

Equipo Probabilístico de Algoritmos Evolutivos Multiobjetivo en Computación Paralela

José Manuel Fernandez Giangreco^{1,2} y Benjamín Barán^{1,2}

¹ Centro Nacional de Computación - Universidad Nacional de Asunción
P. O. Box 1439 – Paraguay. Tel/Fax: (+595-21) 585550
{jmfernandez, bbaran}@cnc.una.py
<http://www.cnc.una.py>

² Universidad Católica “Nuestra Señora de la Asunción”
<http://www.uca.edu.py>

Resumen. Con el uso cada vez más extendido de Algoritmos Evolutivos para Optimización Multiobjetivo en problemas reales de optimización, se hace necesario mejorar el desempeño de los mismos sacando el máximo provecho de las bondades de cada algoritmo. Para ello, una reconocida alternativa es la incorporación de conceptos de paralelismo. En consecuencia, los algoritmos evolutivos multiobjetivo paralelos se constituyen en un área de creciente interés para aplicaciones prácticas, sobre todo en problemas industriales, financieros y de ingeniería aplicada. Dado que estos algoritmos difieren en su desempeño según el tipo del problema, se propone un *Equipo Probabilístico de Algoritmos Evolutivos Multiobjetivo* en el contexto de la Computación Paralela. Resultados experimentales demuestran las ventajas (y desventajas) de esta nueva propuesta, al compararla con implementaciones paralelas de diversos algoritmos evolutivos multiobjetivos conocidos.

Palabras Claves: Algoritmos Evolutivos, Optimización Multiobjetivo, Paralelismo, Equipo de Algoritmos.

1 Introducción

En la búsqueda de soluciones a problemas del mundo real puede ser necesario satisfacer de manera simultánea múltiples objetivos, los cuales pueden ser contradictorios [22]. De existir la posibilidad de combinar los diferentes objetivos y conociendo la mejor manera de hacerlo, se puede simplemente considerar la existencia de un único objetivo a optimizar, resultante de la combinación de todos los objetivos considerados. Sin embargo, lo usual es que no se conozca de que manera combinar los diferentes objetivos, o esto sea inadecuado, cuando no imposible. Entonces, se trata de un Problema de Optimización Multiobjetivo (*Multiobjective Optimization Problem* - MOP) [8,20, 22, 26].

En los problemas de optimización multiobjetivo con objetivos contradictorios no existe una solución única que pueda ser considerada la mejor, sino un conjunto de alternativas representando los mejores compromisos entre todos los objetivos, en el

sentido que cada solución es mejor que las otras en algún objetivo, pero ninguna es mejor que otra en todos los objetivos simultáneamente [22]. Dicho conjunto es llamado conjunto de soluciones Pareto óptimo y sus correspondientes vectores en el espacio objetivo constituyen el denominado Frente Pareto [20].

Los Algoritmos Evolutivos (*Evolutionary Algorithms* - EAs) [1] han demostrado ser especialmente adecuados para la optimización multiobjetivo [10, 11, 16, 19, 27]. En la actualidad, existe un gran número de Algoritmos Evolutivos para Optimización Multiobjetivo (*MultiObjective Evolutionary Algorithms* - MOEAs) [9]. Con el uso cada vez más extendido de estos algoritmos en problemas reales de optimización, se hace necesario mejorar el desempeño de los mismos [2, 18]. Por lo tanto, para asegurar la aplicabilidad de la técnica de optimización evolutiva multiobjetivo a problemas de complejidad creciente, es necesario mejorar tanto la efectividad como la eficiencia de los métodos evolutivos. Para ello, una alternativa es la incorporación de conceptos de paralelismo al diseño de estos algoritmos [18, 25], así tenemos a los algoritmos evolutivos multiobjetivo paralelos (*parallel Multiobjective Evolutionary Algorithms* - pMOEAs).

Por otra parte tenemos a los algoritmos en equipo (*Team Algorithm* - TA), cuya idea central es combinar algoritmos diferentes que interactúen en la solución de un mismo problema global que por su complejidad o tamaño no puede ser resuelto de manera eficiente o práctica, con un único algoritmo [5]. Los TA han demostrado ser una excelente técnica computacional para combinar una variedad de algoritmos corriendo en diferentes procesadores de una red típicamente asíncrona, como las actuales redes de área local conformadas por computadores personales [2, 3, 4, 6, 7].

Ya que se dispone de diversos algoritmos pMOEAs, surge el *Team Algorithm Multiobjective Evolutionary Algorithms* (TA-MOEA) como una alternativa válida para mejorar tanto la efectividad como en la eficiencia al poder explorar diferentes porciones del espacio de búsqueda, posiblemente disjuntas, con características variadas.

Siendo el área de algoritmos evolutivos multiobjetivo un área de importancia y reciente interés, este trabajo propone determinar las ventajas y desventajas existentes en el desarrollo y aplicación de un TA-MOEA que combine las mejores características de cada MOEA, conformando un TA paralelo de algoritmos evolutivos multiobjetivo. A fin de determinar la efectividad de la técnica propuesta, inspirado en [20, 26, 30], el presente trabajo realiza una comparación de distintos pMOEAs y el TA-MOEA utilizando diferentes métricas experimentales usualmente utilizadas para medir su desempeño [17, 26].

Entonces, el presente trabajo propone básicamente un TA-MOEA que utiliza un criterio de selección probabilístico, basado en el número de soluciones aportadas por cada pMOEA. El trabajo está organizado de la siguiente manera: En la Sección 2 se presenta la definición general de un problema de optimización multiobjetivo. La formulación de los problemas ZDT con sus funciones objetivo son dadas en la Sección 3. Los algoritmos en equipo son explicados en la Sección 4. Los resultados experimentales son mostrados en la Sección 5. Por último, las conclusiones y los trabajos futuros son presentados en la Sección 6.

2 Optimización Multiobjetivo

Un Problema de Optimización Multiobjetivo (MOP) general incluye un conjunto de n parámetros (variables de decisión), un conjunto de k funciones objetivo, y un conjunto de m restricciones. Las funciones objetivo y las restricciones son funciones de las variables de decisión. Luego, un MOP puede expresarse como:

$$\begin{aligned}
 &\text{Optimizar} && \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \\
 &\text{sujeta a} && \mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_m(\mathbf{x})) \geq \mathbf{0} \\
 &\text{donde} && \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbf{X} \\
 &&& \mathbf{y} = (y_1, y_2, \dots, y_k) \in \mathbf{Y}
 \end{aligned} \tag{2.1}$$

siendo \mathbf{x} el vector de decisión e \mathbf{y} el vector objetivo. El espacio de decisión se denota por \mathbf{X} , y al espacio objetivo por \mathbf{Y} . Optimizar, dependiendo del problema, puede significar igualmente, minimizar o maximizar. El conjunto de restricciones $\mathbf{e}(\mathbf{x}) \geq \mathbf{0}$ determina el conjunto de soluciones factibles \mathbf{X}_f y su correspondiente conjunto de vectores objetivo factibles \mathbf{Y}_f .

El problema de optimización consiste en hallar el vector \mathbf{x} que tenga el “mejor valor” de $\mathbf{f}(\mathbf{x})$. En general, no existe un único “mejor valor”, sino un conjunto de soluciones óptimas. Entonces, un nuevo concepto de optimalidad debe ser establecido para MOPs. Dados dos vectores de decisión $\mathbf{u}, \mathbf{v} \in \mathbf{X}$, puede darse una de las tres condiciones siguientes:

$$\begin{aligned}
 &\mathbf{f}(\mathbf{u}) = \mathbf{f}(\mathbf{v}) \text{ si y solo si } \forall i \in \{1, 2, \dots, k\}: f_i(\mathbf{u}) = f_i(\mathbf{v}) \\
 &\mathbf{f}(\mathbf{u}) \leq \mathbf{f}(\mathbf{v}) \text{ si y solo si } \forall i \in \{1, 2, \dots, k\}: f_i(\mathbf{u}) \leq f_i(\mathbf{v}) \\
 &\mathbf{f}(\mathbf{u}) < \mathbf{f}(\mathbf{v}) \text{ si y solo si } \mathbf{f}(\mathbf{u}) \leq \mathbf{f}(\mathbf{v}) \wedge \mathbf{f}(\mathbf{u}) \neq \mathbf{f}(\mathbf{v})
 \end{aligned} \tag{2.2}$$

En un contexto de minimización, esta situación se expresa con los siguientes símbolos y términos:

$$\begin{aligned}
 &\mathbf{u} \succ \mathbf{v} \text{ (u domina a v) si y solo si } \mathbf{f}(\mathbf{u}) < \mathbf{f}(\mathbf{v}) \\
 &\mathbf{v} \succ \mathbf{u} \text{ (v domina a u) si y solo si } \mathbf{f}(\mathbf{v}) < \mathbf{f}(\mathbf{u}) \\
 &\mathbf{u} \sim \mathbf{v} \text{ (u y v no son comparables) si y solo si } \mathbf{f}(\mathbf{u}) \not\leq \mathbf{f}(\mathbf{v}) \wedge \mathbf{f}(\mathbf{v}) \not\leq \mathbf{f}(\mathbf{u})
 \end{aligned} \tag{2.3}$$

Alternativamente, $\mathbf{u} \triangleright \mathbf{v}$ denota que \mathbf{u} domina o es igual a \mathbf{v} .

Dado un vector de decisión $\mathbf{x} \in \mathbf{X}_f$, se dice que \mathbf{x} es no dominado respecto a un conjunto $\mathbf{V} \subseteq \mathbf{X}_f$ si y solo si $\mathbf{x} \succ \mathbf{v}$ o $\mathbf{x} \sim \mathbf{v}$, $\forall \mathbf{v} \in \mathbf{V}$. En caso que \mathbf{x} sea no dominado respecto a todo el conjunto \mathbf{X}_f , y solo en ese caso, se dice que \mathbf{x} es una solución Pareto óptima. Por lo tanto, el conjunto Pareto óptimo \mathbf{X}_{true} puede ser definido formalmente de la siguiente manera:

$$\mathbf{X}_{true} = \{ \mathbf{x} \in \mathbf{X}_f \mid \mathbf{x} \text{ es no dominado con respecto a } \mathbf{X}_f \} \tag{2.4}$$

El correspondiente conjunto de vectores objetivo $\mathbf{Y}_{true} = \mathbf{f}(\mathbf{X}_{true})$ constituye el Frente Pareto óptimo [12].

Usualmente, \mathbf{Y}_{true} no puede ser calculado en forma exacta en tiempos razonables por lo que para fines prácticos, se intentará calcular al menos una buena aproximación al Frente Pareto óptimo, conocido en la literatura como \mathbf{Y}_{known} [26].

3 Problemas ZDT

Los problemas ZDT constituyen un conjunto de seis problemas desarrollados en [30] que contemplan distintas posibilidades y dificultades al considerar problemas multiobjetivo [15]. Todos son problemas de minimización y están estructurados de igual manera sobre tres funciones f_1 , g y h :

$$\begin{array}{ll} \text{Minimizar} & \mathbf{F}(\mathbf{x}) = (f_1(x_1), f_2(\mathbf{x})) \\ \text{sujeta a} & f_2(\mathbf{x}) = g(x_2, \dots, x_n) \cdot \mathbf{h}(f_1(x_1), g(x_2, \dots, x_n)) \\ \text{donde} & \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbf{X} \end{array} \quad (3.1)$$

La función f_1 es una función que depende únicamente de la primera variable de decisión, g es una función de las $n-1$ variables de decisión restantes y los parámetros de h son los valores de las funciones f_1 y g . Las funciones de prueba difieren en estas tres funciones así como en el número de variables n y en los valores que éstas pueden tomar.

El problema ZDT1 tiene 30 variables en el rango $[0,1]$. Su frente de Pareto es convexo. Es el problema más sencillo del conjunto, tiene un frente de Pareto continuo y una distribución uniforme de soluciones a lo largo del frente, donde $\mathbf{x} = (x_1, \dots, x_n)$, $n = 30$ y $x_i \in [0,1]$.

$$f_1(x_1) = x_1; g(x_2, \dots, x_n) = 1 + 9 \cdot \sum_{i=2}^n \frac{x_i}{(n-1)}; h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \quad (3.2)$$

El problema ZDT2 tiene 30 variables en el rango $[0,1]$. Su frente de Pareto es no convexo. La distribución de soluciones a lo largo del frente de Pareto es uniforme, donde $\mathbf{x} = (x_1, \dots, x_n)$, $n = 30$ y $x_i \in [0,1]$.

$$f_1(x_1) = x_1; g(x_2, \dots, x_n) = 1 + 9 \cdot \sum_{i=2}^n \frac{x_i}{(n-1)}; h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2 \quad (3.3)$$

El problema ZDT3 tiene 30 variables en el rango $[0,1]$. Su frente de Pareto es discontinuo. La distribución de soluciones a lo largo del frente es uniforme, donde $\mathbf{x} = (x_1, \dots, x_n)$, $n = 30$ y $x_i \in [0,1]$.

$$\begin{array}{l} f_1(x_1) = x_1; g(x_2, \dots, x_n) = 1 + 9 \cdot \sum_{i=2}^n \frac{x_i}{(n-1)} \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \left(\frac{f_1}{g}\right) \sin(10\pi f_1) \end{array} \quad (3.4)$$

El problema ZDT4 tiene 10 variables en el rango $[0,1]$. Su frente de Pareto es convexo. La complejidad de este problema es que contiene 21^9 frentes Pareto-óptimos locales [30] y, por tanto, prueba a los algoritmos evolutivos con relación a su capacidad de lidiar con la multimodalidad. En este caso, $\mathbf{x} = (x_1, \dots, x_n)$, $n = 10$, $x_1 \in [0,1]$, $x_2, \dots, x_n \in [-5,5]$.

$$\begin{aligned}
f_1(\mathbf{x}_1) &= x_1; \mathbf{g}(\mathbf{x}_2, \dots, \mathbf{x}_n) = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)) \\
h(f_1, \mathbf{g}) &= 1 - \sqrt{\frac{f_1}{\mathbf{g}}}
\end{aligned} \tag{3.5}$$

El problema ZDT5 utiliza funciones booleanas definidas sobre *strings*. Constituye un caso de problema *deceptivo* ya que la forma de la función auxiliar determina que la mayor parte del espacio de búsqueda se concentra cerca de óptimos locales, mientras que el óptimo global se halla relativamente aislado. Aquí $\mathbf{x} = (x_1, \dots, x_n)$, $n = 11$, $x_1 \in \{0,1\}^{30}$, $x_2, \dots, x_n \in \{0,1\}^5$.

$$\begin{aligned}
f_1(\mathbf{x}_1) &= 1 + u(\mathbf{x}_1); \mathbf{g}(\mathbf{x}_2, \dots, \mathbf{x}_n) = \sum_{i=2}^n v(u(\mathbf{x}_i)); h(f_1, \mathbf{g}) = \frac{1}{f_1} \\
u(\mathbf{x}_i) &= \text{"Número de unos en } \mathbf{x}_i \text{"} \\
v(u(\mathbf{x}_i)) &= \begin{cases} 2 + u(\mathbf{x}_i) & \text{si } u(\mathbf{x}_i) < 5 \\ 1 & \text{si } u(\mathbf{x}_i) = 5 \end{cases}
\end{aligned} \tag{3.6}$$

El problema ZDT6 tiene 10 variables en el rango $[0,1]$. Su frente de Pareto es no convexo. 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, donde $\mathbf{x} = (x_1, \dots, x_n)$, $n = 10$, $x_i \in [0,1]$.

$$\begin{aligned}
f_1(\mathbf{x}_1) &= 1 - e^{(-4x_1)} \sin^6(6\pi x_1); \mathbf{g}(\mathbf{x}_2, \dots, \mathbf{x}_n) = 1 + 9 \cdot \left(\sum_{i=2}^n \frac{x_i}{(n-1)} \right)^{\frac{1}{4}} \\
h(f_1, \mathbf{g}) &= 1 - \left(\frac{f_1}{\mathbf{g}} \right)^2
\end{aligned} \tag{3.7}$$

4 Algoritmos en Equipo

El desempeño de un algoritmo se mide generalmente por la calidad de los resultados obtenidos y por la rapidez en la respuesta. En este aspecto, los *Team Algorithms* han demostrado ser muy eficientes superando muchas veces en desempeño a los métodos tradicionales [3, 4, 5, 6]. Debido a estas ventajas en combinar diferentes métodos, el presente trabajo combina siete Algoritmos Evolutivos Multiobjetivos:

- *Multiobjective Genetic Algorithm* (MOGA) [19].
- *Nondominated Sorting Genetic Algorithm* (NSGA) [24].
- *Niched Pareto Genetic Algorithm* (NPGA) [21].
- *Nondominated Sorting Genetic Algorithm II* (NSGA2) [13].
- *Controlled Elitist NSGA II* (CNSGA2) [14].
- *Strength Pareto Evolutionary Algorithm* (SPEA) [29].
- *Strength Pareto Evolutionary Algorithm II* (SPEA2) [28].

Los *Team Algorithm* han surgido como una novedosa herramienta que no solo mejora los tiempos de respuesta del algoritmo sino que se ha comprobado que tiene un efecto sinérgico [3, 4, 5, 6].

El modelo utilizado se denomina *Team Algorithm* of pMOEAs y esta formado por un proceso Coordinador y siete procesos paralelos esclavos en los cuales se pueden ejecutar uno de los siete MOEAs disponibles.

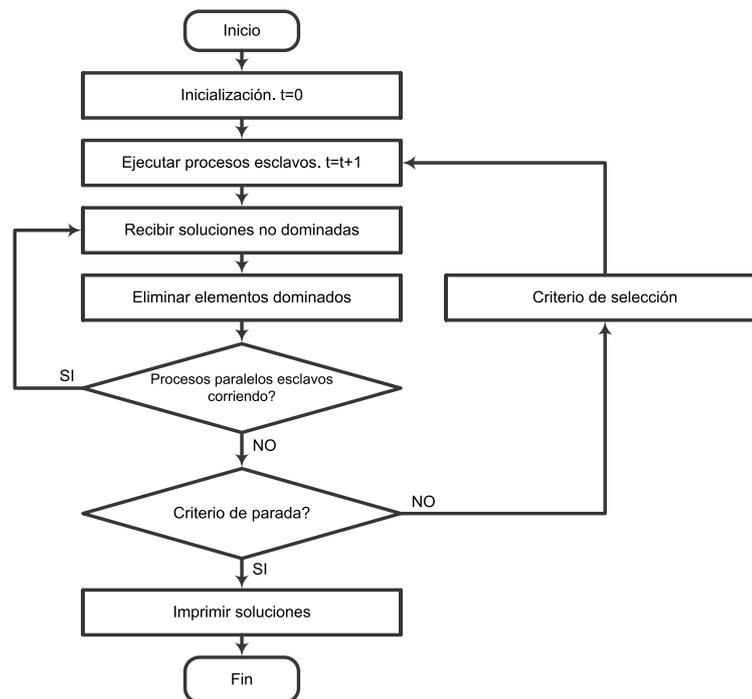


Figura 1. Diagrama de flujo del Coordinador

El diagrama de flujo del proceso Coordinador se observa en la Figura 1. El proceso Coordinador primeramente crea las estructuras para almacenar los resultados provenientes de los distintos procesos MOEA en su población, luego se agrega a un grupo de trabajo e inicia cada uno de los procesos MOEA con sus parámetros específicos. Esto es implementado utilizando primitivas de comunicación de grupo proveídas por librerías de paso de mensajes [23], lo que facilita la comunicación entre los distintos procesos utilizados.

Durante la evolución, cada uno de los MOEAs que interviene en la búsqueda envía a todos los elementos que componen el equipo de trabajo un porcentaje de las mejores soluciones obtenidas. Cuando el Coordinador recibe estas soluciones las almacena en su población. A fin de mantener sólo las mejores soluciones, se eliminan las soluciones cubiertas.

Cuando la cuenta de MOEAs en ejecución es igual a cero, se verifica el criterio de parada, de no ser alcanzado aún se procede a seleccionar los algoritmos esclavos para la siguiente iteración.

El criterio utilizado por el proceso Coordinador para la selección de los algoritmos esclavos es asignarle a cada uno una probabilidad basada en el número de soluciones que introduce en la población Coordinador. La probabilidad de elección del algoritmo Alg_i se denomina p_i y para nuestro modelo en particular se determina por:

$$p_i = \frac{PC_i + 1}{PC_T + 7} \quad (4.1)$$

Siendo PC_i el número de soluciones no dominadas aportadas por el Alg_i a la población del Coordinador y PC_T el número total de soluciones que posee la población del Coordinador.

Una vez seleccionado los algoritmos, se repite el ciclo hasta que el criterio de parada (500 iteraciones) se cumple, entonces el Coordinador imprime las soluciones contenidas en su población y finaliza.

5 Resultados Experimentales

Los siete algoritmos elegidos para este trabajo fueron implementados, para la resolución de los seis problemas de prueba presentadas en la Definición 3.1. Los algoritmos seleccionados se implementaron de acuerdo a la literatura original de referencia y utilizando los parámetros de la Figura 2. Así se tienen 8 conjuntos, 7 de ellos conformados por pMOEAs y 1 por el TA-MOEA. Para el cálculo de las métricas que requieren el conocimiento del frente Pareto óptimo Y_{true} , se utilizó como aproximación el conjunto de soluciones no dominadas Y_{known} obtenido de la unión de todos los resultados experimentales, eliminando las soluciones dominadas.

probabilidad de migración	0,5	probabilidad de mutación	0,01
número máximo de soluciones migradas	10	radio de nicho	0,41
tamaño de la población genética	100	presión de dominancia	10
tamaño de la población externa	100	tasa de reducción	0,7
probabilidad de cruzamiento	0,8	número de iteraciones	1000

Figura 2. Parámetros utilizados en los resultados experimentales.

5.1 Métricas Utilizadas

Razón de Error (E). Reporta la proporción de vectores objetivo en Y_{known} que no son miembros de Y_{true} [26]. Por ello una razón cercana a 1 indica una baja correspondencia entre el frente obtenido y el real; $E = 0$ es lo óptimo. La definición matemática es:

$$E = \frac{\sum_{i=1}^N e_i}{|Y_{known}|_c} \quad e_i = \begin{cases} 0 & \text{si un vector en } Y_{known} \text{ esta también en } Y_{true} \\ 1 & \text{de otro modo} \end{cases} \quad (5.1)$$

Distancia generacional (G). Presentada primeramente en [26]; esta métrica es un valor que representa que tan lejos está Y_{known} de Y_{true} . Se define como:

$$G = \frac{\left(\sum_{i=1}^N d_i^2 \right)^{\frac{1}{2}}}{|Y_{known}|_c} \quad (5.2)$$

donde d_i es la distancia euclidiana (en el espacio objetivo) entre cada vector objetivo $y \in Y_{known}$ y su miembro correspondiente más cercano en el frente Pareto óptimo real Y_{true} . Un valor grande de G indica que Y_{known} está alejado de Y_{true} ; $G = 0$ es la situación ideal.

5.2 Resultados de los experimentos sobre los problemas ZDT

Basados en los valores óptimos de las métricas utilizadas se procede a clasificar los 8 conjuntos de algoritmos.

Conjunto	ZDT1	Conjunto	ZDT2	Conjunto	ZDT3
1 CNSGA2	0,0185418	1 SPEA	0,06277056	1 SPEA2	0,02111801
2 TAMOEa	0,03262335	2 NSGA2	0,09090909	2 TAMOEa	0,17653027
3 SPEA2	0,06945917	3 TAMOEa	0,15606242	3 SPEA	0,19920319
4 SPEA	0,18105209	4 SPEA2	0,20116279	4 NSGA2	0,20762155
5 NSGA2	0,1823913	5 CNSGA2	0,20697954	5 CNSGA2	0,21927791
6 NSGA	0,99333333	6 NSGA	0,98818898	6 NSGA	0,89332728
7 NPGA	0,997669	7 NPGA	0,99647266	7 NPGA	0,95016611
8 MOGA	1	8 MOGA	0,99799197	8 MOGA	0,98551449

Figura 3. Resultados de la métrica E sobre los problemas ZDT1, ZDT2 y ZDT3

En la Figura 3 se observa las posiciones en los problemas ZDT1, ZDT2 y ZDT3 utilizando la métrica E, como vemos para el problema ZDT1 el mejor fue el CNSGA2 quedando en segundo lugar el TA-MOEa, en el problema ZDT2 quedó primero el SPEA mientras el TA-MOEa obtiene el tercer lugar, en el problema ZDT3 el SPEA2 obtuvo el primer lugar y el TA-MOEa quedó segundo.

Conjunto	ZDT4	Conjunto	ZDT5	Conjunto	ZDT6
1 CNSGA2	0,11342812	1 SPEA2	0,25995316	1 TAMOEa	0,06883365
2 SPEA	0,1618705	2 TAMOEa	0,27017763	2 SPEA	0,12797075
3 TAMOEa	0,18345324	3 CNSGA2	0,27649208	3 NSGA2	0,19253731
4 NSGA2	0,23017903	4 SPEA	0,31393298	4 SPEA2	0,33478261
5 SPEA2	0,24914286	5 NSGA2	0,31652989	5 CNSGA2	0,35118823
6 NSGA	0,96350365	6 NSGA	0,96020761	6 MOGA	0,94278607
7 NPGA	0,97968684	7 NPGA	0,97586207	7 NPGA	0,95189003
8 MOGA	0,99001664	8 MOGA	0,98684829	8 NSGA	0,96020761

Figura 4. Resultados de la métrica E sobre los problemas ZDT4, ZDT5 y ZDT6

En la Figura 4 se observa las posiciones en los problemas ZDT4, ZDT5 y ZDT6 utilizando la métrica E, como vemos para el problema ZDT4 el mejor fue el CNSGA2 y el TA-MOEa se posiciona en el tercer lugar, en el problema ZDT5 quedó primero el SPEA2 mientras el TA-MOEa obtiene el segundo lugar, en el problema ZDT6 el TA-MOEa queda primero.

Conjunto		ZDT1
1	TAMOEa	0,00015946
2	NSGA2	0,00014576
3	SPEA2	0,00024714
4	SPEA	0,00025936
5	CNSGA2	0,00034837
6	NSGA	0,05687313
7	NPGA	0,28573518
8	MOGA	0,39595684

Conjunto		ZDT2
1	CNSGA2	0,00023762
2	TAMOEa	0,00026216
3	NSGA2	0,00026633
4	SPEA	0,00027105
5	SPEA2	0,00036343
6	NSGA	0,02506101
7	NPGA	0,03055195
8	MOGA	0,05624174

Conjunto		ZDT3
1	TAMOEa	0,00098158
2	SPEA2	0,00147351
3	NSGA2	0,00164381
4	SPEA	0,00797476
5	NSGA	0,01033311
6	CNSGA2	0,01544077
7	NPGA	0,13675506
8	MOGA	0,14065879

Figura 5. Resultados de la métrica G sobre los problemas ZDT1, ZDT2 y ZDT3

En la Figura 5 se observa las posiciones en los problemas ZDT1, ZDT2 y ZDT3 utilizando la métrica G, como vemos para el problema ZDT1 el mejor fue el TAMOEa, en el problema ZDT2 quedó primero el CNSGA2 mientras el TAMOEa obtiene el segundo lugar, en el problema ZDT3 el TAMOEa vuelve a estar primero.

Conjunto		ZDT4
1	TAMOEa	0,01606166
2	CNSGA2	0,02096637
3	NSGA2	0,34635745
4	SPEA	0,69991634
5	SPEA2	0,72846653
6	NSGA	0,8657436
7	NPGA	0,86594995
8	MOGA	0,89584308

Conjunto		ZDT5
1	NSGA2	0,00185052
2	TAMOEa	0,00293943
3	CNSGA2	0,00325496
4	NSGA	0,00904485
5	SPEA2	0,03014446
6	SPEA	0,03455024
7	MOGA	0,09824596
8	NPGA	0,10006193

Conjunto		ZDT6
1	SPEA	0,00215552
2	SPEA2	0,0043351
3	TAMOEa	0,01203456
4	CNSGA2	0,06705503
5	NSGA2	0,08066857
6	NSGA	0,09106741
7	NPGA	0,10205026
8	MOGA	0,20012904

Figura 6. Resultados de la métrica G sobre los problemas ZDT4, ZDT5 y ZDT6

En la Figura 6 es utilizada la métrica G y vemos que para el problema ZDT4 el mejor fue el TAMOEa, en el problema ZDT5 quedó primero el NSGA2 mientras el TAMOEa obtiene el segundo lugar, en el problema ZDT6 el primer lugar lo ocupa el SPEA, quedando el TAMOEa en tercera posición.

Razón de Error (E)			Distancia generacional (G)		
	Conjunto	PROMEDIO		Conjunto	PROMEDIO
1	TAMOEa	0,14794676	1	TAMOEa	0,00540648
2	SPEA	0,17446668	2	CNSGA2	0,01788385
3	SPEA2	0,18926977	3	NSGA2	0,07182207
4	CNSGA2	0,19765128	4	SPEA	0,12418788
5	NSGA2	0,20336136	5	SPEA2	0,12750503
6	NSGA	0,95979474	6	NSGA	0,17635385
7	NPGA	0,97529112	7	NPGA	0,25351739
8	MOGA	0,98385957	8	MOGA	0,29784591

Figura 7. Posiciones según la métrica E y la métrica G

Haciendo un promedio del conjunto global de problemas tratados, el TAMOEa en la métrica E obtuvo el primer lugar, seguido del SPEA y el SPEA2, esto se aprecia en la Figura 7 a la izquierda, a la derecha se observa que el primer lugar de nuevo es ocupado por el TAMOEa, esta vez utilizando la métrica G, seguido del CNSGA2 y el NSGA2.

Concluyendo se puede afirmar que el TA-MOEA presenta soluciones más robustas ante un conjunto de problemas con características y dificultades diferentes, convirtiéndose en una opción válida para la resolución de problemas de optimización multiobjetivo.

6 Conclusiones y Trabajos Futuros

A partir de los distintos resultados experimentales obtenidos, se han propuesto conclusiones parciales sobre la implementación del TA-MOEA con criterio de selección probabilístico en la resolución de cada uno de los problemas considerados. Con estos, es posible arribar a las siguientes conclusiones finales de este trabajo:

- El diseño e implementación de un equipo probabilístico de algoritmos evolutivos multiobjetivo paralelos es un problema complejo. Existen varias decisiones que tomar. El rango de éstas va desde el tipo de plataforma paralela en que se realizará la implementación, hasta la determinación de diversos parámetros de configuración y migración de datos.
- Teniendo en cuenta las métricas utilizadas para medir la calidad del conjunto aproximación, se puede establecer que el TA-MOEA ha sido el que logró, en promedio, un mejor desempeño. Esta diferencia se extiende a toda la gama de problemas resueltos en este trabajo.
- Al utilizar el TA-MOEA, se extiende el espacio de búsqueda. La recepción de elementos provenientes de distintos algoritmos introduce información genética en forma aleatoria que es útil para la obtención de mejores soluciones.

En resumen, a partir de los distintos resultados presentados queda claro que la utilización del Equipo Probabilístico de Algoritmos Evolutivos Multiobjetivo Paralelos es adecuado para la búsqueda de soluciones en espacios de búsqueda complejos y de alta dimensionalidad.

De forma a continuar con el trabajo iniciado, los siguientes tópicos son propuestos como trabajos futuros:

- Modelado matemático de los algoritmos presentados de manera a hallar la complejidad de los mismos
- Implementación de TA-MOEA que incorporen criterios de selección más refinados y dinámicos
- Estudio e implementación de nuevos problemas de ingeniería que sirvan de base comparativa y experimental para el TA-MOEA.

Referencias

1. T. Bäck, D. B. Fogel, y Z. Michalewicz, editors. "Handbook of Evolutionary Computation". Institute of Physics Publishing and Oxford University Press, 1997.
2. B. Barán, J. Vallejos, R. Ramos, y U. Fernández. "Multi-objective reactive power compensation". En 2001 IEEE/PES Transmission and Distribution Conference and Exposition, volume 1, págs. 97–101. IEEE, 2001.

3. B. Barán, E. Kaszkurewicz y D. M. Falcão. "Team Algorithms in Distributed Load Flow Computations". IEE Proceeding on Generation, Transmission and Distribution, Vol. 142, No. 6, pg. 583-588, noviembre 1995. Londres - Gran Bretaña.
4. B. Barán, N. Cáceres y E. Chaparro. "Reducción del Tiempo de Búsqueda utilizando una Combinación de Algoritmos Genéticos y Métodos Numéricos". XV International Conference of the Chilean Computer Science Society. Arica - Chile, 1995.
5. B. Barán, E. Kaszkurewicz y A. Bhaya. "Parallel Asynchronous Team Algorithms: Convergence and Performance Analysis". IEEE Transactions on Parallel & Distributed Systems, Vol. 7, No. 7, pg. 677-688, julio 1996. Estados Unidos.
6. B. Barán, E. Chaparro y N. Cáceres. "A-Teams en la Optimización del Caudal Turbinado de una Represa Hidroeléctrica". IBERAMIA-98, Lisboa-Portugal. 1998.
7. B. Barán y F. Laufer. "Topological Optimization of Reliable Networks using A-Teams". Systemics, Cybernetics and Informatics SCI99, Orlando - Florida, Estados Unidos. 1999.
8. C. A. Coello Coello y G. Toscano Pulido. "A Micro-Genetic Algorithm for Multiobjective Optimization". En E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, y D. Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, págs. 126-140. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
9. C. A. Coello Coello y C. E. Mariano Romero. "Evolutionary Algorithms and Multiple Objective Optimization". En M. Ehrgott y X. Gandibleux, editors, Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys, págs. 277-331. Kluwer Academic Publishers, Boston, 2002.
10. C. A. Coello Coello. "An Updated Survey of GA-Based Multiobjective Optimization Techniques". Technical Report Lania-RD-98-08, Laboratorio Nacional de Informática Avanzada (LANIA), Xalapa, Veracruz, México, December 1998.
11. C. A. Coello Coello. "Constraint handling through a multiobjective optimization technique". En A. S. Wu, editor, Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program, págs. 117-118. Orlando, Florida, Julio 1999.
12. J. Crichigno y B. Barán, "Multiobjective Multicast Routing Algorithm". IEEE ICT'2004, Ceará, Brasil, 2004.
13. K. Deb, S. Agrawal, A. Pratab, y T. Meyarivan. "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II". KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.
14. K. Deb y T. Goel. "Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence". En E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, y D. Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, págs. 67-81. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
15. K. Deb. "Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems", Technical Report CI-49/98, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany, 1998.
16. K. Deb. "Evolutionary algorithms for multi-criterion optimization in engineering design". En K. Miettinen, M. M. Mäkelä, P. Neittaanmäki, y J. Periaux, editors, Evolutionary Algorithms in Engineering and Computer Science, págs. 135-161, Chichester, UK, 1999. JohnWiley & Sons, Ltd.
17. N. M. Duarte, A. E. Ruano, C. M. Fonseca, y P. J. Fleming. "Accelerating Multi-Objective Control System Design Using a Neuro-Genetic Approach". En 2000 Congress on Evolutionary Computation, volume 1, págs. 392-397, Piscataway, New Jersey, Julio 2000. IEEE Service Center.

18. S. Duarte y B. Barán. "Multiobjective Network Design Optimisation Using Parallel Evolutionary Algorithms". En XXVII Conferencia Latinoamericana de Informática CLEI-2001, Mérida, Venezuela, 2001.
19. C. M. Fonseca y P. J. Fleming. "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization". En S. Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, págs. 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.
20. D. E. Goldberg. "Genetic Algorithms in Search, Optimization and Machine Learning". Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
21. J. Horn y N. Nafpliotis. "Multiobjective Optimization using the Niche Pareto Genetic Algorithm". Technical Report IlliGAL Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.
22. K. Miettien. "Some methods for nonlinear multi-objective optimization". En E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, y D. Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
23. A. G. et al. "PVM:Paralell Virtual machine - A user's guide and Tutorial for Networked parallel Computing". M.I.T. press, Cambridge, MA, 1994.
24. N. Srinivas y K. Deb. "Multiobjective optimization using nondominated sorting in genetic algorithms". Technical report, Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, India, 1993.
25. D. A. van Veldhuizen, J. B. Zydallis, y G. B. Lamont. "Issues in Parallelizing Multiobjective Evolutionary Algorithms for Real World Applications". En Proceedings of the 17th ACM Symposium on Applied Computing, págs. 595–602, Madrid, Spain, 2002. ACM Press.
26. D. A. van Veldhuizen. Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Mayo 1999.
27. E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, y D. Cornde, editors. "Proceedings of the First International conference on EMOO, 2001", Berlin, Germany, Marzo 2001. Springer-Verlag.
28. E. Zitzler, M. Laumanns, y L. Thiele. "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", in K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty (eds.) EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, pp. 95--100, Athens, Greece, 2002.
29. E. Zitzler y L. Thiele. "An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach". Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Mayo 1998.
30. E. Zitzler, K. Deb, y L. Thiele. "Comparison of Multiobjective Evolutionary Algorithms on Test Functions of Different Difficulty". En A. S. Wu, editor, Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program, págs. 121–122, Orlando, Florida, Julio 1999.