Autor del trabajo: Carmen Gádor Garzón Escamilla.
Título: Resolución de problemas de optimización vía algoritmos evolutivos.
Fecha/Convocatoria de defensa: Septiembre-2012
Directoras: Pilar Martínez Ortigosa y Juana López Redondo

PROYECTO FIN DE MÁSTER

POSGRADO EN INFORMÁTICA

UNIVERSIDAD DE ALMERIA

ESCUELA POLITÉCNICA SUPERIOR

MÁSTER EN INFORMÁTICA INDUSTRIAL

RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN MULTIOBJETIVO VÍA ALGORITMOS EVOLUTIVOS

Curso 2011/2012

Alumno/a:
Carmen Gádor Garzón Escamilla

Director/es:
Dra. Pilar Martínez Ortigosa
Dra. Juana López Redondo

UNIVERSIDAD DE ALMERÍA
ESCUELA POLITÉCNICA SUPERIOR
Departamento de Lenguajes y Computación

TRABAJO FIN DE MÁSTER
MÁSTER EN INFORMÁTICA INDUSTRIAL
POSGRADO EN INFORMÁTICA

RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN MULTIOBJETIVO VÍA ALGORITMOS EVOLUTIVOS

Carmen Gádor Garzón Escamilla

Dirigida por: Dra. Pilar Martínez Ortigosa y Dra. Juana López Redondo

Almería, Septiembre 2012

TRABAJO FIN DE MÁSTER
MÁSTER EN INFORMÁTICA INDUSTRIAL
POSGRADO EN INFORMÁTICA

RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN MULTIOBJETIVO VÍA ALGORITMOS EVOLUTIVOS

por
Carmen Gádor Garzón Escamilla

Para la obtención del
Título del Máster en Informática Industrial
Posgrado en Informática

Directora	**Directora**	**Autora**
Dra. Pilar Martínez Ortigosa	Dra. Juana López Redondo	Carmen Gádor Garzón Escamilla

El éxito no se logra sólo
con cualidades especiales.
Es sobre todo un trabajo
de constancia, de método y
de organización (J.P.Sergent)

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mis directoras de Proyecto, Pilar y Juani. Es muy difícil intentar plasmar en un papel mi agradecimiento hacia ellas porque me han convertido un camino que parecía difícil, con su paciencia, ánimo y confianza, en un camino mucho más ameno y confortable por el que caminar.

A Jose, profesor de la Universidad de Murcia al que no conozco personalmente, pero que a través de *skype* me ha ayudado y acercado un poco más a la estadística.

A Vicky, una de mis compañera de despacho, agradecerle su paciencia en mis momentos de agobio y pésimo, porque con ella muchos momentos se volvieron dulces.

A los profesores que me han impartido clase en el máster, por introducirme en la informática industrial. A mis compañeros de máster, porque juntos hemos vivido tristezas, alegrías, agobios y cada uno de esos momentos me han ayudado a crecer como persona. Quería hacer una distinción especial a una compañera, porque ha sido la mejor compis que me podía tener, porque aunque nos conocemos de siempre nunca habíamos sido compis de clase y ha sido una grata experiencia, es mi hermana, Bea.

Tal vez llegue a la parte más difícil de expresar mi agradecimiento, porque hasta cuando yo creía que me iba a caer y no creía en mí misma, ellos creyeron, me apoyaron y me dieron el aliento que nadie mejor me podía dar para continuar, quiero darle las gracias a mis padres. A mi "Mama Carmen" porque ella ha sido una clave fundamental en mi educación y en los valores que me ha inculcado como persona.

A mis dos loquillas, una llegada de tierras manchegas (Angelitas) y otra de la sierra jienense (Ochi), son mis compis de piso, porque han sido un gran apoyo en las noches cuando llegaba agotada y con el ánimo por los suelos.

A mi gente de la "doble" que aunque hemos seguido este año caminos diferentes, siempre los he tenido ahí, y esa amistad que se fraguó durante los años de estudios juntos, no ha desaparecido con la distancia.

A cada una de esas personas, familiares y amigos, que estáis siempre en los buenos y malos momentos, en las alegrías y las tristezas, porque la vida a veces es dura pero gracias a vosotros es diferente.

El agradecimiento es la parte principal de un hombre de bien (Francisco de Quevedo)

Índice

Resolución de problemas de optimización multiobjetivo vía algoritmos evolutivos

Carmen Gádor Garzón Escamilla
Máster en Informática Industrial.
Postgrado en Informática.
Escuela Superior de Ingeniería, Universidad de Almería.

Abstract—This paper intends to implement a multi-objective optimization meta-heuristic algorithm. The multi-objective optimization is a very important area of research because most real-world problems have multiple objectives. Instead of providing a single solution, the procedures that apply to multi-objective optimization problems generate a set of compromise solutions, generally known as Pareto optimal solutions, from which a decision maker chooses one. Different meta-heuristics, including evolutionary algorithms, have become the main method to explore the Pareto front in multi-objective optimization problems that are too complex to be solved by exact methods.
Given the many heuristics to solve multi-objective problems, in this paper, besides the implementation of a new algorithm, efficiency and effectiveness metrics will be defined and used to compare our implementation with some of the best known in the literature.
Resumen—En este trabajo se pretende implementar un algoritmo metaheurístico de optimización multiobjetivo. La optimización multiobjetivo es un área de investigación muy importante debido a que la mayoría de los problemas del mundo real tienen objetivos múltiples. En lugar de proporcionar una única solución, los procedimientos que se aplican a problemas de optimización multiobjetivo generan un conjunto de soluciones compromiso, generalmente conocidas como soluciones Pareto óptimas, de entre las cuales un agente decisor escogerá una. Diferentes metaheurísiticas, incluyendo los algoritmos evolutivos, se han convertido en el principal método para explorar el frente de Pareto en los problemas de optimización multiobjetivo que son demasiado complejos para ser resueltos por métodos exactos.
Dado que existen numerosas heurísticas para resolver problemas multiobjetivo, en este trabajo, aparte de la implementación de un algoritmo nuevo, se definirán métricas de eficiencia y eficacia y se utilizarán para comparar nuestra implementación con algunoss de los algoritmos más conocidos en la literatura.

Keywords—Multiobjective evolutionary algorithms, FEMOEA, Pareto-front.

Palabras clave—Algoritmos evolutivos multiobjetivo, FEMOEA, frontera de Pareto.

1 Introducción

Existen áreas del conocimiento y del quehacer humano donde surgen problemas relacionados con la mejora de determinados procesos o la obtención de mejores soluciones (generales o particulares). La disciplina que estudia este tipo de problemas y sus respectivas alternativas es conocida como *optimización*.

Fecha(Septiembre 14, 2012)

1.1 Algoritmos evolutivos

Los algoritmos evolutivos (en inglés Evolutionary Algorithms, EA) [20] se consideran como técnicas de aprendizaje no supervisado que, en general, no requieren de ningún tipo de conocimiento ya que el propio algoritmo es capaz de generar por sí mismo el conocimiento que requiere. Se consideran como técnicas meta-heurísticas de búsqueda que usan operadores probabilísticos y que funcionan como cajas negras, pues no requieren conocimiento especifico del dominio para actuar, y que son aplicables a un gran número de aplicaciones. La computación evolutiva [20] interpreta la naturaleza como una inmensa máquina de resolver problemas, y trata de encontrar los principios en que se basa su comportamiento para utilizarlos en los procedimientos de optimización. En la naturaleza todos los seres vivos

se enfrentan a problemas que deben resolver con éxito, como obtener alimento o conseguir más luz solar. El origen de esta capacidad está en la evolución producida por la selección natural que favorece la perpetuación de los individuos más adaptados a su entorno. La principal diferencia conceptual entre la selección natural que se produce sin intervención del hombre, y la selección artificial que establecemos en nuestros programas, es que mientras nosotros podemos seleccionar para la reproducción los seres que más nos interesan, en la naturaleza no existe una inteligencia exterior que determine la dirección de la evolución.

Algoritmos evolutivos multiobjetivo: En el contexto multi-objetivo no existe un solo óptimo global, que optimice todos los objetivos de forma simultánea, sino que cada objetivo pueda alcanzar su valor óptimo en un punto diferente.

1.1.1 Optimización multiobjetivo basada en frentes de Pareto

Los problemas en los que se obtienen múltiples soluciones de forma que ninguna de ellas se pueda probar mejor que otras suelen tratarse mediante técnicas multi-objetivo específicas, destacando sobremanera las conocidas como técnicas de optimización basadas en frentes de Pareto [21], [22] . Dicho concepto, que se describe formalmente con posterioridad, trata de encontrar todas las soluciones pertenecientes al conocido como frente Pareto-óptimo [22]. Dicho frente se puede definir como el conjunto de soluciones que no pueden ser mejoradas en todos los objetivos o igualadas en unos y mejoradas en otros por ninguna otra solución que también cumpla las restricciones del problema.

Consideremos un problema multiobjetivo:

min f(x)

s.t. $x \in X \subseteq \Re^m$

donde $f = (f_1, \cdots, f_n)$ consta de n funciones objetivo.

***Definición** 1:* Un vector factible $x^* \in X$ se dice que es *eficiente* si y sólo si no existe otro vector factible $x \in X$ tal que $f_l(x) \leq f_l(x^*)$ para todo $l = 1, \ldots, n$, y $f_j(x) < f_j(x^*)$ para al menos un índice $j \in \{1, \ldots, n\}$. El conjunto de todos los puntos eficientes se denomina: *conjunto eficiente* o *de Pareto*. Si x_1 y x_2 son dos puntos factibles y $f_l(x_1) \leq f_l(x_2), l = 1, \ldots, n$, siendo al menos una de las desigualdades estricta, entonces se dice que x_1 *domina* a x_2, y se denota por $x_1 \prec x_2$.

La eficiencia se define en el espacio de decisión. La definición correspondiente en el espacio imagen es la siguiente:

***Definición** 2:* Un vector objetivo $z^* = f(x^*)$ se denomina *no-dominado* (o también *eficiente*) si y sólo si x^* es eficiente. El conjunto de todos los vectores no-dominados se denomina *conjunto no-dominado* o *frente de Pareto*. Si x_1 y x_2 son dos puntos factibles y $x_1 \prec x_2$, entonces se dice que $f(x_1)$ domina a $f(x_2)$, y se denotará por $f(x_1) \prec f(x_2)$.

***Definición** 3:* Un vector factible $x^* \in X$ se dice que es *débilmente eficiente* si y sólo si no existe otro vector factible $x \in X$ tal que $f_l(x) \leq f_l(x^*)$ para todo $l = 1, \ldots, n$. Si x_1 y x_2 son dos puntos factibles y $f_l(x_1) \leq f_l(x_2)$, $l = 1, \ldots, n$, entonces se dice que x_1 *domina débilmente* a x_2, y se denota por $x_1 \preceq x_2$. Análogamente, en el espacio imagen, se dice que $f(x_1)$ domina débilmente a $f(x_2)$, $f(x_1) \preceq f(x_2)$.

***Definición** 4:* Se dice que dos puntos factibles x_1 y x_2 son *indeterminados (o incomparables)* siempre si ni $x_1 \not\preceq x_2$ ni $x_2 \not\preceq x_1$.

A continuación, se extiende algunas de estas definiciones a conjuntos:

***Definición** 5:* Se dice que el conjunto A *domina (débilmente)* al conjunto B, $A \prec B$ ($A \preceq B$), si todo punto $x_2 \in B$ es dominado (débilmente) por al menos un punto $x_1 \in A$.

***Definición** 6:* Se dice que los conjuntos A y B son *indiferentes*, $A \sim B$, si y sólo si $A \preceq B$ y $B \preceq A$.

***Definición** 7:* Se dice que el conjunto A es *mejor* que el conjunto B, denotado por $A \lhd B$, si y sólo si $A \preceq B$ y $A \not\sim B$.

2 ALGORITMOS

2.1 FEMOEA

FEMOEA es un algoritmo evolutivo ideado para hacer frente a problemas multiobjetivo no lineales. Su principal objetivo es proporcionar una aproximación de la frontera de Pareto de tamaño predeterminado, es decir, un número fijo de soluciones bien distribuidas y

no dominadas. Para esta finalidad, se combinan ideas procedentes de algoritmos clásicos que se encuentran descritos en la literatura: la utilización de un fichero externo (*external archive*) para almacenar la soluciones no-dominadas más *preferibles* (ver definición 8) [1], [2], y el uso del operador de comparación *crowded* para guiar el algoritmo hacia la obtención de una aproximación del frente de Pareto uniforme. Adicionalmente, también hereda algunos conceptos procedentes de otros algoritmos evolutivos pensados para la resolución de problemas de optimización con un único objetivo. Más exactamente, FEMOEA comparte algunas ideas con UEGO, un algoritmo general que ha sido utilizado para resolver problemas de localización [5], [6]. En particular, el concepto de radio decreciente, como un mecanismo para mantener un equilibrio entre exploración y explotación de la búsqueda en el espacio, también se considera aquí. Sin embargo, FEMOEA incorpora nuevos mecanimos los cuales aceleran el proceso de optimización y mejora la calidad (eficiencia) de las soluciones. El uso del método *de mejora* o del criterio de parada son dos de estas contribuciones específicas.

El concepto más importante en FEMOEA es el de subpoblación. Una subpoblación está definida por un centro y un radio. El centro es una solución y el radio es un número positivo el cual determina la subregión del espacio de búsqueda cubierta por la subpoblación. El objetivo principal del radio es *enfocar* los operadores que realizan la búsqueda en las subregiones correspondientes. Es necesario mencionar que el valor del radio de una subpoblación puede no ser constante a lo largo de la ejecución de FEMOEA ni el mismo para cada subpoblación. En particular, el radio es una función monótona que decrece cuando el proceso de optimización avanza. De este modo, en cada etapa del algoritmo pueden coexistir simultáneamente varias subpoblaciones con diferentes radios. El uso de diferentes radios en todo el proceso de optimización permite, que cuando los radios sean grandes, se pueda identificar rápidamente las regiones del espacio de búsqueda con alta probabilidad de contener soluciones de buena calidad mientras que se descartan las zonas con pocas probabilidades. Por el contrario, cuando los radios son pequeños, se pueden realizar búsquedas mucho más precisas que permiten alcanzar soluciones con una gran precisión[7]. Esta idea de dismunición del radio es un legado de UEGO [5].

Aparte del radio y el centro, una subpoblación tiene dos atributos relativos al espacio objetivo: el *rango de dominación*(d_{rank}) y la *distancia crowding* (c_{dist}), ver [12]. El rango de dominación indica el número de subpoblaciones que dominan a esa subpoblación en particular. En este sentido, un valor cero significa que una subpoblación no es dominada por ninguna de las restantes subpoblaciones vigentes, y es por tanto candidata a formar parte de la frontera de Pareto (ver figura 1). Por otro lado, la distancia

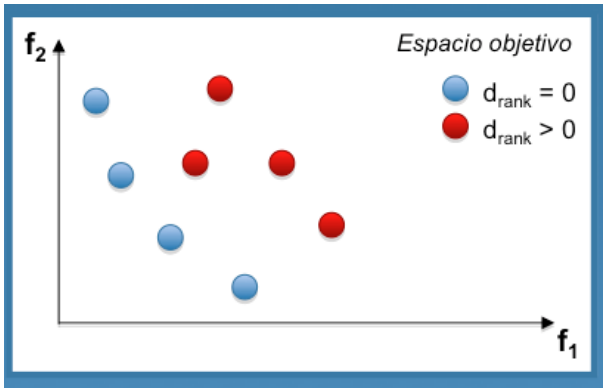

Fig. 1. Rango de dominación (d_{rank})

crowding es una estimación de la densidad de soluciones alrededor de una solución particular en una población. En este trabajo, se calcula como la distancia euclídea entre las dos soluciones más cercanas a cada lado del punto en el espacio criterio normalizado. En [12] se propuso un algoritmo que calcula la distancia *crowding* de cada punto en una población P, pero usando la distancia rectangular en lugar de la euclídea. Para el problema tratado, el algoritmo fue simplificado para el caso donde sólo dos objectivos son considerados y se modifica para trabajar con la distancia Euclídea, ver Algoritmo 1. Hemos utilizado la distancia euclídea puesto que representa la distancia *crowding* mejor que la distancia rectangular.

En el algoritmo 1, f_m^i (con m=1,2, …, n) se refiere al valor de la $m-$ésimo función objetivo

en el punto $i-$ésimo del conjunto P, y f_m^{max} y f_m^{min} se refieren al valor máximo y mínimo de la función objetivo $m-$ésima, respectivamente.

Algoritmo 1 Asignación de la distancia *crowding*

1: $p = |P|$
2: P=sort(P) Sort ordena la función del valor de la primera función objetivo
3: $c_{dist}^1 = c_{dist}^p = \infty$
4: **para** $i = 2 : p - 1$ **hacer**
5: $c_{dist}^i = \sqrt{\sum_{j=1:2}(\frac{f_j^{i-1} - f_j^{i+1}}{f_j^{max} - f_j^{min}})^2}$
6: **fin para**

FEMOEA trabaja con dos listas denominadas *population_list* y *external_list*. El número de soluciones máximo que se mantiene en las dos listas es M, un parámetro de entrada del algoritmo. El parámetro M hace referencia al número de soluciones deseadas en el frente de Pareto final. La *population_list* está compuesta de M subpoblaciones diversas con diferentes atributos, por ejemplo: varios radios, rangos de dominación y distancia *crowding*. FEMOEA es de hecho un método para gestionar estas listas (por ejemplo; creación, eliminación y mejora de las subpoblaciones). La *external_list*, puede entenderse como un depósito para guardar las soluciones no-dominadas. Obsérvese que el número de puntos no-dominados podría ser menor que M durante las etapas tempranas del algoritmo de optimización y por consiguiente, la *external_list* podría contener menos elementos que los deseados. De hecho, no se puede garantizar que se hayan encontrado un total de M soluciones no-dominadas, una vez que el criterio de parada ha sido satisfecho. En dicho caso, la *external_list* se completa entonces hasta M elementos con las soluciones más preferibles de la *population-list*.

***Definición** 8:* Una solución i es preferible a una solución j, $i \succ j$, si

- $d_{rank}(i) < d_{rank}(j)$, o
- $d_{rank}(i) = d_{rank}(j)$ y $c_{dist}(i) > c_{dist}(j)$.

La relación anterior se conoce como operador de comparación *crowded*. Para acelarar el proceso de selección, ambas listas se ordenan siempre de acuerdo con el operador de comparación *crowded*, es decir, en orden ascendente según el rango de dominanción y en orden descendente de la distancia *crowding* cuando varios elementos comparten el mismo rango de dominación.

En FEMOEA, cada subpoblación pretende obtener una solución eficiente. Para este propósito, FEMOEA ordena y dirige las subpoblaciones durante el proceso de búsqueda hacia la región más apropiada. Por lo tanto, una subpoblación particular no es una parte fija del dominio de búsqueda sino que puede moverse por el espacio según la búsqueda continúa. *Subpopulation_management* es una de las partes fundamentales de FEMOEA. Se trata de un conjunto de procedimientos para creación y selección de subpoblaciones durante todo el proceso de optimización (búsqueda de la frontera óptima de Pareto). Adicionalmente, FEMOEA incluye un método 'de mejora', el cuál se ha separado lógica y modularmente del proceso de manejo de subpoblaciones. Esto significa que para cada problema a resolver se podría elegir el método de improving más adecuado sin necesidad de modificar la estructura del algoritmo de optimización. Como consecuencia se puede decir que FEMOEA puede adaptarse fácilmente a cualquier problema multiobjetivo. En la fase de inicialización se crea un conjunto de M subpoblaciones mediante la generación aleatoria de M nuevas soluciones. La fase inicial finaliza con la construcción de la *population_list*, que incluye todas las soluciones, y la *external_list*, la cual sólo contiene las no dominadas. Después de este procedimiento; comienza el bucle principal de FEMOEA, el cual básicamente consta de tres procedimientos: creación, mejora y selección de subpoblaciones. Este bucle se ejecuta hasta que se cumple algún criterio de parada. Actualmente se ha diseñado para que se produzca la parada del bucle siempre que no se obtenga una mejora considerable entre tres frentes de Pareto consecutivos (depositados en *external_list*) o cuando se alcanza un nivel o número de iteraciones máximo. Dicho valor máximo se proporciona mediante el parámetro de entrada L.

El bucle principal comienza con la creación de un nueva descendencia de subpoblaciones, que se insertará en *population_list* dependiendo de si cumplen una serie de condiciones. Cada vez que una nueva subpoblación se crea, se le asociará un radio, cuyo valor depende del nivel actual i (con $i \in [1, L]$). De este modo, las subpoblaciones que se han creado en diferentes niveles tienen diferentes radios. El primer radio R_1 se define como el diámetro del espacio de búsqueda, y el radio más pequeño correspondiente al último nivel o iteración L, R_L, es un parámetro de entrada del algoritmo. El resto de valores del radio se calcula como la función exponencial decreciente que se indica en la siguiente ecuación:

$$R_i = R_1(\frac{R_L}{R_1})^{\frac{i}{L-1}}$$

en donde R_i es el radio asociado al nivel i. Se puede observar que un punto en el espacio de búsqueda puede pertenecer a diferentes subpoblaciones con diferentes radios. Por la tanto, subpoblaciones con radio pequeño examinan un área relativamente pequeña, su movimiento en el espacio es lento, pero ello permite diferenciar entre la soluciones eficientes que se encuentran muy cerca. Al contrario, las subpoblaciones con radio grande estudian un área más grande, pueden recorrer grandes distancias y descubrir nuevas áreas prometedoras, que puede ser analizadas a conciencia en las últimas etapas del algoritmo. El proceso de creación también incluye una fase de actualización, que consiste en copiar las nuevas soluciones no-dominadas procedentes de la *population_list* a la *external_list*.

El procedimiento de mejora es una de las principales etapas del algoritmo. La idea que hay detrás de este método es el de uso un optimizador local con el objetivo, por un lado, de aproximar las soluciones almacenadas en ambas listas (*population_list* y *external_list*) al conjunto eficiente real y, por otro lado, de obtención de soluciones no-dominadas en los alrededores de las subpoblaciones para mejorar la distribución uniforme de estas.

Después de los procedimientos de creación y/o *de mejora*, puede ocurrir que el número de subpoblaciones en alguna de las dos listas sea mayor que el parámetro M. Entonces hay que hacer una selección de las soluciones más preferibles, mientras que el resto se descarta.

En el caso de que el número final de subpoblaciones en la *external_list* sea M, entonces la solución proporcionada por el algoritmo FEMOEA son las M subpoblaciones de dicha lista. Sin embargo, en el caso de que el número de subpoblaciones de dicha lista sea menor que M, entonces el algoritmo elige el número necesario de subpoblaciones de la *population_list* que se añadirán a las de la *external_list* hasta formar un total de M subpoblaciones. El criterio para elegir dichas subpoblaciones es el de máxima preferencia.

Algoritmo 2 Algoritmo FEMOEA

```
1:  Init_supopulation_list
2:  mientras no se cumpla el criterio de parada hacer
3:     Create_new_subpopulation
4:     si tamaño(population_list) > M entonces
5:        Select_subpopulation(population_list)
6:     fin si
7:     si tamaño(external_list) > M) entonces
8:        Select_supopulation(external_list)
9:     fin si
10:    Improve_subpopulation(population_list)
11:    Update_external_list
12:    si tamaño(external_list) > M entonces
13:       Compose_Pareto(external_list)
14:    fin si
15:    Improve_subpoblación (external_list)
16: fin mientras
17: si tamaño(external_list) < M) entonces
18:    Obtener_Pareto
19: fin si
```

2.1.1 El algoritmo

En el Algoritmo 2 se proporciona la descripción de la estructura básica del algoritmo FEMOEA. A continuación se describen las diferentes etapas clave:

- ***Init_subpopulation_list***(Inicialización de la listas de subpoblaciones) En este procedimiento, se crean tantas

subpoblaciones como indique el parámetro M. El centro de la subpoblaciones se crea aleatoriamente, mientras los radios serán asociados al nivel 1. Dicho radio inicial coincide con el diámetro del espacio de búsqueda, de este modo todo el área de búsqueda quedará cubierta. La *population_list* se inicializa a partir de este conjunto de subpoblaciones, mientras la *external_list* se compone únicamente de las subpoblaciones no-dominadas.

- ***Create_new_subpopulation(evals)***(Creación de nueva subpoblación) Por cada subpoblación en la *population_list*, se crean $\frac{evals}{2}$ puntos aleatorios de prueba en el área definida por el radio. El parámetro *evals* hace referencia al presupuesto del número de evaluaciones de las funciones objetivo disponible para cada subpoblación existente. En este trabajo se ha definido $evals = 20$. Además, para cada nueva solución candidata aleatoria, se calcula el punto más cercano (en el espacio de objetivo) en la *external_list*. Entonces, se calcula un punto aleatorio nuevo en el segmento que une la solución candidata con el punto más cercano de la *external_list*. Observar que el punto intermedio se puede situar fuera del área cubierta por la subpoblación original. Si el punto intermedio domina a la solución candidata, entonces será incluido en la *population_list* como una nueva subpoblación. En caso contrario, si la solución candidata es la que domina a la otra, será ésta la insertada en la población. Adicionalmente, si los dos puntos son *indeterminados* (es decir, ninguno domina al otro), entonces ambos se insertan como nuevas subpoblaciones. Los radios asignados a cada nueva subpoblación están asociados al nivel actual *i*. Como resultado de este procedimiento, la lista que se obtiene contiene varias subpoblaciones con diferentes niveles (por lo tanto, diferentes radios). La motivación detrás de este método es doble: por un lado, permite explorar el área definida por el radio de la subpoblación en la búsqueda de puntos no-dominados que ayudan a componer el conjunto de Pareto; por otro lado, permite descubrir nuevas áreas prometedoras fuera del radio de la subpoblación, que se analizan en futuras iteraciones del algoritmo. Además, se calculan el rango de dominación y la distancia *crowding* asociada a cada subpoblación. La *population_list* se ordena según el operador de comparación *crowding*. Después se estudia, si alguna de las subpoblaciones no-dominadas en la *population_list* merece ser incluirla en la *external_list*. Sólo aquellos que no están dominados por ningún elemento de la *external_list* se copian. Debemos tener en cuenta que algunos puntos en la *external_list* pueden ser dominados por los nuevos elementos insertados. En ese caso, se eliminan.
- ***Select_subpopulation(list)***(Selección de subpoblación) Si la *lista*(list) alcanza su capacidad máxima permitida, algunos individuos deben ser eliminados. La estrategia de selección usada en este trabajo está basada en el operador de comparación *crowded* [12]. De este modo siempre se seleccionan las subpoblaciones más preferibles. En la bibliografía se consideran otros criterios de selección como el *density estimation criterion* de SPEA2 [4]. Esta técnica también se ha estudiado con un amplio conjunto de problemas de prueba. Sin embargo, los resultados obtenidos, en términos de la distribución de puntos en el frente de Pareto, fueron similares para ambos métodos.
- ***Improve_subpopulation(list)*** (Método *improving* a las subpoblaciones) Los algoritmos de optimización multi-objetivo más clásicos utilizan métodos *de mejora* que se basan en operadores de mutación para conducir una solución hacia el frente de Pareto óptimo. En dichos métodos solamente se utilizan los valores de la función objetivo para guiar la estrategia de búsqueda [12], [2], [4]. Tales algoritmos parten de una solución. A continuación, basándose en una regla de transición pre-especificada, el algoritmo in-

dica una dirección de búsqueda. Se realiza entonces una mutación a lo largo de la dirección de búsqueda para encontrar la mejor solución. Si se encuentra una solución mejor, se convierte en la nueva solución y el procedimiento anterior se itera un número de veces. Estos métodos *de mejora* son generalmente lentos, porque requiere muchas evaluaciones de la función para la convergencia, aunque se puede aplicar a muchos problemas sin cambios importantes en el algoritmo. En este trabajo el método *de mejora* (Algoritmo 4), se basa en una mejora del algoritmo de búsqueda aleatoria MO_Random (Algoritmo 3). *Improve_subpopulation* se aplica el método *de mejora* a todas las subpoblaciones de la *list*. Como puede observarse en el Algoritmo 2, esta técnica se aplica tanto a la *population_list* como a la *external_list* en diferentes étapas del proceso de optimización (etapas 10 y 15, respectivamente). Dependiendo de la lista de entrada, hay pequeños cambios en el método *de mejora*. Una vez que a todas las subpoblaciones de la lista de entrada se les ha aplicado el método *de mejora*, se reordenan según el operador de comparación *crowded*.

- ***Update_external_list*** (Actualizar *external_list*). Durante el proceso anterior, se generan nuevos puntos no-dominados. En el Paso 11 del Algoritmo 2, la *external_list* se actualiza mediante la copia de la soluciones no-dominadas de la *population_list*. Por supuesto, esto implica reordenar la *external_list* de acuerdo con los nuevos valores del operador de comparación crowded.
- ***Compose_Pareto*** (Componer el Pareto). La solución aportada por el algoritmo debe incluir M soluciones, ya que es el requisito impuesto por el usuario. Si el número de soluciones en la *external_list* alcanza ese valor, el conjunto de Pareto se presentará como la solución final que se mantuvo en esta lista. Observar que, en la *external_list*, se han almacenado las soluciones no-dominadas que tienen mejor distribución durante el proceso de optimización. Sin embargo, puede ocurrir que el número de soluciones no-dominadas encontradas por el algoritmo sea menor que M. En dicho caso, se deberá componer una lista conjunta considerando todos los elementos en la *population_list* y en la *external_list*, y las M soluciones más preferibles entre ellas deberán ofrecerse como resultado del algoritmo.

2.1.2 *El método* improving

Los métodos *aleatorios de mejora* (Algoritmo 3), calculan tosdos los puntos de manera aleatoria, lo que aumenta la posibilidades de que los puntos candidatos dominen al centro actual. Esto aumenta la probabilidad de que los puntos que forman la aproximación del frente de Pareto se encuentren más cerca del frente de Pareto óptimo, pero no favorece una buena distribución de los puntos. El método de mejora que se propone como alternativa (Algoritmo 5) busca encontrar un equilibrio de modo que el frente de Pareto obtenido sea competitivo con las métricas estudiadas.

Algoritmo 3 Algoritmo MO_Random1(nf, σ_{ub})

1: Hacer $ic = 1$, $nf^{(ic)} = nf$, $scnt = 0$, $fcnt = 0$
2: Fijar $ex,\ ct,\ Scnt,\ Fcnt,\ Maxfcnt,\ ic_{max}$
3: **mientras** $ic < ic_{max}$ y $fcnt < Maxfcnt$ **hacer**
4: Generar un vector aleatorio $\xi_{aux}^{(ic)}$ en la zona definida por el radio de las especies σ_{ub}
5: **si** $\xi^{(ic)}$ dominan $nf^{(ic)}$ **entonces**
6: $nf^{(ic+1)} = \xi^{(ic)}$; $scnt = scnt+1, fcnt = 0$
7: **si no**
8: **si** $\xi^{(ic)}$ no es dominado por ningún punto de la *external_list* **entonces**
9: Incluir $\xi^{(ic)}$ en la *external_list*; $scnt = 0, fcnt = fcnt + 1$
10: **fin si**
11: **fin si**
12: $ic = ic + 1$
13: **fin mientras**
14: Return $nf^{(ic)}$

- **Algoritmo MO_SASS**
  Este algoritmo busca una dirección de búsqueda. Además, para moverse en la

Algoritmo 4 Algoritmo Improve_subpopulation

1: **para** para cada subpoblación *nf* en *list* **hacer**
2: nf=MO_Random1(nf, σ_{ub})
3: **fin para**
4: Return *list*

dirección *de mejora*, se ha desarrollado una variante multi-objetivo del procedimiento SASS (ver [18]) , Algoritmo 5. Tradicionalmente la optimización aleatoria se ha basado en estrategias de Búsqueda Estocástica con Agente Único (Sigle Agent Stochastic Search) o abreviadamente SASS. Como método directo, SASS no requiere propiedades especiales (diferenciabilidad, continuidad) de la función objetivo a optimizar por lo que presenta la ventaja de ser fácilmente implementable.

La modificación de SASS presentada en este trabajo se puede observar en el Algoritmo 5, donde σ es la varianza del tamaño de la perturbación que se controla mediante el número de éxitos (*scnt*) o fallos (*fcnt*) sucesivos que se producen.

Los parámetros de usuario del algoritmo son: nf y el límite superior (σ_{ub}). Hay una serie de parámetros predeterminados por el algoritmo que son: el número mínimo de éxitos, *Scnt*, y fallos, *Fcnt*, (para aplicar una expansión *ex* o una contracción *ct*), el número máximo de fallos consecutivos, *MaxFcnt* y el número máximo de iteracciones, ic_{max}. La desviación estándar σ especifica el tamaño de esfera que más probablemente contiene el vector de perturbación. El sesgo b localiza el centro de tal esfera basado en la dirección del éxito pasado.

El criterio de parada está basado en el número de iteraciones (ic_{max}) y el número máximo de fallos consecutivos (*MaxFcnt*).

2.1.3 Criterio de parada

Por lo general, los criterios de parada definidos por los algoritmos de optimización multi-objetivo descritos en la literatura se basan en el número de evaluaciones de la función [12], [2],

Algoritmo 5 Algorithm MO_SASS(nf, σ_{ub})

1: Hacer $ic = 1$, $nf^{(ic)} = nf$, $b^{(ic)} = 0$, $scnt = 0$, $fcnt = 0$, $\sigma^{(0)} = \sigma_{ub}$, $\sigma_{lb} = \max\{\sigma_{ub}/1000, 10^{-5}\}$
2: Fijar $ex,\ ct,\ Scnt,\ Fcnt,\ Maxfcnt,\ ic_{max}$
3: **mientras** $ic < ic_{max}$ y $fcnt < Maxfcnt$ **hacer**
4: $\sigma^{(ic)} = \sigma^{(ic-1)}$
5: **si** $scnt > Scnt$ **entonces**
6: $\sigma^{(ic)} = ex \cdot \sigma^{(ic-1)}$
7: **fin si**
8: **si** $fcnt > Fcnt$ **entonces**
9: $\sigma^{(ic)} = ct \cdot \sigma^{(ic-1)}$
10: **fin si**
11: **si** $\sigma^{(ic)} < \sigma_{lb}$ **entonces**
12: $\sigma^{(ic)} = \sigma_{ub}$ and $b^{(ic)} = 0$
13: **fin si**
14: **si** $\sigma^{(ic)} > \sigma_{ub}$ **entonces**
15: $\sigma^{(ic)} = \sigma_{ub}$
16: **fin si**
17: Generar un vector multivariable aleatorio gaussiano $\xi^{(ic)}_{aux} = N(b^{(ic)}, \sigma^{(ic)} I)$
18: **si** $nf^{(ic)} + \xi^{(ic)}$ domina $nf^{(ic)}$ **entonces**
19: $nf^{(ic+1)} = nf^{(ic)} + \xi^{(ic)}$; $scnt = scnt + 1, fcnt = 0$
20: **si no**
21: **si** $nf^{(ic)} + \xi^{(ic)}$ no es dominado por ningún punto de la *external_list* **entonces**
22: Incluir $nf^{(ic)} + \xi^{(ic)}$ en *external_list*; $scnt = 0, fcnt = fcnt + 1$; $b^{(ic+1)} = 0.4\xi^{(ic)}_{aux} + 0.2b^{(ic)}$
23: **si no**
24: **si** $nf^{(ic)} - \xi^{(ic)}$ domina a $nf^{(ic)}$ **entonces**
25: $nf^{(ic+1)} = nf^{(ic)} - \xi^{(ic)}$; $scnt = scnt + 1, fcnt = 0$
26: **si no**
27: **si** $nf^{(ic)} - \xi^{(ic)}$ no es dominado por ningún punto de la *external_list* **entonces**
28: Incluir $nf^{(ic)} - \xi^{(ic)}$ en *external_list*; $scnt = 0, fcnt = fcnt + 1$; $b^{(ic+1)} = b^{(ic)} - 0.4\xi^{(ic)}_{aux}$
29: **si no**
30: $b^{(ic+1)} = 0.5b^{(ic)}$, $fcnt = fcnt + 1$, $scnt = 0$
31: **fin si**
32: **fin si**
33: **fin si**
34: **fin si**
35: $ic = ic + 1$
36: **fin mientras**

[4]. De este modo, un algoritmo se para cuando se alcanza un número máximo de evaluaciones. Sin embargo, en este trabajo, se propone otro criterio de parada basado en la distancia Hausdorff. Informalmente, se mide la distancia entre dos conjuntos. Matemáticamente, la modificación de la distancia Hausdorff *hf* que se utiliza en FEMOEA viene dada por:

$$hd(F_1, F_2) = \frac{\frac{\sum_{x \in F_1} \min\{d(x,y): y \in F_2\}}{\max\{d(x,x'): x, x' \in F_1\}} + \frac{\sum_{y \in F_2} \min\{d(x,y): x \in F_1\}}{\max\{d(y,y'): y, y' \in F_2\}}}{2}$$

donde F_1 y F_2 son dos conjuntos discretos propuestos.
Para el problema multi-objetivo actual, el algoritmo finaliza si durante tres iteraciones consecutivas, los cambios que se experimentan en la *external_list* son despreciables (en términos de los valores de la función objetivo), para una tolerancia dada (*tol*), para este trabajo, *tol*$= 10^{-7}$, es decir; si:

$$hd(f(\text{external\_list}^{iter}), f(\text{external\_list}^{iter-1})) < tol$$

$$hd(f(\text{external\_list}^{iter-1}), f(\text{external\_list}^{iter-2})) < tol$$

Sin embargo, como una garantía, se considera un segundo criterio de parada basado en el número de iteracciones ejecutadas por FEMOEA. Entonces, el algoritmo para cuando se cumple el criterio anterior o si se alcanza el número máximo de iteraciones. El valor máximo se representa por el parámetro de entrada L.
Se puede observar que el primer criterio de parada permite que el algoritmo se pare cuando se obtiene una aproximación buena de un conjunto eficiente. Esto permite el ahorro de tiempo de CPU en algunos casos.

2.1.4 Parámetros de entrada

Hay tres parámetros de entrada que han de ser proporcionados por el usuario:

- M: El número de soluciones que deberán componer el frente de Pareto.
- L: El máximo número de niveles o iteraciones..
- R_L: El radio que se asocian con el máximo nivel.

Se debe tener en cuenta que los parámetros que deben estar bien ajustados son L y R_L.El parámetro M se determina según la experiencia del usuario basandose en los requerimientos o necesidades de la frontera de Pareto final.

2.2 NSGA-II

Algoritmo 6 Algoritmo NSGA-II

1: Generar una población P de tamaño N.
2: Identificar los frentes de dominancia
3: Evaluar las distancias de crowded de cada frente.
4: **mientras** no se cumpla el criterio de parada **hacer**
5: Usando selección, cruzamiento y mutación se genera una población descendiente del mismo tamaño que P.
6: Reunir padres e hijos en un conjunto de tamaño $2N$, clasificar los frentes de dominancia y distancias crowding.
7: Determinación del Conjunto Descendiente Final.
8: **fin mientras**

2.2.1 El algoritmo [12]

- **Identificar los frentes de dominancia**: para esto primero, por cada solución, se calculan 2 valores:
  1) n_i, el número de soluciones que domina a la solución i.
  2) S_i, un conjunto de soluciones las cuales la solución i domina.

  Posteriormente se identifican todos los puntos que tienen $n_i = 0$ y se almacenan en una lista F_1. Se llama F_1 a la frontera actual. Ahora, para cada solución en la frontera actual se visita cada miembro (j) en su conjunto S_i y reduce su n_j uno a uno. De esta forma, si para cualquier miembro j el valor se vuelve cero, se pone en una lista separada H. Cuando todos los miembros de la frontera actual han sido chequeados, se declara a los miembros de la lista F_1 como miembros de la primera frontera. Luego se continúa el proceso usando la nueva frontera H como la frontera actual.
- **Evaluar las distancias de crowding de cada frente**: (ver algoritmo 1)
- **Determinación del Conjunto Descendiente Final**: seleccionando los frentes de

mejor rango de dominación. Si se supera el límite de población N, eliminar las soluciones con menor distancia crowding en el último frente seleccionado.

2.3 SPEA2

Algoritmo 7 Algorithm SPEA2

```
1: Inicialización de población
2: mientras No se cumpla el criterio de parada hacer
3:    Calcular Fitness
4:    Reemplazo
5:    si T ≤ t o se satisface otro criterio de parada entonces
6:       Finalizar
7:    si no
8:       Selección
9:       Variación
10:   fin si
11: fin mientras
```

2.3.1 El algoritmo

En este apartado se describen las etapas más relevantes del algoritmo SPEA2 [9] Algoritmo (7):

- **Inicialización de población**. Genera una población inicial P_0 y crea un fichero vacío (*external_list)*. $\bar{P} = \emptyset$. Conjunto $t = 0$.
- **Calcular Fitness**.Calcular el valor fitness de los individuos en P_t y $\bar{P}_t$
  SPEA2 emplea un archivo externo $\bar{P}_t$ (también llamada población secundaria) para almacenar los individuos no dominados. A cada individuo i perteneciente a la población principal P_t o al archivo $\bar{P}_t$, se le asigna una valor de fuerza (*strength*) $S(i)$ que se calcula como:
  $$S(i) = |\{j | j \in P_t \cap \bar{P}_t \wedge i \succ j\}|$$
  Es decir, $S(i)$ representa el número de individuos en $P + \bar{P}$ que son dominados por el individuo *i*. A partir de $S(i)$ se calculan el valor de $R(i)$ mediante:
  $$R(i) = \sum_{j \in P_t + \bar{P}_t, j \succ i} S(j)$$
  $R(i)$ representa la suma de las fuerzas de los conjuntos que dominan al individuo *i*. El mecanismo de preservación de diversidad utilizado por SPEA2 se basa en el cálculo de la densidad $D(i)$ de cada individuo:
  $$D(i) = \frac{1}{\sigma_i^k + 2}$$
  donde σ_i^k corresponde a la distancia (media en el espacio objetivo) entre el individuo i y su *k*-ésimo vecino más cercano. Para mantener el número de individuos constante, el archivo se filtra en cada iteración, eliminando los individuos que presenten mayor densidad. El valor de k generalmente se toma como el entero más cercano a $\sqrt{|P| + |\bar{P}|}$. Finalmente, el fitness de cada individuo se obtiene:
  $$F(i) = R(i) + D(i)$$
- **Reemplazo**. Copia todos los individuos no-dominados en P_t y $\bar{P}_t$ a $\bar{P}_{t+1}$. Si el tamaño de $\bar{P}_{t+1}$ supera al valor $\bar{M}$ (tamaño del archivo), entonces se reduce $\bar{P}_{t+1}$ mediante el operador de truncamiento; si por el contrario el tamaño de $\bar{P}_{t+1}$ es menor que $\bar{N}$ entronces se completa $\bar{P}_{t+1}$ con individuos dominados en P_t y $\bar{P}_t$.
- **Finalizar**. Si se verifica el criterio de parada, entonces el conjunto A (conjunto de salida) se compone con los individuos no-dominados en $\bar{P}_{t+1}$.
- **Selección**. Realizar la selección binaria por torneos (*tournament*) con el reemplazo de $\bar{P}_{t+1}$ con la finalidad de completar la población de padres potenciales (la que tendrá posibilidades de crear descendencia).
- **Variación**. Se aplican los operadores recombinación y mutación a población de padres potenciales y el conjunto P_{t+1} es la población resultante. Se incrementa el contador de iteraciones ($t = t + 1$).

3 PROBLEMAS

En este apartado se presentan los diferentes problemas sin restricciones que se han resuelto en este trabajo. Son funciones que se encuentran presentes en la literatura. Estas funciones las vamos a dividir en 3 grandes grupos, dependiendo si el número de variables ó el número de funciones objetivo son parametrizables (Ver Anexo A).

- Número de variables y dimensión predeterminadas:

  - deb
  - deb1
  - fonseca
  - kursave
  - poloni
  - schaffer
  - viennet
  - viennet2
- Número de funciones objetivo predeterminadá y número de variables modificable:
  - qv
  - zdt1
  - zdt2
  - zdt3
  - zdt4
  - zdt6
- Número de variables de decisión y de funciones objetivo modificables
  - dtlz1
  - dtlz2
  - dtlz3
  - dtlz5
  - dtlz6
  - dtlz7

4 PLATAFORMA PISA

4.1 ¿Qué es PISA?

PISA consta de dos partes:

- Una interfaz basada en texto para los algoritmos de búsqueda. Se divide un proceso de optimización en dos módulos. Un módulo contiene todas las partes específicas para el problema de optimización (por ejemplo, la evaluación de soluciones, la representación problema, la variación de las soluciones). El otro módulo contiene las partes que son independientes del problema de optimización (principalmente el proceso de selección). Estos dos módulos se implementan como programas separados que se comunican a través de archivos de texto.
- Una biblioteca de módulos que se pueden seleccionar, concretamente, problemas de optimización (problemas de test y de referencia), los módulos de selección (optimizadores evolutivos multi-objetivo) y módulos de evaluación de la ejecución.

4.2 ¿Por qué es útil PISA?

En PISA se puede combinar libremente un módulo de selección con cualquier módulo problema, y viceversa. De este modo, para una aplicación específica, cuando un ingeniero de dicha aplicación quiera optimizarla, su trabajo de programación se puede reducir a escribir o programar el módulo que define la función objetivo para dicha aplicación. Una vez definido ese módulo, puede utilizar los módulos existentes en PISA con el diseño de los algoritmos multiobjetivo. Por lo tanto, no necesita ser un experto en la búsqueda multi-objetivo. Como PISA trabaja mediante el intercambio de archivos, los módulos son totalmente independientes del lenguaje de programación utilizado o el sistema operativo subyacente. El experto en una aplicación específica puede probar fácilmente un nuevo algoritmo multi-objetivo de optimización en los diferentes problemas de prueba y los problemas del mundo real.

4.3 Variator y Selector

PISA es una interfaz basada en texto para los algoritmos de búsqueda. Se divide un proceso de optimización en dos módulos. Estos dos módulos se implementan como programas separados que se comunican a través de archivos de texto.

Todas las partes con problemas específicos se encuentran en el módulo problema, *variator*. El módulo de selección, *selector*, de una solución candidata (llamado individuo) puede representarse por un identificador (ID) y un conjunto de valores objetivos que describen la calidad de este individuo. Esta es la única información que pasa desde el módulo problema al módulo de selección.

El *variator* crea una colección inicial de individuos y calcula los valores objetivos. El módulo *selector* elige a los padres y genera una descendencia, utilizando los operadores genéticos del algoritmo y esta descencencia se la pasa al *variator*. Todo este intercambio de datos se establece mediante archivos de texto. La sincronización de los dos módulos se consigue escribiendo una variable de estado en un archivo de texto que ambos programas pueden leer y actualizar. Los módulos se

pueden programar en diferentes lenguajes de programación o incluso ejecutar en diferentes plataformas (distribuida).

Los programas *variator* y *selector*, se pueden utilizar de dos formas:

- Aplicar el programa *monitor* existente, junto con los programas *variator* y *selector*.
- Agregar un nuevo programa *variator* y/o *selector* que sean compatibles.

4.4 Monitor

El monitor se encarga de realizar varias ejecuciones de un problema y de registrar las aproximaciones del frente de Pareto obtenidas [19]. En particular, los efectos de la herramienta de monitor son los siguientes:

- Observar la comunicación entre el *variator* y el *selector* con el fin de almacenar información sobre las poblaciones generadas en archivos de texto.
- Invocar automáticamente la optimización de varios procesos.
- Controlar los programas *variator* y *selector*.

4.5 El parámetro wait

Como se ha explicado anteriormente, en la plataforma PISA los programas se comunican a través de ficheros de texto, donde se define el parámetro "wait" que especifica el tiempo que un módulo se mantiene en reposo antes de volver a leer un archivo. De este modo, cuando un módulo intenta leer un archivo proveniente de otro módulo pero dicho archivo todavía no se ha actualizado, entonces tiene que esperar dicho tiempo "wait" antes de volver a intentar leer dicho archivo.

5 J-METAL

jMetal es sinómino de Algoritmos Metaheurísticos en Java (en inglés: Metaheuristic Algorithms in Java). Es un marco orientado a objetos basado en Java dedicado al desarrollo, la experimentación y estudio de metaheurísticas para la resolución de problemas multi-objetivo. jMetal proporciona un amplio conjunto de clases que pueden ser utilizados como bloques de construcción de metaheurísticas multiobjetivo, tomando ventaja de la reutilización de código. Así los algoritmos pueden compartir los mismos componentes básicos, tales como las implementaciones de los operadores genéticos y los estimadores de densidad, lo que facilita el desarrollo de los nuevos algoritmos con múltiples objetivos.

6 INDICADORES DE CALIDAD

En el campo de investigación de la optimización multiobjetivo se definen diferentes indicadores de calidad que se utilizan simultáneamente como medida de eficiencia de los algoritmos multiobjetivo. Estas medidas también se utilizan para comparar los resultados proporcionados por diferentes algoritmos cuando resuelven un mismo problema. En esta sección se presentan los indicadores más utilizados en la literatura y que se han empleado en este trabajo para la evaluación de la calidad de los resultados obtenidos por los algoritmos presentados anteriormente.

Dado que los algoritmos presentados son metaheurísticos, la metodología obliga a realizar diferentes ejecuciones para la resolución de un problema con un algoritmo dado. Por tanto, para la resolución de un problema dado, para cada algoritmo i, se realizan r_i ejecuciones diferentes, obteniendo por tanto r_i soluciones diferentes $A_1^i, ..., A_{r_i}^i$. Donde cada solución consiste en el conjunto de puntos que forman la aproximación a la frontera de Pareto. Se denota por *C* el conjunto de todos los conjuntos obtenidos: $C = \{A_1^1, ..., A_{r_1}^1, ..., A_1^q, ..., A_{r_q}^q\}$, siendo q, el número de algoritmos que vamos a comparar.

En algunos indicadores es necesario comparar la aproximación de la frontera de Pareto obtenida con la frontera óptima de Pareto. Sin embargo, en la mayoría de los casos dicha frontera óptima no es conocida, bien porque tenga infinitos puntos, bien porque no haya podido ser resuelta por ningún algoritmo determinístico. En tales casos se utiliza como aproximación un conjunto de referencia, *R*, formado por puntos conocidos no dominados. En nuestro caso, dado que vamos a comparar los

diferentes algoritmos para un amplio conjunto de funciones de prueba, para cada problema a resolver, se utilizará como conjunto de referencia el subconjunto de los puntos no dominados de C.

El estudio de las métricas se realiza a partir de la normalización de los valores de la función objetivo. Para realizar dicha normalización se considera el rango obtenido considerando todos los puntos que componen el conjunto C (en PISA el rango se calcula con el programa 'bound.c'). La normalización estándar para la $l-$ésima función objetivo en el punto $a_i$es:

$$f_l(a_i)' = \frac{f_l(a_i) - z_l^{(min)}}{z_l^{(max)} - z_l^{(min)}}$$

donde $z_l^{(min)}$ y $z_l^{(max)}$ son los valores mínimo y máximo, respectivamente, de f_l obtenidos por los puntos de C.
Pero en PISA la normalización se realiza en el intervalo $[1, 2]$ (con el programa 'normalize.c'), entonces la fórmula de normalización variaría de la siguiente forma:

$$f_l(a_i)'' = 1 + \frac{f_l(a_i) - z_l^{(min)}}{z_l^{(max)} - z_l^{(min)}}$$

6.1 Proximidad al frente de Pareto

Se mide la distancia mínima de la aproximación del frente de Pareto al conjunto de referencia R. Por lo tanto, los frentes de Pareto con mejor calidad estarán más próximos al conjunto de referencia, en consecuencia el valor del indicador obtenido será menor.

6.1.1 *Average distance [15]*

$$D_{av}(A_i) = \frac{\sum_{a^r \in R} \min_{a^i \in A_i} d_\infty(f(a^i)', f(a^r)')}{|R|}$$

donde

$$d_\infty(f(a^i)', f(a^r)') = \max_{l=1,\ldots,n}\{|f_l(a^i)' - f_l(a^r)'|\}$$

6.1.2 *Additive ϵ-indicator [17]*

Este indicador calcula la distancia mínima (en cualquier objetivo) que habría que desplazar cada solución para ser no-dominada con respecto a otro frente de Pareto (el conjunto de referencia R).

$$I_{\epsilon+}(A_i, A_j) = \min_{\epsilon \in \mathbb{R}}\{\forall a^j \in A_j \ \exists a^i \in A_i :$$

$$f_l(a^i)' - \epsilon \leq f_l(a^j)' \ \forall l \in \{1 \ldots, n\}\}$$

De este modo proporciona la mínima distancia que A_i necesita ser transladado en cada dimensión en el espacio objetivo tal que A_j es débilmente dominado por A_i.

El Additive ϵ-indicator viene dado por la siguiente fórmula:

$$I^1_{\epsilon+}(A) = I_\epsilon(A, R)$$

dond R es un conjunto de referencia.

6.2 Diversidad

Estos indicadores calculan la distribución de los puntos a lo largo del frente de Pareto. Los valores obtenidos por estos indicadores o métricas muestran la distancia entre los puntos del frente de Pareto, en consecuencia, si los puntos se encuentran mejor distribuidos el valor de la distancia será menor.

6.2.1 *Spread*

- Dimensión=2 [12]:

  $$\Delta(A_i) = \frac{d_u + d_l + \sum_{j=1}^{N-1} |d_j - \bar{d}|}{d_u + d_l + (N-1)\bar{d}}$$

  NOTA: En esta métrica es necesario ordenar los puntos.
  - d_j : distancia entre un punto y su consecutivo, siendo $\bar{d}$ la medida de estas distancias.
  - d_u y d_l es la distancia Euclidea entre los extremos de la solución y los extremos del conjunto de soluciones no-dominadas.

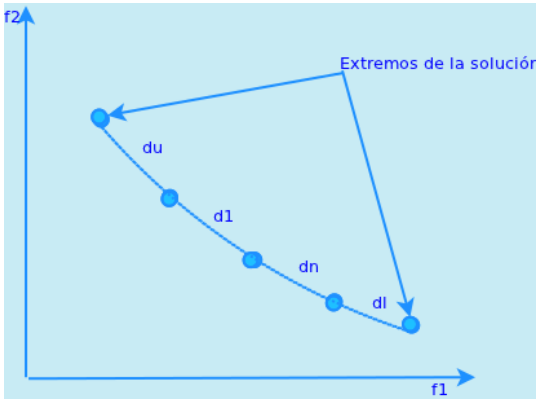

Fig. 2. Spread (dimensión 2)

- Dimensión m [2]:

$$\Delta(A_i) = \frac{\sum_{j=1}^{m} d(e_j,S)+\sum_{X\in A_i}|d(X,A_i)-\bar{d}|}{\sum_{j=1}^{m} d(e_j,A_i)+|A_i|*\bar{d}}$$

No es necesario ordenar los elementos:

  - $(e_1, ..., e_m)$ son los m extremos de R.

$$d(X, A_i) = \min_{Y\in A_i, Y\neq X} ||f(X) - f(Y)||^2$$
$$\bar{d} = \sum_{X\in R} d(R, A_i)$$

6.2.2 Spacing [16]

Viene definido por la fórmula:

$$TS(A_i) = \sqrt{\frac{1}{|A_i|}\sum_{a^j\in A_i}\frac{(D_j-\overline{D})^2}{\overline{D}}}$$

donde $\overline{D} = \sum_{a^j\in A_i} D_j/|A_i|$ y D_j es la distancia Euclídea en el espacio objetivo entre la solución a^j y la solución más próxima.

6.3 Globales

En estas métricas se ofrece una medida general que nos aporta información global, acerca de la calidad del frente obtenido cuando comparamos con otros algoritmos.

6.3.1 Dominance ranking [13]

El dominance ranking de un conjunto $A_i \in C$ viene dado por: $rank(A_i) = 1 + |A_j \in C : A_j \lhd A_i|$.

Este indicador o métrica compara conjuntos de frentes de Pareto, midiendo el número de conjuntos de frentes de Pareto que son mejores que uno dado. Si un conjunto es dominado por otros conjuntos un mayor número de veces, el valor de su *dominance ranking* será también mayor. Pero puede ocurrir que ningún frente de Pareto domine al frente analizado. En dicho caso el valor del *dominance ranking* será 1.

6.3.2 Hypervolume [14]

Esta métrica cálcula el volumen (en el espacio objetivo) que cubren los miembros de un conjunto no-dominado de soluciones A_i, para problemas donde los objetivos se minimizan. Matemáticamente, para cada solución $a_j \in A_i$, un hipercubo v_j se construye teniendo por referencia el punto de *nadir* y la solución a_j como las esquinas de las diagonales del hipercubo. El punto de *nadir* viene dado por los peores valores de la función objetivo. A continuación, se realiza la unión de todos los hipercubos y el hypervolume es:

$$HV = volume\left(\bigcup_{j=1}^{|A_i|} v_j\right)$$

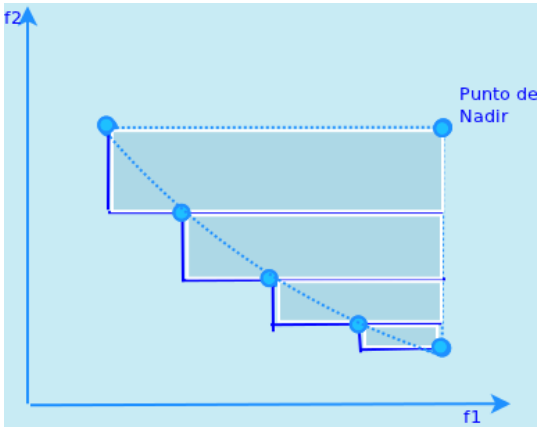

Fig. 3. hypervolume

6.4 Pruebas de hipótesis estadísticas

Dado que para la evaluación de los algoritmos se utilizan diferentes indicadores o métricas basados en diferentes ejecuciones de un mismo problema, se hace necesario confirmar que las diferencias entre las medias obtenidas no son fruto del azar y que los algoritmos realmente producen resultados diferentes. Para ello se ha realizado un estudio estadístico de contraste de hipótesis.

La técnica de contraste de hipótesis proporcionan procedimientos para aceptar o rechazar hipótesis estadísticas emitidas acerca de un parámetro, u otra característica de la población.

A la hipótesis que se desea contrastarse la denominaremos *Hipótesis nula*, y la denotaremos por H_0. Esta hipótesis nula es la que se somete a comprobación, y es la que se acepta o rechaza, como la conclusión final de un contraste. En nuestro caso, dado un un indicador de calidad IQ, se denotamos por μ_i^Q la media del valor del indicador IQ obtenida por el algoritmo i, $i = 1, \ldots, q$, el contraste que se hará para determinar si hay una diferencia significativa entre alguna de dichas medias es el contraste de igualdad de medias dado por $H_0 : \mu_1^{IQ} = \ldots = \mu_q^{IQ}$.

Cuando se realiza la comparación de dos o más algoritmos (medias), se pueden utilizar:

- Muestras independientes: cada ejecución de cada algoritmo utiliza una población inicial aleatoria, y el resto de variables aleatorias se obtienen de forma independiente.
- Muestras apareadas: la población inicial utilizada por los algoritmos es la misma en las diferentes ejecuciones.

En el caso de muestras independientes se pueden obtener diferencias entre la distribución de los valores de los indicadores en términos generales. Por el contrario, el uso de muestras apareadas permite obtener conclusiones sobre la capacidad de los algoritmos a la hora de obtener una buena solución a partir de un conjunto de partida dado.

6.4.1 Muestras independientes

En nuestro estudio se han empleado muestras independientes. Asi que, como paso previo a la resolución del contraste de igualdad de medias necesitamos saber si la poblaciones bajo estudio siguen o no una distribución normal. Para ello, y para cada algoritmo, resolvemos el contraste. H_0: Losa datos del indicador IQ siguen una distribución normal. Para verificar la hipótesis de normalidad hemos aplicado el test de Kolmogorov-Smirnov y el test de Shapiro-Will, y hemnos supuesto normalidad si al menos uno de ellos ha dado positivo.

6.4.1.1 Todas las muestras siguen una distribución normal: Entonces, se realiza el contraste de **homocedasticidad** o igualdad de varianzas:
$H_0 : \sigma_1^2 = \sigma_2^2 = ... = \sigma_q^2$ aplicando los test de LEvene y de Barlett. La homocedasticidad significa que la varianza entre los distintos grupos no es muy diferente.Hemos supuesto homocedasticidad si alguno de dichos tests da positivo.

- No existe diferencia significativa entre las varianzas: se resuelve entonces el test de igualdad de medias ($H_0 : \mu_1 = \mu_2 = ... = \mu_n$) aplicando ANOVA.
- Existe diferencia significativa entre las varianzas: se resuelve entonces el contraste de igualdad de medias mediante el test de Welch.

6.4.1.2 Al menos una de las muestras no sigue una distribución normal: Para resolver el contraste de igualdad de medias se aplica entonces el test no paramétrico de Kruskal-Wallis. En la figura 4 se presenta un resumen del análisis estadístico:

Fig. 4. Análisis estadístico

En este trabajo se considera un nivel de confianza del 95% (es decir, nivel de significación del 5% o p-valor inferior a 0.05) en los tests estadísticos. Si el test ha sido éxitoso se marca con el símbolo "+" en la última columna de todas las tablas; por el contrario, se marca con el símbolo "-", que la confianza estadísitca no se encontro (p-valor>0.05).

7 Estudio computacional

La finalidad del estudio computacional es exponer la competitividad de FEMOEA. Para ello se ha comparado con dos algoritmos de referencia, y que se han descrito anteriormente: NSGA-II y SPEA2.

Los parámetros que caracterizan al algoritmo FEMOEA están presentes en su descripción. Pero para los otros algoritmos se utilizarán los parámetros incluidos en [2]. Los operadores para cruce y mutación son SBX y mutación polinomial, con distribuciones indexadas $\eta_c = 20$ y $\eta_m = 20$, respectivamente. La probabilidad de cruce $\rho = 0.9$ y la probabilidad de mutación $\rho = \frac{1}{n}$ (donde n es el número de variables de decisión). El tamaño de las aproximaciones del frente de Pareto es de 100 individuos.

Se han realizado 100 ejecuciones para la obtención de un conjunto de frentes de Pareto para cada uno de los problemas con el algoritmo FEMOEA. Además de la obtención de los frentes de Pareto, se obtienen el número de evaluaciones de la función y el tiempo empleado por el algoritmo.

A continuación se ha realizado 100 ejecuciones de FEMOEA pero sin incorporar el método de mejora, y se han utilizado 2 versiones (random1 y random2) diferentes del algoritmo de búsqueda.

Para la ejecución de los algoritmos NSGA-II y SPEA2, se han utilizado las implementaciones de PISA. Su criterio de parada se basa en el número de evaluaciones de la función; pero se ha realizado un variación para poder utilizarlo también cuando el criterio de parada es un tiempo dado. Tenemos dos criterios de parada:

1) El número de evaluaciones de la función realizadas por FEMOEA (EVAL).
2) El tiempo de ejecución de FEMOEA (TIME).

A continuación se muestra una tabla con el número de ejecuciones realizadas con cada uno de los 2 algoritmos:

Criterio de Parada	Valor *wait*	Num.Ejecuciones
EVAL	0.1	10
EVAL	0.01	10
TIME	0.1	100
TIME	0.01	100

7.1 El método improving

En esta sección se analiza la bondad del método de mejora. Los resultados obtenidos se muestran en las tablas 1 y 2. Aunque los valores de las métricas obtenidos son bastantes similares, el método de mejora siempre obtiene mejores voalores en media. Además, es relevante el hecho de que se reduzca el tiempo de ejecución y el número de evaluaciones de la función.

7.2 Tiempos de CPU y número de evaluaciones de la función

En la tabla 3 se muestra los tiempos obtenidos por los algoritmos NSGA-II y SPEA2 en las plataformas PISA (con diferentes valores de wait=0.1 ó 0.01) y J-metal, siendo el criterio de parada el número de evaluaciones medias obtenidas por el algoritmo FEMOEA. El tiempo de ejecución de los algoritmos NSGA-II y SPEA2 en la plataforma PISA aumenta considerablemente respecto al valor de tiempo de ejecución obtenido por FEMOEA, pudiendo concluir que el tiempo de ejecución de FEMOEA es una ventaja que podemos destacar de este algoritmo; aunque si la comparación se realiza en la plataforma J-metal ofrece tiempos de ejecución más cercanos a FEMOA, siendo el tiempo de ejecución de NSGA-II inferior al de FEMOEA y el de SPEA2 superior.

En la tabla 4 se muestra el número de evaluaciones de la función requeridas por los algoritmos NSGA-II y SPEA2 en la plataforma PISA siendo el criterio de parada el tiempo de ejecución empleado por FEMOEA. Se puede observar que FEMOEA puede realizar un número elevado de evaluaciones, respecto a los otros dos algoritmos.

7.3 Resultados para indicadores de calidad

En esta sección se muestra los valores obtenidos por los indicadores de calidad aplicados a los frentes de Pareto obtenidos por los 3 algoritmos que se comparan en este trabajo (FEMOEA, NSGA-II y SPEA2). Se debe tener en cuenta que existen dos criterios de parada (EVAL y TIME) expuestos anteriormente, para los algoritmos NSGA-II y SPEA2 con diferentes valores de wait (0.1 ó 0.01).

Para comentar los resultados obtenidos vamos a seguir el mismo esquema de presentación de los indicadores de calidad.

7.3.1 Proximidad al frente de Pareto

Los indicadores que estudian la proximidad al frente de Pareto, nos indican de diferentes formas que distancia debería transladarse el frente de Pareto para llegar a ser el frente de Pareto óptimo, lo que implica que si las distancias son menores, el movimiento que debe realizar el Pareto también debería ser menor y por lo tanto, el frente de Pareto de mayor calidad, será el de distancia menor.

Si se observan las tablas relativas a *Average distance* (ver tablas 5 y 6), se puede determinar que en la tabla 5 las funciones tests alcanzan los valores menores de la tabla en el algoritmo FEMOEA en el 55% de los casos, siendo este valor superior cuando se observa la tabla 6 que alcanza el 80%. Si observamos los valores medios FEMOEA es del orden de 10^{-4}, mientras que el resto de algoritmos oscila entre 10^{-3} y 10^{-2}.

Tabla 1
Resultados con ó sin el método improving (Algoritmo 3)

	FEMOEA sin el método improving (Algoritmo 3)					FEMOEA con el método improving (Algoritmo 5)				
	time	eval	hyper	Average $I^1_{\epsilon+}$	Spread	time	eval	hyper	Average $I^1_{\epsilon+}$	Spread
deb	3.71956e+01	4.13551e+06	9.35260e-01	8.22500e-08	3.76264e-01	2.01000e+01	5.57064e+05	9.35263e-01	8.44271e-06	3.84189e-01
deb1	7.56257e+01	1.08597e+06	8.48828e-01	4.08988e-06	1.19193e+00	3.77600e+01	4.61618e+05	8.48828-01	4.09519e-06	1.19042e+00
dtlz1	7.30804e+00	1.38762e+06	9.99986e-01	0.00000e+00	1.69490e-01	8.55000e+00	5.26461e+05	9.99986e-01	6.40000e-10	1.63882e-01
dtlz2	9.98711e+01	2.40041e+06	4.93214e-01	0.00000e+00	1.94050e-01	3.54500e+01	5.04502e+05	4.93190e-01	0.00000e+00	1.91944e-01
dtlz3	6.35731e+00	1.28181e+06	9.99983e-01	0.00000e+00	1.89906e-01	7.17000e+00	4.66509e+05	9.99983e-01	9.99999e-12	1.71601e-01
dtlz5	1.00428e+02	2.40075e+06	4.87034e-01	0.00000e+00	1.92882e-01	3.83100e+01	5.04501e+05	4.87016e-01	0.00000e+00	1.91814e-01
dtlz6	4.80515e+01	5.27613e+05	8.02381e-01	0.00000e+00	1.93142e-01	5.25700e+01	4.37460e+05	8.02383e-01	0.00000e+00	1.89002e-01
dtlz7	6.56522e+01	7.70247e+05	9.34784e-01	6.77681e-05	8.98643e-01	4.86100e+01	4.51530e+05	9.34784e-01	6.79749e-05	8.98444e-01
fonseca	6.65097e+01	2.98211e+06	3.13180e-01	8.12220e-05	1.92806e-01	3.03900e+01	5.49244e+05	3.13121e-01	5.50361e-06	1.91634e-01
kur1	4.84199e+01	3.84421e+06	4.12887e-01	4.52829e-04	4.35391e-01	2.08400e+01	6.36392e+05	4.12935e-01	5.03549e-05	4.35793e-01
poloni	8.28791e+01	1.10380e+06	2.19623e-01	0.00000e+0	4.70152e-01	4.43600e+01	4.73881e+05	2.19623e-01	0.00000e+0	4.69062e-01
qv	7.81108e+01	2.86427e+06	8.32325e-02	2.15091e-04	2.49089e-01	3.31800e+01	4.91466e+05	8.31280e-02	2.41073e-04	2.44725e-01
schaffer	1.02771e+02	2.67304e+06	8.29745e-01	4.27392e-05	1.92734e-01	2.55800e+01	4.68853e+05	8.29755e-01	4.78947e-05	1.92099e-01
viennet	1.06540e+02	2.00665e+06	9.27349e-01	1.76749e-03	2.23472e-01	4.64300e+01	4.82751e+05	9.27351e-01	1.76749e-03	2.23493e-01
viennet2	1.67724e+02	2.51350e+06	8.43355e-01	1.54020e-05	3.88963e-01	4.54300e+01	4.84278e+05	8.43651e-01	2.14198e-05	3.50534e-01
zdt1	7.90105e+01	9.35909e+05	9.63504e-01	0.00000e+00	3.37348e-01	2.60300e+01	4.88530e+05	9.63496e-01	0.00000e+00	3.29624e-01
zdt2	7.95567e+01	9.36696e+05	9.05323e-01	0.00000e+00	2.79272e-01	2.45700e+01	4.88260e+05	9.05314e-01	0.00000e+00	2.74026e-01
zdt3	1.15831e+02	1.69189e+06	5.16340e-01	2.46928e-06	7.00156e-01	2.32100e+01	5.08962e+05	5.16257e-01	2.62124e-06	7.00431e-01
zdt4	4.63553e+01	4.09076e+06	9.92221e-01	0.00000e+00	3.49892e-01	2.46000e+00	1.63815e+05	9.92220e-01	1.11000e-09	3.42832e-01
zdt6	7.64356e+01	9.18442e+05	9.41081e-01	0.00000e+00	1.97997e-01	2.23100e+01	4.38316e+05	9.4108e-01	0.00000e+00	2.02945e-01
Average	7.45317e+01	2.02756e+06	7.22466e-01	1.32459e-04	3.71179e-01	2.96655e+01	4.79220e+05	7.22468e-01	1.10843e-04	3.66925e-01

Tabla 2
Resultados con ó sin el método improving (random 2)

	FEMOEA sin el método improving (random 2)					FEMOEA con el método improving (Algoritmo 5)				
	time	eval	hyper	Average $I^1_{\epsilon+}$	Spread	time	eval	hyper	Average $I^1_{\epsilon+}$	Spread
deb	1.10276e+01	4.64545e+05	9.34973e-01	1.02211e-05	3.94262e-01	2.01000e+01	5.57064e+05	9.35263e-01	8.44271e-06	3.84189e-0
deb1	1.73853e+01	4.90532e+05	8.48828e-01	4.35335e-06	1.19243e+00	3.77600e+01	4.61618e+05	8.48828e-01	4.09519e-06	1.19042e+00
dtlz1	1.21254e+00	8.76580e+04	9.99986e-01	0.00000e+00	1.76232e-01	8.55000e+00	5.26461e+05	9.99986e-01	6.40000e-10	1.63882e-01
dtlz2	1.90402e+01	5.13417e+05	4.93203e-01	1.99999e-11	1.92762e-01	3.54500e+01	5.04502e+05	4.93190e-01	0.00000e+00	1.91944e-01
dtlz3	1.32724e+00	9.71476e+04	9.99983e-01	1.95000e-08	2.03518e-01	7.17000e+00	4.66509e+05	9.99983e-01	9.99999e-12	1.71601e-01
dtlz5	1.91880e+01	5.13455e+05	4.87028e-01	1.40000e-10	1.92698e-01	3.83100e+01	5.04501e+05	4.87016e-01	0.00000e+00	1.91814e-01
dtlz6	2.53815e+01	4.37670e+05	8.02382e-01	0.00000e+00	1.93002e-01	5.25700e+01	4.37460e+05	8.02383e-01	0.00000e+00	1.89002e-01
dtlz7	2.47097e+01	4.71138e+05	9.34783e-01	6.86948e-05	8.99998e-01	4.86100e+01	4.51530e+05	9.34784e-01	6.79749e-05	8.98444e-01
fonseca	1.97448e+01	5.16449e+05	3.13173e-01	9.72875e-05	1.93071e-01	3.03900e+01	5.49244e+05	3.13121e-01	5.50361e-06	1.91634e-01
kur1	1.11259e+01	5.35233e+05	4.12488e-01	5.25358e-04	4.36016e-01	2.08400e+01	6.36392e+05	4.12935e-01	5.03549e-05	4.35793e-01
poloni	2.42673e+01	4.76430e+05	2.19626e-01	0.00000e+00	4.69745e-01	4.43600e+01	4.73881e+05	2.19623e-01	0.00000e+00	4.69062e-01
qv	2.58895e+01	5.16538e+05	8.31656e-02	2.11620e-04	2.6131e-01	3.31800e+01	4.91466e+05	8.31280e-02	2.41073e-04	2.44725e-01
schaffer	3.27715e+01	4.93085e+05	8.29770e-01	3.78935e-05	1.85439e-01	2.55800e+01	4.68853e+05	8.29755e-01	4.78947e-05	1.92099e-01
viennet	2.78066e+01	5.00125e+05	9.27347e-01	1.74981e-03	2.23321e-01	4.64300e+01	4.82751e+05	9.27351e-01	1.76749e-03	2.23493e-01
viennet2	2.84784e+01	5.07743e+05	8.43237e-01	1.43052e-05	3.76836e-01	4.54300e+01	4.84278e+05	8.43651e-01	2.14198e-05	3.50534e-01
zdt1	1.92696e+01	4.96125e+05	9.63502e-01	0.00000e+00	3.32021e-01	2.60300e+01	4.88530e+05	9.63496e-01	0.00000e+00	3.29624e-01
zdt2	1.91385e+01	4.96593e+05	9.05321e-01	0.00000e+00	2.756370e-01	2.45700e+01	4.88260e+05	9.05314e-01	0.00000e+00	2.74026-01
zdt3	1.38459e+01	5.15621e+05	5.16307e-01	1.17460e-05	7.00329e-01	2.32100e+01	5.08962e+05	5.16257e-01	2.62124e-06	7.00431e-01
zdt4	2.50702e+00	1.40207e+05	9.92220e-01	0.00000e+00	2.94407e-01	2.46000e+00	1.63815e+05	9.92220e-01	1.11000e-09	3.42832e-01
zdt6	2.11704e+01	4.59514e+05	9.41083e-01	0.00000e+00	1.96055e-01	2.23100e+01	4.38316e+05	9.41080e-01	0.00000e+00	2.02945e-01
Average	1.82644e+01	4.36461e+05	7.22420e-01	1.36566e-04	3.69454e-01	2.96655e+01	4.79220e+05	7.22469e-01	1.10843e-04	3.66925e-01

Para mostrar los resultados de forma más gráfica se realiza la representación de un trozo del frente de Pareto del problema *dtlz2* en la gráfica 5. El color rojo representa el conjunto de referencia, el color verde representa al algoritmo FEMOEA y el color azul al algoritmo SPEA2. Se puede observar a simple vista que los puntos obtenidos por el algoritmo FEMOEA están más próximos al conjunto de referencia, por eso el valor obtenido por la métrica *average distance* es menor.

La métrica Additive ϵ-indicator (ver tablas 7 y 8), muestra resultados similares a la métrica estudiada anteriormente porque estudia la proximidad al frente de pareto óptima. Nos corrobora la competitividad del algoritmo FEMOEA. Si observamos la tabla 7, el 65% funciones que componen el conjunto de funciones tests tienen un valor de Additive ϵ-indicator inferior en el algoritmo FEMOEA. Pero si observamos la tabla 8, cuyo criterio de parada es TIME, es decir, el tiempo que ejecución que emplea FEMOEA; aumenta el porcentaje de funciones tests que alcanza el éxito con eÃś algoritmo FEMOEA, llegando a un valor del 85%. Si comparamos los dos criterios de

Tabla 3
Tiempo medio de ejecución (Av(Time)) obtenido por los algoritmos cuando se ejecutan el número de evaluciones empleadas por FEMOEA(Av(NumberEval)). $M = 100$ para todos los algoritmos

	FEMOEA		PISA				jMETAL	
Problem	Av(NumberEval)	Av(Time)	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	NSGA-II	SPEA2
deb	5.57e+05	2.01e+01	2.12e+03	2.23e+03	1.46e+03	1.56e+03	1.12e+01	1.10e+02
deb1	4.62e+05	3.77e+01	3.10e+03	3.07e+03	2.47e+03	2.42e+03	1.03e+01	9.01e+02
dtlz1	5.26.5e+05	8.50e+00	1.98e+03	2.08e+03	1.31e+03	1.41e+03	1.23e+01	1.04e+02
dtlz2	5.05.1e0+5	3.54e+01	1.90e+03	1.98e+03	1.25e+03	1.29e+03	9.9e+00	1.16e+02
dtlz3	4.67.8e+05	7.10e+00	1.76e+03	1.87e+03	1.16e+03	1.24e+03	1.09e+01	9.01e+01
dtlz5	5.05.1e+05	3.83e+01	1.89e+03	1.97e+03	1.25e+03	1.29e+03	9.6e+00	1.14e+02
dtlz6	4.37.9e+05	5.25e+01	2.13e+03	3.17e+03	1.63e+03	2.19e+03	9.3e+00	1.13e+02
dtlz7	4.52.1e+05	4.86e+01	4.20e+03	4.49e+03	3.74e+03	3.93e+03	9.2e+00	1.06e+02
fonseca	5.49.3e+05	3.03e+01	1.96e+03	2.09e+03	1.23e+03	1.30e+03	1.32e+01	8.39e+01
kursave	6.36.7e+05	2.08e+01	2.26e+03	2.36e+03	1.40e+03	1.46e+03	1.48e+01	9.91e+01
poloni	4.74.2e+05	4.43e+01	1.74e+03	1.82e+03	1.13e+03	1.17e+03	1.37e+01	1.01e+02
qv	4.91.0e+05	3.31e+01	1.77e+03	1.87e+03	1.12e+03	1.17e+03	1.26e+01	1.21e+02
schaffer	4.69.8e+05	2.55e+01	4.88e+03	4.82e+03	4.96e+03	5.06e+03	1.10e+01	3.92e+01
viennet	4.83.2e+05	4.64e+01	1.92e+03	2.25e+03	1.28e+03	1.58e+03	1.91e+01	1.47e+02
viennet2	4.84.5e+05	4.54e+01	2.55e+03	2.87e+03	2.86e+03	3.68e+03	1.91e+01	1.07e+02
zdt1	4.89.3e+05	2.60e+01	6.56e+03	6.56e+03	5.79e+03	5.87e+03	9.4e+00	1.19e+02
zdt2	4.88.3e+05	2.45e+01	6.42e+03	6.31e+03	5.66e+03	5.68e+03	9.5e+00	1.25e+02
zdt3	5.09.8e+05	2.32e+01	2.92e+03	3.42e+03	2.06e+03	2.66e+03	1.48e+01	9.32e+01
zdt4	1.64.1e+05	2.40e+00	6.03e+02	6.26e+02	3.89e+02	4.04e+02	4.1e+00	3.06e+01
zdt6	4.38.9e+05	2.23e+01	2.11e+03	2.23e+03	1.58e+03	1.65e+03	9.4e+00	1.20e+02
Medias	4.79.6e+05	2.96e+01	2.74e+03	2.90e+03	2.19e+03	2.35e+03	1.17e+01	1.01e+02

Tabla 4
Número de evaluaciones medias de la función obtenidas por los algoritmos cuando el criterio de parada se basa en el tiempo de ejecución empleado por FEMOEA. $M = 100$ para todos los algoritmos

	FEMOEA		PISA			
Problem	Av(NumberEval)	Av(Time)	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$
deb	5.57e+05	2.01e+01	9.84e+03	9.48e+03	4.16e+04	3.97e+04
deb1	4.62e+05	3.78e+01	1.77e+04	1.70e+04	5.83e+04	6.27e+04
dtlz1	5.26e+05	8.55e+00	4.68e+03	4.55e+03	2.29e+04	1.84e+04
dtlz2	5.05e+05	3.55e+01	1.70e+04	1.62e+04	6.73e+04	5.61e+04
dtlz3	4.67e+05	7.17e+00	4.20e+03	4.10e+03	2.10e+04	1.69e+04
dtlz5	5.05e+05	3.83e+01	1.84e+04	1.76e+04	7.23e+04	6.05e+04
dtlz6	4.37e+05	5.26e+01	2.46e+04	2.34e+04	8.55e+04	6.94e+04
dtlz7	4.52e+05	4.86e+01	2.21e+04	2.11e+04	6.02e+04	5.62e+04
fonseca	5.49e+02	3.04e+01	1.49e+04	1.45e+04	6.83e+04	5.42e+04
kursave	6.36e+05	2.08e+01	1.03e+04	1.00e+04	5.14e+04	4.02e+04
poloni	4.74e+05	4.44e+01	2.13e+04	2.05e+04	8.79e+04	7.54e+04
qv	4.91e+05	3.32e+01	1.62e+04	1.58e+04	7.71e+04	6.11e+04
schaffer	4.69e+05	2.56e+01	1.20e+04	1.15e+04	3.59e+04	3.27e+04
viennet	4.83e+05	4.64e+01	2.19e+04	2.03e+04	7.94e+04	6.42e+04
viennet2	4.84e+05	4.54e+01	2.11e+04	1.99e+04	6.90e+04	6.03e+04
zdt1	4.89e+05	2.60e+01	1.25e+04	1.19e+04	3.78e+04	3.45e+04
zdt2	4.88e+05	2.46e+01	1.17e+04	1.11e+04	3.63e+04	3.24e+04
zdt3	5.09e+05	2.32e+01	1.15e+04	1.10e+04	4.09e+04	3.62e+04
zdt4	1.64e+05	2.46e+00	1.83e+03	1.80e+03	9.49e+03	7.770e+03
zdt5	4.38e+05	2.23e+01	1.11e+04	1.09e+04	4.32e+04	3.61e+04
Medias	4.83e+05	2.89e+01	1.42e+04	1.36e+04	5.33e+04	4.57e+04

parada podemos concluir que los algoritmos NSGA-II y SPEA2 obtienen mejores resultados cuando el criterio de parada es el número de evaluaciones. Por lo tanto, se puede determinar que FEMOEA ofrece frentes de Pareto más próximos al óptimo que el resto de algoritmos

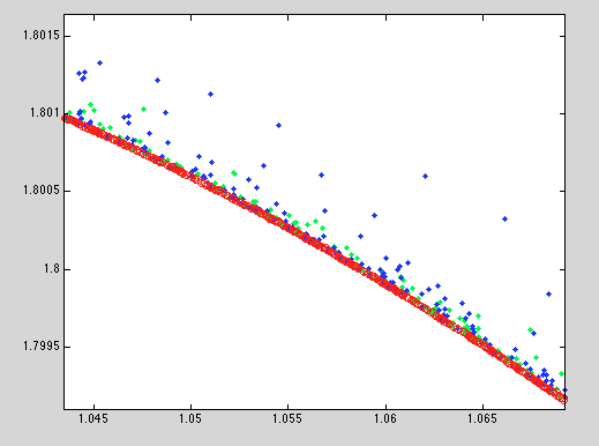

Fig. 5. Aproximación del frente de Pareto (dtlz2)

estudiados.

7.3.2 Diversidad

Los indicadores de *diversidad*, estudian la distribución de los puntos a lo largo del frente de Pareto, es decir, se mide la distancia entre los puntos que componen el frente de Pareto. En este trabajo, cada uno de los paretos que se han obtenido en las ejecuciones está compuesto por 100 individuos. Si los puntos están mejor distribuidos a lo largo del frente, quiere decir que la distancia obtenida por las métricas será inferior.
Vamos a analizar los valores obtenidos por el indicador *spread*. Los resultados se muestran en las tablas 9 y 10. Aunque todos los valores son del orden de 10^{-1}, el valor medio inferior lo obtiene el algoritmo FEMOEA (3.6710^{-1}). El porcentaje de éxito de FEMOEA, alcanza un porcentaje del 55% cuando el resto de algoritmos tiene como criterio de parada el número de evaluaciones, mientras que si el criterio de parada depende del tiempo alcanza un valor superior con un éxito del 60%.

La otra métrica que compone esta sección es el *Spacing*. Los resultados obtenidos por los diferentes algoritmos se muestran en las tablas 11 y 12. Se determina que cuando el resto de algoritmos tiene como criterio de parada el número de evaluaciones, FEMOEA obtiene en el 50% de las funciones tests el valor menor; pero cuando el criterio de parada atiende al tiempo, entonces este porcentaje alcanza un valor del 60%. Aunque si observamos los valores numéricos obtenidos se puede determinar que todos son del orden de 10^{-1}, FEMOEA obtiene un valor medio $= 5.57 * 10^{-1}$.

Se puede concluir que FEMOEA obtiene un frente de Pareto con los puntos mejor distribuidos que el resto de algoritmos estudiados en este trabajo.

7.3.3 Globales

Para concluir el estudio de los indicadores de calidad, finalizamos con los indicadores globales. El *dominance ranking* nos muestra si un Pareto es dominado por otro, es decir, si cada uno de los puntos que componen un Pareto domina al menos un punto de otro Pareto. Para todos los algoritmos estudiados en este trabajo se obtiene el valor de *dominance ranking* 1. Como se explicó anteriormente el valor 1 significa que ningún frente de Pareto es dominado por ninguno de los otros frentes de Pareto Por lo tanto, no se puede concluir que los frentes de Pareto obtenidos por un algoritmo sean estrictamente mejores que los obtenidos por otro algoritmo.

El último indicador que nos muestra el estudio estadístico es el *hypervolume*, que indica el área o volumen que puede cubrir el frente de Pareto. Si observamos las tablas 13 y 14, podemos podemos confirmar los resultados obtenidos en los indicadores ó métricas anteriores. Es decir, que el algoritmo FEMOEA nos ofrece frentes de Pareto mejores respecto a los otros 2 algoritmos estudiados. Aunque todos los valores obtenidos son del orden de 10^{-1}, el valor medio superior lo obtiene el agoritmo FEMOEA ($7.224682 * 10^{-1}$), debemos tener en cuenta que estamos midiendo el área o volumen dependiendo de la dimensión que cubre el frente de Pareto obtenido. Si se analiza el porcentaje de éxito obtenido por el algoritmo FEMOEA respecto a los otros algoritmos, se puede comprobar que el 65% de las funciones tests alcanza un valor superior en la métrica hypervolume cuando el criterio de parada del resto de los algoritmos consiste en el número de evaluaciones coincida con el obtenido por FEMOEA. Por otro lado, si se considera como criterio de parada el tiempo requerido por FEMOEA, dicho porcentaje de éxito se eleva hasta un 75% de las funciones.

8 CONCLUSIONES

Se ha desarrollado un algoritmo multiobjetivo nuevo, FEMOEA, se ha realizado un estudio comparativo dde FEMOEA con los algoritmos estándares multiobjetivo NSGA-II y SPEA2, utilizando para ella la plataforma PISA, que nos permite descargar la implementación de los algoritmos mencionados anteriormente. Para ello ha sido necesario comprender y entender su funcionamiento; con la finalidad de adaptarla a nuestras necesidades. Para el estudio se han seleccionado un conjunto de problemas tests presentes en la literatura que han sido resueltos por los tres algoritmos. Los problemas tests se han clasificado en función del número de funciones objetivo y del número de variables. También se ha tenido en cuenta incluir algunas funciones con frente de Pareto discontinuo. Al resolver los problemas tests se ha obtenido un conjunto de frentes de Pareto. Para poder comparar la calidad de los algoritmos ha sido necesario seleccionar un conjunto de indicadores de calidad o métricas. En el proceso de selección de los indicadores de calidad se ha optado por un conjunto compuesto por 6 indicadores que nos permiten conocer la distancia al frente de Pareto óptimo, la distribución de los puntos que componen el frente de Pareto y su calidad global. Los resultados medios obtenidos por las métricas tienen valores significativamente diferentes, ver la última columna de las tablas 5 a 14.

8.1 Aportaciones

Se ha implementado un algoritmo multiobjetivo: FEMOEA. Se han realizado algunas modificaciones en las implementaciones propuestas por PISA, para cambiar el criterio de parada y añadir nuevas funciones tests que no se encontraban presentes en la plataforma. Se ha adaptado la plataforma PISA para trabajar con frentes de Pareto que no son obtenidos por los algoritmos que se encuentran implementados en ella.

8.2 Futuros trabajos

Como algunas de las líneas de investigación en trabajos futuros proponemos incluir el método *calidad* como mecanismo extra en otros algoritmos para tratar de conseguir mejor calidad en la solución y una convergencia más temprana. Otro de los posibles trabajos futuros sería la adaptación de FEMOEA para la resolución de problemas con restricciones.

9 AGRADECIMIENTOS

Este trabajo ha sido financiado por la Junta de Andalucía (P08-TIC3518).

Tabla 5
Valores medios de *Average distance*. Algoritmos NSGA-II y SPEA2 con criterio de parada EVAL

	FEMOEA	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	
deb	3.51e-04	6.95e-02	9.39e-02	4.52e-02	4.57e-02	+
deb1	1.10e-05	2.55e-07	4.48e-07	2.73e-07	4.85e-07	+
dtlz1	2.51e-04	2.41e-04	3.03e-02	1.89e-04	7.53e-02	+
dtlz2	1.09e-04	5.11e-04	4.00e-04	5.39e-04	3.33e-04	+
dtlz3	2.34e-04	5.01e-02	5.90e-03	2.45e-04	7.17e-03	+
dtlz5	1.03e-04	5.69e-04	3.57e-04	5.13e-04	3.82e-04	+
dtlz6	3.42e-10	6.26e-04	1.96e-04	8.28e-04	2.79e-04	+
dtlz7	6.13e-06	4.79e-06	2.14e-07	3.46e-07	1.04e-07	+
fonseca	1.53e-04	1.10e-03	5.60e-04	1.01e-03	5.35e-04	+
kur1	4.40e-04	1.13e-03	6.91e-04	1.16e-03	6.94e-04	+
poloni	4.58e-05	1.50e-03	1.38e-03	1.06e-03	9.79e-04	+
qv	3.12e-04	8.23e-04	5.62e-04	7.81e-04	5.67e-04	+
schaffer	8.67e-07	8.90e-07	8.40e-07	1.34e-06	8.35e-07	+
viennet	2.60e-03	2.96e-03	3.09e-03	2.91e-03	3.24e-03	+
viennet2	3.94e-04	9.72e-04	7.48e-04	1.12e-03	7.46e-04	+
zdt1	5.95e-05	2.42e-07	1.60e-07	2.18e-07	1.60e-07	+
zdt2	4.94e-05	6.76e-09	2.88e-10	9.97e-10	1.48e-09	+
zdt3	7.17e-03	7.01e-03	7.39e-03	7.49e-03	7.30e-03	+
zdt4	3.20e-04	2.36e-04	4.21e-03	1.76e-04	3.46e-04	+
zdt6	1.70e-03	7.64e-05	1.52e-04	7.80e-05	1.06e-04	+
Medias	7.16e-04	6.87e-03	7.49e-03	3.16e-03	7.18e-03	

Tabla 6
Valores medios de *Average distance*. Algoritmos NSGA-II y SPEA2 con criterio de parada TIME

	FEMOEA	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	
deb	3.501e-04	1.31e-01	1.48e-01	8.27e-02	1.01e-01	+
deb1	1.11e-05	1.510e-03	5.89e-03	2.35e-06	1.21e-05	+
dtlz1	2.51e-04	6.63e-01	4.57e-01	4.44e-01	3.58e-01	+
dtlz2	1.09e-04	1.79e-03	8.48e-04	5.47e-04	7.11e-04	+
dtlz3	2.34e-04	6.13e-01	4.84e-01	4.80e-01	2.96e-01	+
dtlz5	1.03e-04	1.63e-03	1.07e-03	5.31e-04	7.84e-04	+
dtlz6	3.42e-10	5.23e-03	8.31e-03	7.28e-04	1.04e-03	+
dtlz7	6.13e-06	2.28e-04	5.99e-03	1.79e-06	1.32e-05	+
fonseca	1.53e-04	1.08e-03	6.31e-04	1.04e-03	5.72e-04	+
kur1	4.40e-04	1.51e-03	1.30e-03	1.22e-03	8.14e-04	+
poloni	4.58e-05	1.13e-03	9.93e-04	1.15e-03	9.41e-04	+
qv	3.12e-04	7.60e-04	5.60e-04	8.53e-04	5.82e-04	+
schaffer	8.67e-07	1.66e-06	1.92e-06	1.13e-06	9.74e-07	+
viennet	2.60e-03	3.15e-03	3.32e-03	3.24e-03	3.20e-03	+
viennet2	3.93e-04	9.04e-04	7.77e-04	9.64e-04	7.85e-04	+
zdt1	5.95e-05	2.60e-03	9.54e-03	9.69e-06	6.070e-05	+
zdt2	4.94e-05	7.29e-03	1.14e-02	1.95e-05	6.36e-04	+
zdt3	7.17e-03	7.19e-03	7.33e-03	7.17e-03	7.31e-03	+
zdt4	3.20e-04	3.95e-01	3.24e-01	3.12e-01	2.11e-01	+
zdt6	1.70e-03	1.44e-02	1.51e-02	3.12e-01	2.11e-01	+
Medias	7.16e-04	9.26e-02	7.43e-02	3.12e-01	2.11e-01	

Tabla 7
Valores medios: Additiveϵ-indicator. Algoritmos NSGA-II y SPEA2 con criterio de parada EVAL

	FEMOEA	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	
deb	8.44e-06	3.04e-02	4.16e-02	3.45e-02	1.25e-02	+
deb1	4.10e-06	4.04e-07	8.56e-08	0.00e+00	0.00e+00	+
dtlz1	6.40e-10	2.83e-06	1.27e-06	9.78e-07	9.17e-07	+
dtlz2	0.00e+00	2.32e-04	5.92e-04	3.77e-04	1.95e-04	+
dtlz3	1.00e-12	4.62e-07	1.15e-07	5.01e-07	4.96e-08	+
dtlz5	0.00e+00	3.87e-04	2.40e-04	1.95e-04	4.33e-05	+
dtlz6	0.00e+00	1.16e-04	5.41e-05	1.20e-04	5.98e-05	+
dtlz7	6.80e-05	6.65e-08	0.00e+00	0.00e+00	0.00e+00	+
fonseca	5.50e-06	1.02e-03	7.93e-04	1.17e-03	5.42e-04	+
kur1	5.04e-05	8.67e-04	1.53e-03	1.18e-03	1.22e-03	+
poloni	0.00e+00	2.77e-03	3.98e-04	7.32e-04	3.05e-04	+
qv	2.41e-04	5.03e-04	1.11e-03	1.87e-03	7.01e-04	+
schaffer	4.79e-05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	+
viennet	1.77e-03	1.82e-03	1.67e-03	2.97e-03	3.62e-03	+
viennet2	2.14e-05	9.61e-04	3.05e-03	2.89e-03	1.65e-03	+
zdt1	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	+
zdt2	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	+
zdt3	2.62e-06	1.830e-04	1.95e-08	8.42e-03	2.48e-03	+
zdt4	1.11e-09	1.54e-05	4.55e-06	1.82e-05	2.02e-05	+
zdt6	0.00e+00	1.35e-05	3.18e-05	1.87e-05	1.85e-05	+
Medias	1.11e-04	1.96e-03	2.55e-03	2.72e-03	1.17e-03	

Tabla 8
Valores medios: Additiveϵ-indicator. Algoritmos NSGA-II y SPEA2 con criterio de parada TIME

	FEMOEA	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	
deb	8.44e-06	5.77e-02	6.83e-02	3.34e-02	4.22e-02	+
deb1	4.10e-06	1.28e-05	2.40e-05	1.67e-06	2.62e-06	+
dtlz1	6.40e-10	7.32e-03	2.64e-04	1.170e-03	1.88e-03	+
dtlz2	0.00e+00	6.73e-04	4.51e-04	5.47e-04	6.48e-04	+
dtlz3	1.00e-11	5.82e-03	9.81e-04	1.55e-03	1.26e-03	+
dtlz5	0.00e+00	2.92e-03	4.52e-04	3.98e-04	8.31e-04	+
dtlz6	0.00e+00	4.51e-04	1.42e-03	1.24e-04	8.10e-05	+
dtlz7	6.80e-05	1.38e-05	4.41e-06	4.92e-07	6.15e-07	+
fonseca	5.50e-06	1.24e-03	7.53e-04	1.33e-03	6.80e-04	+
kur1	5.04e-05	1.46e-03	1.32e-03	1.25e-03	9.80e-04	+
poloni	0.00e+00	5.04e-04	4.72e-04	7.14e-04	4.12e-04	+
qv	2.41e-04	1.11e-03	8.48e-04	1.43e-03	8.83e-04	+
schaffer	4.79e-05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	+
viennet	1.77e-03	2.18e-03	4.95e-03	4.02e-03	3.96e-03	+
viennet2	2.14e-05	1.54e-03	6.42e-04	1.54e-03	9.98e-04	+
zdt1	0.00e+00	1.42e-05	1.71e-05	1.36e-06	2.75e-05	+
zdt2	0.00e+00	5.21e-05	1.08e-05	7.47e-06	2.00e-05	+
zdt3	2.62e-06	6.38e-03	6.17e-03	3.14e-03	5.70e-03	+
zdt4	1.11e-09	6.43e-03	1.15e-03	2.21e-02	3.84e-05	+
zdt6	0.00e+00	1.06e-04	1.16e-04	6.41e-05	4.90e-05	+
Medias	1.11e-04	4.80e-03	4.42e-03	3.63e-03	3.03e-03	

Tabla 9
Valores medios de *spread*. Algoritmos NSGA-II y SPEA2 con criterio de parada EVAL.

	FEMOEA	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	
deb	3.84e-01	8.04e-01	7.59e-01	8.00e-01	8.02e-01	+
deb1	1.19e+00	1.23e+00	1.23e+00	1.23e+00	1.23e+00	+
dtlz1	1.64e-01	4.56e-01	6.25e-01	4.23e-01	6.85e-01	+
dtlz2	1.92e-01	3.86e-01	1.68e-01	3.75e-01	1.72e-01	+
dtlz3	1.72e-01	5.21e-01	4.00e-01	4.51e-01	3.89e-01	+
dtlz5	1.92e-01	4.06e-01	1.76e-01	4.08e-01	1.67e-01	+
dtlz7	8.98e-01	1.01e+00	9.55e-01	1.00e+00	9.62e-01	+
fonseca	1.92e-01	4.13e-01	1.55e-01	4.38e-01	1.59e-01	+
kur1	4.36e-01	5.65e-01	4.38e-01	5.68e-01	4.36e-01	+
poloni	4.69e-01	6.11e-01	5.02e-01	6.06e-01	4.97e-01	+
qv	2.45e-01	4.09e-01	1.54e-01	4.06e-01	1.52e-01	+
schaffer	1.92e-01	3.92e-01	1.71e-01	4.35e-01	1.66e-01	+
viennet	2.23e-01	4.61e-01	2.38e-01	4.92e-01	2.14e-01	+
viennet2	3.56e-01	5.91e-01	5.93e-01	6.16e-01	6.01e-01	+
zdt1	3.30e-01	5.04e-01	3.17e-01	5.18e-01	3.13e-01	+
zdt2	2.74e-01	4.91e-01	2.53e-01	4.92e-01	2.63e-01	+
zdt3	7.00e-01	7.54e-01	7.04e-01	7.53e-01	7.042e-01	+
zdt4	3.43e-01	5.26e-01	3.27e-01	5.21e-01	3.10e-01	+
zdt6	2.03e-01	4.45e-01	2.27e-01	4.26e-01	2.22e-01	+
Medias	3.67e-01	5.71e-01	4.28e-01	5.70e-01	4.31e-01	

Tabla 10
Valores medios de *spread*. Algoritmos NSGA-II y SPEA2 con criterio de parada TIME.

	FEMOEA	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	
deb	3.84e-01	8.64e-01	8.59e-01	8.32e-01	8.30e-01	+
deb1	1.19e+00	1.23e+00	1.25e+00	1.23e+00	1.23e+00	+
dtlz1	1.64e-01	1.40e+00	1.41e+00	1.26e+00	1.34e+00	+
dtlz2	1.92e-01	4.12e-01	1.90e-01	4.00e-01	1.86e-01	+
dtlz3	1.72e-01	1.44e+00	1.46e+00	1.30e+00	1.27e+00	+
dtlz5	1.92e-01	4.09e-01	1.88e-01	4.03e-01	1.88e-01	+
dtlz6	1.89e-01	4.69e-01	4.34e-01	4.34e-01	2.14e-01	+
dtlz7	8.98e-01	9.96e-01	9.78e-01	9.95e-01	9.59e-01	+
fonseca	1.92e-01	4.20e-01	1.57e-01	4.18e-01	1.58e-01	+
kur1	4.36e-01	5.74e-01	4.68e-01	5.73e-01	4.44e-01	+
poloni	4.69e-01	6.05e-01	4.93e-01	6.11e-01	4.91e-01	+
qv	2.45e-01	4.01e-01	1.86e-01	4.03e-01	1.62e-01	+
schaffer	1.92e-01	3.98e-01	1.68e-01	3.90e-01	1.67e-01	+
viennet	2.23e-01	4.84e-01	2.17e-01	4.77e-01	2.17e-01	+
viennet2	3.51e-01	5.80e-01	5.88e-01	5.82e-01	5.87e-01	+
zdt1	3.30e-01	5.09e-01	3.61e-01	4.90e-01	3.14e-01	+
zdt2	2.74e-01	5.07e-01	3.36e-01	4.74e-01	2.74e-01	+
zdt3	7.00e-01	7.62e-01	7.06e-01	7.65e-01	7.05e-01	+
zdt4	3.43e-01	1.16e+00	1.19e+00	7.87e-01	5.27e-01	+
zdt6	2.03e-01	4.73e-01	3.28e-01	4.32e-01	2.44e-01	+
Medias	3.67e-01	7.05e-01	5.99e-01	6.63e-01	5.25e-01	

Tabla 11
Valores medios *Spacing*. Algoritmos NSGA-II y SPEA2 con criterio de parada EVAL.

	FEMOEA	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	
deb	6.16e-01	8.90e-01	8.65e-01	8.89e-01	8.90e-01	+
deb1	1.09e+00	1.11e+00	1.10e+00	1.11e+00	1.11e+00	+
dtlz1	4.03e-01	6.72e-01	6.80e-01	6.47e-01	6.94e-01	+
dtlz2	4.36e-01	6.18e-01	4.07e-01	6.09e-01	4.12e-01	+
dtlz3	4.12e-01	7.143e-01	5.78e-01	6.67e-01	5.61e-01	+
dtlz5	4.37e-01	6.34e-01	4.17e-01	6.35e-01	4.07e-01	+
dtlz6	4.32e-01	6.52e-01	4.09e-01	6.52e-01	4.14e-01	+
dtlz7	9.43e-01	1.00e+00	9.73e-01	9.96e-01	9.76e-01	+
fonseca	4.35e-01	6.39e-01	3.91e-01	6.58e-01	3.97e-01	+
kur1	6.57e-01	7.48e-01	6.58e-01	7.49e-01	6.57e-01	+
poloni	6.81e-01	7.77e-01	7.05e-01	7.74e-01	7.01e-01	+
qv	4.56e-01	6.36e-01	3.90e-01	6.34e-01	3.87e-01	+
schaffer	4.36e-01	6.22e-01	4.11e-01	6.56e-01	4.05e-01	+
viennet	2.38e-01	6.05e-01	2.58e-01	6.48e-01	2.36e-01	+
viennet2	5.21e-01	7.82e-01	6.69e-01	8.19e-01	6.86e-01	+
zdt1	5.71e-01	7.06e-01	5.59e-01	7.16e-01	5.56e-01	+
zdt2	5.21e-01	6.97e-01	5.00e-01	6.98e-01	5.10e-01	+
zdt3	8.33e-01	8.64e-01	8.35e-01	8.63e-01	8.35e-01	+
zdt4	5.82e-01	7.21e-01	5.68e-01	7.17e-01	5.54e-01	+
zdt6	4.48e-01	6.64e-01	4.74e-01	6.49e-01	4.69e-01	+
Medias	5.57e-01	7.37e-01	5.92e-01	7.39e-01	5.93e-01	

Tabla 12
Valores medios *Spacing*. Algoritmos NSGA-II y SPEA2 con criterio de parada TIME.

	FEMOEA	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	
deb	6.16e-01	9.18e-01	9.13e-01	9.05e-01	9.03e-01	+
deb1	1.09e+00	1.11e+00	1.11e+00	1.10e+00	1.10e+00	+
dtlz1	4.03e-01	1.23e+00	1.23e+00	1.14e+00	1.18e+00	+
dtlz2	4.36e-01	6.30e-01	4.23e-01	6.29e-01	4.20e-01	+
dtlz3	4.12e-01	1.24e+00	1.26e+00	1.16e+00	1.12e+00	+
dtlz5	4.36e-01	6.29e-01	4.18e-01	6.31e-01	4.20e-01	+
dtlz6	4.32e-01	6.64e-01	6.06e-01	6.52e-01	4.41e-01	+
dtlz7	9.43e-01	9.93e-01	9.80e-01	9.85e-01	9.75e-01	+
fonseca	4.35e-01	6.44e-01	3.94e-01	6.43e-01	3.96e-01	+
kur1	6.57e-01	7.54e-01	6.80e-01	7.53e-01	6.63e-01	+
poloni	6.81e-01	7.74e-01	6.99e-01	7.77e-01	6.97e-01	+
qv	4.56e-01	6.24e-01	4.12e-01	6.31e-01	3.97e-01	+
schaffer	4.36e-01	6.27e-01	4.08e-01	6.21e-01	4.07e-01	+
viennet	2.38e-01	6.56e-01	2.37e-01	6.52e-01	2.38e-01	+
viennet2	5.21e-01	7.81e-01	6.75e-01	7.87e-01	6.71e-01	+
zdt1	5.71e-01	7.08e-01	5.90e-01	6.96e-01	5.57e-01	+
zdt2	5.21e-01	7.04e-01	5.64e-01	6.85e-01	5.19e-01	+
zdt3	8.33e-01	8.68e-01	8.36e-01	8.70e-01	8.35e-01	+
zdt4	5.82e-01	1.09e+00	1.10e+00	8.72e-01	7.05e-01	+
zdt6	4.48e-01	6.81e-01	5.53e-01	6.53e-01	4.88e-01	+
Medias	5.57e-01	8.16e-01	7.04e-01	7.93e-01	6.57e-01	

Tabla 13
Valores medios *hypervolume*. Algoritmos NSGA-II y SPEA2 con criterio de parada EVAL.

	FEMOEA	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	
deb	9.352634e-01	9.121948e-01	9.053117e-01	9.197051e-01	9.201750e-01	+
deb1	8.488281e-01	8.487460e-01	8.488098e-01	8.487483e-01	8.488032e-01	+
dtlz1	9.999863e-01	9.999863e-01	9.999863e-01	9.999863e-01	9.999864e-01	+
dtlz2	4.931895e-01	4.922199e-01	4.928555e-01	4.922970e-01	4.929600e-01	+
dtlz3	9.999830e-01	9.999830e-01	9.999830e-01	9.999830e-01	9.999831e-01	+
dtlz5	4.870162e-01	4.859695e-01	4.867252e-01	4.860355e-01	4.866835e-01	+
dtlz6	8.023829e-01	8.020251e-01	8.023405e-01	8.020064e-01	8.023233e-01	+
dtlz7	9.347838e-01	9.347081e-01	9.347747e-01	9.3470735e-01	9.347727e-01	+
fonseca	3.131206e-01	3.101001e-01	3.124008e-01	3.101630e-01	3.124325e-01	+
kur1	4.129349e-01	4.110217e-01	4.124322e-01	4.109804e-01	4.124480e-01	+
poloni	2.196234e-01	2.161355e-01	2.163939e-01	2.173501e-01	2.171983e-01	+
qv	8.312798e-02	8.174594e-02	8.266643e-02	8.180099e-02	8.259053e-02	+
schaffer	8.297551e-01	8.292326e-01	8.297972e-01	8.290636e-01	8.297677e-01	+
viennet	9.273507e-01	9.219175e-01	9.270828e-01	9.230030e-01	9.268743e-01	+
viennet2	8.436510e-01	8.363732e-01	8.376190e-01	8.355039e-01	8.381381e-01	+
zdt1	9.634960e-01	9.633687e-01	9.635100e-01	9.633665e-01	9.635107e-01	+
zdt2	9.053143e-01	9.051593e-01	9.053263e-01	9.051522e-01	9.053305e-01	+
zdt3	5.162569e-01	5.159868e-01	5.162907e-01	5.159253e-01	5.162913e-01	+
zdt4	9.922198e-01	9.921978e-01	9.922297e-01	9.922003e-01	9.922320e-01	+
zdt6	9.410801e-01	9.409267e-01	9.410580e-01	9.409491e-01	9.410664e-01	+
Medias	7.224682e-01	7.199999e-01	7.203797e-01	7.204464e-01	7.211784e-01	

Tabla 14
Valores medios *hypervolume*. Algoritmos NSGA-II y SPEA2 con criterio de parada TIME.

	FEMOEA	NSGA-II$_{0.1}$	SPEA2$_{0.1}$	NSGA-II$_{0.01}$	SPEA2$_{0.01}$	
deb	9.352634e-01	8.980572e-01	8.937976e-01	9.091941e-01	9.056390e-01	+
deb1	8.488281e-01	8.487347e-01	8.487930e-01	8.487412e-01	8.488060e-01	+
dtlz1	9.999863e-01	9.999849e-01	9.999805e-01	9.999862e-01	9.999863e-01	+
dtlz2	4.931895e-01	4.920618e-01	4.928217e-01	4.921810e-01	4.928338e-01	+
dtlz3	9.999830e-01	9.999814e-01	9.999796e-01	9.999829e-01	9.999828e-01	+
dtlz5	4.870162e-01	4.858719e-01	4.866265e-01	4.860160e-01	4.866413e-01	+
dtlz6	8.023829e-01	8.018812e-01	8.022268e-01	8.019987e-01	8.022889e-01	+
dtlz7	9.347838e-01	9.347090e-01	9.346518e-01	9.343689e-01	9.347727e-01	+
fonseca	3.131206e-01	3.101664e-01	3.122558e-01	3.102846e-01	3.123668e-01	+
kur1	4.129349e-01	4.101068e-01	4.104965e-01	4.107755e-01	4.120535e-01	+
poloni	2.196234e-01	2.169578e-01	2.173535e-01	2.168089e-01	2.175375e-01	+
qv	8.312798e-02	8.200318e-02	8.276793e-02	8.179511e-02	8.264829e-02	+
schaffer	8.297551e-01	8.292254e-01	8.297922e-01	8.292855e-01	8.297928e-01	+
viennet	9.273507e-01	9.224266e-01	9.270283e-01	9.222818e-01	9.270006e-01	+
viennet2	8.436510e-01	8.370034e-01	8.371788e-01	8.370148e-01	8.371964e-01	+
zdt1	9.634960e-01	9.633612e-01	9.634987e-01	9.633761e-01	9.635077e-01	+
zdt2	9.053143e-01	9.051287e-01	9.053038e-01	9.051585e-01	9.053225e-01	+
zdt3	5.162569e-01	5.158970e-01	5.162385e-01	5.158970e-01	5.162763e-01	+
zdt4	9.922198e-01	9.905372e-01	9.905533e-01	9.921474e-01	9.920972e-01	+
zdt6	9.410801e-01	9.408504e-01	9.409791e-01	9.409020e-01	9.410368e-01	+
Medias	7.224682e-01	7.192473e-01	7.196162e-01	7.199098e-01	7.203893e-01	

ANEXO A
FUNCIONES TEST

Número de variables y de funciones objetivo predeterminadas: [11]

Nombre	Definición	Límites
deb	$f_1(x,y) = x$ $f_2(x,y) = (1+10y)[1-(\frac{x}{1+10y})^{\alpha} - \frac{x}{1+10y}\sin(2\pi q x)]$	$0 \leq x, y \leq 1$ $q = 4, \alpha = 2$ n° variables =2
deb-Bimodal	$f_1(x,y) = x$ $f_2(x,y) = \frac{g(y)}{x}$ $g(y) = 2 - \exp(-(\frac{y-0.2}{0.004})^2) - 0.8\exp(-(\frac{y-0.6}{0.4})^2)$	$0.1 \leq x, y \leq 1$ n° variables=2
fonseca	$f_1(x) = 1 - \mathrm{e}^{-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2}$ $f_2(x) = 1 - \mathrm{e}^{-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2}$	$-4 \leq x_i \leq 4$ n° variables=3
kursave	$f_1(x) = \sum_{i=1}^{n-1}(-10\mathrm{e}^{-0.2\sqrt{x_i^2+x_{i+1}^2}})$ $f_2(x) = \sum_{i=1}^{n}(\lvert x_i\rvert^{0.8} + 5\sin x_i^3)$	$-5 \leq x_i \leq 5$ n° variables=3
poloni	$f_1(x) = -[1 + (A_1 - B_1)^2 + (A_2 - b_2)^2]$ $f_2(x) = -[(x+3)^2 + (y+1)^2]$ $A_1 = 0.5\sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(1.5)$ $A_2 = 1.5\sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2)$ $B_1 = 0.5\sin(x) - 2\cos(x) + \sin(y) - 0.5\cos(y)$ $B_2 = 1.5\sin(x) - \cos(x) + 2\sin(y) - 0.5\cos(y)$	$-\pi \leq x, y \leq \pi$ n° variables=3
schaffer	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$	$-10^5 \leq x \leq 10^5$
viennet	$f_1(x,y) = \frac{(x-2)^2}{2} + \frac{(y+1)^2}{13} + 3$ $f_2(x,y) = \frac{(x+y-3)^2}{36} + \frac{(-x+y+2)^2}{8} - 17$ $f_3(x,y) = \frac{(x+2y-1)^2}{175} + \frac{(2y-x)^2}{17} - 13$	$-4 \leq x \leq 4$ n° variables=2
viennet2	$f_1(x,y) = 0.5(x^2+y^2) + \sin(x^2+y^2)$ $f_2(x,y) = \frac{(3x2u+4)^2}{8} + \frac{(x-y+1)^2}{27} + 15$ $f_3(x,y) = \frac{1}{x^2+y^2+1} - 1.1\exp(-x^2-y^2)$	$-3 \leq x \leq 3$ n° variables=2

Número de funciones objetivo predeterminadoy número de variables modificables[9], [10]

Nombre	Definición	Límites
qv	$f_1(x) = (\frac{1}{n}\sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i)) + 10)^{1/4}$ $f_2(x) = (\frac{1}{n}\sum_{i=1}^{n}((x_i - 1.5)^2 - 10\cos(2\pi(x_i - 1.5)) + 10))^{1/4}$	
zdt1	$f_1 = x_1$ $f_2 = g(x)[1 - \sqrt{\frac{x_1}{g(x)}}]$ $g(x) = 1 + (9\frac{\sum_{i=2}^{n} x_i}{n-1})$	$0 \leq x_i \leq 1$
zdt2	$f_1 = x_1$ $f_2 = g(x)[1 - (\frac{x_1}{g(x)})^2]$ $g(x) = 1 + (9\frac{\sum_{i=2}^{n} x_i}{n-1})$	$0 \leq x_i \leq 1$
zdt3	$f_1 = x_1$ $f_2 = g(x)[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)}sin(10\pi x_1)]$ $g(x) = 1 + (9\frac{\sum_{i=2}^{n} x_i}{n-1})$	$0 \leq x_i \leq 1$
zdt4	$f_1 = x_1$ $f_2 = g(x)[1 - (\frac{x_1}{g(x)})^2]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[x_i^2 - 10cos(4\pi x_i)]$	$0 \leq x_1 \leq 1$ $-5 \leq x_i \leq 5$
zdt6	$f_1 = 1 - exp(-4x_1)\sin^6(6\pi x_1)$ $f_2 = g(x)[1 - (x_1/g(x))^2]$ $g(x) = 1 + (9\frac{\sum_{i=2}^{n} x_i}{n-1})$	$0 \leq x_i \leq 1$

Número de variables de decisión y de funciones objetivo modificables[8]

Nombre	Definición	Límites
dtlz1	$f_1(x) = \frac{1}{2}x_1x_2...x_{M-1}(1+g(x_M))$ $f_2(x) = \frac{1}{2}x_1x_2...x_{M-1}(1-x_{M-1})(1+g(x_M))$ ... $f_{M-1}(x) = \frac{1}{2}x_1(1-x_2)(1+g(x_M))$ $f_M(x) = \frac{1}{2}(1-x_1)(1+g(x_M))$ La función $g(x_M)$ requiere que $\|x_M\| = k$ variables y deben tomar una función con $g >= 0$. Sugierén: $g(x_M) = 100[\|x_M\| + \sum_{x_i \epsilon x_M}((x_i-0.5)^2 - \cos(20\pi(x_i-0.5))]$	s.a. $0 \leq x_i \leq 1$ para $i = 1, 2, ..., n$.
dtlz2	$f_1(x) = (1+g(x_M))\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2})...\cos(\frac{x_{M-2}\pi}{2})\cos(\frac{x_{M-1}\pi}{2})$ $f_2(x) = (1+g(x_M))\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2})...\cos(\frac{x_{M-2}\pi}{2})\sin(\frac{x_{M-1}\pi}{2})$ $f_3(x) = (1+g(x_M))\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2})...\sin(\frac{x_{M-2}\pi}{2})$ ..... $f_{M-1}(x) = (1+g(x_M))\cos(\frac{x_1\pi}{2})\sin(\frac{x_2\pi}{2})$ $f_M(x) = (1+g(x_M))\sin(\frac{x_1\pi}{2})$ donde $g(x_M) = \sum_{x_i \epsilon X_M}(x_i-0.5)^2$	$0 \leq x_i \leq 1$ $i = 1, 2, ..., n,$
dtlz3	$f_1(x) = (1+g(x_M))\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2})...\cos(\frac{x_{M-2}\pi}{2})\cos(\frac{x_{M-1}\pi}{2})$ $f_2(x) = (1+g(x_M))\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2})...\cos(\frac{x_{M-2}\pi}{2})\sin(\frac{x_{M-1}\pi}{2})$ $f_3(x) = (1+g(x_M))\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2})...\sin(\frac{x_{M-2}\pi}{2})$ ..... $f_{M-1}(x) = (1+g(x_M))\cos(\frac{x_1\pi}{2})\sin(\frac{x_2\pi}{2})$ $f_M(x) = (1+g(x_M))\sin(\frac{x_1\pi}{2})$ donde $g(x_M) = 100[\|x_M\| + \sum_{x_i \epsilon X_M}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))]$	$0 \leq x_i \leq 1$ $i = 1, 2, ..., n,$
dtlz5	$f_1(x) = (1+g(x_M))\cos(\frac{\theta_1\pi}{2})\cos(\frac{\theta_2\pi}{2})...\cos(\frac{\theta_{M-2}\pi}{2})\cos(\frac{\theta_{M-1}\pi}{2})$ $f_2(x) = (1+g(x_M))\cos(\frac{\theta_1\pi}{2})\cos(\frac{\theta_2\pi}{2})...\cos(\frac{\theta_{M-2}\pi}{2})\sin(\frac{\theta_{M-1}\pi}{2})$ $f_3(x) = (1+g(x_M))\cos(\frac{\theta_1\pi}{2})\cos(\frac{\theta_2\pi}{2})...\sin(\frac{\theta_{M-2}\pi}{2})$ ..... $f_{M-1}(x) = (1+g(x_M))\cos(\frac{\theta_1\pi}{2})\sin(\frac{\theta_2\pi}{2})$ $f_M(x) = (1+g(x_M))\sin(\frac{\theta_1\pi}{2})$ donde $g(x_M) = \sum_{x_i \epsilon X_M}(x_i-0.5)^2$ $\theta_i = \frac{\pi}{4(1+g(r))}(1+2f(r)x_i) i = 1, 2, ..., (M-1)$	$0 \leq x_i \leq 1, i = 1, 2, ..., n,$
dtlz6	$f_1(x) = (1+g(x_M))\cos(\frac{\theta_1\pi}{2})\cos(\frac{\theta_2\pi}{2})...\cos(\frac{\theta_{M-2}\pi}{2})\cos(\frac{\theta_{M-1}\pi}{2})$ $f_2(x) = (1+g(x_M))\cos(\frac{\theta_1\pi}{2})\cos(\frac{\theta_2\pi}{2})...\cos(\frac{\theta_{M-2}\pi}{2})\sin(\frac{\theta_{M-1}\pi}{2})$ $f_3(x) = (1+g(x_M))\cos(\frac{\theta_1\pi}{2})\cos(\frac{\theta_2\pi}{2})...\sin(\frac{\theta_{M-2}\pi}{2})$ ..... $f_{M-1}(x) = (1+g(x_M))\cos(\frac{\theta_1\pi}{2})\sin(\frac{\theta_2\pi}{2})$ $f_M(x) = (1+g(x_M))\sin(\frac{\theta_1\pi}{2})$ donde $g(x_M) = \sum_{x_i \epsilon X_M} x_i 0.1$ $\theta_i = \frac{\pi}{4(1+g(r))}(1+2f(r)x_i) i = 1, 2, ..., (M-1)$	$0 \leq x_i \leq 1, i = 1, 2, ..., n,$
dtlz7	$f_1(X_1) = x_1,$ $f_2(X_2) = x_2$ .... $f_{M-1}(X_{M-1}) = x_{M-1}$ $f_M(X) = (1+g(X_M)h(f_1, f_2, ..., f_{M-1}, g)$ donde $g(X_M) = 1 + \frac{9}{\|X_M\|}\sum_{x_i \epsilon X_M} x_i$ $h(f_1, f_2, ..., f_{M-1}, g) = M - \sum_{i=1}^{M-1}[\frac{f_i}{1+g}(1+\sin(3\pi f_i)]$	s.a. $0 \leq x_i \leq 1, i = 1, 2, ..., n$

Nota: Cuando el número de variables o el número de funciones objetivo son modificables, en este trabajo se ha determinado que se va a utilizar el valor 2.

REFERENCIAS

[1] J.D.Knowles and D.W.Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimization, volume 1, pages 98-105. IEEE Service Center, 1999.

[2] A.J.Nebro, F.Luna, E.Alba, B.Dorronsoro, J.J.Durillo, and A. Beham. AbYSS: Adapting scatter search to multiobjective optimization. IEEE Transactions on Evolutionary Computation, 12(4): 439-457, 2008.

[3] K.Deb, A.Pratap, S.Agarwal and T.Meyarivan. A fast and elistist multiobjetive genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182-197, 2002.

[4] E.Zitzler, M.Laumanns, and L.Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K.C. Giannakoglou, D.T.Tsahalis, J.Périaux, K.D.Papailiou, and T.Fogarty, editors,Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, pages 95-100, Athens,Greece,2001. International Center for Numerical Methods in Engineering.

[5] P.M.Ortigosa, I.García, and M.Jelasity. Reliability and performance of UEGO, a clustering-based global optimizer. Journal of Global Optimization, 19(3):265-289, 2001.

[6] J.L.Redondo, J.Fernández, A.G.Arrondo, I.García and P.M.Ortigosa. Fixed or variable demand? Does it matter when locating a facility? OMEGA-International Journal of Management Science, 40(1):9-20, 2012.

[7] C.Blum and A.Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys, 35(3):268-308,2003.

[8] K.Deb,L.Thiele, m.Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Zurich, Switzerland, Tech.Rep.112,2001

[9] E.Zitzler, M.Laumanns and T.Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Zurich, Switzerland, Tech.Rep.103,2001

[10] E.Zitzler,K.Deb, and L.Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation, vol.8n0.2,pp.173-195,200

[11] A.Herreros López. Diseño de controladores robustos multiobjetivo por medio de algoritmos genéticos (Tesis doctoral).

[12] K. Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Evolutionary Computation, 7(3):205-230, Fall 1999.

[13] J.D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK-Report 214, 2006.

[14] E. Zitzler. Evolutionary Algorithms for Multiobjective Optimization: Meth- ods and Applications. PhD thesis, ETH Zurich, Switzerland, Germany, 1999.

[15] P. Czyz-zak and A. Jaszkiewicz. Pareto simulated annealing | a metaheuris-tic technique for multiple-objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis, 7(1):34-47, 1998.

[16] K.C. Tan, C.K. Goh, Y.J. Yang, and T.H. Lee. Evolving better population distribution and exploration in evolutionary multi-objective optimization. European Journal of Operational Research, 171(2):463-495, 2006.

[17] E. Zitzler, L. Thiele, M. Laumanns, C. M. Foneseca, and V. Grunert da Fon- seca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation, 7(2):117- 132, 2003.

[18] F.J.Solis and R.J.B. Wets. Minimization by random search techniques. Mathematics of Operations Research. 6(1):19-30,1981.

[19] L.Thiele. Using the Monitor in PISA.
http://www.tik.ee.ethz.ch/pisa/monitor/monitor_documentation.pdf

[20] D.B.Fogel, Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, Wiley-IEEE Press, 2006.

[21] 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), Mexico, 1998.

[22] D.E.Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, New York, 1989.

UNIVERSIDAD DE ALMERÍA

Resumen/Abstract

This paper intends to implement a multi-objective optimization meta-heuristic algorithm. The multi-objective optimization is a very important area of research because most real-world problems have multiple objectives. Instead of providing a single solution, the procedures that apply to multi-objective optimization problems generate a set of compromise solutions, generally known as Pareto optimal solutions, from which a decision maker chooses one. Different meta-heuristics, including evolutionary algorithms, have become the main method to explore the Pareto front in multi-objective optimization problems that are too complex to be solved by exact methods.

Given the many heuristics to solve multi-objective problems, in this paper, besides the implementation of a new algorithm, efficiency will be defined and effectiveness metrics and used to compare our implementation with some of the best known in the literature.

En este trabajo se pretende implementar un algoritmo metaheurístico de optimización multiobjetivo. La optimización multiobjetivo es un área de investigación muy importante debido a que la mayoría de los problemas del mundo real tienen objetivos múltiples. En lugar de proporcionar una única solución, los procedimientos que se aplican a problemas de optimización multiobjetivo generan un conjunto de soluciones compromiso, generalmente conocidas como soluciones Pareto óptimas, de entre las cuales un agente decisor escogerá una. Diferentes metaheurísiticas, incluyendo los algoritmos evolutivos, se han convertido en el principal método para explorar el frente de Pareto en los problemas de optimización multiobjetivo que son demasiado complejos para ser resueltos por métodos exactos.

Dado que existen numerosas heurísticas para resolver problemas multiobjetivo, en este trabajo, aparte de la implementación de un algoritmo nuevo, se definirán métricas de eficiencia y eficacia y se utilizarán para comparar nuestra implementación con algunas de las más conocidas en la literatura..