

Strength By Objective: Una nueva estrategia de asignación de fitness para Algoritmos Evolutivos Multi-Objetivos.

Guillermo García, Wilmer Pereira y Gonzalo Ron

Escuela de Ingeniería de Informática,
Universidad Católica Andrés Bello,
Caracas, Venezuela, 1020
ggarciao@gmail.com,
gonzalo_ron@hotmail.com
wpereira@ucab.edu.ve

Abstract

Nowadays, a lot of effort has been invested in determining heuristic techniques for finding better solutions to multiobjective optimizations. Among the variety of proposals, the evolutionary algorithm SPEA2 can be named for its proved superiority over other techniques in a wide range of experiments. One of its milestones lies in the *Fitness* assignment strategy, called *Strength*, which classifies the population using a niche mechanism, based in the concept of *Pareto Front*. Although this strategy is considered one of the best because of the obtained results, it presents a weakness, normally handled adding a density estimator to the fitness. This work presents a new strategy, called *Strength by Objective*, which seeks to mitigate said weakness, including within the mechanism those elements from the population that do not dominate nor are dominated by others, which are completely ignored by the *Strength* technique.

Keywords: Multiobjective Optimization, Evolving Algorithms, EMO, SPEA2.

Resumen

En la actualidad, se ha invertido un importante esfuerzo en la determinación de una técnica heurística que encuentre mejores soluciones a optimizaciones multiobjetivo. Entre la gran variedad de propuestas resalta el algoritmo evolutivo SPEA2, por su demostrada superioridad sobre otras técnicas en un surtido grupo de pruebas experimentales. Uno de los pilares de SPEA2 es su estrategia de asignación de *fitness*, llamada *Strength*, que clasifica a la población con un mecanismo de nicho, basados en el concepto de Frente Pareto. Aunque esta estrategia se considera una de las mejores por los resultados obtenidos, ella presenta una debilidad que es manejada agregando un estimador de densidad al *fitness*. El presente trabajo propone una estrategia, denominada *Strength By Objective*, que busca mitigar dicha debilidad, incluyendo dentro del mecanismo a aquellos individuos que no dominan ni son dominados por otros, los cuales son ignorados completamente por *Strength*.

Palabras claves: Optimización Multiobjetivo, Algoritmos Evolutivos, EMO, SPEA2.

1. INTRODUCCIÓN

Cuando un problema matemático consiste en optimizar diversas funciones que comparten el mismo espacio de búsqueda, y entre dichas funciones no se puede establecer ningún tipo de ponderación o prioridad, se dice que es un problema de optimización multiobjetivo. Además, en muchos casos, estas funciones pueden ser contradictorias. Tomemos como ejemplo la función Schaffer F_2 (**Figura 1**), que consiste en conseguir un mismo número real X , que minimice a $h(x)$ y a $g(x)$. A simple vista, se puede observar que si $h(x)$ tiende a su mínimo global, $g(x)$ se aleja de su mínimo global, y viceversa.

$$f(x) = \begin{cases} g(x) = x^2 & \rightarrow \text{Mínimo global } g(0) = 0 \\ h(x) = (x-2)^2 & \rightarrow \text{Mínimo global } h(2) = 0 \end{cases}$$

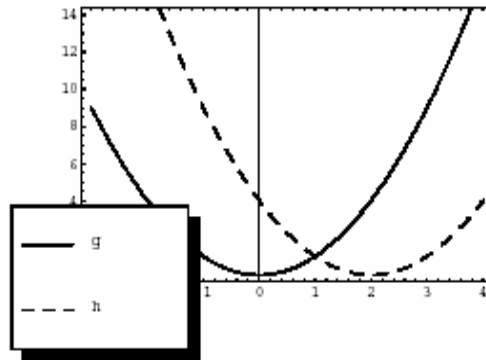


Figura. 1. Schaffer f_2

Observemos la siguiente comparación:

$$\text{Para } X = 0 \rightarrow f(0) = (0,4)$$

$$\text{Para } X = 2 \rightarrow f(2) = (4,0)$$

Si mantenemos el principio de no ponderación de los objetivos, no se puede establecer cual de las dos soluciones es mejor. Por esta razón, la solución a un problema multiobjetivo suele ser un conjunto, al cual se le denomina Frente Pareto. Este conjunto esta formado por aquellas soluciones que no consiguen ninguna otra solución que las mejore en todas las funciones que se busca optimizar. Estas soluciones pareto óptimas también son llamadas soluciones no dominadas. Las soluciones dominadas, son aquellas que si consiguen a otra solución que las mejora en todos los objetivos.

La segunda versión de *Strength Pareto Evolutionary Algorithm* (SPEA2)[2], es una de las técnicas heurísticas mas utilizada en la resolución de optimizaciones multiobjetivo. Su demostrada superioridad sobre otras estrategias, como NSGA-II [3] y su antecesor SPEA [1], lo han colocado como una de las primeras opciones a considerar cuando se presenta un problema de optimización de este tipo.

SPEA2 se diferencia de otros algoritmos evolutivos multiobjetivos por varios factores, entre los cuales tenemos dos de suma importancia ya que permiten determinar la bondad o fitness de un individuo:

- Una estrategia de asignación de *fitness* crudo o *raw fitness*, llamada en este trabajo *Strength*, que toma en cuenta para cada individuo cuantos individuos domina (fuerza) y por cuantos otros es dominado (golpes recibidos).
- Utiliza una técnica de estimación de densidad denominada el *K-ésimo vecino más cercano*, que refina el *raw fitness* y otorga una guía mas precisa para el proceso de búsqueda.

La estrategia de asignación de *fitness* crudo se divide en dos partes: La primera determina la fuerza de cada individuo, que es igual a la cantidad de individuos que son dominados por él. En la segunda, cada individuo recibirá golpes por parte de todos los individuos que lo dominan. La magnitud de esos golpes vienen definidos por la fuerza del golpeador.

La suma de las magnitudes de los golpes recibidos es el *raw fitness* del individuo. Nótese que a menos golpes recibidos, el individuo es mejor, lo que indica que la función *fitness* crudo es para minimizar.

Entendiendo bien esta estrategia, se puede observar claramente un detalle. Si la mayoría de los individuos no se dominan entre si, existirán grandes grupos de individuos que poseen el mismo *fitness*, lo cual no permite clasificar correctamente la población y por ende los individuos serán escogidos prácticamente de forma aleatoria. Para mitigar este detalle, SPEA2 agrega al *fitness* crudo el estimador de densidad, el cual permite desempatar a aquellos individuos con el mismo *fitness*.

Pero además, esta estrategia no toma en cuenta a los individuos que no son dominados ni dominan a otros, ya que estos no le dan fuerza a ningún individuo y además no golpean a nadie. Se puede hasta decir, que la existencia de estos individuos no afecta para nada la estrategia de asignación de fitness del resto de la población. Definitivamente, aquí se esta perdiendo información.

2. SPEA2: STRENGTH PARETO EVOLUTIONARY ALGORITHM 2

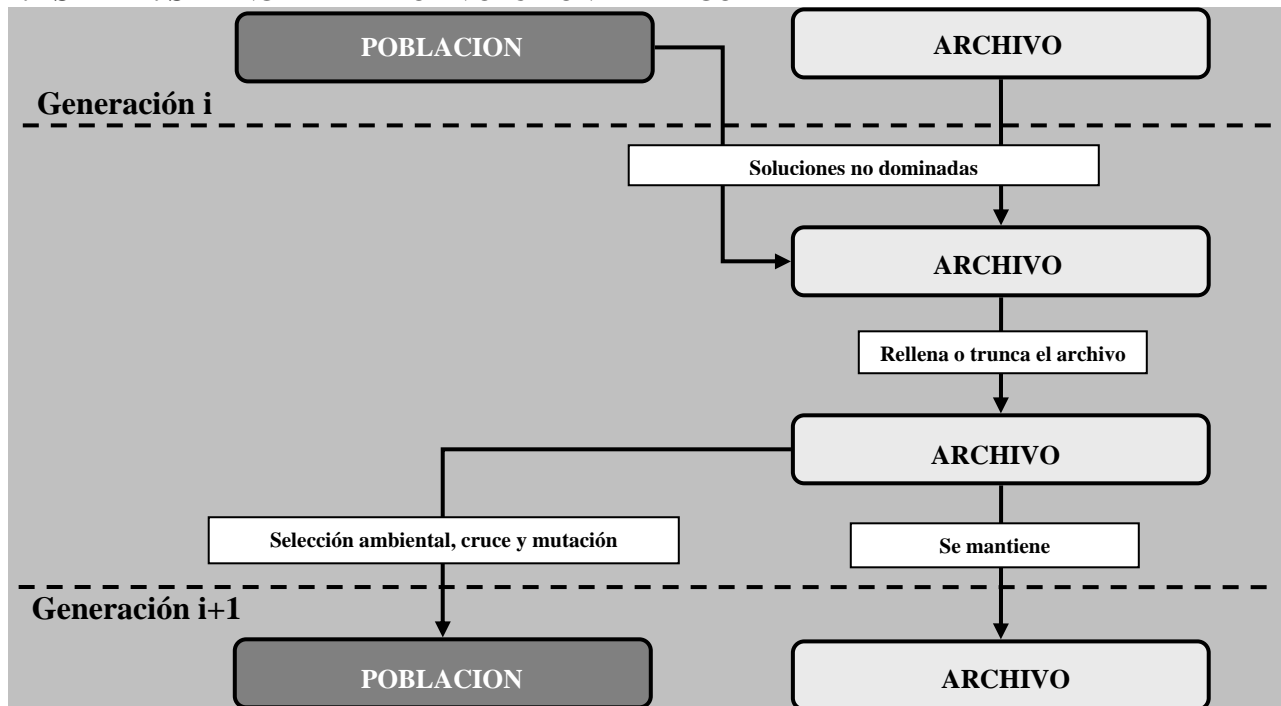


Figura 2. Algoritmo principal de SPEA2

Esta técnica heurística presenta un algoritmo evolutivo sencillo (Figura 2) que posee el siguiente hilo de ejecución principal:

1. Calcula la bondad o *fitness* de la **Población** y del **Archivo externo**.
2. Actualiza el **Archivo externo**.
3. Realiza **torneo binario** sólo con los elementos del **Archivo externo**.
4. Cruza a los vencedores del torneo, obteniendo una nueva población.
5. Vuelve al paso 1, hasta que se cumpla la condición de parada.

2.1. La estrategia de asignación de fitness de SPEA2

Este es el punto fuerte de la heurística. SPEA2 ya que presenta un sistema de valoración de soluciones basado en la dominancia entre vectores. Cada individuo posee dos atributos:

- a) **Fuerza** o *Strength*: Representa la cantidad de individuos dominados por él.
- b) **Raw Fitness**: Es la suma de las fuerzas de los individuos que lo dominan.

Para hacer el cálculo del fitness de todos los individuos (los de la Población y los del Archivo externo), primero se le asigna a cada individuo su fuerza o *strength*, determinando a cuántos individuos domina. Luego, a cada individuo se le asigna su *fitness crudo*, sumando las fuerzas de los individuos que lo dominan. Nótese que el *fitness* es para minimizar. Como se puede ver en la Figura 3, a los individuos pertenecientes al frente pareto no los domina nadie, por lo cual, su *raw fitness* es cero (0). Pero si los individuos de la población establecen pocas relaciones de dominancia entre sí (son todos del frente pareto, por ejemplo), se formarán grandes grupos de individuos con el mismo *fitness*, lo cual hará que el algoritmo escoja a los individuos con un alto grado de aleatoriedad. Para mitigar este detalle, a este raw fitness se le suma el estimador de densidad denominado *K-ésimo vecino más cercano*. Primero, para cada individuo se debe calcular la distancia cartesiana que existe entre él y cada otro individuo de la población y del archivo externo en el espacio solución.

Con esto obtenemos un vector de distancias ($dist(x)$), el cual debe estar ordenado de menor a mayor. Luego se toma el K-ésimo elemento del vector (donde $k = \sqrt{\text{tamañoDeLaPoblacion} + \text{tamañoDelArchivo}}$) y se coloca en la siguiente formula:

$$D(i) = \frac{1}{dist(k) + 2}$$

La constante 2 que se suma en el denominador es para garantizar que el estimador de densidad siempre sea menor que cero. Con esto se puede decir que todo individuo pareto óptimo tiene un fitness menor que uno (1).

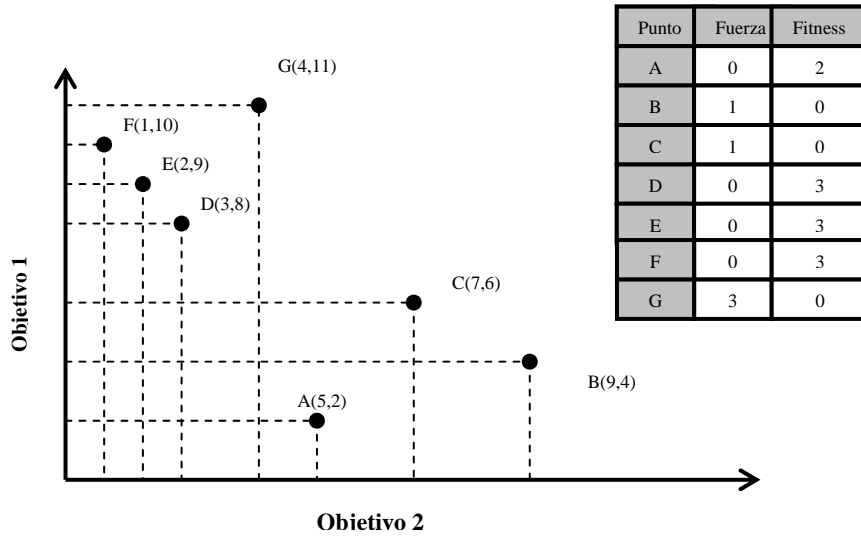


Figura 3. Asignación de fitness usada por SPEA2

2.2. La actualización del archivo externo

El Archivo externo es una lista de las mejores soluciones que se han conseguido hasta la generación actual. El proceso de actualización de este archivo externo cumple con los siguientes pasos:

- Se calculan las soluciones no dominadas de la Población actual y del Archivo externo.
- Se genera un nuevo archivo externo con las soluciones conseguidas en el paso anterior.
- Si el tamaño del nuevo Archivo Externo sobrepasa un tamaño preestablecido, se utiliza un operador de truncamiento basado en distancia cartesiana de vectores. Si el tamaño del archivo es menor que dicho tamaño, se rellena el Archivo Externo con los mejores individuos de la Población, basándose en la bondad o *fitness* de éstos.

2.3. El torneo binario (con remplazamiento)

La selección se realiza única y exclusivamente con los individuos que se encuentran en el Archivo externo. Los individuos generados por los cruces son considerados la población para la próxima iteración del algoritmo. El operador de cruce usado es el *single point crossover* con punto aleatorio.

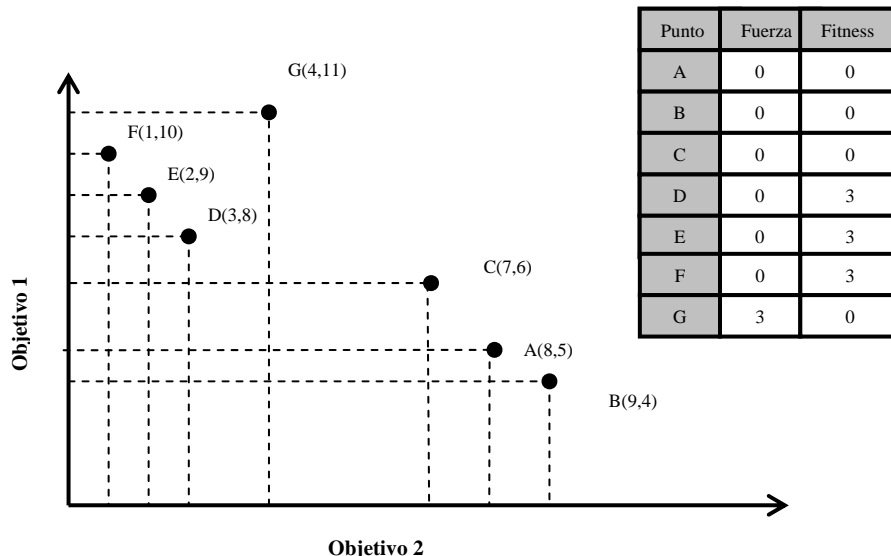


Figura 4. Un detalle en la asignación de fitness de SPEA2

3. STRENGTH BY OBJECTIVE

En el ejemplo de la **Figura 4** se puede observar como los individuos A, B y C no dominan ni son dominados por nadie. Por ende su fuerza y sus *fitness* son cero (0). Pero el resto de la población si establece alguna relación de dominancia con al menos un individuo de la población.

Cuando se calcula el raw fitness de todos los individuos podemos observar que los individuos A, B y C son ignorados completamente por el algoritmo. Tanto así que si elimináramos esos puntos y recalculáramos el *fitness* crudo de la población restante, esta no cambiaría. Esto lleva a pensar que en el algoritmo existe una cierta pérdida de información, ya que existen individuos que son incapaces de informar al resto de la población de su existencia.

Strength by Objective es nueva estrategia de asignación de *fitness* busca minimizar las debilidades presentadas por *Strength*, enfocándose estos dos problemas:

- Grandes grupos de individuo con el mismo *fitness*.
- Individuos que son ignorados por el algoritmo.

Esta estrategia de asignación de *fitness* que aborda el problema de forma más granular, tomando en cuenta la calidad de individuos en cada uno de las funciones a optimizar.

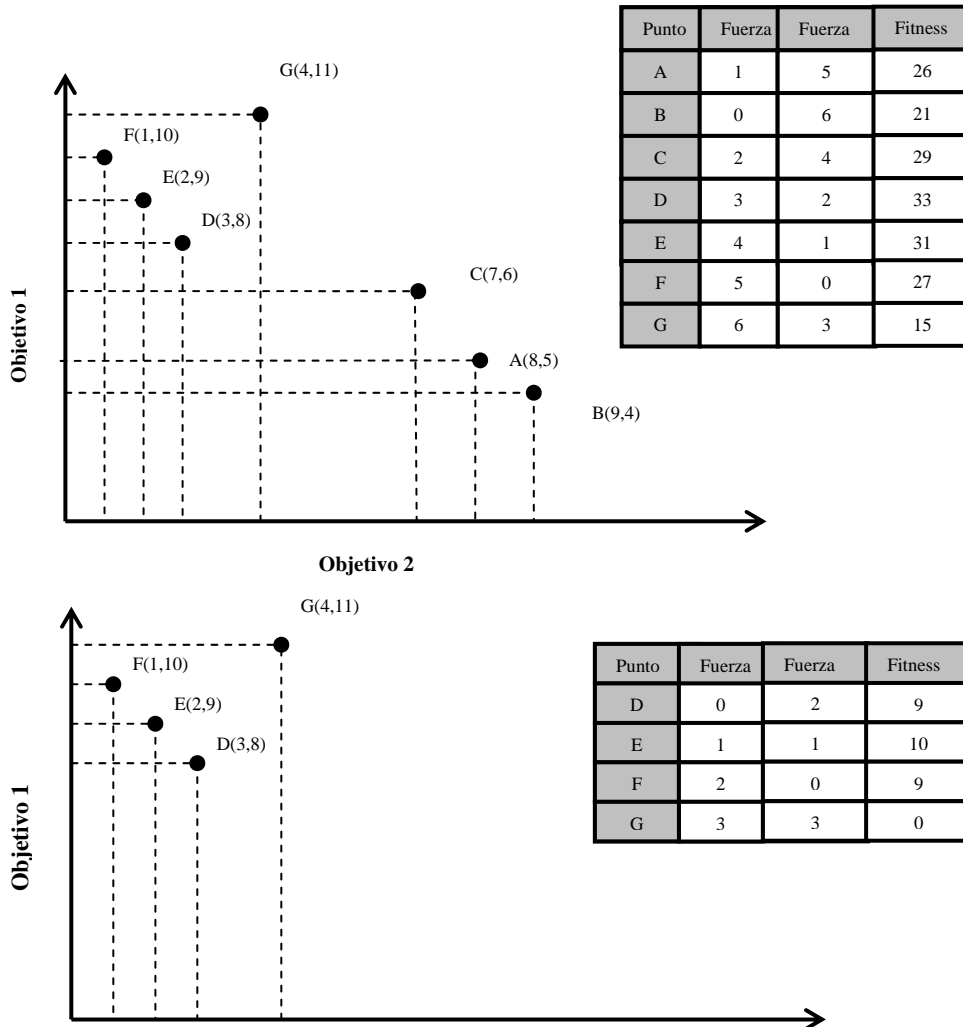


Figura 5. Strength By Objective

Strength by Objective utiliza atributos equivalentes a los de SPEA2 para cada individuo, aunque se calculan de forma distinta:

1. *Fuerza por objetivo*: Este atributo indica a cuántos individuos de la población vence el individuo en un objetivo en particular. Nótese que se tiene una fuerza por cada objetivo a optimizar.
2. *Golpes recibidos*: Cada vez que un individuo es vencido por otro en un objetivo recibe un golpe. La magnitud de este golpe depende de la fuerza del vencedor para ese objetivo. Este atributo es la suma de los golpes recibidos por un individuo.

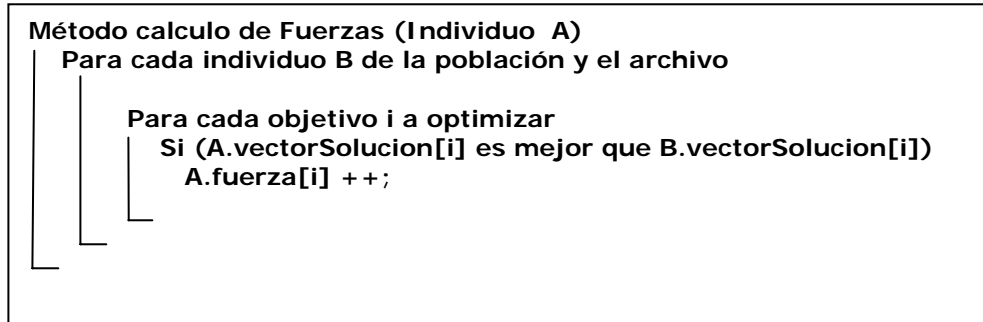


Figura 6. Algoritmo para el cálculo de Fuerzas

Primero se calculan las fuerzas de los individuos (Figura 6). Para esto, se compara a cada uno de ellos con el resto de la población y se determina a cuántos individuos vence en cada objetivo a optimizar. Luego, se determinan cuántos golpes recibe un individuo comparando a cada individuo con el resto de la población (Figura 7). En cada comparación, si el individuo vence a otro de la población en un objetivo, lo golpea con la fuerza que tenga para ese objetivo. La bondad o *fitness* de un individuo es la cantidad de golpes recibidos. A menor *fitness*, el individuo es mejor.

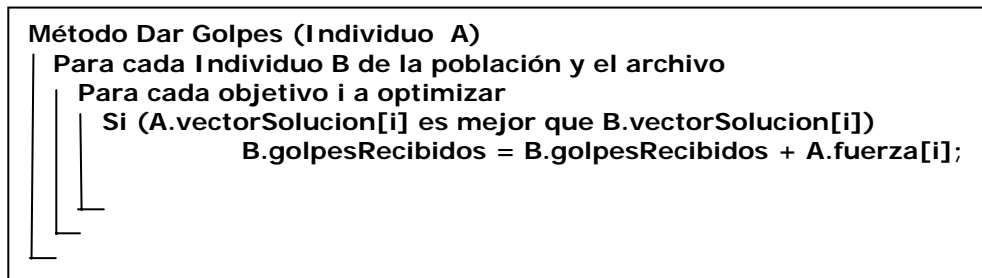


Figura 7. Algoritmo de asignación de *fitness*

Nótese que con esta estrategia la población completa participa en el proceso, porque todo individuo cumple con la condición de perder o ganar en un objetivo determinado. Además, este enfoque disminuye las posibilidades de que grandes grupos individuos tengan el mismo *fitness*.

Otro detalle que se debe resaltar acerca de *Strength By Objective* es que determina un nuevo orden a la población. Si se observa a los individuos A, B, C y G de la figura 5, notamos que son del frente pareto, pero sus *fitness* son completamente distintos, sin necesidad de agregar alguna otra estrategia, como el estimador de densidad.

5. STRENGTH VS STRENGTH BY OBJECTIVE

La forma de determinar si la nueva propuesta aborda correctamente el problema, es sometándolo a una comparación experimental. Como punto de comparación se va a utilizar la estrategia de asignación de *fitness Strength*, de SPEA2.

5.1. El algoritmo

Se hizo una implementación de SPEA2 siguiendo las especificaciones del trabajo de Zitler, Leummans y Thile, que pueda utilizar *Strength* o *Strength by Objective* como estrategia de asignación de *fitness* crudo. Es importante resaltar que ambas estrategias no usaran el estimador de densidad para así realizar una comparación exclusiva entre *raw fitness*.

5.2. Descripción de la prueba

Se usaron ambas estrategias para resolver tres planteamientos del “problemas de la mochila” (problema de combinatoria) con 750 elementos cuyos valores y pesos son generados de forma aleatoria. Los individuos son representados con un cromosoma binario. Los parámetros están especificados en la **Tabla 1**.

	<i>Dos(2) Knapsacks</i>	<i>Tres(3) Knapsacks</i>	<i>Cuatro(4) Knapsacks</i>
Tamaño población	250	300	400
Tamaño archivo	250	300	400
Mutación	0.006	0.006	0.006
Generaciones	50	50	50

Tabla 1. Parámetros de las pruebas a realizar (Knapsacks problems)

5.3. Variables de medición

- *Área cubierta*: Este criterio indica la cantidad de área del espacio de solución cubierta por el frente pareto dado. En un espacio de dos dimensiones, el área cubierta será igual al área de la unión de los rectángulos formados por cada uno de los vectores del frente pareto y el origen (0,0). Este criterio puede ser canónicamente extendido a N dimensiones. La **Figura 8** muestra un ejemplo.
- *Dominancia*: Dado dos conjuntos frente pareto A y B, el porcentaje de dominancia de A/B indica el porcentaje de soluciones de B que son dominadas por al menos una de las soluciones de A.

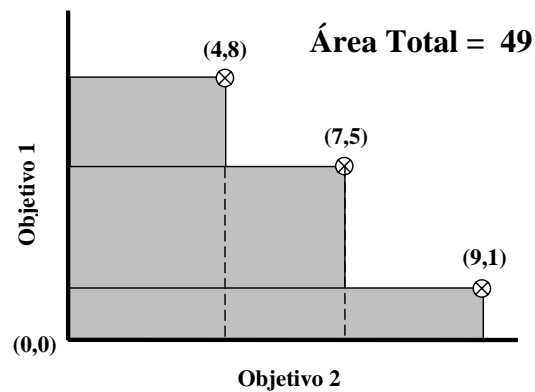


Figura 8. Área cubierta

5.4. Resultados

Estos resultados son el promedio obtenido de diez (10) corridas independientes del algoritmo para cada planteamiento.

- *Área cubierta*

<i>Estrategia</i>	<i>Dos(2) Knapsacks</i>	<i>Tres(3) Knapsacks</i>	<i>Cuatro(4) Knapsacks</i>
<i>Strength</i>	6,62E+10	1,42E+16	3,04E+21
<i>Strength by Objective</i>	6,58E+10	1,41E+16	2,93E+21
<i>Diferencia</i>	4,55E+08	1,25E+14	1,06E+20

Tabla 2. Resultados promedios de área cubierta

- *Dominancia*

<i>Estrategia/Estrategia</i>	<i>Dos(2) Knapsacks</i>	<i>Tres(3) Knapsacks</i>	<i>Cuatro(4) Knapsacks</i>
<i>Strength / Strength by Objective</i>	45,38%	28,25%	21,50%
<i>Strength by objective / Strength</i>	44,50%	29,38%	62,50%
<i>Diferencia</i>	0,88%	1,13%	41,00%

Tabla 3. Resultados promedios de dominancia

5.5. Análisis

Los resultados obtenidos en la prueba experimental ponen a *Strength by Objective* en muy buena posición. Usando las dos variables de medición, se puede observar que *Strength by Objective* mejora su desempeño con respecto a *Strength* al aumentar el número de objetivos.

En el promedio de área cubierta (**Tabla 2**) se observa que *Strength* se mantuvo por arriba de *Strength by Objective* por una mínima diferencia. Con esto se puede decir que ambas estrategias se comportan igual. Pero la dominancia nos revela que si existe una diferencia. En la **Tabla 3**, se puede observar como *Strength by Objective*, comienza a ganarle terreno a *Strength* al aumentar el número de objetivos, llegando al punto de sacar una diferencia de 41 puntos porcentuales para el problema de cuatro mochilas (*Four Knapsacks*).

6. CONCLUSIÓN

Strength By Objective es una estrategia de asignación de *fitness* para algoritmos multiobjetivos que evalúa a la población tomando en cuenta la calidad de un individuo para cada función a optimizar. Basándose en *Strength* de SPEA2, esta nueva propuesta establece una nueva clasificación a las poblaciones, la cual busca que todos los individuos de la población colaboren con su evaluación, tratando así de no perder la valiosa información que representa la existencia de un individuo. Pero además, *Strength by Objective* garantiza que la información se propague, ya que un individuo influenciara a todos los demás, porque siempre establecerán una relación de ganar o perder en un determinado objetivo.

Las comparaciones experimentales entre *Strength* de SPEA2 y *Strength by Objective* permiten decir que la nueva propuesta mejora a la de SPEA2 al aumentar la cantidad de funciones a optimizar, y en el peor de los casos ambas estrategias se comportan de la misma forma.

7. EL FUTURO DE LA INVESTIGACION

Este trabajo es solo el comienzo de una serie de investigaciones que buscan determinar el desempeño de *Strength by Objective* en la rama de los EMOs. (Evolutionary Multiobjective Optimization). En los siguientes pasos, se utilizara esta nueva propuesta para la resolución de un amplio grupo de problemas, de espacios de búsqueda continuos y discretos. Otro experimento que ayudara a definir la calidad de *Strength by Objective* es compararlo con el estimador de densidad usado por SPEA2 para corregir su debilidad.

Referencias

- [1] Eckart Zitzler and Lothar Thiele. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. May 1998.
- [2] Eckart Zitzler, Marco Laumanns, y Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. May 2001 (Errata added Sept 27, 2001).
- [3] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. 2001.