

ALGORITMO MULTI OBJETIVO NSGA-II APLICADO AL PROBLEMA DE LA MOCHILA.**Multiobjective knapsack problem using NSGA-II algorithm****RESUMEN**

Este trabajo presenta la filosofía del algoritmo multiobjetivo elitista NSGA-II, explicando su esquema de funcionamiento y los mecanismos especiales que permiten la preservación y la evolución de soluciones Pareto-óptimas. Este algoritmo se aplica sobre el problema de optimización clásico correspondiente al problema de la mochila adaptado para optimización multiobjetivo. También se desarrolla una propuesta alternativa para ser comparada con el esquema básico del NSGA-II.

PALABRAS CLAVES: Optimización Multiobjetivo, Optimalidad de Pareto, Problema de la Mochila.

ABSTRACT

This paper presents the basic concepts of NSGA-II elitist multiobjective algorithm, showing its working principles and the special mechanisms that lead to preservation and evolution towards Pareto-optimum solutions. This algorithm is applied on the knapsack classical optimization problem adapted for multiobjective optimization. An alternative approach is also proposed, aiming to be compared with the basic NSGA-II procedure.

KEYWORDS: Dominance, Pareto optimality, knapsack problem, multiobjective optimization.

1. INTRODUCCIÓN

El problema clásico de la mochila (Knapsack problem) bajo el esquema mono-objetivo tiene como fin determinar cuáles elementos deben ser adicionados en una mochila con el propósito de obtener el máximo beneficio nutricional pero garantizando que no se exceda el volumen o capacidad de la mochila o recipiente de almacenamiento. Las variables son binarias y tanto las restricciones como la función objetivo son ampliamente desarrolladas por Gallego *Et al* y ha sido tratado de manera amplia dentro del Grupo de Planeamiento de Sistemas Eléctricos de la Universidad Tecnológica de Pereira [1].

En la literatura especializada se documentan y exponen diversos tamaños para este problema específico y debido a que se trata de un problema de tipo NP, es sabido que el tamaño y complejidad del problema aumenta a medida que aumenta el número de variables [2-3]. Adicionalmente, el problema puede ser modificado considerando varias mochilas con diferentes capacidades y modelado mediante un esquema multiobjetivo por lo que no existe una solución óptima sino un conjunto de posibles combinaciones que permitan diferentes beneficios al introducir unos u otros elementos entre las diferentes mochilas respetando para cada una la capacidad de almacenamiento o volumen. Así, la decisión final está encaminada a almacenar la mayor cantidad de elementos posibles en las diferentes mochilas

con el objetivo de alcanzar el máximo beneficio combinado con la totalidad de elementos almacenados.

En este trabajo se implementa un esquema multiobjetivo (MO) *Non Dominated Sorting Genetic Algorithm* (NSGA-II). Adicionalmente, se adapta el esquema del NSGA-II con base en la metodología del algoritmo especializado de Chu & Beasley con el fin de comparar los resultados obtenidos mediante las dos metodologías multiobjetivo propuestas [6].

En el cuerpo del artículo se efectúan consideraciones de la filosofía general de la optimización MO en la sección 2. En la sección 3 se presenta el algoritmo NSGA-II con sus variaciones y algunas formas de efectuar las mutaciones. Ya en la sección 4 y 5 se muestra la implementación de la metodología junto con sus resultados y pruebas [5].

2. OPTIMIZACIÓN MULTI OBJETIVO

Cuando en un problema de optimización se tienen varias funciones objetivo, la tarea de encontrar una o más soluciones óptimas se denominada optimización multiobjetivo o toma de decisiones multicriterio. Bajo este enfoque la atención no se puede centrar solo en uno de los objetivos dejando a un lado los otros. Diferentes soluciones pueden producir soluciones *trade off* (escenarios en conflicto) entre diferentes objetivos. Una solución que es óptima con respecto a un objetivo puede

CARLOS ADRIÁN CORREA FLÓREZ

Ingeniero Electricista
Profesional Dirección Técnica de
Gestión de Energía
Superintendencia de Servicios
Públicos Domiciliarios
correa@utp.edu.co

RICARDO ANDRÉS BOLAÑOS

Ingeniero Electricista
Analista Programación Operación
XM – Compañía de Expertos en
Mercados S.A. “ESP”
rbolanos@utp.edu.co

ALEXÁNDER MOLINA CABRERA

Ingeniero Electricista, M.Sc.
Profesor Auxiliar
Universidad Tecnológica de Pereira
almo@utp.edu.co

no serlo para el resto, por lo tanto sería impropio escoger dicha solución como óptima del problema y se crea la necesidad de establecer un compromiso entre los objetivos.

Los problemas con múltiples objetivos se encuentran presentes en la mayoría de las disciplinas y su solución siempre ha representado un reto para los investigadores. La principal motivación para el uso de novedosos algoritmos, como los de la familia de los evolutivos, en la solución de problemas de este tipo radica en que al estar basados en población para realizar búsqueda de soluciones, éstos pueden retornar un conjunto de soluciones de buena calidad, hecho que es de alta importancia en este tipo de enfoques.

2.1 Conceptos básicos acerca de la optimización multiobjetivo

El modelo matemático general de un problema de optimización MO se presenta en la ecuación 1.

$$\begin{aligned} \min/\max : f_m(x), \quad m=1,2,\dots,M \\ \text{s.a.} \\ g_j(x) \geq 0, \quad j=1,2,\dots,J \\ h_k(x) = 0, \quad k=1,2,\dots,K \\ x_i^l \leq x \leq x_i^u, \quad i=1,2,\dots,n \end{aligned} \quad (1)$$

Para resolver el anterior problema se deben definir algunos criterios con el fin de determinar cuáles soluciones se consideran de buena calidad y cuáles no. Para lo anterior se introduce el concepto de dominancia, que coadyuva en el proceso de clasificar las diferentes soluciones y de encontrar buenas alternativas teniendo en cuenta la presencia y la cuantificación de los M objetivos del problema [2].

Se dice que una solución $x^{(1)}$ domina a otra solución $x^{(2)}$ si se cumplen las siguientes condiciones:

1. La solución $x^{(1)}$ no es peor que $x^{(2)}$ en todos los objetivos.
2. La solución $x^{(1)}$ es estrictamente mejor que $x^{(2)}$ en por lo menos un objetivo.

Si alguna de las condiciones es violada, la solución $x^{(1)}$ no domina la solución $x^{(2)}$.

El concepto de dominancia se puede extender para encontrar un conjunto de soluciones no dominadas dentro de una población. Si se considera una población de N soluciones cada una con M valores de funciones objetivo, se usa el siguiente procedimiento para encontrar el conjunto no dominado de soluciones

1. Hacer $i=1$.

2. Para todos los $j \neq i$, comparar las soluciones $x^{(i)}$ y $x^{(j)}$ para determinar dominancia, usando las 2 condiciones mencionadas anteriormente para todos los M objetivos.
3. Si para algún j , $x^{(i)}$ es dominado por $x^{(j)}$, marcar $x^{(i)}$ como dominado. Incrementar i en uno e ir al paso 2.
4. Si todas las soluciones (es decir, cuando se llega a $i=j$) en el conjunto son consideradas, ir al paso 5; de lo contrario incrementar i en uno e ir al paso 2.
5. Todas las soluciones que no son marcadas como dominadas son soluciones no-dominadas.

Cuando se realiza un análisis de dominancia entre dos soluciones y se encuentra que la primera condición de dominancia no se cumple para ninguna de las dos soluciones, no se puede concluir acerca de la dominancia de una con respecto a otra. Cuando esto sucede, se dice que las soluciones son no-dominadas. De esta forma, si se tiene un conjunto finito de soluciones y se realiza una comparación de todos los pares posibles, al final se va a tener un conjunto de soluciones que son no dominadas entre sí y dicho conjunto tiene la propiedad de dominar al resto de soluciones que no pertenezcan a él. Por lo tanto, si se tiene un conjunto de soluciones P , el conjunto no dominado de soluciones P' lo forman aquellas que no son dominadas por ningún miembro del conjunto P , a este conjunto se le llama frente de Pareto. Cuando el conjunto P es el espacio de búsqueda, el conjunto no dominado P' resultante es denominado frente óptimo de Pareto [2-3].

En la figura 1 se muestran los conjuntos óptimos de Pareto para diferentes escenarios con dos objetivos y para el mismo espacio de soluciones. En cualquier caso el óptimo de Pareto siempre está compuesto por soluciones ubicadas en el borde de la región factible del espacio de soluciones.

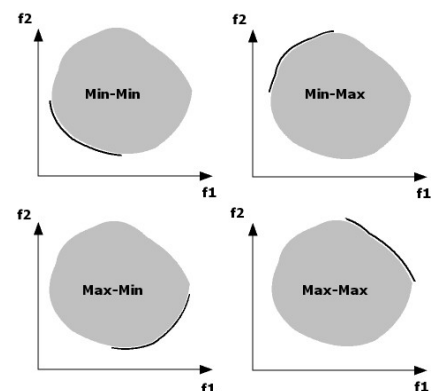


Figura 1. Frentes Óptimos de Pareto para el mismo espacio de soluciones.

Una población puede ser clasificada en diferentes niveles de no dominancia. Cuando el procedimiento de 5 pasos descrito para encontrar el conjunto no dominado es aplicado por primera vez en una población, el conjunto resultante es el conjunto no dominado de mejor nivel

(primer frente de Pareto). Para obtener otras clasificaciones, estas soluciones no dominadas pueden ser temporalmente omitidas del conjunto original para aplicar nuevamente el procedimiento. Lo que resulta es un conjunto no dominado de soluciones de segundo nivel (o el siguiente mejor nivel). Este proceso puede ser repetido hasta que todos los miembros de la población sean clasificados dentro de un nivel o frente de Pareto.

Uno de los objetivos de la optimización para problemas MO es encontrar tantas soluciones Pareto-óptimas como sea posible. Dado que los algoritmos evolutivos (EA por sus siglas en inglés) utilizan una población de soluciones, se pueden hacer algunos cambios a estos algoritmos de tal forma que tras su ejecución se puedan encontrar soluciones no-dominadas de buena calidad. Con el objeto de desarrollar la anterior idea se describe a continuación un algoritmo elitista para solucionar problemas MO.

3. ALGORITMO NSGA-II

El NSGA-II (*Elitist Non-Dominated Sorting Genetic Algorithm*) fue propuesto por Deb y sus estudiantes en el año 2000 [3]. En este algoritmo, la población descendiente Q_t (tamaño N) es creada en primera instancia usando la población de padres P_t (tamaño N). Después de esto, las dos poblaciones son combinadas para formar R_t de tamaño $2N$. Después de lo anterior, mediante un ordenamiento no dominado se clasifica la población R_t en diferentes frentes de Pareto. Aunque esto requiere un mayor esfuerzo, se justifica por el hecho de permitir una verificación global de dominancia entre la población de padres y descendientes. Una vez el proceso de ordenamiento no dominado ha finalizado, la nueva población es generada a partir de las configuraciones de los frentes no dominados. Esta nueva población empieza a ser construida con el mejor frente no dominado (F_1), continúa con las soluciones del segundo frente (F_2), tercero (F_3) y así sucesivamente. Como la población R_t es de tamaño $2N$, y solamente existen N configuraciones que conforman la población descendiente, no todas la configuraciones de los frentes pertenecientes a la población R_t podrán ser acomodados en la nueva población. Aquellos frentes que no pueden ser acomodados desaparecen [3].

Cuando se está considerando el último frente, la soluciones que hacen parte de éste pueden exceder las restantes por acomodar en la población descendiente, dicha situación se ilustra en la figura 2. En este caso resulta útil emplear alguna estrategia que permita seleccionar las configuraciones situadas en un área poco poblada (alejada de otras soluciones) para llenar las posiciones restantes de la población descendiente a cambio de optar por escoger configuraciones de forma aleatoria.

Lo anterior es poco relevante en los primeros ciclos generacionales del algoritmo, ya que en esta etapa existen

muchos frentes que sobreviven hacia la siguiente generación, pero a medida que el proceso avanza, muchas configuraciones pasan a ser parte del primer frente inclusive haciendo que dicho frente tenga más de N individuos, por lo que se hace importante que las configuraciones no rechazadas sean de buena calidad y escogidas mediante una metodología que garantice la diversidad. La idea es que siempre se promuevan las configuraciones que aseguren diversidad dentro del mismo frente de Pareto. Cuando la población en su totalidad converge al frente de Pareto óptimo, el algoritmo asegura que las soluciones estén distanciadas una de otra.

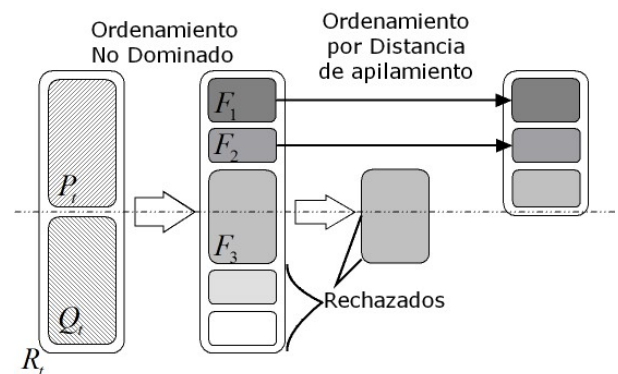


Figura 2. Diagrama esquemático del mecanismo de promoción de individuos del NSGA-II

En el algoritmo NSGA-II inicialmente se crea una población (aleatoria o mediante una técnica de inicialización) de padres P_0 . La población se ordena de acuerdo a los niveles de no dominancia (ordenamiento de los frentes de Pareto, F_1, F_2, \dots). A cada solución se le asigna una función *fitness* de acuerdo a su nivel de no dominancia (1 es el mejor nivel) y se entiende que durante el proceso de selección por torneo (empleando un operador de torneo para apilamiento, descrito posteriormente), el cruzamiento y la mutación son utilizados para crear la población de descendientes Q_0 de tamaño N . Los principales pasos del algoritmo NSGA-II son descritos a continuación:

1. Combinar las poblaciones de padres y descendientes para crear $R_t = P_t \cup Q_t$. Realizar el ordenamiento no dominado a R_t e identificar los frentes $F_i, i=1,2,\dots,etc$.
2. Hacer $P_{t+1} = \emptyset$, e $i=1$. Mientras $|P_{t+1}| + |F_i| < N$ hacer $|P_{t+1}| = |P_{t+1}| \cup |F_i|$ e $i=i+1$.
3. Realizar ordenamiento por apilamiento ($F_i' < C$ descrito posteriormente) e incluir en P_t las $N - |P_{t+1}|$ soluciones más esparcidas usando los valores de distancia de apilamiento asociadas al frente F_i .

4. Crear la población descendiente Q_{i+1} a partir de P_{i+1} usando selección por torneo para apilamiento, cruzamiento y mutación.

3.1 Selección por torneo para apilamiento

Este operador ($F_i' < C$) compara dos soluciones y elige un ganador del torneo. Se considera que una solución tiene un rango de no dominancia asociado (r_i) y una distancia de apilamiento (d_i). La distancia de apilamiento d_i de una solución i es la medida del espacio de búsqueda alrededor de i que no está ocupado por otra solución en la población. Por medio de estos dos atributos se puede encontrar la mejor configuración a través del operador de selección por torneo para apilamiento como aquella que tiene mejor rango de no-dominancia. En el caso de estar situadas en el mismo frente de Pareto la ganadora del torneo es aquella que tiene mejor (más alta) distancia de apilamiento. La distancia de apilamiento $d_{I_j^m}$ para cada solución j , según un índice I , es determinada algorítmicamente, haciendo uso de la ecuación 2.

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{\max} - f_m^{\min}} \quad (2)$$

donde f_m^{\max} , f_m^{\min} son el valor máximo y mínimo de la función objetivo m , $f_m^{(I_{j+1}^m)}$, $f_m^{(I_{j-1}^m)}$ son las soluciones vecinas a la configuración j para cada una de las funciones objetivo m . Las distancias consideran todas las funciones objetivo y se asigna el valor de infinito a las soluciones extremas del frente de Pareto considerado. Por ser éstas las que cuentan con el mejor valor de una de las funciones objetivo del frente, la distancia resultante es la suma de las distancias en cada una de las direcciones de las funciones objetivo del problema.

De esta manera el algoritmo utiliza los operadores genéticos básicos y promueve hacia el siguiente ciclo generacional las configuraciones que ocupen los mejores frentes, y las más diversas, a través de las distancias de apilamiento.

3.2 Algoritmo NSGA-II modificado

Como propuesta alternativa, se plantea el algoritmo NSGA-II, incluyendo una variante para reducir el tiempo computacional. Como se ha visto, el NSGA-II crea una población de descendientes de tamaño N , para después ser combinada con la población de padres P , lo que implica que por cada ciclo generacional se deben realizar por lo menos N evaluaciones de la función objetivo. Para reducir este inconveniente se puede importar la lógica del Algoritmo Genético de Chu-Beasley (CBGA) que solamente introduce un individuo a la población por cada ciclo generacional, disminuyendo sustancialmente el esfuerzo computacional [5]. Dicha lógica implementada sobre el NSGA-II consiste en que la población

descendiente es reemplazada por un solo individuo, resultante del operador cruzamiento, que es posteriormente sometido a mutación para ser incluido en la población total R . Para este caso la población R tendrá tamaño $N+1$ (al contrario del planteamiento original en donde el tamaño es $2N$) y de cumplir con las condiciones necesarias de optimalidad de Pareto el individuo será promovido hacia el siguiente ciclo generacional como miembro de la población P . Los cambios que sufren los operadores bajo este enfoque son los siguientes:

3.2.1 Selección

Se llevan a cabo dos torneos. Estos torneos se realizan utilizando la lógica del NSGA-II, es decir, en cada torneo se selecciona la configuración que esté ubicada en el mejor frente de Pareto y si las configuraciones participantes de dicho torneo están en el mismo frente se utiliza la mejor distancia de apilamiento para determinar la configuración ganadora. Este proceso es denominado selección por torneo para apilamiento y el resultado de su aplicación son dos configuraciones padres.

3.2.2 Cruzamiento y Mutación

Este operador se emplea como es usual y se utilizan los dos padres obtenidos del operador selección. De los dos descendientes generados se escoge uno de la siguiente manera: Los dos hijos son introducidos temporalmente en la población, de tal manera que la población queda con $N+2$ individuos y después se realiza el ordenamiento no dominado para clasificar las configuraciones de acuerdo al frente al que pertenezcan y además se asignan las respectivas distancias de apilamiento. A partir de los valores *fitness* asignados anteriormente se elige el mejor hijo, es decir, el hijo que se encuentre en el mejor frente no dominado o de ser necesario, el que posea menor distancia de apilamiento. De igual manera sucede con el operador mutación, en donde la configuración obtenida después de la mutación se compara con la original para así decidir sobre cual elegir.

Como se observa, el esquema propuesto es en realidad un híbrido del algoritmo NSGA-II y el CBGA, además, para aceptar una configuración dentro de la población, dicha solución debe cumplir con unas características especiales. Si la solución se encuentra dentro del primer frente de Pareto, es aceptada dentro de la población. Por el contrario, si la solución pertenece a un frente diferente al primero, para que ésta sea aceptada debe ser diversa con respecto a los demás miembros de la población, es decir, debe diferenciarse en un número determinado de bits del resto de configuraciones.

Otro aspecto importante de este enfoque consiste en que se incrementa el número de veces que se realiza el proceso de clasificación de las configuraciones en frentes de Pareto por cada ciclo generacional.

4. APLICACIÓN DEL NSGA-II AL PROBLEMA DE LA MOCHILA 0/1 MULTIOBJETIVO

Aunque este problema es de fácil comprensión, sigue siendo considerado como un problema de difícil solución por pertenecer al grupo de los problemas NP completos (*Non-Polynomial Hard*). Este problema ha sido utilizado para mostrar resultados de diferentes EA's y algoritmos MO [2-7].

Generalmente, el problema de la mochila consiste en una serie de productos o ítems con un peso y un beneficio asociado a cada uno de ellos, y un límite de capacidad para la mochila. La tarea consiste en hallar el subconjunto de ítems que maximicen el beneficio total y que puedan ser ubicados en la mochila sin exceder su límite de capacidad o peso. El problema mono-objetivo puede ser extendido directamente al caso multiobjetivo si se considera un número determinado de mochilas. De esta forma, el problema es definido así: dado un conjunto de m ítems y un conjunto de n mochilas, donde: $p_{i,j}$ es el beneficio del ítem j en la mochila i ; $w_{i,j}$ es el peso del ítem j en la mochila i ; y c_i es la capacidad de la mochila i , encontrar un vector $x=(x_1,x_2,\dots,x_m) \in \{0,1\}^m$, tal que

$$\forall i \in \{1,2,\dots,n\} : \sum_{j=1}^m w_{i,j} \cdot x_j \leq c_i \tag{2}$$

Y para el cual $f(x)=(f_1(x),f_2(x),\dots,f_n(x))$ es máximo, donde

$$f_i(x) = \sum_{j=1}^m p_{i,j} \cdot x_j \tag{3}$$

y $x_j=1$ si el ítem j es seleccionado.

Para este problema se utiliza la codificación binaria, en donde cada cromosoma representa un conjunto de ítems para ser ubicados en las mochilas. Como muchas configuraciones pueden llevar a soluciones que excedan la capacidad de una o varias mochilas, se aplica un método para corregir dichas infactibilidades [2]. Este método remueve ítems de una solución paso a paso hasta que todas las restricciones de capacidad sean satisfechas. El orden en que los ítems son removidos es determinado por la máxima relación beneficio/peso por ítem. Así, para el ítem j la máxima relación q_j es dada por:

$$q_j = \max_{i=1}^n \left\{ \frac{p_{i,j}}{w_{i,j}} \right\} \tag{4}$$

Los ítems son considerados en orden ascendente del valor q_j , es decir, aquellos con menor beneficio por unidad de peso son removidos primero, de esta manera se trata de alcanzar la factibilidad disminuyendo el beneficio tan poco como sea posible.

Este problema es resuelto para el caso de prueba de 100 ítems y dos mochilas (denominado 2-100 en [2]), lo que corresponde a un problema de 2 objetivos. Tanto el esquema básico del algoritmo NSGA-II como el modificado fueron implementados para este caso.

5. PRUEBAS Y RESULTADOS

Con el fin de establecer una metodología válida para establecer comparaciones se realizaron múltiples pruebas para cada algoritmo teniendo como criterio de parada un número determinado de evaluaciones de la función objetivo, ya que la utilización de ciclos generacionales puede llevar a establecer medidas desiguales por la naturaleza de los dos enfoques tratados, debido a que en un ciclo generacional del algoritmo NSGA-II básico se realizan muchas más evaluaciones de la función objetivo en comparación con el esquema modificado. De esta forma, después de correr ambos algoritmos por un número de evaluaciones de función objetivo, se comparan a través de la determinación del número de individuos del frente óptimo que pudieron alcanzar.

Para el enfoque básico se utilizó una población de 150 individuos, tasa de mutación de 0.01 y tasa de cruzamiento de 0.95 mientras que para el esquema modificado se utilizó una población de 80 individuos, tasa de mutación y cruzamiento de 0.02 y 1 respectivamente.

El límite de evaluaciones de la función objetivo se fija en 1 millón. Después de correr el esquema básico con los parámetros ya mencionados se encuentra que en promedio el número de individuos encontrados que pertenecen al frente óptimo es de 36, en donde se encuentra una buena distribución de las soluciones encontradas gracias a la cuantificación de las distancias de apilamiento y su inclusión en el proceso de selección. En la figura 3 se muestra un frente de Pareto típico para este problema. Además, se presenta una relación comparable entre el tiempo de evaluación de la función objetivo y el gastado en el ordenamiento no dominado.

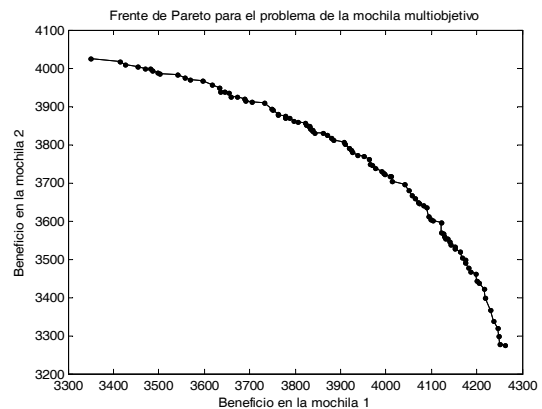


Figura 3: Frente de Pareto para el Esquema Básico.

Para el NSGA-II modificado se encuentra que el número de individuos encontrados pertenecientes al frente óptimo es de 13 en promedio, y encontrándose que gran parte del esfuerzo computacional se centra en la realización del ordenamiento no dominado, de tal forma que el tiempo invertido en esta tarea representa casi el 94% del tiempo de cómputo total. Lo anterior es de importancia, ya que en problema tratado, la evaluación de la función objetivo no representa una alta dificultad o una alta inversión de tiempo de cómputo.

Como se observa, el número de individuos del frente óptimo encontrados difiere bastante entre los dos enfoques, lo que indica que el esquema modificado tiene que realizar un esfuerzo muy grande cuando se trata de encontrar un conjunto no dominado de relativo alto tamaño. De hecho, cuando el NSGA-II modificado se prueba con un número alto de individuos (más de 100), encuentra mucha dificultad para encontrar soluciones de buena calidad, por lo que se tuvo que reducir el número de individuos para este enfoque. Un frente de Pareto típico del algoritmo NSGA-II modificado se muestra en la figura 4, y se observa que existe una leve diferencia.

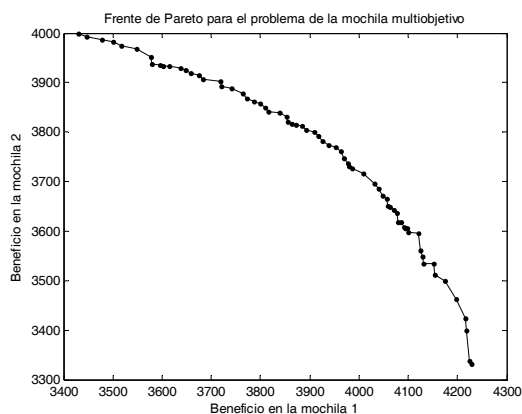


Figura 4: Frente de Pareto para el Esquema Modificado.

6. CONCLUSIONES Y RECOMENDACIONES

Se observa que el algoritmo presenta un buen desempeño al enfrentarse a problemas de optimización multiobjetivo. Su esquema de ordenamiento y la conservación de las mejores soluciones garantiza la presencia de soluciones de buena calidad y su posterior evolución hacia zonas Pareto-óptimas.

En el problema resuelto y utilizando el NSGA-II básico, el tiempo de cómputo al realizar la evaluación de la función objetivo y el ordenamiento no dominado es comparable. Mientras que esta relación es bastante desigual para el NSGA-II modificado, ya que en cada ciclo generacional se debe realizar de forma adicional el ordenamiento después de cruzamiento y mutación.

El algoritmo NSGA-II modificado propuesto como alternativa presenta un alto componente elitista al evaluar una configuración por cada ciclo generacional y permitiendo su entrada de acuerdo al criterios de optimalidad de Pareto y diversidad. Este hecho puede ser de importancia en problemas en los cuales el frente óptimo de Pareto no contenga un alto número de individuos y en donde la evaluación de la función objetivo tenga un peso desde el punto de vista del esfuerzo computacional alto, ya que como se vio, en el problema tratado la evaluación de la función objetivo no representaba un alto costo computacional, por lo que el esquema modificado se mostraba ineficiente al gastar mucho más tiempo realizando el ordenamiento no dominado de las soluciones que evaluando la misma función objetivo.

Cuando el frente óptimo de Pareto tiene asociados alto número de puntos, es más factible que el esquema básico encuentre puntos de alta calidad con mayor eficiencia.

7. BIBLIOGRAFÍA

- [1] R. A. Gallego, A. Escobar, R. A. Romero, "Técnicas de Optimización Combinatorial". *Primera Edición*, Abril de 2006.
- [2] Eckart Zitzler and Lothar Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, november 1999.
- [3] K. Deb and S. Agrawal and A. Pratap and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", Technical report, Indian Institute of Technology, Kanpur: Kanpur Genetic Algorithms Laboratory (KanGAL), Vol 15, 2000.
- [4] K. Deb, "Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems", *Evolutionary Computation*, 1999, 7(3).
- [5] K. Deb and S. Agrawal and A. Pratap and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multiobjective optimization: NSGA-II", *Proceedings of the Parallel Problem Solving from Nature VI*, 2000.
- [6] John Beasley and P. C. Chu, "A Genetic Algorithm for the Generalized Assignment Problem", *Computers Operations Research*, 1997.
- [7] S. Khuri, T. Bäck, and J. Heitkötter, "The zero/one multiple knapsack problem and genetic algorithms," in *Proc. 1994 ACM Symp. Applied Computing*, E. Deaton, D. Oppenheim, J. Urban, and H. Berghel, Eds. New York: ACM-Press, 1994, pp. 188–193.