



Título del artículo.

Algoritmo evolutivo multi-objetivo basado en descomposición con múltiples funciones de escalarización.

Título del artículo en idioma Inglés.

Multiobjective evolutionary algorithm based on decomposition with multiple scalarization functions.

Autores.

Cirilo Tino Salgado
Carlos Artemio Coello Coello
Miriam Pescador Rojas

Referencia bibliográfica:

MLA

Tino Salgado, Cirilo, Carlos Artemio Coello Coello y Miriam Pescador Rojas. Algoritmo evolutivo multi-objetivo basado en descomposición con múltiples funciones de escalarización. *Tlamati* 9.2, (2018): 5-13. Print.

APA

Tino Salgado, C., Coello Coello, C. A. y Pescador Rojas, M. (2018). Algoritmo evolutivo multi-objetivo basado en descomposición con múltiples funciones de escalarización. *Tlamati*, 9(2), 5-13.

ISSN: 2007-2066.

Publicado el 31 de diciembre del 2018

© 2018 Universidad Autónoma de Guerrero

Dirección General de Posgrado e Investigación

Dirección de Investigación

TLAMATI, es una publicación semestral de la Dirección de Investigación de la Universidad Autónoma de Guerrero. El contenido de los artículos es responsabilidad exclusiva de los autores y no refleja de manera alguna el punto de vista de la Dirección de Investigación de la UAGro. Se autoriza la reproducción total o parcial de los artículos previa cita de nuestra publicación.



Algoritmo evolutivo multi-objetivo basado en descomposición con múltiples funciones de escalarización

Cirilo Tino Salgado^{1*}
 Carlos Artemio Coello Coello²
 Miriam Pescador Rojas².

¹Universidad Autónoma de Guerrero. Unidad Académica de Ciencias y Tecnologías de la Información. Av. de las Colinas #37-A, Fracc. Las Playas, C. P. 39390, Acapulco, Guerrero-México. Tel. +(52) 7444872655

²Instituto Politécnico Nacional. Centro de Investigaciones y Estudios Avanzados. Av. Instituto Politécnico Nacional 2508, San Pedro Zacatenco, C. P. 07360, Ciudad de México, CDMX

*Autor de correspondencia
 citino@uagro.mx

Resumen

Para Problemas de Optimización Multi-Objetivo [POM], es decir, problemas con más de una función objetivo, una alternativa de solución es el uso de los Algoritmos Evolutivos, basados en métodos como: Pareto, Indicadores y en Descomposición.

Los Algoritmos Evolutivos Basados en Descomposición [MOEA/D] dividen el POM en un conjunto de Problemas de Optimización de un Solo Objetivo [POS], los cuales son optimizados simultáneamente haciendo uso de operadores genéticos y funciones de escalarización.

En el presente artículo proponemos la implementación de un MOEA/D con múltiples funciones de escalarización, en el cual se generan vectores de peso a través de tres diferentes métodos: *Simplex Lattice Design* [SLD], *Uniform Design* [UD] y la combinación del método *UD* con el método *Hammersley* [UDH]. Además, consideramos la creación de subpoblaciones continuas y aleatorias para mantener la diversidad de soluciones.

Las funciones de escalarización implementadas en el MOEA/D con múltiples funciones de escalarización son: *Weighted Sum* [WS], *Weighted Tchebycheff* [TCH], *Augmented Weighted Tchebycheff* [AT], *Achievement Scalarization* [ASF] y *Penalty-Based Boundary Intersection* [PBI]. Es importante enfatizar que a cada subpoblación le es asignada una función de escalarización. Asimismo, se explora y se explota el espacio de soluciones mediante el uso de operadores genéticos como: *Simulated Binary Crossover* [SBX] y *Polynomial Mutation* (PM). Finalmente se realizan pruebas de escalabilidad en problemas de prueba definidas como DTLZ1 y DTLZ2 para conocer la eficiencia del algoritmo.

Palabras clave: Algoritmos Evolutivos. Problemas de Optimización Multi-Objetivo. Funciones de Escalarización, Algoritmos Evolutivos Multi-Objetivo

Como citar el artículo:

Tino Salgado, C., Coello Coello, C. A. y Pescador Rojas, M. (2018). Algoritmo evolutivo multi-objetivo basado en descomposición con múltiples funciones de escalarización. *Tlamati*, 9(2), 5-13.

Abstract

For Multi-Objective Optimization Problems [POM] as problems with more than one objective function, an alternative solution is using Evolutionary Algorithms, based on methods such as: Pareto, Indicators and Decomposition.

Decomposition-Based Evolutionary Algorithms [MOEA/D] divide the POM into a set of Single Objective Optimization Problems [POS], which are optimized simultaneously using genetic operators and scalarization functions.

In this study, authors propose implementation of a MOEA/D with multiple scalarization functions, in which weight vectors are generated through three different methods: Simplex Lattice Design [SLD], Uniform Design [UD] and the combination of the method UD with the Hammersley method [UDH]. In addition, we consider the creation of continuous and random subpopulations to maintain the diversity of solutions.

Scalarization functions implemented in the MOEA/D with multiple scalarization functions are: Weighted Sum [WS], Weighted Tchebycheff [TCH], Augmented Weighted Tchebycheff [AT], Achievement Scalarizing [ASF] and Penalty-Based Boundary Intersection [PBI]. It is important to emphasize that to each subpopulation is assigned a scalarization function. Also, the solution space is explored and exploited through using of genetic operators such as follows: Simulated Binary Crossover [SBX] and Polynomial Mutation (PM). Finally, scalability tests are performed on test problems defined as DTLZ1 and DTLZ2 to know efficiency of algorithm.

Keywords: Evolutionary Algorithms, Optimization Problems Multi-Objective, Scalarization functions, Evolutionary Algorithms Multi-Objective

Introducción

Los algoritmos evolutivos son una técnica de las metaheurísticas bio–inspiradas y se basan en la teoría de la evolución propuesta por Charles Darwin. Buscan generar mejores soluciones en cada generación haciendo uso de operadores genéticos, tales como: selección, cruza y mutación.

Un Problemas de Optimización Multi–Objetivo [POM] es un conjunto de Problemas de Optimización de un Solo Objetivo [POS] y puede ser establecido por la ecuación 1.

$$\begin{aligned} \text{minimizar } F(x) &= (f_1(x), \dots, f_m(x))^T \\ \text{sujeto a: } x &\in \Omega \end{aligned} \quad (1)$$

Donde $x=(x_1, \dots, x_n)^T$ es un vector de n dimensiones en un espacio de decisión \mathbf{R}^n y $F(x)$ es un vector objetivo de dimensión m en el espacio objetivo \mathbf{R}^m . Entonces, $\Omega \in \mathbf{R}^n$ determina la región de las variables de decisión.

Sean x^1 y $x^2 \in \Omega$ dos soluciones de la Ecuación 1, x^1 domina a x^2 si y solo si $f_i(x^1) \leq f_i(x^2)$ para todo $i \in \{1, \dots, m\}$ y $f_i(x^1) \neq f_i(x^2)$. Sintetizando, cualquier solución x^* es la solución óptima de Pareto si y solo si no es dominada por alguna otra solución presente en Ω .

Para la resolución de POM se han utilizado diversos tipos de Algoritmos Evolutivos Multi–Objetivo [MOEA], estos basado en métodos como: Pareto, indicadores, y en descomposición, éste último es abordado en el presente artículo.

La idea central de los Algoritmos Evolutivos Multi–Objetivo basados en descomposición [MOEA/D] es dividir el POM en POS haciendo uso de vectores de peso. Cada elemento de dicho vector representa el nivel de importancia de un objetivo.

Por cada POS es obtenida una solución óptima de Pareto, lo anterior mediante la implementación de funciones de escalarización. A partir de las soluciones óptimas de Pareto dadas en cada POS se establece un Conjunto óptimo de

Pareto [CP] para el POM. Cada elemento de CP está asociado a un vector objetivo, lo cual permite la generación de posibles soluciones que pueden ser vistas en un Frente de Pareto [FP].

Para estudiar la escalabilidad del número de objetivos y de variables de decisión que puedan ser resueltas por el MOEA/D con múltiples funciones de escalarización, hemos optado por realizar pruebas en problemas propuestos por Deb, Thiele, Laumanns y Zitzler (2011), conocidos como test de problemas DTLZ1 y DTLZ2, lo anterior debido a que en estos problemas se conoce *a priori* la forma y la localización exacta del FP.

Resumiendo, nuestro objetivo principal es brindar soluciones a problemas multi–objetivo (DTLZ1, DTLZ2) a través de la generación de subpoblaciones, estableciendo múltiples funciones de escalarización y aplicando operadores genéticos como SBX y PM.

La estructura del presente artículo se observa a continuación. Inicialmente abordamos los métodos implementados en el MOEA/D con múltiples funciones de escalarización. En seguida, se presentan los resultados obtenidos. Posteriormente se compara el MOEA/D con múltiples funciones de escalarización contra otros algoritmos presentes en la literatura actual y además, se dan a conocer las conclusiones de la investigación y finalizamos expresando brevemente los trabajos a futuro surgidos a partir de la presente investigación.

Métodos y materiales

Como parte del desarrollo del MOEA/D con múltiples funciones de escalarización, se presentan a continuación los métodos involucrados en su funcionamiento. Tales métodos fueron desarrollados en lenguaje C y en casos particulares realizando scripts en lenguajes como: Bash y Python. Inicialmente se describen los problemas de optimización a resolver. En seguida se muestran los métodos empleados en la generación de vectores de peso. Posteriormente se describe la creación de subpoblaciones continuas y aleatorias y brevemente, son presentadas las funciones de escalarización implementadas en el presente algoritmo. Posteriormente, detallamos el proceso de cruza y el proce-

so mutación. Finalmente, para mejor comprensión del lector se presenta el funcionamiento general del MOEA/D con múltiples funciones de escalarización en pseudocódigos.

Problemas DTLZ1 y DTLZ2

Los problemas DTLZ1 son establecidos con base a la ecuación 2, donde una de sus principales características se encuentra en su FP, el cual corresponde al hiperplano lineal $\sum_{m=1}^M f_m = 0.5$ asimismo los óptimos de Pareto de $x_i=0.5$ para todo $x_i \in X_M$. Véase el FP del problema DTLZ1 en la figura 1.

$$\left\{ \begin{array}{l} \text{minimizar } f_1(X) = 1/2 x_1 x_2 \dots x_{M-1} (1 + g(X_M)), \\ \text{minimizar } f_2(X) = 1/2 x_1 x_2 \dots (1 - x_{M-1}) (1 + g(X_M)), \\ \vdots \\ \text{minimizar } f_{M-1}(X) = 1/2 x_1 (1 - x_2) (1 + g(X_M)), \\ \text{minimizar } f_M(X) = 1/2 (1 - x_1) (1 + g(X_M)), \end{array} \right. \quad (2)$$

$$\left[\text{Donde } g(X_M) = 100 \left[|X_M| + \sum_{x_i \in X_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right] \right]$$

En contraparte, los problemas DTLZ2 se definen por la Ecuación 3 y tienen un FP continuo, no convexo y en forma esférica, pueden ser descritos por $\sum_{m=1}^M f_m^2 = 1$, en otras palabras, se puede ver como una hiperesfera de radio 1 con centro en el origen. El FP del problema DTLZ2 puede ser visto en la Figura 2.

Generación de vectores de peso

Un proceso esencial en la resolución de POM es la generación de vectores de peso. En gran parte de la literatura referente a los MOEA/D se ha optado por generar

$$\left\{ \begin{array}{l} \text{minimizar } f_1(X) = (1 + g(X_M)) \cos(x_1\pi/2) \dots \cos(X_M - 2\pi/2) \cos(X_M - 1\pi/2), \\ \text{minimizar } f_2(X) = (1 + g(X_M)) \cos(x_1\pi/2) \dots \cos(X_M - 2\pi/2) \sin(X_M - 1\pi/2), \\ \text{minimizar } f_3(X) = (1 + g(X_M)) \cos(x_1\pi/2) \dots \sin(X_M - 2\pi/2), \\ \vdots \\ \text{minimizar } f_M(X) = (1 + g(X_M)) \sin(X_1\pi/2), \end{array} \right. \quad (3)$$

$$\left[\text{donde } g(X_M) = \sum_{x_i \in X_M} (x_i - 0.5)^2 \right]$$

vectores de peso uniformemente distribuidos a través del método *Simplex Lattice Design (SLD)*, el cual se muestra en la ecuación 4. Donde q es el número de dimensiones y m el grado de latencia.

$$x_i = 0, 1/m, 2/m, \dots, 1 \text{ para } i = 1, 2, \dots, q \quad (4)$$

En la presente investigación se generaron vectores de peso para valores mínimos y máximos en dos y tres dimensiones con tres diferentes métodos: *Uniform Design (UD)* (Fang y Wang, 1994), *SLD* y la combinación del método *UD* con el método *Hammersley (UDH)* (Molinet Berenguer y Coello Coello, 2015), los resultados obtenidos a partir de la Tabla 1 pueden ser vistos en la Figura 3.

Generación de subpoblaciones continuas y aleatorias

Considerando la distribución uniforme de los vectores de peso generados por el método *SLD* y en busca de mejores soluciones se ha optado por realizar subpoblaciones continuas y aleatorias para 100 vectores de peso con dimensionalidad 2 y 120 vectores de peso con dimensionalidad 3 en problemas DTLZ1 y DTLZ2. Dichos vectores son divididos en dos, tres, cuatro y cinco subpoblaciones. A través del indicador *Spacing (SP)* se observa en las tablas 2 y 3 que la generación de subpoblaciones aleatorias puede

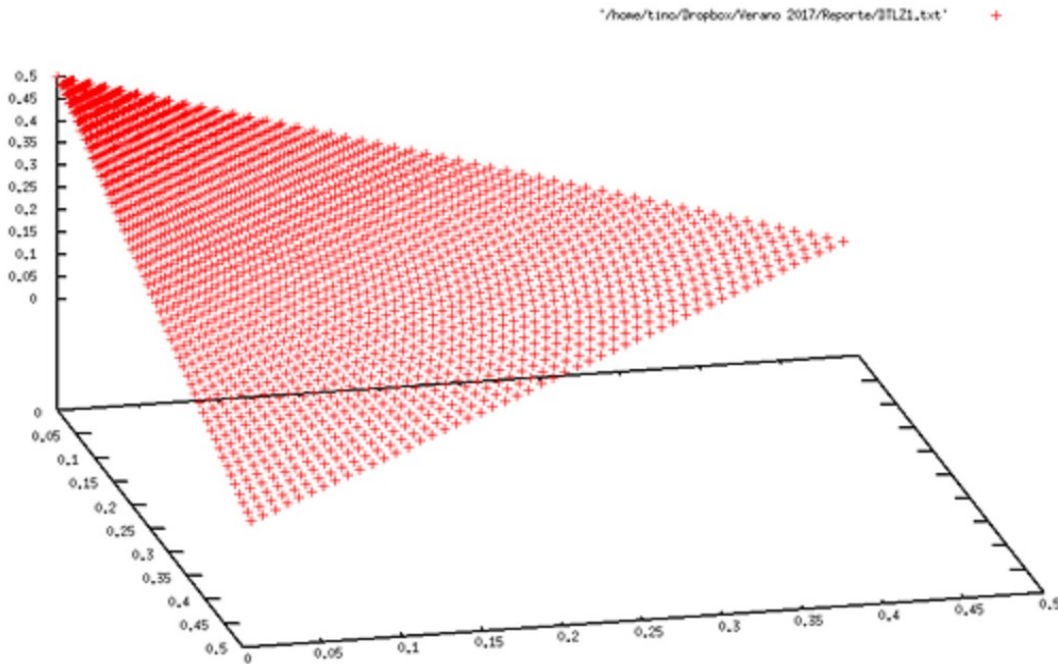


Figura 1. FP del problema DTLZ1.

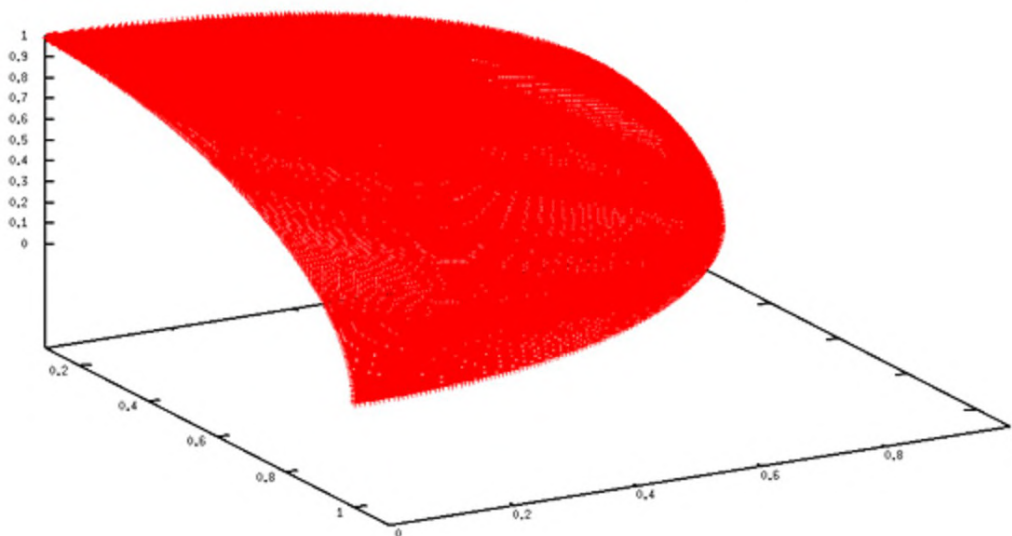


Figura 2. FP del problema DTLZ2.

retornar mejores soluciones al explorar y explotar el espacio de soluciones mediante el proceso de SBX y PM, respectivamente.

Implementación de funciones de escalarización

Weighted Sum (WS) y *Weighted Tchebycheff (TCH)* son las funciones de escalarización más utilizadas. Sin embargo, en el presente artículo, *WS*, *TCH*, *Augmented Weighted Tchebycheff (AT)*, *Achievement Scalarizing (ASF)* y *Penalty-Based Boundary Intersection (PBI)* son las funciones de escalarización implementadas. *WS* puede ser vista en la Ecuación 5,

$$g^{WS}(x|\lambda) = \lambda_1 \cdot f_1(x), \dots, \lambda_m \cdot f_m(x) \quad (5)$$

Donde λ_i es un número positivo para el objetivo i , asimismo $\lambda_1 + \lambda_2 + \dots + \lambda_m = 1$ y $\lambda \geq 0$ para todo $i=1,2,\dots,m$.

A diferencia de *WS*, algunas funciones de escalarización como *TCH*, *AT* y *ASF* hacen uso de un conjunto de referencia $z^*=(z_1, z_2, \dots, z_m)$. Estos puntos de referencia comúnmente son considerados como un punto de aspiración o incluso puntos inalcanzables. Las funciones de escalarización que hacen uso de este conjunto de referencia se caracterizan por pequeñas variaciones. *TCH* (Ishibuchi, Sakane, Tsukamoto y Nojima, 2010). puede verse en la ecuación 6, donde λ_i al igual que en *WS* es un número po-

$$g^{TE}(x|\lambda, z^*) = \max_{i=1,2,\dots,m} \{\lambda_i \cdot |z_i^* - f_i(x)|\} \quad (6)$$

$$g^{AT}(x|\lambda, z^*) = \max_{i=1,2,\dots,m} \{\lambda_i \cdot |z_i^* - f_i(x)|\} + p \sum_{j=1}^m |f_j(x) - z_j^*| \quad (7)$$

sitivo para el objetivo i y $\lambda_1 + \lambda_2 + \dots + \lambda_m = 1$.

Una variación de *TCH* es *AT* (Ishibuchi et al., 2010), ésta puede ser vista en ecuación 7, en la cual p es usualmente un número constante positivo y es multiplicado por la sumatoria de la diferencia entre la función objetivo j y el punto de referencia j .

A diferencia de *TCH* y *AT*, *ASF* (Gonçalves, Kuk, Almeida y Venske, 2015) considera la diferencia entre la función objetivo i y la partición entre el punto de referencia i con el peso asociado a dicho objetivo.

Véase *ASF* en la ecuación 8, donde λ_i es un número entero positivo y se considera $\lambda_1 + \lambda_2 + \dots + \lambda_m = 1$.

Una función de escalarización también utilizada en la presente investigación es *PBI*. De acuerdo con Zhang y Li (2007), *PBI* busca encontrar puntos de intersección entre un límite superior y un conjunto de líneas. Si estas líneas son distribuidas uniformemente en un sentido, uno puede esperar que los puntos de intersección resultantes proporcionan una buena aproximación a todo el FP. La función de escalarización *PBI* puede ser vista en la Ecuación 9, donde $\theta > 0$ es un parámetro de penalización previamente establecido.

A cada subpoblación se le asigna aleatoriamente una función de escalarización, la cual tiene como objetivo principal encontrar la solución óptima local, dichas soluciones serán alteradas a través de operadores genéticos como: SBX y PM.

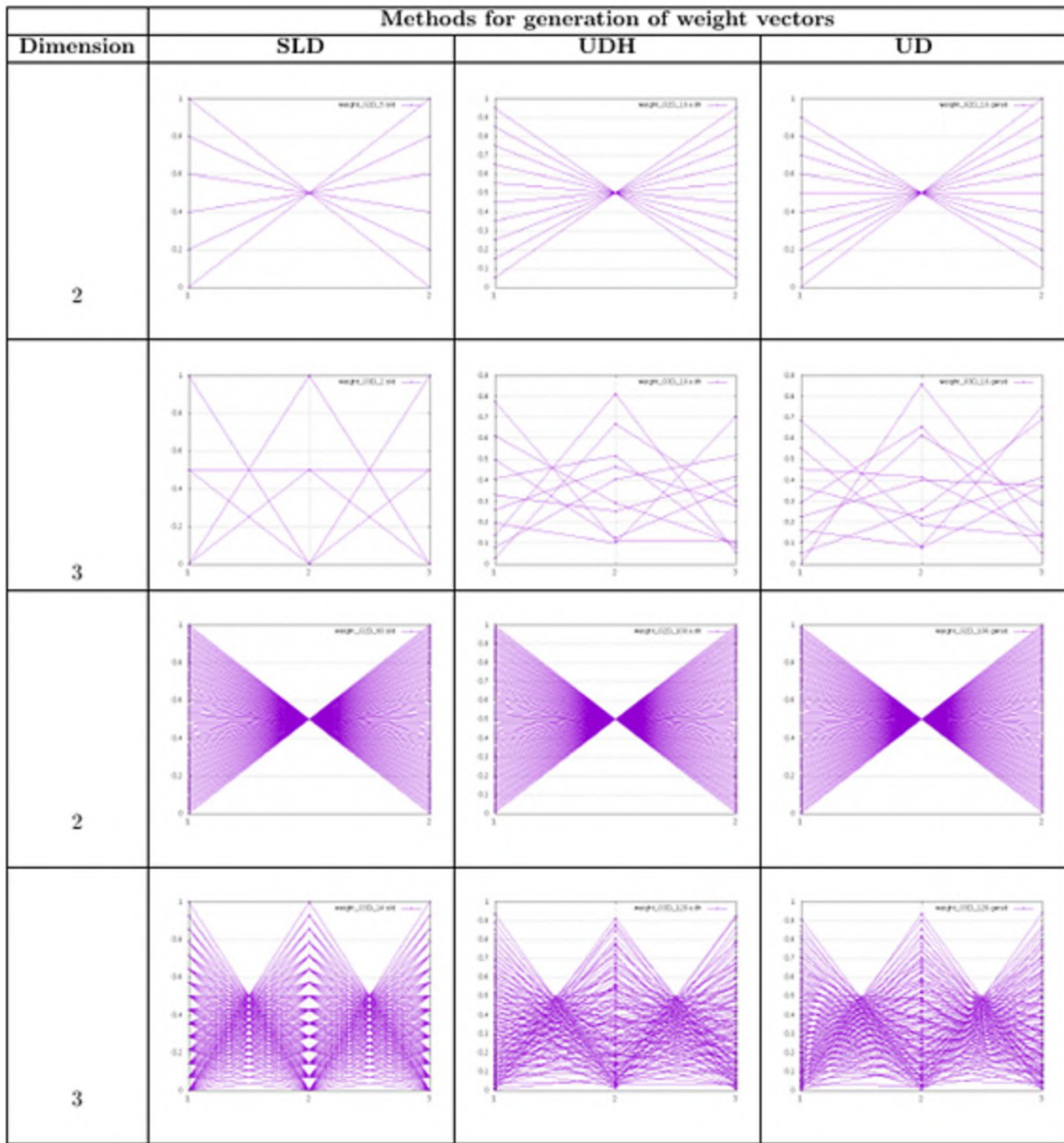


Figura 3. Generación de vectores de peso con valores mínimos y máximos.

Simulated Binary Crossover

El método SBX está basado en el operador genético de recombinación o cruce y tiene como objetivo principal la creación de dos nuevos individuos (soluciones) a partir de la combinación de dos individuos previamente selecciona-

dos de manera aleatoria.

Considere dos cadenas reales padres de longitud l , las cuales son cruzadas en un determinado punto. Si codificamos los k valores de más a la derecha como A_1 y al resto $(l-k)$ como B_1 , el valor decodificado de la cadena es: $x_1 = B_1 2^k + A_1$. Podemos codificar el valor de la segunda cadena padre considerando que A_2 y B_1 son los valores de los k valores de la derecha y el resto $(l-k)$, respectivamente: $x_2 = B_2 2^k + A_2$. De esta manera las cadenas hijas decodi-

$$\begin{aligned} \text{Minimizar } ASF(x) &= \max_{j=1}^M (f_j(x) - z_j / \lambda_j) \quad (8) \\ \text{sujeto a: } x &\in \Omega \end{aligned}$$

DIM	MÉTODOS DE GENERACIÓN DE VECTORES DE PESO						
	SLD			UD		UD-Hammersly	
	minimo	medio	máximo	minimo	máximo	minimo	máximo
02D	5 => 6	9 => 10	99 => 100	10	100	10	100
03D	2 => 6	3 => 10	14 => 120	10	120	10	120

Tabla 1. Valores mínimos y máximos para la generación de vectores de peso.

$$\begin{aligned} \text{minimizar } g^{pbi}(x, \lambda, z^*) &= d_1 + \theta d_2, \\ \text{sujeto a: } x &\in \Omega \end{aligned} \quad (9)$$

donde

$$\begin{aligned} d_1 &= \| (F(X) - z^*)^T \lambda \| / \| \lambda \| \\ d_2 &= \| F(X) - (z^* + d_1 \lambda) \| \end{aligned}$$

$$\begin{aligned} y_1 &= B_1 2^k + A_2, \\ y_2 &= B_2 2^k + A_1. \end{aligned} \quad (10)$$

ficadas resultantes pueden ser vistas en la Ecuación 10.

En busca de variedad en las soluciones se realiza una selección previa al método de *PM*. El proceso de selección es realizado a las soluciones hijas generadas por *SBX*. Posteriormente, a la mejor solución se le aplica *PM* y la solución restante es desechada.

Polynomial Mutation

Posterior al proceso de *SBX*, se procede a realizar una alteración genética en la solución hija denominada *PM*, ésta considera a δ como un factor de perturbación dado por la ecuación 11.

$$\delta = c - p / \Delta_{MAX} \quad (11)$$

Donde Δ_{MAX} es un valor fijo, representando la máxima perturbación permisible en el valor padre c y p es el valor mutado. El valor de mutación es calculado con una probabilidad de distribución que depende del factor de perturbación δ , dicho valor puede ser observado en la ecuación 12.

Seguidamente, para la creación de un valor mutado, un número aleatorio u es creado en el rango $[0,1]$.

La ecuación 13 puede ser utilizada para calcular la perturbación del factor $\bar{\delta}$ correspondiente para u . Finalmente el valor de mutación puede ser calculado mediante la ecuación 14.

$$\begin{aligned} \mathcal{P}(\delta) &= 0.5(n+1)(1-|\delta|)^n, \\ \text{sujeto a: } \delta &\in (-1, 1). \end{aligned} \quad (12)$$

$$\bar{\delta} = \begin{cases} (2u)(1/n+1) - 1 & \text{si } u < 0.5 \\ 1 - [2(1-u)](1/n+1) & \text{si } u > 0.5 \end{cases} \quad (13)$$

$$c = p + \bar{\delta} \Delta_{MAX} \quad (14)$$

dim	num subp	SP/continuous	SP/random
02	2	2.6698e-03(9.0310e-03)	3.7059e-02(1.1382e-01)
	3	1.7729e-03(3.0980e-04)	3.2446e-02(3.4590e-02)
	4	4.4015e-02(1.8762e-02)	2.2839e-01(1.0696e-01)
	5	2.9238e-02(2.1065e-02)	1.2785e-01(7.1220e-02)
03	2	2.6889e-02(2.0797e-04)	3.2404e-02(3.0317e-02)
	3	2.5874e-02(4.8350e-04)	8.7992e-02(8.0092e-02)
	4	5.7766e-02(5.5921e-02)	2.8479e-01(1.4371e-01)
	5	3.8545e-01(1.5089e-01)	6.8811e-01(2.3672e-01)

Tabla 2. Indicador SP en problema DTLZ1. Media y Desviación estándar.

dim	num subp	SP/continuous	SP/random
02	2	6.5862e-03(1.5918e-04)	6.5285e-03(1.4987e-03)
	3	6.6344e-03(5.3443e-04)	7.4157e-03(7.8878e-04)
	4	6.5132e-03(4.3688e-04)	8.8356e-03(6.6497e-04)
	5	6.5678e-03(2.4115e-04)	1.1317e-02(5.5152e-04)
03	2	7.6872e-02(7.4491e-04)	7.6921e-02(1.2638e-03)
	3	7.7158e-02(4.4869e-04)	7.6957e-02(1.0523e-03)
	4	7.5929e-02(1.4154e-03)	7.6168e-02(1.5887e-03)
	5	7.7987e-02(8.8753e-04)	7.7180e-02(1.5271e-03)

Tabla 3. Indicador SP en Problema DTLZ2. Media y Desviación estándar.

Funcionamiento general del MOEA/D con múltiples funciones de escalarización

En párrafos anteriores, se han presentado los diferentes métodos implementados en el MOEA/D con múltiples funciones de escalarización. Buscando una comprensión clara del lector puede verse el funcionamiento de la generación de subpoblaciones en la figura 4.

Es importante resaltar el establecimiento de parámetros para el MOEA/D. Este método asigna alguna de las funciones de escalarización vistas en la sección 2.4 en una subpoblación, además de configurar el número de ejecuciones del MOEA/D con múltiples funciones de escalarización. En la línea 4 de la figura 5 se asigna un porcentaje de 0.9 para el operador *SBX* y un porcentaje de $1/n$, donde n es el número de variables del POM al operador *PM*. Los parámetros asignados al POM son establecidos previamente por el usuario al ser recibidos como parámetros, posteriormente son implementados en la línea 3.

ENTRADA:		Problema, Nobj, Neje, Salida, Nsubpo, TipoSubpo
SALIDA:		Subpoblaciones almacenadas en archivos
1	Wfile ←	Vector de prefijos de archivos
2	Para S=0 hasta S < Nsubpo	
3		Establecer parámetros para el MOP
4		Establecer parámetros para operadores genéticos
5		Establecer parámetros para el MOEA/D (Función de escalarización)
6		S++
7		Fin
8	Para R=1 hasta R ≤ Neje	
9		Para S = 0 hasta S < Nsubpo
10		Establecer condición de parada
11		Inicializar población
12		Ejecutar MOEA/D (Véase Figura 7)
13		Guardar parámetros
14		S++
15		Fin
16		Unir variables
17		Unir objetivos
18		Generar archivo de subpoblaciones
19		R++
20		Fin

Figura 4. Generación de subpoblaciones

ENTRADA:		alg, param, pop, mop
SALIDA:		Archivos de puntos ideal y nadir
1		Calcular puntos ideal y nadir
2		Mientras no se cumpla la condición de parada
3		Hacer l = 0 hasta l < pop -> tam
4		Simulated Binary Crossover
5		Mutación polinomial
6		Evaluar población
7		Actualizar puntos ideal y nadir
8		Actualizar soluciones vecinas
		l++
9		Fin
10		Generar archivo de puntos nadir
11		Generar archivo de puntos ideal
12		Fin

Figura 5. Funcionamiento de MOEA/D

Resultados

Para la solución de problemas DTLZ1, el MOEA/D con múltiples funciones de escalarización responde de forma efectiva a configuraciones con 2 y 3 objetivos para 2, 3, 4 y 5 subpoblaciones. En la figura 6 se observa la comparación los FP generados a partir de la ejecución 1 y 30 para el problema DTLZ1 con 2 subpoblaciones y 2 objetivos.

Respecto a la solución de problemas DTLZ2, los cuales presentan variaciones considerables a los problemas DTLZ1, los resultados dados por el MOEA/D con múltiples funciones de escalarización se consideran aceptables para configuraciones con 3 objetivos en 2, 3, 4 y 5 subpoblaciones. Muestra de ello puede ser visto en la figura 7, donde se presenta la comparación de los FP generados a partir de las ejecuciones 1 y 30 con 3 subpoblaciones y 3 objetivos.

Conclusiones y discusiones

El MOEA/D con múltiples funciones de escalarización es capaz de generar soluciones para problemas DTLZ1 y DTLZ2 a partir de la generación de subpoblaciones, estableciendo en ellas alguna de las funciones de escalariza-

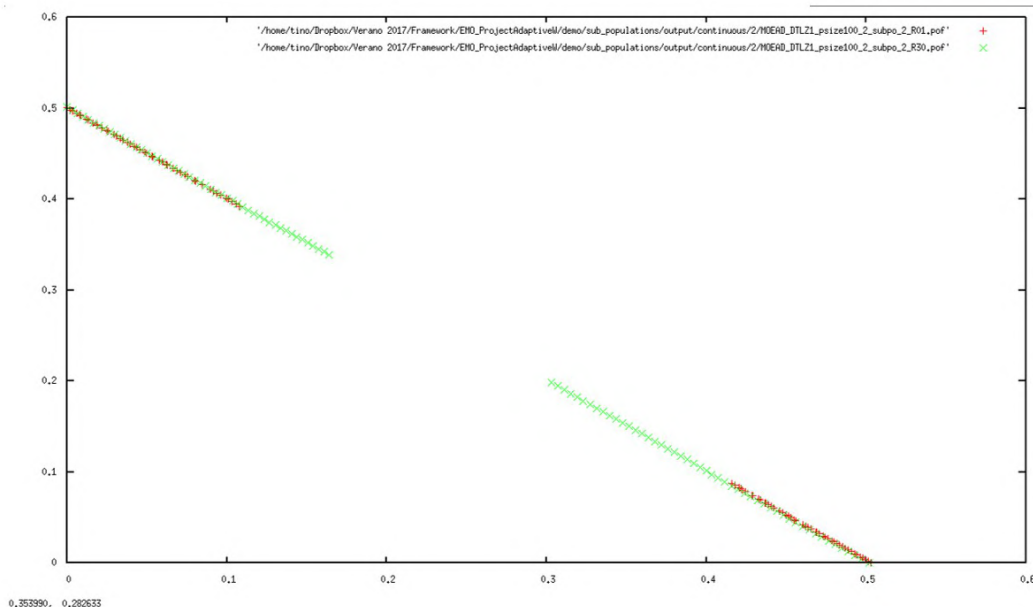


Figura 6. Comparación entre FP para problemas DTLZ1

ción vistas y realizando procesos genéticos. Además, el uso de metaheurísticas bio-inspiradas resulta útil para la solución de POM, lo anterior debido a la búsqueda de mejores soluciones en cada iteración del proceso evolutivo. Asimismo, considerando el indicador *SP* a partir de la generación de subpoblaciones aleatorias se pueden obtener mejores soluciones.

Con relación a la resolución de POM mediante funciones de escalarización, en la literatura actual es común ver la comparación entre dichas funciones, nosotros hacemos uso de cada una de ellas al asignarlas a una subpoblación. A diferencia de Castro Jr., Pozo, Lozano y Santana (2017), en donde propone la implementación de algoritmos de agrupamiento (clusters) para mantener la diversidad de las soluciones, el MOEA/D con múltiples funciones de escalarización no considera una distancia entre las soluciones.

Sin embargo, ha optado por la generación de subpoblaciones continuas y aleatorias, en esta última obteniendo buenos resultados.

Trabajos futuros

Con base a los resultados obtenidos de la presente investigación, consideramos la implementación del operador genético denominado Differential Evolution (ED) en el MOEA/D con múltiples funciones de escalarización, este operador basado en la mutación a posteriori puede realizar recombinación, lo cual permite explotar y explorar aún más el espacio de soluciones. Además, valorando la literatura actual, consideramos viable la implementación de adaptación de pesos durante la ejecución del MOEA/D con múltiples funciones de escalarización.

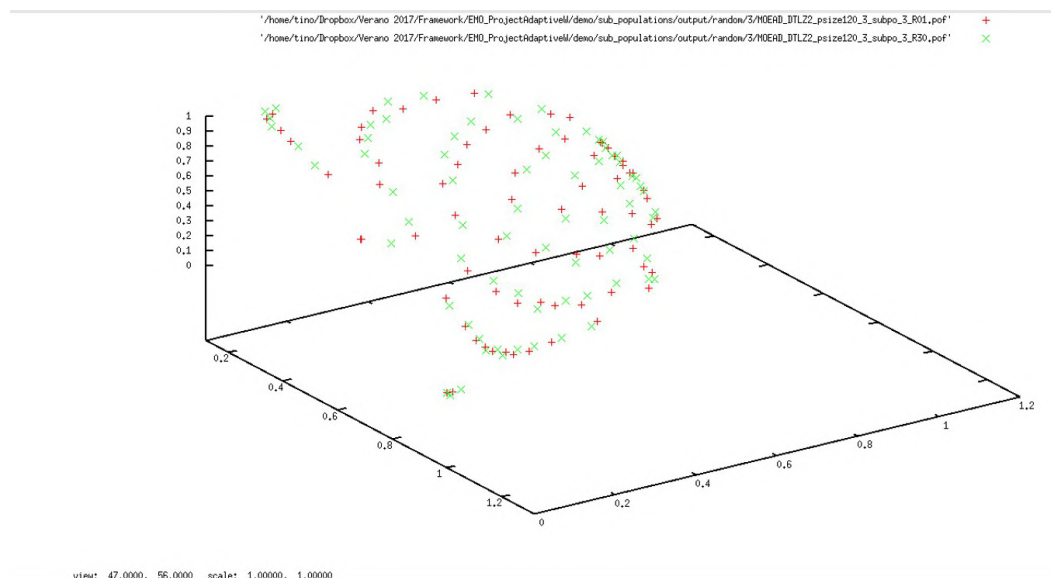


Figura 7. Comparación de FP para problemas DTLZ2

Agradecimientos

Agradezco a la Academia Mexicana de Ciencias y a la Universidad Autónoma de Guerrero por ser parte de mi formación profesional, mediante el apoyo brindado en mi segundo Verano de Investigación Científica. Al Dr. Carlos Artemio Coello Coello y a la M.C Miriam Pescador Rojas por permitirme colaborar en sus líneas de investigación. Asimismo, agradezco al M.C Raúl Juárez Morales y al Dr. Virgilio Cruz Guzmán por su motivación y apoyo brindado durante mi formación académica.

Referencias

- Castro Jr., O. R., Pozo, A., Lozano, J. A. y Santana, R. (2017). An investigation of clustering strategies in many-objective optimization: the I-Multi algorithm as a case study. *Swarm Intelligence*, 11(2), 101–130.
- Deb K., Thiele L., Laumanns M. y Zitzler E. (2005) Scalable Test Problems for Evolutionary Multiobjective Optimization. En: A. Abraham, L. Jain y R. Goldberg (Eds) *Evolutionary Multiobjective Optimization. Advanced Information and Knowledge Processing*, 105-145. London, ENG. Springer,
- Fang, K.T. y Wang, Y. (1994) *Number-Theoretic Methods in Statistics*. London, ENG. Chapman and Hall.
- Gonçalves, R. A., Kuk J. N., Almeida C. P. y Venske S.M. (2015) MOEA/D-HH: A Hyper-Heuristic for Multi-objective Problems; En A. Gaspar-Cunha, C. Henggeler Antunes y C. Coello (Eds) *Evolutionary Multi-Criterion Optimization. EMO 2015. Lecture Notes in Computer Science*, 9018, 94-108.
- Ishibuchi, H., Sakane, Y., Tsukamoto, N. y Nojima, Y. (2010). Simultaneous use of different scalarizing functions in MOEA/D. *Proceedings of the 12th annual conference on Genetic and evolutionary computation (GECCO '10)*. New York, NY, USA, ACM, 519-526.
- Molinet Berenguer, J. A. y Coello Coello C. A. (2015) Evolutionary Many-Objective Optimization Based on Kuhn-Munkres' Algorithm. En: Gaspar-Cunha A., Henggeler Antunes C., Coello C. (eds) *Evolutionary Multi-Criterion Optimization. EMO 2015. Lecture Notes in Computer Science*, 9019, 3-17.
- Zhang, Q. y Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712-731.