

Un nuevo método de estimación de densidad para problemas de optimización con tres objetivos usando metaheurísticas

Alejandro Santiago^{1,3}, Bernabé Dorronsoro¹, Antonio J. Nebro², and Héctor Fraire³

¹ Universidad de Cádiz
alejandro.santiagopi@mail.uca.es,
bernabe.dorronsoro@uca.es,

² Universidad de Málaga
antonio@lcc.uma.es

³ Instituto Tecnológico de Ciudad Madero, México
automatas2002@yahoo.com.mx

Resumen Muchos de los algoritmos de optimización multi-objetivo más populares son poco eficaces al tratar con problemas de tres o más objetivos. Esto se debe en general al uso de estimadores de densidad, como la distancia de *crowding* de NSGA-II, que fueron diseñados cuando el principal reto era optimizar problemas de dos objetivos. En este artículo presentamos un nuevo estimador de densidad que permite a los algoritmos que utilizan archivos externos mejorar notablemente la diversidad de soluciones de los frentes al resolver problemas de tres objetivos. Hemos evaluado el estimador en dos variantes de algoritmos basados en archivo, SMPSO y MOCell, comparado los resultados obtenidos con los de las versiones originales de los mismos sobre un banco de pruebas compuesto por 16 problemas de tres objetivos. Los resultados muestran que los frentes obtenidos con nuestra propuesta mejoran de forma significativa a los que generan los algoritmos originales.

Palabras clave: Optimización multi-objetivo, estimador de densidad, diversidad, problemas 3D.

1. Introducción

La optimización multi-objetivo es la disciplina que trata la resolución de problemas de optimización que se caracterizan por tener dos o más objetivos (o funciones) que están en conflicto entre sí, de forma que para una solución óptima la mejora en uno de ellos implica un empeoramiento del resto. Por este motivo, no suele existir una única solución a estos problemas, sino un conjunto de soluciones denominadas *conjunto de óptimos de Pareto*. La representación de este conjunto en el espacio de objetivos se denomina *frente de Pareto*. Obtener el frente de Pareto es lo que persigue la optimización multi-objetivo. Sin embargo,

A. Santiago, B. Dorronsoro, A. J. Nebro, H. Fraire

en la práctica, cuando se usan técnicas aproximadas como las metaheurísticas, lo que suele buscarse es una aproximación o subconjunto de dicho frente con la suficiente calidad para que un tomador de decisiones (un experto en el problema) tenga un buen abanico de alternativas. Para mejor comprensión debe contener soluciones lo más cercanas posibles al frente de Pareto y a su vez han de estar uniformemente distribuidas. Estas dos propiedades son conocidas como convergencia y diversidad, respectivamente. Además, el número de soluciones del frente está acotado, por lo que el algoritmo debe seleccionar durante la búsqueda aquellas soluciones que optimicen ambas propiedades.

Si nos centramos en el aspecto de la diversidad, la estrategia más habitual usada por las metaheurísticas multi-objetivo es aplicar un estimador de densidad que permita determinar qué soluciones son las mejores (para mantenerlas) o peores (para desecharlas) en vistas a conseguir un frente lo más diverso posible. Probablemente, el método más utilizado es la distancia de *crowding* del algoritmo NSGA-II [2], por su gran rendimiento al resolver problemas de dos objetivos (2D) y a su baja complejidad $O(kn \log n)$, donde n es el número de soluciones y k el de los objetivos. En contraposición, es bien sabido que pierde su eficiencia al aumentar las dimensiones del problema [4] ya que la distancia *crowding* se basa en medir cuboides respecto a los individuos más cercanos, lo que sólo funciona bien con dos objetivos. Los estimadores basados en la contribución al hipervolumen también se han aplicado en muchos algoritmos [1,12,13], ya que suponen una mejora al resolver problemas de tres o más objetivos respecto a la distancia de *crowding*; sin embargo, la complejidad del cálculo del hipervolumen crece exponencialmente con el número de objetivos.

La principal contribución de este artículo es la propuesta de un nuevo estimador de densidad que permite mejorar la diversidad de los frentes en metaheurísticas basadas en archivo externo para problemas de tres objetivos (3D). Nuestra propuesta la Desviación Especial Propagada se basa enteramente en la posición de las soluciones sobre la actual aproximación al frente de Pareto y es de tiempo polinomial. Se han hecho algunas propuestas similares eliminando los vecinos más cercanos en [7,17] pero éstas no toman en cuenta la posición en el espacio objetivo multidimensional (la distancia de *crowding* no toma en cuenta la relación entre diferentes objetivos).

Para evaluar el estimador de densidad que hemos diseñado se ha implementado en dos metaheurísticas del estado del arte que usan un archivo externo basado en la distancia de *crowding*. Los algoritmos son SMPPO [9] y MOCCell [10], que han mostrado una notable capacidad para resolver problemas 2D pero que sufren dificultades para producir frentes 3D con una buena diversidad de soluciones.

El resto del artículo se estructura como sigue. En la Sección 2, se presenta nuestro nuevo estimador de densidad. Después, se explica en la Sección 3 el procedimiento seguido para generar y analizar los resultados presentados en este artículo. Los resultados se presentan y discuten en la Sección 4. Para terminar, el artículo resume nuestras principales conclusiones y sugerencias de trabajo futuro en la Sección 5.

Un nuevo estimador de densidad para metaheurísticas multi-objetivo

2. Estimador de densidad propuesto

El Algoritmo 1 muestra el funcionamiento de la Desviación Espacial Propagada (DEP) propuesta. La variable A es el conjunto de soluciones a evaluar y es de cardinalidad n (línea 1). El algoritmo comienza asignando una aptitud de 0 a todas las soluciones del conjunto (líneas 4 a 6). Después, en las líneas 7-11 la función $\text{ordena}(A,m)$ localiza los extremos por objetivo (peor y mejor solución en el objetivo $m \in K$) asignándoles la mejor aptitud posible $-\infty$. Tras esto se calculan en la línea 12 todas las distancias euclidianas normalizadas con los valores máximos y mínimos encontrados en cada objetivo m en A , almacenándolas en la matriz M . La línea 13 localiza la distancia máxima D_{max} y mínima D_{min} entre soluciones distintas del conjunto A . En las líneas 14 a la 21 se penaliza cada solución considerando el momento central entre las distancias producidas por cada solución i contra las demás soluciones j del conjunto y $(D_{max} - D_{min})$.

Entrada: Un conjunto de soluciones no dominadas A
Salida : Conjunto A con aptitud DEP en cada solución

```

1   $n \leftarrow |A|$ 
2  foreach solución  $i \in A$  do
3  |  $A[i].\text{aptitud} \leftarrow 0$ 
4  end
5  for  $m = 1$  hasta  $k$  do
6  |  $\text{ordena}(A,m)$ 
7  |  $A[1].\text{aptitud} \leftarrow -\infty$ 
8  |  $A[n].\text{aptitud} \leftarrow -\infty$ 
9  end
10 Calcula la matriz de distancias normalizadas  $M$ 
11 Encuentra  $D_{max}$  y  $D_{min}$  en  $M$  entre soluciones  $i$  y  $j$ ,  $i \neq j$ 
12 foreach solución  $i \in A$  do
13 |  $\text{temp} \leftarrow 0,0$ 
14 | foreach solución  $j \neq i \in A$  do
15 | |  $\text{temp} = \text{temp} + (M[i][j] - (D_{max} - D_{min}))^2$ 
16 | end
17 |  $\text{temp} = \frac{\text{temp}}{n-1}$ 
18 |  $A[i].\text{aptitud} = A[i].\text{aptitud} + \sqrt{\text{temp}}$ 
19 end
20 foreach solución  $i \in A$  do
21 |  $\text{ordena}(M[i])$ 
22 | for  $j = k + 1$  hasta 2 do
23 | |  $A[i].\text{aptitud} = A[i].\text{aptitud} + ((D_{max} - D_{min}) / (M[i][j] + D_{min}))$ 
24 | end
25 end

```

Algorithm 1: Desviación Espacial Propagada

A. Santiago, B. Dorronsoro, A. J. Nebro, H. Fraire

Por último las líneas 22 a 27 penalizan la cercanía de las k soluciones más cercanas a cada solución i del conjunto, facilitando la propagación en cada dimensión objetivo. Al utilizarse un archivo de soluciones no-dominadas, esta operación se realiza sobre el actual frente conocido manteniendo las mejores soluciones no repetidas respecto al máximo tamaño del archivo. A diferencia de la distancia *crowding*, se busca minimizar la Desviación Espacial Propagada. La complejidad del algoritmo propuesto está acotada por su operación más costosa, el cálculo de las desviaciones $O(n^2)$ siendo de complejidad polinomial.

3. Análisis experimental

Para probar la calidad del estimador de densidad propuesto se ha diseñado un experimento compuesto por algoritmos que utilizan archivo con distancia *crowding*, reemplazandola por la Desviación Espacial Propagada, y se ha realizado una comparativa de las dos versiones de los algoritmos al resolver un conjunto de problemas de prueba de tres objetivos.

A continuación se detallan los problemas utilizados, los algoritmos que se han seleccionado y la metodología experimental que se ha seguido.

3.1. Problemas

Como bancos de prueba para evaluar la solución que proponemos hemos elegido las familias de problemas conocidas como DTLZ [3] y WFG [6], que constan de 7 y 9 problemas respectivamente. En conjunto abarcan problemas de diferentes características: frentes convexos, cóncavos, lineales, particionados, degenerados (frentes con menor dimensión que el número de objetivos), etc.

Todos los problemas son escalables en el número de objetivos, por lo que en nuestro estudio se han usado sus variantes con tres objetivos, manteniendo el número de variables con los valores propuestos originalmente.

3.2. Algoritmos

Los algoritmos seleccionados para la experimentación son SMPPO [9] y MO-Cell [10], que se describen brevemente a continuación.

SMPPO (Speed-constrained Multi-objective PSO) es un PSO multi-objetivo cuya principal característica es el uso de una estrategia para limitar la velocidad de las partículas con el fin de mantener la posición de las partículas en rangos razonables cuando la velocidad de las partículas es demasiado elevada. Incluye un operador de mutación polinomial que actúa a modo de *turbulencia* y que se aplica un porcentaje de las partículas. Por último, incluye un archivo externo que se usa para almacenar las soluciones no-dominadas que se van encontrando y para la selección de líderes.

MOCeCell (Multi-Objective Cellular genetic algorithm) es un algoritmo evolutivo celular, por lo que tiene de estructura poblacional basada en malla y la selección de individuos se hace dentro de una estructura de vecindario que puede

Un nuevo estimador de densidad para metaheurísticas multi-objetivo

tener diversas topologías; en MOCcell la malla es cuadrada y el vecindario de una solución está compuesto de todas aquellas soluciones que está a distancia 1 en la malla. Al igual que SMPSO, MOCcell usa un archivo basado en *crowding* que usa además para la selección de soluciones para la recombinación: una se toma del vecindario y otra del archivo.

Ambos algoritmos han mostrado en varios trabajos ser competitivos en problemas 2D, tanto de bancos de prueba como en problemas reales, como SMPSO al abordar problemas de acoplamiento molecular (*protein docking*) [5] y MOCcell al resolver problemas de diseño de estructuras civiles [14]. Sin embargo, tienen dificultades para conseguir frentes con buena diversidad en problemas 3D.

Las versiones de estos algoritmos con Desviación Espacial Propagada se han denominado MOCcell-DEP y SMPSO-DEP. Para la implementación tanto de los algoritmos como los problemas de prueba se utilizó jMetal 5 [11].

3.3. Metodología Experimental

Para comparar los algoritmos se han llevado a cabo 50 ejecuciones independientes por configuración y se han aplicado tres indicadores de calidad para determinar el grado de convergencia y diversidad de los frentes obtenidos. En concreto, se ha usado el indicador Epsilon aditivo [16], el denominado *spread* generalizado [8] y el hipervolumen [15]; el primero mide convergencia, el segundo diversidad, y el tercero tiene en cuenta ambas propiedades.

Las siguientes tablas de resultados incluyen las medianas y rangos intercuartílicos de los valores de los indicadores. En ellas los mejores resultados se destacan con un fondo gris. Se ha usado el test de suma de rangos de Wilcoxon con un nivel de significancia estadística del 95%, mostrándose un triángulo negro ▲ cuando algoritmo original es significativamente mejor que su variante con el esquema propuesto es, un triángulo blanco ▽ cuando sucede contrario, y el símbolo - cuando las diferencias no son significativas.

Los algoritmos han sido configurados con los valores por defecto habituales. El tamaño del enjambre en SMPSO y de la malla en MOCcell es de 100 individuos, y los archivos en ambos tienen 100 soluciones. La condición de parada es llevar a cabo 50,000 evaluaciones.

4. Resultados

Los resultados obtenidos al aplicar cada una de las métricas a los frentes obtenidos por los algoritmos se muestran en la Tabla 1. Comenzamos discutiendo los resultados según el hipervolumen. Si nos fijamos en MOCcell, la versión DEP ha mejorado a la original en 12 de los 17 problemas abordados. Los valores obtenidos en los problemas DTLZ3 y DTLZ6 indican que MOCcell ha sido incapaz de encontrar soluciones que dominen al punto de referencia usado para calcular el hipervolumen. Podemos observar que todas las diferencias son significativas de acuerdo al test de Wilcoxon. En el caso de SMPSO, la versión DEP consigue mejorar a la original también en 12 problemas, siendo también significativas

A. Santiago, B. Dorronsoro, A. J. Nebro, H. Fraire

Tabla 1. Resultados experimentales

Hipervolumen: Mediana y rango intercuartílico

Problema	MOCcell	MOCcell-DEP	SMPSO	SMPSO-DEP
DTLZ1_3D	$7,63e-01_{1,9e-02}$	$7,84e-01_{3,7e-03} \nabla$	$7,36e-01_{8,0e-03}$	$7,72e-01_{3,4e-03} \nabla$
DTLZ2_3D	$3,78e-01_{7,0e-03}$	$4,07e-01_{3,3e-03} \nabla$	$3,58e-01_{8,7e-03}$	$3,88e-01_{5,3e-03} \nabla$
DTLZ3_3D	$0,00e+00_{0,0e+00}$	$0,00e+00_{7,6e-02}$	$3,53e-01_{1,7e-02}$	$3,88e-01_{8,1e-03} \nabla$
DTLZ4_3D	$3,76e-01_{7,8e-03}$	$4,03e-01_{4,1e-03} \nabla$	$3,63e-01_{8,8e-03}$	$3,85e-01_{4,8e-03} \nabla$
DTLZ5_3D	$9,39e-02_{26,3e-08}$	$9,29e-02_{2,2e-04} \blacktriangle$	$9,39e-02_{26,0e-05}$	$9,27e-02_{4,9e-04} \blacktriangle$
DTLZ6_3D	$0,00e+00_{0,0e+00}$	$0,00e+00_{0,0e+00}$	$9,49e-02_{4,0e-05}$	$9,37e-02_{7,7e-04} \blacktriangle$
DTLZ7_3D	$2,81e-01_{8,8e-03}$	$2,96e-01_{2,9e-03} \nabla$	$2,78e-01_{7,6e-03}$	$2,99e-01_{2,7e-03} \nabla$
WFG1_3D	$8,90e-01_{6,8e-03}$	$8,96e-01_{7,3e-03} \nabla$	$1,17e-01_{1,8e-02}$	$9,28e-02_{6,6e-02} \blacktriangle$
WFG2_3D	$8,98e-01_{4,9e-03}$	$9,16e-01_{1,2e-03} \nabla$	$8,83e-01_{5,6e-03}$	$9,09e-01_{1,9e-03} \nabla$
WFG3_3D	$3,17e-01_{2,5e-03}$	$3,00e-01_{4,6e-03} \blacktriangle$	$3,01e-01_{5,5e-03}$	$2,75e-01_{7,9e-03} \blacktriangle$
WFG4_3D	$3,78e-01_{5,5e-03}$	$4,07e-01_{1,8e-03} \nabla$	$3,27e-01_{8,7e-03}$	$3,58e-01_{4,6e-03} \nabla$
WFG5_3D	$3,53e-01_{7,5e-03}$	$3,78e-01_{1,9e-03} \nabla$	$3,41e-01_{5,5e-03}$	$3,68e-01_{3,1e-03} \nabla$
WFG6_3D	$3,79e-01_{1,3e-02}$	$4,08e-01_{4,5e-03} \nabla$	$3,47e-01_{1,2e-02}$	$3,87e-01_{5,6e-03} \nabla$
WFG7_3D	$3,73e-01_{4,1e-03}$	$4,01e-01_{2,7e-03} \nabla$	$3,42e-01_{1,1e-02}$	$3,80e-01_{7,3e-03} \nabla$
WFG8_3D	$2,44e-01_{5,7e-03}$	$2,62e-01_{6,4e-03} \nabla$	$2,33e-01_{8,5e-03}$	$2,62e-01_{7,3e-03} \nabla$
WFG9_3D	$3,65e-01_{1,0e-02}$	$3,96e-01_{1,0e-02} \nabla$	$3,55e-01_{9,8e-03}$	$3,83e-01_{3,9e-03} \nabla$

Epsilon: Mediana y rango intercuartílico

Problema	MOCcell	MOCcell-DEP	SMPSO	SMPSO-DEP
DTLZ1_3D	$1,04e-01_{4,6e-02}$	$6,58e-02_{7,0e-03} \nabla$	$1,17e-01_{8,5e-03}$	$7,86e-02_{5,8e-03} \nabla$
DTLZ2_3D	$1,19e-01_{2,7e-02}$	$7,32e-02_{26,0e-03} \nabla$	$1,32e-01_{1,9e-02}$	$8,87e-02_{6,8e-03} \nabla$
DTLZ3_3D	$1,45e+00_{9,6e-01}$	$1,31e+00_{1,8e+00}$	$1,42e-01_{2,8e-02}$	$8,87e-02_{9,4e-03} \nabla$
DTLZ4_3D	$1,17e-01_{2,6e-02}$	$6,76e-02_{7,7e-03} \nabla$	$1,15e-01_{2,2e-02}$	$8,26e-02_{8,8e-03} \nabla$
DTLZ5_3D	$5,86e-03_{39,6e-04}$	$1,16e-02_{1,8e-03} \blacktriangle$	$5,42e-03_{2,6e-04} \blacktriangle$	$1,14e-02_{2,3e-03} \blacktriangle$
DTLZ6_3D	$7,42e-01_{1,4e-01}$	$7,11e-01_{2,2e-01}$	$5,19e-03_{3,6e-04}$	$1,19e-02_{3,6e-03} \blacktriangle$
DTLZ7_3D	$1,08e-01_{4,3e-02}$	$6,25e-02_{9,0e-03} \nabla$	$1,34e-01_{4,4e-02}$	$5,84e-02_{6,0e-03} \nabla$
WFG1_3D	$1,61e-01_{1,2e-02}$	$1,60e-01_{1,0e-02} \nabla$	$7,30e-01_{9,1e-03}$	$7,48e-01_{3,0e-02} \blacktriangle$
WFG2_3D	$1,08e-01_{1,4e-02}$	$6,04e-02_{8,7e-03} \nabla$	$1,07e-01_{1,2e-02}$	$6,29e-02_{8,5e-03} \nabla$
WFG3_3D	$6,69e-02_{2,0e-02}$	$5,24e-02_{7,7e-03} \nabla$	$1,11e-01_{2,2e-02}$	$7,69e-02_{1,2e-02} \nabla$
WFG4_3D	$1,18e-01_{1,6e-02}$	$7,60e-02_{2,8e-03} \nabla$	$1,41e-01_{1,7e-02}$	$9,78e-02_{9,2e-03} \nabla$
WFG5_3D	$1,31e-01_{2,3e-02}$	$8,70e-02_{2,8e-03} \nabla$	$1,36e-01_{1,9e-02}$	$1,03e-01_{6,7e-03} \nabla$
WFG6_3D	$1,20e-01_{2,6e-02}$	$7,96e-02_{0,3e-03} \nabla$	$1,28e-01_{1,7e-02}$	$8,90e-02_{7,1e-03} \nabla$
WFG7_3D	$1,18e-01_{2,1e-02}$	$8,00e-02_{6,4e-03} \nabla$	$1,34e-01_{2,2e-02}$	$8,84e-02_{7,1e-03} \nabla$
WFG8_3D	$2,70e-01_{5,4e-02}$	$2,53e-01_{6,8e-02} \nabla$	$2,78e-01_{2,6e-02}$	$2,41e-01_{3,7e-02} \nabla$
WFG9_3D	$1,16e-01_{1,8e-02}$	$7,84e-02_{25,2e-03} \nabla$	$1,27e-01_{2,3e-02}$	$8,11e-02_{7,0e-03} \nabla$

Spread Generalizado: Mediana y rango intercuartílico

Problema	MOCcell	MOCcell-DEP	SMPSO	SMPSO-DEP
DTLZ1_3D	$4,23e-01_{9,1e-01}$	$1,05e-01_{1,6e+00} \nabla$	$3,74e-01_{4,5e-02}$	$1,01e-01_{1,8e-02} \nabla$
DTLZ2_3D	$4,32e-01_{5,0e-02}$	$1,23e-01_{2,5e-02} \nabla$	$3,97e-01_{3,6e-02}$	$1,32e-01_{1,8e-02} \nabla$
DTLZ3_3D	$9,53e-01_{1,8e-01}$	$8,79e-01_{3,3e-01} \nabla$	$4,00e-01_{6,4e-02}$	$1,29e-01_{1,9e-02} \nabla$
DTLZ4_3D	$4,32e-01_{5,2e-02}$	$1,14e-01_{2,6e-02} \nabla$	$4,13e-01_{4,4e-02}$	$1,37e-01_{2,2e-02} \nabla$
DTLZ5_3D	$1,22e-01_{1,9e-02}$	$1,73e-01_{2,3e-02} \blacktriangle$	$9,66e-02_{1,9e-02}$	$1,46e-01_{2,7e-02} \blacktriangle$
DTLZ6_3D	$6,24e-01_{4,2e-02}$	$3,88e-01_{6,4e-02} \nabla$	$9,94e-02_{1,8e-02}$	$1,43e-01_{2,6e-02} \nabla$
DTLZ7_3D	$4,66e-01_{8,3e-02}$	$1,58e-01_{2,6e-02} \nabla$	$4,65e-01_{4,9e-02}$	$1,42e-01_{2,0e-02} \nabla$
WFG1_3D	$5,23e-01_{6,3e-02}$	$1,68e-01_{2,5e-02} \nabla$	$8,13e-01_{2,7e-02}$	$6,48e-01_{3,3e-02} \nabla$
WFG2_3D	$4,63e-01_{7,0e-02}$	$1,52e-01_{3,2e-02} \nabla$	$4,37e-01_{7,8e-02}$	$1,56e-01_{3,1e-02} \nabla$
WFG3_3D	$5,31e-01_{5,4e-02}$	$1,78e-01_{2,0e-02} \nabla$	$5,33e-01_{3,2e-02}$	$1,88e-01_{2,5e-02} \nabla$
WFG4_3D	$3,70e-01_{6,1e-02}$	$1,05e-01_{2,6e-02} \nabla$	$3,60e-01_{3,7e-02}$	$1,25e-01_{1,7e-02} \nabla$
WFG5_3D	$3,93e-01_{4,2e-02}$	$1,02e-01_{3,6e-02} \nabla$	$3,54e-01_{3,3e-02}$	$1,09e-01_{1,9e-02} \nabla$
WFG6_3D	$4,06e-01_{5,9e-02}$	$1,07e-01_{2,3e-02} \nabla$	$3,97e-01_{4,4e-02}$	$1,05e-01_{2,5e-02} \nabla$
WFG7_3D	$4,05e-01_{6,5e-02}$	$1,13e-01_{3,1e-02} \nabla$	$3,85e-01_{5,0e-02}$	$1,06e-01_{2,3e-02} \nabla$
WFG8_3D	$4,48e-01_{7,2e-02}$	$1,41e-01_{2,7e-02} \nabla$	$4,17e-01_{4,5e-02}$	$1,53e-01_{1,8e-02} \nabla$
WFG9_3D	$3,82e-01_{5,7e-02}$	$1,05e-01_{2,3e-02} \nabla$	$3,48e-01_{4,6e-02}$	$1,06e-01_{1,6e-02} \nabla$

todas las diferencias. Hay que reseñar que los frentes de los problemas DTLZ5 y DTLZ6 son degenerados, lo que implica que son 2D, por lo que en ellos la versión original de SMPSO consigue mejores valores.

Un nuevo estimador de densidad para metaheurísticas multi-objetivo

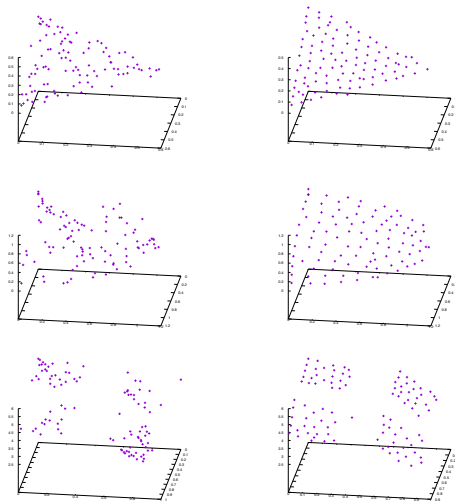


Figura 1. Frentes con mediana en hipervolumen en los problemas DTLZ1, DTLZ3 y DTLZ 7 obtenidos por SMPSO (izquierda) y SMPSO-DEP (derecha).

Si prestamos atención a los valores obtenidos por los indicadores específicos de convergencia y diversidad los resultados de las versiones que utilizan DEP son aún superiores a los mostrados para el hipervolumen. El algoritmo MOCeL-DEP mejora a su versión original en todos los problemas menos el DTLZ5 (uno de los frentes degenerados), tanto para la métrica epsilon como para *spread* generalizado. En el caso de SMPSO, la variante que utiliza DEP mejora también a la original en la mayoría de los problemas para ambas métricas. Como se puede observar en la tabla, la gran mayoría de las diferencias son significativas.

Para ilustrar las mejoras de calidad en los algoritmos con el nuevo estimador de densidad, en la Figura 1 mostramos los frentes correspondientes a la mediana del hipervolumen obtenidos con SMPSO y SMPSO-DEP en los problemas DTLZ1, DTLZ3 y DTLZ7. Se puede observar claramente la mejora en diversidad que conlleva el uso del archivo DEP propuesto: mientras que en SMPSO aparecen muchas zonas vacías y la distribución no es uniforme, la versión DEP sí consigue abarcar todo el frente.

A. Santiago, B. Dorronsoro, A. J. Nebro, H. Fraire

En Figura 2 se incluyen los frentes obtenidos por MOCcell y MOCcell-DEP en los problemas DTLZ1, DTLZ2 y DTLZ7, en la que también se observa la notable mejora en la diversidad de los frentes producidos al utilizar DEP.

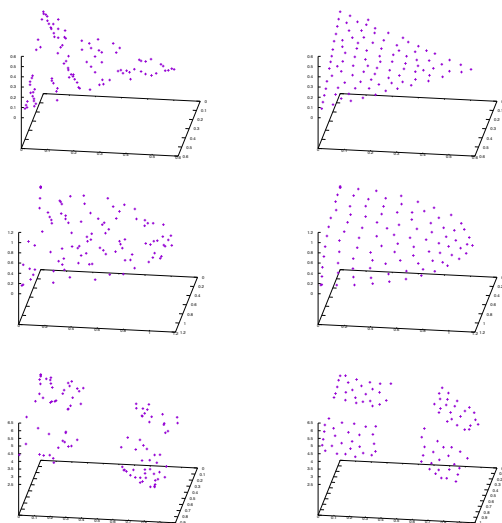


Figura 2. Frentes con mediana en hipervolumen en los problemas DTLZ1, DTLZ2 y DTLZ 7 obtenidos por MOCcell (izquierda) y MOCcell-DEP (derecha).

5. Conclusiones y trabajo futuro

En este artículo hemos presentado un nuevo estimador de densidad, denominado Desviación Espacial Propagada (DEP), para problemas de tres objetivos. Para determinar su efectividad lo hemos incorporado a los archivos externos de dos metaheurísticas, SMPSO y MOCcell, que en su configuración por defecto usan la distancia de *crowding* en dichos archivos. Dado que estos algoritmos son eficaces para problemas de dos dimensiones, nuestro interés ha sido estudiar su

Un nuevo estimador de densidad para metaheurísticas multi-objetivo

rendimiento en problemas de tres objetivos, para los que presentan deficiencias utilizando su archivo original. El estudio de rendimiento se ha basado en resolver los 16 problemas que conforman los bancos de prueba DTLZ y WFG, en su configuración con tres objetivos.

Los resultados del estudio experimental llevado a cabo han mostrado que las versiones de los algoritmos con nuestro estimador de densidad han conseguido mejorar notablemente los frentes producidos por los algoritmos originales en la mayor parte de los casos. La exploración visual de los frentes obtenidos ha corroborado que al incluir DEP en los algoritmos se pasa de obtener unas distribuciones muy pobres a otras muy satisfactorias, en las que las soluciones están uniformemente repartidas.

Podemos concluir por tanto que el uso del estimador Desviación Espacial Propagada permite mejorar claramente los resultados de los algoritmos en lo que respecta a la diversidad de soluciones en el frente.

Como líneas de trabajo futuro se plantea estudiar el rendimiento de nuestro estimador en problemas con más de tres objetivos. También resulta muy interesante la posibilidad de aplicar DEP a algoritmos no basados en archivo, como NSGA-II.

Agradecimientos

Alejandro Santiago agradece a CONACyT México por el apoyo número 360199. Bernabé Dorronsoro agradece al Ministerio de Economía y Competitividad su apoyo mediante los contratos TIN2014-60844-R (el proyecto SAVANT) y el programa Ramón y Cajal (contrato RYC-2013-13355). Antonio J. Nebro agradece los apoyos TIN2011-25840 (Ministerio de Ciencia e Innovación) y P11-TIC-7529, P12-TIC-1519 (Plan Andaluz de Investigación, Desarrollo e Innovación).

Referencias

1. Nicola Beume, Boris Naujoks, and Michael Emmerich. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653 – 1669, 2007.
2. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr 2002.
3. Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multi-objective optimization. 2001.
4. F-A. Fortin and M. Parizéau. Revisiting the nsga-ii crowding-distance computation. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 623–630, New York, NY, USA, 2013. ACM.
5. María Jesús García-Godoy, Esteban López-Camacho, José García-Nieto, Antonio J. Nebro, and José F. Aldana-Montes. Solving molecular docking problems with multi-objective metaheuristics. *Molecules*, 20(6):10154, 2015.

A. Santiago, B. Dorronsoro, A. J. Nebro, H. Fraire

6. Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings*, chapter A Scalable Multi-objective Test Problem Toolkit, pages 280–295. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
7. Saku Kukkonen and Kalyanmoy Deb. A Fast and Effective Method for Pruning of Non-dominated Solutions in Many-Objective Problems. In Thomas Philip Runarsson, Hans-Georg Beyer, Edmund Burke, Juan J. Merelo-Guervós, L. Darrell Whitley, and Xin Yao, editors, *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference*, pages 553–562. Springer. Lecture Notes in Computer Science Vol. 4193, Reykjavik, Iceland, September 2006.
8. 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, Aug 2008.
9. A. J. Nebro, J. J. Durillo, J. García-Nieto, C. A. Coello Coello, F. Luna, and E. Alba. Smpso: A new pso-based metaheuristic for multi-objective optimization. In *2009 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MCDM 2009)*, pages 66–73. IEEE Press, 2009.
10. A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba. Mocell: A cellular genetic algorithm for multiobjective optimization. *International Journal Intelligent Systems*, 24(7):726–746, 2009.
11. Antonio J. Nebro, Juan J. Durillo, and Matthieu Vergne. Redesigning the jmetal multi-objective optimization framework. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO Companion '15*, pages 1093–1100, New York, NY, USA, 2015. ACM.
12. L. While and L. Bradstreet. Applying the WFG algorithm to calculate incremental hypervolumes. In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2012.
13. L. While, L. Bradstreet, and L. Barone. A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95, Feb 2012.
14. Gustavo Zavala, Antonio J. Nebro, Francisco Luna, and Carlos A. Coello Coello. Structural design using multi-objective metaheuristics. comparative study and application to a real-world problem. *Structural and Multidisciplinary Optimization*, 53(3):545–566, 2016.
15. E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
16. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.
17. Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.