0.000022	0.397	0.0636	7	4.76	369	0.000029	0.391	0.0627
8	29.27	372	0.000026	0.398	0.0617	8	5.98	374	0.000027	0.416	0.0611
9	27.58	371	0.000023	0.411	0.0629	9	6.87	373	0.000024	0.436	0.0617
10	27.31	361	0.000024	0.382	0.0637	10	7.38	369	0.000024	0.394	0.0632
11	26.73	365	0.000022	0.397	0.0636	11	6.41	369	0.000024	0.388	0.0628
12	32.01	370	0.000026	0.401	0.0624	12	7.18	375	0.000024	0.410	0.0614
13	31.68	365	0.000025	0.412	0.0618	13	6.16	375	0.000023	0.447	0.0607
14	28.41	377	0.000023	0.399	0.0622	14	4.73	370	0.000024	0.402	0.0623
15	26.27	365	0.000029	0.397	0.0636	15	4.73	373	0.000024	0.413	0.0613
16	23.37	360	0.000025	0.387	0.0638	16	7.68	363	0.000025	0.411	0.0622
17	30.95	376	0.000023	0.404	0.0619	17	7.13	369	0.000024	0.394	0.0620
18	29.25	369	0.000023	0.410	0.0621	18	6.83	373	0.000027	0.364	0.0631
19	28.89	368	0.000024	0.403	0.0618	19	7.97	368	0.000023	0.411	0.0626
20	27.18	368	0.000024	0.400	0.0630	20	5.66	368	0.000024	0.424	0.0618
21	27.28	362	0.000023	0.401	0.0637	21	4.44	373	0.000025	0.402	0.0620
22	27.18	371	0.000028	0.428	0.0618	22	7.11	377	0.000025	0.400	0.0618
23	33.05	364	0.000024	0.383	0.0640	23	6.53	370	0.000024	0.422	0.0609
24	26.45	369	0.000025	0.400	0.0622	24	6.91	371	0.000024	0.405	0.0619
25	30.18	371	0.000028	0.428	0.0618	25	6.21	373	0.000024	0.387	0.0633
26	29.24	369	0.000023	0.410	0.0621	26	5.87	369	0.000026	0.392	0.0624
27	29.46	354	0.000024	0.381	0.0650	27	5.54	371	0.000023	0.390	0.0623
28	30.22	367	0.000021	0.379	0.0641	28	5.92	370	0.000024	0.404	0.0621
29	31.64	367	0.000027	0.421	0.0614	29	6.23	377	0.000025	0.402	0.0618
30	29.76	368	0.000025	0.409	0.0614	30	7.21	361	0.000025	0.369	0.0642
Promedio	29.24	368.0	0.000024	0.401	0.0623	Promedio	6.32	370.0	0.000024	0.402	0.0622
Desv. Est.	2.42	4.6	1.9E-06	0.013	0.0010	Desv. Est.	0.92	4.2	1.4E-06	0.018	0.0008

## Problema ZDT4

Serial					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	29.68	390	0.000024	0.363	0.061
2	29.41	390	0.000026	0.371	0.060
3	26.29	391	0.000039	0.378	0.060
4	23.95	385	0.000026	0.361	0.061
5	23.56	379	0.000006	0.382	0.061
6	26.39	389	0.000067	0.372	0.061
7	23.57	388	0.000186	0.376	0.061
8	23.72	389	0.000033	0.365	0.061
9	26.43	388	0.000082	0.376	0.060
10	30.88	387	0.000069	0.372	0.061
11	30.43	388	0.000186	0.376	0.061
12	30.18	386	0.000054	0.358	0.061
13	30.48	390	0.000007	0.374	0.061
14	30.01	378	0.000022	0.348	0.062
15	25.99	388	0.000186	0.376	0.061
16	21.64	390	0.000054	0.359	0.061
17	21.78	391	0.000044	0.369	0.060
18	21.65	391	0.000067	0.343	0.061
19	21.60	394	0.000057	0.368	0.060
20	27.55	394	0.000070	0.366	0.060
21	23.99	390	0.000042	0.396	0.060
22	26.79	392	0.000125	0.368	0.060
23	24.11	389	0.000042	0.365	0.061
24	25.91	393	0.000154	0.355	0.061
25	26.80	392	0.000125	0.368	0.060
26	24.24	391	0.000067	0.343	0.061
27	26.43	395	0.000072	0.381	0.060
28	26.80	390	0.000029	0.364	0.060
29	26.62	389	0.000100	0.357	0.061
30	28.05	389	0.000033	0.355	0.061
Promedio	26.41	390.0	0.000056	0.368	0.061
Desv. Est.	2.86	3.69	5.2E-05	0.012	0.001

Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	4.99	390	0.000061	0.351	0.061
2	5.11	390	0.000057	0.340	0.061
3	7.86	391	0.000138	0.367	0.060
4	4.66	385	0.000138	0.350	0.062
5	5.93	379	0.000137	0.341	0.061
6	4.73	389	0.000165	0.400	0.059
7	3.91	388	0.000235	0.346	0.061
8	5.04	389	0.000192	0.327	0.063
9	4.35	388	0.000205	0.344	0.062
10	4.94	387	0.000115	0.351	0.061
11	4.53	388	0.000230	0.357	0.061
12	5.20	386	0.000195	0.365	0.060
13	4.23	390	0.000189	0.394	0.060
14	4.89	378	0.000175	0.327	0.062
15	5.00	388	0.000215	0.358	0.061
16	4.28	390	0.000181	0.372	0.060
17	4.55	391	0.000211	0.359	0.060
18	3.64	391	0.000158	0.341	0.061
19	4.67	394	0.000272	0.360	0.060
20	4.96	394	0.000040	0.363	0.061
21	5.80	390	0.000206	0.381	0.060
22	4.49	392	0.000179	0.360	0.061
23	4.81	389	0.000101	0.392	0.060
24	3.92	393	0.000346	0.339	0.062
25	4.98	392	0.000190	0.362	0.061
26	5.32	391	0.000129	0.363	0.060
27	4.27	395	0.000225	0.371	0.061
28	5.06	390	0.000094	0.382	0.060
29	5.35	389	0.000189	0.324	0.062
30	5.45	389	0.000054	0.381	0.061
Promedio	4.92	390.0	0.000180	0.359	0.061
Desv. Est.	0.77	3.69	6.8E-05	0.020	0.001

Problema ZDT5

Serial				Paralelo (4 procesos)			
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional
1	35.47	25	0.3445	1	8.92	24	0.1920
2	30.63	25	0.3445	2	9.69	25	0.2669
3	34.22	25	0.3445	3	9.35	22	0.2497
4	36.87	25	0.3445	4	10.61	24	0.1969
5	37.34	25	0.3445	5	8.92	25	0.1636
6	30.82	25	0.3445	6	9.76	22	0.2326
7	32.99	25	0.3445	7	11.10	26	0.2784
8	32.18	25	0.3445	8	9.85	28	0.2003
9	31.10	25	0.3445	9	10.02	27	0.1084
10	33.66	25	0.3445	10	8.55	24	0.0609
11	50.87	25	0.3445	11	8.67	23	0.1199
12	33.53	25	0.3445	12	7.95	22	0.1825
13	34.11	25	0.3445	13	10.44	26	0.2449
14	33.66	25	0.3445	14	8.72	26	0.2912
15	35.15	25	0.3445	15	9.53	24	0.3113
16	29.98	25	0.3445	16	10.44	22	0.3236
17	34.82	25	0.3445	17	8.17	24	0.1892
18	34.49	25	0.3445	18	9.02	26	0.1217
19	33.90	25	0.3445	19	8.15	26	0.2912
20	34.88	25	0.3445	20	8.01	25	0.3369
21	33.56	25	0.3445	21	10.95	24	0.2021
22	35.12	25	0.3445	22	7.76	27	0.2255
23	36.73	25	0.3445	23	8.05	21	0.2523
24	34.80	25	0.3445	24	11.43	26	0.1179
25	35.44	25	0.3445	25	9.24	25	0.2312
26	35.24	25	0.3445	26	8.80	26	0.2674
27	35.41	25	0.3445	27	7.47	24	0.2782
28	35.15	25	0.3445	28	10.29	27	0.1523
29	33.51	25	0.3445	29	8.95	25	0.1734
30	34.93	25	0.3445	30	9.49	24	0.3335
Promedio	34.65	25.0	0.3445	Promedio	9.13	25.0	0.2283
Desv. Est.	3.54	0.0	0.00	Desv. Est.	1.06	1.75	0.0724

Problema ZDT5 (100 individuos, 200 generaciones)

Serial			
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional
1	2.11	23	0.5912
2	2.11	23	0.5912
3	2.10	23	0.5912
4	2.10	23	0.5912
5	2.09	23	0.5912
6	2.08	23	0.5912
7	2.12	23	0.5912
8	2.08	23	0.5912
9	2.11	23	0.5912
10	2.08	23	0.5912
11	2.01	23	0.5912
12	2.03	23	0.5912
13	2.08	23	0.5912
14	2.10	23	0.5912
15	2.09	23	0.5912
16	2.07	23	0.5912
17	2.10	23	0.5912
18	2.08	23	0.5912
19	2.00	23	0.5912
20	2.09	23	0.5912
21	2.08	23	0.5912
22	2.00	23	0.5912
23	2.01	23	0.5912
24	2.10	23	0.5912
25	2.12	23	0.5912
26	2.02	23	0.5912
27	2.02	23	0.5912
28	2.00	23	0.5912
29	2.10	23	0.5912
30	2.00	23	0.5912
Promedio	2.08	23	0.5912
Desv. Est.	0.04	0	0.0

Paralelo (4 procesos)			
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional
1	2.41	31	0.0270
2	2.45	30	0.0511
3	2.67	30	0.0056
4	2.85	30	0.0479
5	2.73	30	0.0511
6	2.24	29	0.0229
7	2.53	30	0.0511
8	2.28	30	0.0511
9	2.49	30	0.0511
10	2.55	26	0.0356
11	3.02	30	0.0511
12	2.50	30	0.0511
13	2.63	30	0.0511
14	2.45	30	0.0511
15	2.53	30	0.0511
16	2.98	30	0.0511
17	2.32	30	0.0511
18	2.58	31	0.0301
19	2.44	30	0.0511
20	2.56	30	0.0511
21	2.55	30	0.0511
22	2.43	27	0.0468
23	2.74	30	0.0511
24	2.16	29	0.0251
25	2.42	30	0.0511
26	2.32	28	0.0464
27	2.54	27	0.0481
28	2.40	29	0.0478
29	2.91	30	0.0511
30	2.37	30	0.0511
Promedio	2.51	30	0.0511
Desv. Est.	0.21	1	0.011

Problema ZDT6

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	24.48	361	0.0764	0.491	0.107	1	3.35	350	0.073006	0.432	0.056
2	24.69	356	0.0679	0.446	0.113	2	2.98	359	0.066536	0.405	0.056
3	25.35	370	0.0683	0.469	0.113	3	3.00	362	0.071217	0.474	0.054
4	25.46	348	0.0785	0.490	0.105	4	2.96	360	0.073436	0.459	0.054
5	26.16	360	0.0681	0.454	0.109	5	3.00	356	0.073278	0.422	0.056
6	25.99	358	0.0802	0.447	0.111	6	3.17	366	0.065414	0.401	0.056
7	25.87	357	0.0762	0.469	0.112	7	3.08	368	0.066118	0.418	0.055
8	26.25	359	0.0659	0.461	0.109	8	3.23	363	0.065839	0.435	0.055
9	25.88	371	0.0761	0.461	0.108	9	3.13	355	0.075617	0.442	0.056
10	26.04	358	0.0651	0.519	0.105	10	3.02	353	0.064947	0.433	0.056
11	25.82	357	0.0762	0.469	0.112	11	3.10	363	0.066331	0.450	0.054
12	26.10	365	0.0637	0.502	0.105	12	3.04	366	0.072949	0.443	0.055
13	26.21	367	0.0607	0.511	0.104	13	2.97	357	0.06998	0.445	0.056
14	26.04	358	0.0674	0.456	0.108	14	3.08	366	0.074959	0.462	0.054
15	25.90	357	0.0762	0.469	0.112	15	3.00	359	0.069922	0.446	0.055
16	26.19	370	0.0692	0.416	0.110	16	3.05	352	0.070782	0.449	0.056
17	25.79	355	0.0673	0.486	0.107	17	3.10	373	0.070676	0.431	0.055
18	25.98	366	0.0682	0.425	0.114	18	3.00	361	0.074108	0.459	0.055
19	26.02	362	0.0713	0.482	0.108	19	3.12	363	0.064584	0.425	0.055
20	25.86	364	0.0527	0.452	0.106	20	3.09	348	0.071358	0.434	0.057
21	25.90	369	0.0710	0.481	0.110	21	3.01	369	0.062859	0.439	0.055
22	25.70	370	0.0807	0.463	0.110	22	2.95	355	0.070045	0.416	0.057
23	25.93	356	0.0717	0.453	0.109	23	3.12	360	0.066592	0.422	0.056
24	25.97	363	0.0807	0.468	0.112	24	3.08	368	0.069924	0.432	0.055
25	25.70	370	0.0682	0.463	0.110	25	3.09	362	0.072774	0.433	0.056
26	25.96	366	0.0781	0.425	0.114	26	2.98	353	0.072411	0.457	0.056
27	25.84	365	0.0748	0.460	0.114	27	3.69	363	0.062781	0.403	0.056
28	25.85	355	0.0785	0.475	0.111	28	3.06	367	0.067833	0.428	0.055
29	25.85	366	0.0733	0.491	0.110	29	3.04	363	0.06692	0.417	0.056
30	26.07	360	0.0724	0.411	0.115	30	3.02	369	0.059112	0.443	0.054
Promedio	25.90	361.5	0.0715	0.466	0.110	Promedio	3.06	362.0	0.069952	0.433	0.056
Desv. Est.	0.39	5.84	0.0065	0.026	0.003	Desv. Est.	0.14	6.23	0.00409731	0.018	0.001



Problema ZDT6 (100 individuos, 200 generaciones)

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	0.76	90	0.07911	0.491	0.107	1	1.19	96	0.00027	0.418	0.108
2	0.77	88	0.06389	0.446	0.113	2	1.17	88	0.06955	0.381	0.117
3	0.75	87	0.08035	0.469	0.113	3	1.12	89	0.00021	0.401	0.114
4	0.76	94	0.07385	0.490	0.105	4	1.11	89	0.00020	0.532	0.107
5	0.76	89	0.06051	0.454	0.109	5	1.08	88	0.00024	0.409	0.113
6	0.76	93	0.07087	0.447	0.111	6	1.13	91	0.07616	0.442	0.111
7	0.76	87	0.07169	0.469	0.112	7	1.08	89	0.00021	0.413	0.113
8	0.75	89	0.08138	0.461	0.109	8	1.16	91	0.00019	0.463	0.108
9	0.76	90	0.06987	0.461	0.108	9	1.08	85	0.00022	0.436	0.113
10	0.75	95	0.07596	0.519	0.105	10	1.12	89	0.00024	0.382	0.116
11	0.78	87	0.07169	0.469	0.112	11	1.16	90	0.00026	0.427	0.109
12	0.75	89	0.07431	0.502	0.105	12	1.10	90	0.00020	0.463	0.110
13	0.76	91	0.07731	0.511	0.104	13	1.13	90	0.00021	0.446	0.107
14	0.76	93	0.08144	0.456	0.108	14	1.15	86	0.00026	0.367	0.118
15	0.77	87	0.07169	0.469	0.112	15	1.18	90	0.00026	0.510	0.107
16	0.76	91	0.07501	0.416	0.110	16	1.05	88	0.00022	0.440	0.114
17	0.76	92	0.07404	0.486	0.107	17	1.09	89	0.00023	0.358	0.115
18	0.76	85	0.07572	0.425	0.114	18	1.13	92	0.07272	0.410	0.112
19	0.76	93	0.07129	0.482	0.108	19	1.09	87	0.00024	0.404	0.115
20	0.77	97	0.07401	0.452	0.106	20	1.12	90	0.00132	0.523	0.115
21	0.77	87	0.07561	0.481	0.110	21	1.11	88	0.00122	0.525	0.115
22	0.76	90	0.07239	0.463	0.110	22	1.14	88	0.00023	0.406	0.114
23	0.76	90	0.07459	0.453	0.109	23	1.13	87	0.06393	0.421	0.115
24	0.75	88	0.07699	0.468	0.112	24	1.16	95	0.00025	0.376	0.112
25	0.76	90	0.07239	0.463	0.110	25	1.09	90	0.00025	0.381	0.116
26	0.76	85	0.07572	0.425	0.114	26	1.14	92	0.00022	0.471	0.108
27	0.76	83	0.07350	0.460	0.114	27	1.12	86	0.00019	0.406	0.118
28	0.76	89	0.07191	0.475	0.111	28	1.13	88	0.00023	0.398	0.116
29	0.76	88	0.08193	0.491	0.110	29	1.11	87	0.00023	0.397	0.117
30	0.76	89	0.07636	0.411	0.115	30	1.15	90	0.00021	0.452	0.108
Promedio	0.76	89	0.07418	0.466	0.110	Promedio	1.12	89	0.00024	0.415	0.113
Desv. Est.	0.01	3.09	0.00463	0.026	0.003	Desv. Est.	0.03	2.41	0.02436	0.047	0.004

## Problema KUR

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	20.83	391	0.0100	0.548	0.316	1	3.53	387	0.0092	0.475	0.161
2	20.57	391	0.0093	0.548	0.296	2	3.01	385	0.0118	0.453	0.167
3	20.71	388	0.0108	0.467	0.313	3	3.02	389	0.0094	0.422	0.162
4	20.55	385	0.0094	0.483	0.325	4	2.93	381	0.0095	0.479	0.162
5	20.57	387	0.0091	0.459	0.331	5	3.02	385	0.0096	0.457	0.163
6	20.57	390	0.0111	0.462	0.328	6	3.12	387	0.0106	0.453	0.163
7	20.55	391	0.0092	0.494	0.322	7	3.10	385	0.0094	0.475	0.161
8	20.58	389	0.0093	0.504	0.321	8	2.95	389	0.0097	0.475	0.164
9	20.59	390	0.0107	0.519	0.325	9	3.11	388	0.0098	0.434	0.164
10	20.77	391	0.0094	0.545	0.312	10	3.01	383	0.0096	0.454	0.163
11	20.55	391	0.0092	0.494	0.322	11	3.09	386	0.0102	0.441	0.164
12	20.54	386	0.0093	0.478	0.321	12	3.02	393	0.0096	0.477	0.159
13	20.56	378	0.0097	0.544	0.331	13	2.99	390	0.0097	0.482	0.164
14	20.56	384	0.0096	0.469	0.329	14	3.00	393	0.0095	0.451	0.163
15	20.70	391	0.0092	0.494	0.322	15	3.02	389	0.0095	0.420	0.160
16	20.55	387	0.0095	0.468	0.326	16	3.16	386	0.0095	0.438	0.164
17	20.87	381	0.0089	0.483	0.336	17	2.96	388	0.0094	0.467	0.163
18	20.56	387	0.0091	0.423	0.323	18	2.98	386	0.0090	0.441	0.164
19	20.68	389	0.0107	0.496	0.310	19	3.01	390	0.0095	0.441	0.163
20	20.64	386	0.0101	0.479	0.320	20	3.03	391	0.0095	0.471	0.161
21	20.72	390	0.0096	0.450	0.326	21	3.15	387	0.0093	0.445	0.164
22	20.58	390	0.0105	0.463	0.317	22	3.05	384	0.0098	0.438	0.165
23	20.60	386	0.0098	0.485	0.323	23	3.01	388	0.0100	0.432	0.165
24	20.53	390	0.0088	0.501	0.321	24	2.96	388	0.0097	0.460	0.159
25	20.53	390	0.0105	0.463	0.317	25	3.02	390	0.0098	0.416	0.167
26	20.55	387	0.0091	0.423	0.323	26	3.22	389	0.0095	0.427	0.160
27	20.77	387	0.0094	0.404	0.332	27	2.98	393	0.0100	0.446	0.165
28	20.73	383	0.0090	0.515	0.325	28	3.05	391	0.0092	0.495	0.159
29	20.57	383	0.0105	0.515	0.315	29	3.03	387	0.0105	0.463	0.166
30	20.54	378	0.0093	0.413	0.325	30	3.02	384	0.0100	0.433	0.164
Promedio	20.57	387.5	0.0094	0.483	0.322	Promedio	3.02	388.0	0.0096	0.452	0.163
Desv. Est.	0.10	3.73	0.0006	0.038	0.008	Desv. Est.	0.11	2.97	0.0005	0.021	0.002

Problema KUR (100 individuos, 200 generaciones)

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	2.25	98	0.0108	0.9214	0.316	1	3.53	99	0.014	0.869	0.269
2	2.26	97	0.0133	0.9171	0.296	2	3.01	98	0.014	0.863	0.292
3	2.25	98	0.0131	0.9080	0.313	3	3.02	99	0.013	0.880	0.258
4	2.31	96	0.0123	0.9073	0.325	4	2.93	95	0.017	0.875	0.289
5	2.24	98	0.0116	0.8995	0.331	5	3.02	95	0.018	0.879	0.318
6	2.24	97	0.0105	0.9007	0.328	6	3.12	97	0.014	0.873	0.287
7	2.24	98	0.0128	0.9077	0.322	7	3.10	96	0.019	0.879	0.283
8	2.24	99	0.0167	0.9129	0.321	8	2.95	97	0.012	0.880	0.265
9	2.24	96	0.0110	0.9128	0.325	9	3.11	100	0.012	0.874	0.264
10	2.24	98	0.0142	0.9210	0.312	10	3.01	97	0.021	0.859	0.318
11	2.24	98	0.0128	0.9077	0.322	11	3.09	96	0.024	0.888	0.291
12	2.25	97	0.0147	0.9096	0.321	12	3.02	97	0.016	0.874	0.307
13	2.24	98	0.0205	0.9147	0.331	13	2.99	98	0.014	0.865	0.297
14	2.25	96	0.0130	0.9034	0.329	14	3.00	97	0.013	0.871	0.267
15	2.26	98	0.0128	0.9077	0.322	15	3.02	97	0.020	0.847	0.310
16	2.25	96	0.0140	0.9059	0.326	16	3.16	99	0.014	0.868	0.277
17	2.68	97	0.0121	0.9000	0.336	17	2.96	97	0.016	0.870	0.320
18	2.26	98	0.0114	0.8940	0.323	18	2.98	99	0.015	0.888	0.302
19	2.25	98	0.0121	0.9113	0.310	19	3.01	96	0.017	0.861	0.309
20	3.02	98	0.0137	0.9081	0.320	20	3.03	99	0.018	0.897	0.310
21	2.57	97	0.0123	0.8989	0.326	21	3.15	98	0.016	0.866	0.275
22	2.23	98	0.0172	0.9092	0.318	22	3.05	98	0.012	0.881	0.263
23	2.25	97	0.0121	0.9068	0.323	23	3.01	98	0.013	0.879	0.259
24	2.24	96	0.0113	0.9122	0.321	24	2.96	99	0.015	0.850	0.286
25	2.45	98	0.0172	0.9092	0.318	25	3.02	99	0.016	0.866	0.293
26	3.01	98	0.0114	0.8940	0.323	26	3.22	97	0.019	0.889	0.293
27	3.00	99	0.0163	0.8886	0.332	27	2.98	98	0.016	0.877	0.316
28	2.27	97	0.0152	0.9130	0.325	28	3.05	97	0.012	0.877	0.273
29	2.24	95	0.0124	0.9184	0.315	29	3.03	99	0.011	0.885	0.283
30	2.24	98	0.0158	0.8950	0.325	30	3.02	100	0.014	0.883	0.262
Prom	2.25	98	0.0128	0.9079	0.322	Prom	3.02	98	0.015	0.874	0.288
Desv. Est.	0.24	0.97	0.0023	0.0081	0.008	Desv. Est.	0.11	1.34	0.003	0.011	0.020

## Problema CEX

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	21.62	362	0.00044	0.736	0.126	1	2.97	350	0.00060	0.962	0.111
2	21.59	352	0.00035	0.737	0.126	2	2.88	337	0.00055	0.878	0.124
3	21.64	355	0.00036	0.711	0.126	3	2.85	354	0.00073	1.006	0.113
4	21.64	361	0.00042	0.741	0.121	4	3.02	349	0.00052	0.948	0.115
5	21.57	362	0.00034	0.734	0.124	5	2.85	360	0.00066	0.964	0.113
6	21.57	359	0.00054	0.709	0.125	6	3.00	352	0.00059	0.945	0.113
7	21.61	353	0.00052	0.734	0.127	7	2.91	361	0.00049	0.963	0.109
8	21.62	354	0.00042	0.738	0.125	8	2.97	351	0.00052	0.927	0.115
9	21.56	358	0.00047	0.737	0.124	9	2.91	352	0.00059	0.983	0.113
10	21.59	348	0.00033	0.727	0.126	10	2.96	357	0.00054	0.953	0.115
11	21.60	353	0.00052	0.734	0.127	11	2.84	351	0.00067	0.908	0.116
12	21.71	359	0.00040	0.764	0.123	12	2.90	343	0.00059	0.939	0.114
13	21.63	345	0.00053	0.706	0.131	13	2.87	360	0.00057	0.983	0.109
14	21.56	359	0.00042	0.719	0.125	14	2.81	353	0.00067	0.956	0.114
15	21.60	353	0.00052	0.734	0.127	15	2.86	344	0.00078	0.938	0.115
16	21.63	357	0.00037	0.739	0.127	16	3.09	349	0.00064	0.942	0.119
17	21.62	364	0.00032	0.731	0.124	17	2.88	359	0.00069	0.923	0.114
18	21.62	359	0.00048	0.740	0.123	18	2.87	353	0.00063	0.918	0.114
19	21.64	351	0.00048	0.713	0.125	19	2.95	346	0.00051	0.943	0.115
20	21.68	354	0.00041	0.724	0.128	20	2.94	352	0.00054	0.929	0.114
21	21.66	353	0.00054	0.725	0.125	21	2.95	352	0.00061	0.930	0.116
22	21.62	350	0.00034	0.731	0.127	22	2.99	343	0.00045	0.993	0.116
23	21.60	367	0.00059	0.756	0.123	23	2.87	355	0.00066	0.939	0.113
24	21.58	345	0.00031	0.690	0.131	24	2.85	351	0.00061	0.927	0.116
25	21.60	350	0.00034	0.731	0.127	25	2.92	354	0.00081	0.923	0.117
26	21.62	359	0.00048	0.740	0.123	26	2.86	359	0.00057	0.913	0.113
27	21.68	363	0.00046	0.735	0.124	27	2.86	363	0.00059	0.948	0.111
28	21.61	356	0.00054	0.745	0.124	28	3.03	351	0.00071	0.936	0.114
29	21.63	361	0.00043	0.736	0.122	29	2.93	340	0.00053	0.945	0.119
30	21.62	352	0.00042	0.734	0.125	30	2.85	360	0.00054	0.984	0.111
Promedio	21.62	355.5	0.00043	0.734	0.125	Promedio	2.90	352.0	0.00059	0.942	0.114
Desv. Est.	0.04	5.52	7.9E-05	0.015	0.002	Desv. Est.	0.07	6.41	8.4E-05	0.027	0.003

## Problema BNH

Serial					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	28.95	397	0.0142	0.454	0.572
2	27.32	393	0.0142	0.448	0.585
3	27.30	396	0.0130	0.458	0.576
4	27.33	397	0.0125	0.461	0.575
5	27.31	392	0.0118	0.443	0.580
6	27.33	387	0.0148	0.446	0.591
7	27.48	399	0.0133	0.424	0.583
8	27.27	397	0.0138	0.461	0.573
9	27.45	395	0.0127	0.444	0.584
10	27.29	395	0.0134	0.442	0.574
11	27.28	399	0.0133	0.424	0.583
12	27.28	394	0.0139	0.416	0.588
13	27.29	396	0.0122	0.434	0.578
14	27.37	396	0.0131	0.443	0.577
15	27.40	399	0.0133	0.424	0.583
16	27.30	393	0.0147	0.466	0.572
17	27.30	392	0.0144	0.455	0.573
18	27.39	390	0.0109	0.453	0.584
19	27.28	395	0.0146	0.433	0.571
20	27.41	393	0.0130	0.442	0.578
21	27.30	396	0.0137	0.437	0.579
22	27.38	396	0.0119	0.456	0.570
23	27.28	397	0.0140	0.440	0.581
24	27.41	397	0.0126	0.464	0.577
25	27.33	396	0.0119	0.456	0.570
26	27.44	390	0.0109	0.453	0.584
27	27.29	391	0.0144	0.471	0.574
28	27.30	396	0.0116	0.461	0.578
29	27.29	394	0.0124	0.452	0.574
30	27.33	395	0.0123	0.483	0.573
Promedio	27.31	395.5	0.0132	0.450	0.577
Desv. Est.	0.30	2.85	0.0011	0.015	0.006

Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	5.47	398	0.0155	0.423	0.583
2	5.15	397	0.0168	0.460	0.579
3	4.16	394	0.0142	0.443	0.581
4	5.59	397	0.0135	0.441	0.573
5	6.65	394	0.0133	0.469	0.579
6	4.65	398	0.0142	0.433	0.575
7	4.78	396	0.0156	0.430	0.584
8	5.44	398	0.0141	0.459	0.578
9	5.10	395	0.0155	0.447	0.576
10	4.81	397	0.0151	0.446	0.580
11	5.05	392	0.0130	0.460	0.578
12	5.15	395	0.0169	0.442	0.585
13	4.57	393	0.0154	0.475	0.573
14	5.61	394	0.0160	0.449	0.579
15	3.18	394	0.0147	0.437	0.581
16	3.39	398	0.0144	0.461	0.569
17	3.26	393	0.0156	0.461	0.578
18	3.34	398	0.0158	0.446	0.579
19	3.20	397	0.0128	0.449	0.578
20	3.54	394	0.0166	0.463	0.579
21	3.10	396	0.0155	0.448	0.576
22	3.08	396	0.0145	0.444	0.573
23	3.13	394	0.0156	0.437	0.571
24	3.31	393	0.0123	0.444	0.576
25	3.34	394	0.0144	0.415	0.582
26	3.37	396	0.0121	0.420	0.580
27	3.09	393	0.0157	0.437	0.583
28	3.30	398	0.0155	0.458	0.570
29	3.06	395	0.0132	0.445	0.583
30	3.53	392	0.0166	0.442	0.586
Promedio	3.54	395.0	0.0152	0.445	0.579
Desv. Est.	1.05	1.97	0.0013	0.014	0.004

## Problema VI2

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	25.84	394	0.0015	0.5898	0.095	1	3.49	392	0.0014	0.6718	0.087
2	25.84	391	0.0014	0.6153	0.099	2	3.44	391	0.0018	0.6103	0.089
3	25.84	396	0.0021	0.6411	0.100	3	3.22	393	0.0018	0.6314	0.089
4	25.93	396	0.0013	0.6002	0.096	4	3.22	395	0.0012	0.6222	0.085
5	25.86	393	0.0014	0.6558	0.097	5	3.24	390	0.0014	0.5948	0.089
6	25.84	392	0.0015	0.6280	0.100	6	3.32	395	0.0013	0.6189	0.086
7	25.84	390	0.0015	0.5777	0.097	7	3.59	396	0.0014	0.6316	0.086
8	25.85	389	0.0016	0.6613	0.096	8	3.70	387	0.0015	0.6056	0.087
9	25.82	397	0.0017	0.5940	0.097	9	3.30	386	0.0013	0.5876	0.087
10	25.83	394	0.0016	0.6020	0.099	10	3.18	395	0.0017	0.6424	0.093
11	25.82	390	0.0015	0.5777	0.097	11	3.23	391	0.0015	0.6224	0.089
12	25.87	389	0.0017	0.6503	0.096	12	4.29	397	0.0015	0.6157	0.086
13	25.84	391	0.0017	0.5945	0.096	13	3.24	389	0.0018	0.6314	0.089
14	25.83	391	0.0014	0.5939	0.097	14	3.24	393	0.0015	0.6719	0.089
15	25.83	390	0.0015	0.5777	0.097	15	3.42	390	0.0015	0.6080	0.090
16	25.94	392	0.0014	0.6077	0.097	16	3.39	395	0.0017	0.6402	0.088
17	25.91	390	0.0014	0.6124	0.096	17	3.28	390	0.0017	0.6615	0.088
18	25.84	390	0.0015	0.6150	0.096	18	3.21	397	0.0016	0.6381	0.086
19	25.85	389	0.0016	0.6054	0.098	19	3.23	394	0.0013	0.6092	0.086
20	25.84	387	0.0016	0.6440	0.099	20	3.28	391	0.0014	0.6365	0.086
21	25.86	391	0.0015	0.6259	0.098	21	3.31	390	0.0014	0.5703	0.091
22	25.83	391	0.0017	0.6211	0.101	22	3.19	396	0.0018	0.6278	0.087
23	25.84	396	0.0013	0.6016	0.096	23	3.22	390	0.0015	0.5751	0.086
24	25.81	388	0.0013	0.5821	0.098	24	3.23	393	0.0017	0.6752	0.087
25	25.85	391	0.0017	0.6211	0.101	25	3.16	396	0.0013	0.5395	0.086
26	25.85	390	0.0015	0.6150	0.096	26	3.24	398	0.0014	0.6322	0.087
27	25.84	394	0.0013	0.6441	0.098	27	3.28	391	0.0017	0.6388	0.090
28	25.85	397	0.0017	0.6329	0.100	28	3.21	387	0.0016	0.6363	0.088
29	25.91	389	0.0016	0.5778	0.097	29	3.19	395	0.0016	0.6400	0.087
30	25.87	391	0.0013	0.6805	0.097	30	3.16	392	0.0013	0.5959	0.085
Promedio	25.84	391.0	0.0015	0.6137	0.097	Promedio	3.24	392.5	0.0015	0.6296	0.087
Desv. Est.	0.03	2.72	0.0002	0.027	0.002	Desv. Est.	0.22	3.20	0.0002	0.030	0.002

Problema VI3

Serial						Paralelo (4 procesos)					
Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing	Ejecución	Tiempo (seg.)	Puntos ND	Distancia Generacional	Delta	Spacing
1	23.25	375	0.0217	0.7506	0.154	1	3.93	368	0.0220	0.8130	0.157
2	23.28	381	0.0212	0.7737	0.155	2	3.88	378	0.0227	0.7074	0.156
3	23.27	378	0.0226	0.7563	0.154	3	4.05	376	0.0220	0.7737	0.154
4	23.28	382	0.0221	0.7288	0.151	4	3.97	376	0.0227	0.7372	0.156
5	24.13	370	0.0227	0.7678	0.160	5	3.92	379	0.0228	0.7697	0.153
6	23.29	372	0.0231	0.6886	0.155	6	3.84	379	0.0213	0.7571	0.155
7	23.28	381	0.0218	0.7322	0.151	7	4.08	382	0.0219	0.7177	0.153
8	23.26	373	0.0226	0.6844	0.159	8	3.91	377	0.0221	0.7421	0.153
9	23.25	373	0.0223	0.7629	0.154	9	4.08	364	0.0226	0.8035	0.157
10	23.30	374	0.0241	0.6973	0.151	10	3.95	375	0.0220	0.7759	0.158
11	23.26	381	0.0218	0.7322	0.151	11	4.09	372	0.0225	0.7785	0.155
12	23.28	373	0.0222	0.7253	0.154	12	3.82	376	0.0224	0.7699	0.155
13	23.28	377	0.0221	0.7806	0.154	13	3.87	376	0.0222	0.7360	0.152
14	23.29	374	0.0234	0.7112	0.153	14	3.97	375	0.0222	0.7454	0.153
15	23.27	381	0.0218	0.7322	0.151	15	3.96	374	0.0226	0.7725	0.156
16	23.27	376	0.0231	0.7174	0.154	16	3.94	380	0.0223	0.7136	0.153
17	23.27	374	0.0222	0.7628	0.156	17	3.97	375	0.0230	0.7144	0.154
18	23.27	376	0.0225	0.7235	0.150	18	3.95	378	0.0224	0.7133	0.153
19	23.28	369	0.0222	0.7726	0.157	19	3.97	374	0.0218	0.7688	0.158
20	23.30	376	0.0223	0.7810	0.153	20	3.97	375	0.0213	0.7833	0.155
21	23.31	377	0.0222	0.7794	0.154	21	3.88	369	0.0220	0.7246	0.154
22	23.29	367	0.0218	0.7739	0.156	22	3.95	370	0.0223	0.7432	0.151
23	23.31	378	0.0223	0.7455	0.150	23	3.86	378	0.0222	0.7657	0.153
24	23.28	368	0.0227	0.7567	0.156	24	3.96	380	0.0220	0.8038	0.151
25	23.27	367	0.0218	0.7739	0.156	25	3.94	371	0.0221	0.7671	0.159
26	23.33	376	0.0225	0.7235	0.150	26	3.97	371	0.0223	0.7258	0.155
27	23.28	368	0.0222	0.7361	0.157	27	3.99	387	0.0218	0.7621	0.152
28	23.33	372	0.0222	0.7635	0.158	28	3.94	368	0.0228	0.7678	0.159
29	23.28	374	0.0225	0.7658	0.156	29	3.90	367	0.0233	0.7362	0.155
30	23.28	381	0.0224	0.7532	0.151	30	3.96	379	0.0222	0.7656	0.152
Promedio	23.28	374.5	0.0223	0.7519	0.154	Promedio	3.95	375.5	0.0222	0.7638	0.155
Desv. Est.	0.16	4.47	0.0006	0.028	0.003	Desv. Est.	0.07	4.96	0.0004	0.029	0.002

## Anexo II : Tablas de Resultados para el problema GSP

Este anexo reúne los resultados obtenidos para cada una de las 30 ejecuciones independientes de las versiones serial y paralela del algoritmo NSGA-II aplicadas a la resolución del problema de Steiner generalizado.

Grafo 50-15

Serial			
Ejecución	Tiempo	Puntos	Puntos ND
1	148.58	200	9
2	148.16	200	15
3	150.33	200	20
4	150.06	200	13
5	150.27	200	13
6	150.25	200	15
7	150.20	200	15
8	149.71	200	11
9	149.07	200	21
10	150.32	200	19
11	148.59	200	16
12	151.00	200	18
13	148.74	200	18
14	149.35	200	12
15	149.54	200	13
16	149.13	200	13
17	148.13	200	18
18	148.62	200	17
19	150.93	200	19
20	150.24	200	13
21	149.95	200	18
22	148.35	200	7
23	150.22	200	11
24	148.31	200	24
25	149.93	200	10
26	149.75	200	26
27	148.72	200	18
28	150.26	200	13
29	149.54	200	17
30	149.90	200	18
Promedio	149.73	200.0	15.5
Desv. Est.	0.84	0.00	4.29

Paralelo			
Ejecución	Tiempo	Puntos	Puntos ND
1	68.78	47	14
2	70.63	42	11
3	69.34	57	16
4	69.60	52	21
5	68.71	48	14
6	68.07	48	17
7	68.88	60	18
8	69.97	52	18
9	71.61	57	20
10	69.91	37	11
11	70.44	52	18
12	68.16	46	16
13	69.24	52	23
14	71.10	57	12
15	68.98	61	23
16	69.46	42	18
17	69.00	38	13
18	70.01	61	11
19	69.13	46	14
20	69.27	52	15
21	70.99	50	11
22	69.64	37	16
23	68.14	56	18
24	69.98	53	17
25	72.41	52	14
26	69.08	58	15
27	68.90	43	19
28	70.74	39	13
29	71.03	55	20
30	71.07	55	13
Promedio	69.53	52.0	16.0
Desv. Est.	1.08	7.19	3.46



Grafo 75-25

Serial			
Ejecución	Tiempo	Puntos	Puntos ND
1	643.92	200	19
2	643.92	200	13
3	631.88	200	16
4	633.56	200	18
5	643.39	200	20
6	645.02	200	15
7	648.33	200	17
8	644.98	200	14
9	642.67	200	20
10	628.24	200	16
11	631.78	200	13
12	643.61	200	12
13	643.68	200	12
14	643.97	200	12
15	628.65	200	18
16	643.69	200	13
17	631.04	200	14
18	650.29	200	19
19	643.52	200	11
20	643.11	200	20
21	636.41	200	13
22	648.33	200	19
23	642.22	200	19
24	644.51	200	15
25	649.71	200	16
26	653.10	200	14
27	645.13	200	13
28	631.39	200	13
29	651.32	200	17
30	654.50	200	25
Promedio	643.68	200.0	15.5
Desv. Est.	7.29	0.00	3.30

Paralelo			
Ejecución	Tiempo	Puntos	Puntos ND
1	179.00	51	15
2	180.42	42	12
3	179.94	44	18
4	180.00	50	10
5	180.34	71	8
6	176.68	57	17
7	182.31	46	17
8	182.13	48	16
9	180.35	50	17
10	180.13	47	15
11	179.32	56	19
12	180.90	59	13
13	184.99	49	14
14	175.87	45	13
15	179.10	39	15
16	181.59	58	16
17	182.40	49	13
18	179.56	53	18
19	179.83	46	13
20	171.93	59	14
21	183.02	52	17
22	178.37	57	16
23	185.51	36	16
24	180.08	59	17
25	183.12	48	9
26	176.22	46	13
27	184.76	48	13
28	181.74	57	18
29	176.26	59	17
30	176.93	59	22
Promedio	180.10	50.0	15.5
Desv. Est.	2.97	7.35	3.02

Grafo 100-10

Serial			
Ejecución	Tiempo	Puntos	Puntos ND
1	145.20	200	6
2	146.32	200	11
3	143.68	200	15
4	144.75	200	11
5	145.68	200	14
6	146.01	200	11
7	145.07	200	12
8	144.89	200	9
9	143.58	200	11
10	144.99	200	8
11	145.98	200	10
12	145.86	200	9
13	144.78	200	9
14	144.70	200	10
15	142.92	200	14
16	144.32	200	8
17	145.83	200	15
18	143.74	200	13
19	143.83	200	6
20	144.01	200	14
21	144.72	200	12
22	144.44	200	8
23	144.23	200	14
24	146.04	200	9
25	145.79	200	15
26	147.25	200	8
27	146.50	200	12
28	144.63	200	8
29	145.12	200	9
30	145.43	200	9
Prom	144.94	200.0	10.5
Desv. Est.	0.99	0.00	2.68

Paralelo			
Ejecución	Tiempo	Puntos	Puntos ND
1	74.80	31	13
2	74.99	57	11
3	73.93	51	11
4	76.41	44	9
5	75.04	43	16
6	76.34	50	11
7	75.16	45	15
8	75.68	45	13
9	74.74	26	11
10	76.13	25	10
11	74.15	41	17
12	76.18	52	12
13	75.34	40	14
14	74.72	37	10
15	72.94	42	7
16	75.35	35	14
17	76.01	29	14
18	75.97	40	8
19	76.06	42	14
20	74.34	43	11
21	75.14	38	9
22	74.75	47	7
23	77.04	41	7
24	74.80	58	12
25	74.72	43	10
26	75.74	39	14
27	74.31	43	13
28	76.60	44	9
29	74.51	45	13
30	75.18	49	12
Prom	75.15	43.0	11.5
Desv. Est.	0.90	7.84	2.65