

Integración de las preferencias de los usuarios en el proceso de toma de decisiones para problemas de optimización de muchos objetivos discretos

Nancy Pérez¹, Oliver Cuate², Oliver Schütze², Alejandro Alvarado¹

¹ Universidad Autónoma de Ciudad Juárez,
México

² Cinvestav-IPN, Ciudad de México,
México

nancy.perez.glez@gmail.com, ocuate@computacion.cs.cinvestav.mx,
schuetze@cs.cinvestav.mx, alejandro.alvarado@uacj.mx

Resumen. En muchas aplicaciones, uno se enfrenta con el problema de que muchos objetivos tienen que ser optimizados simultáneamente lo que conlleva a tener un problema de optimización de muchos objetivos (MaOP, por sus siglas en inglés). Una característica importante de los problemas de optimización de muchos objetivos discretos es que su conjunto de soluciones, el llamado conjunto de Pareto, consiste de demasiados puntos para ser calculados de manera eficiente. Por lo tanto, aunque los algoritmos evolutivos especializados son en principio capaces de calcular un conjunto S de soluciones candidato bien esparcidas a lo largo del conjunto de Pareto, no se garantiza que el tomador de decisiones del problema en cuestión encontrará la solución 'ideal' dentro de S para su problema. Se argumenta en este trabajo que tiene sentido llevar a cabo una especie de post procesamiento para una solución dada $s \in S$. Específicamente, proponemos dos diferentes métodos que permiten dirigir la búsqueda desde s a lo largo del conjunto de Pareto en direcciones especificadas por el usuario. Resultados numéricos, en casos del problema de enrutamiento de vehículos con ventanas de tiempo, muestran la efectividad de los novedosos métodos propuestos.

Palabras clave. Optimización de muchos objetivos, toma de decisiones, problema de enrutamiento de vehículos, optimización discreta, computación evolutiva.

Including Users Preferences in the Decision Making for Discrete Many Objective Optimization Problems

Abstract. In many applications one is faced with the problem that many objectives have to be optimized concurrently leading to a many objective optimization problem (MaOP). One important characteristic of discrete MaOPs is that its solution set, the so-called Pareto set, consists of too many elements to be efficiently computed. Thus, though specialized evolutionary algorithms are in principle capable of computing a set S of well spread candidate solutions along the Pareto set, it is not guaranteed that the decision maker of the underlying problem will find the 'ideal' solution within S for his or her problem.

We argue in this paper that it makes sense to perform a kind of post-processing for a selected solution $s \in S$. More precisely, we will propose two different methods that allow to steer the search from s along the Pareto set into user specified directions. Numerical results on instances of the vehicle routing problem with time windows will show the effectivity of the novel methods.

Keywords. Many objective optimization, decision making, vehicle routing problem, discrete problem, evolutionary computation.

1. Introducción

En muchos problemas de la vida real, tal como los de Ingeniería o Finanzas, uno se enfrenta con el problema de que varios objetivos tienen que ser optimizados simultáneamente. Si más de tres objetivos están involucrados, el problema resultante se denomina como un problema de optimización de muchos objetivos (MaOP, por sus siglas en inglés) en la literatura. Aunque, el trato a los MaOPs es un campo de investigación relativamente nuevo (hasta el momento, se han estudiado principalmente problemas con dos o tres objetivos) se le considera muy importante ya que hoy en día los procesos de toma de decisiones son cada vez más y más importantes.

Por ejemplo, el objetivo más importante en el problema de enrutamiento de vehículos, el cual es utilizado como demostrador en este estudio, probablemente sea la minimización de costos totales. Sin embargo, otros aspectos importantes (entre otros) son la utilización de los vehículos, el número de vehículos usados, el tiempo total, la distancia total, el balance de ruta, la satisfacción del cliente y el tiempo de retraso. Todos estos aspectos son tan importantes que pueden ser planteados como objetivos individuales en un problema de optimización.

Una característica importante de los MaOPs es que su conjunto de soluciones, el denominado conjunto de Pareto, no consiste en una solución única, como en el problema 'clásico' de optimización escalar (SOPs, por sus siglas en inglés) donde sólo un objetivo es considerado. En lugar de ello, el conjunto de Pareto de un MaOP típicamente forma una superficie de dimensión $(k - 1)$, donde k es el número de objetivos implicados en el problema. Para MaOPs discretos, como el que es considerado en este estudio, la magnitud del conjunto de soluciones se incrementa con k , y este crecimiento incluso es exponencial en k para ciertos casos.

Para lidiar con los problemas de optimización multi-objetivo (MOPs) existen diferentes algoritmos evolutivos que han captado el interés de muchos investigadores en las últimas décadas (ver [11, 9, 51, 22] y sus referencias). Dentro de las razones para esto se encuentra que estos

algoritmos son aplicables a una amplia gama de problemas y son de naturaleza global; por lo tanto, en principio no depende del conjunto de candidatos iniciales (es decir, la población inicial). Además, debido a su enfoque basado en conjuntos, permite calcular una representación de tamaño finito del conjunto de Pareto entero en una sola ejecución del algoritmo. La mayoría de estos algoritmos especializados, llamados EMOAs (Algoritmos evolutivos multi-objetivos) son diseñados para tratar con problemas con sólo unos cuantos objetivos (típicamente, de 2 a 4). A pesar del poco tiempo en el tratamiento de MaOPs, debido a su importancia, cada vez más y más algoritmos han sido desarrollados para tratar con muchos objetivos. Como ejemplo, se encuentra el uso de EMOAs con un tamaño de población grande (por ejemplo, 10,000 individuos) [29], el algoritmo evolutivo multi-objetivo dinámico (DMOEA) [53], y el algoritmo evolutivo basado en redes (GrEA) [50].

No obstante, los conjuntos de Pareto de los MaOPs; por lo general, no se pueden calcular en su totalidad de manera eficiente debido a sus enormes magnitudes, y consecuentemente, estos conjuntos no pueden ser visualizados apropiadamente. Por lo tanto, incluso con la ayuda de algoritmos evolutivos, si el número de objetivos es alto, no se puede esperar que la solución seleccionada por el tomador de decisiones (TD) sea, de hecho, la solución 'ideal' del MaOP —la solución dentro de todo el conjunto de Pareto que satisfaga en mayor medida al TD— con respecto a la configuración dada.

En este artículo, se argumenta que tiene sentido realizar pequeños ajustes a una solución óptima seleccionada de un determinado MaOP. Específicamente, se proponen dos enfoques diferentes, donde, comenzando de una solución inicial x_0 , se generen otras soluciones x_i , $i = 1, \dots, N$, de tal manera que la secuencia de las soluciones candidato realicen un movimiento en las direcciones especificadas por el usuario. Para obtener las soluciones deseadas x_i , una secuencia específica de SOPs tiene que ser resuelta. En este estudio, se utiliza el problema de enrutamiento de vehículos con ventanas de tiempo como ejemplo demostrativo y un

algoritmo genético, el cual es capaz de resolver problemas de optimización combinatoria utilizando un ordenamiento de permutaciones y operadores evolutivos especiales para su representación, para la resolución de los SOPs, ya que ha demostrado ser eficiente en esta clase de problemas. Los resultados numéricos en diferentes escenarios muestran los beneficios de este nuevo enfoque.

Es importante dejar en claro que la idea de dirigir la búsqueda a lo largo del conjunto o frente de Pareto hacia una determinada dirección no es nueva. El método de búsqueda dirigida descendente [44] y el Pareto Tracer [34] son capaces de realizar esta búsqueda; sin embargo, ambos están restringidos a problemas continuos y su enfoque no puede extenderse a dominios discretos mientras se utilice la información del gradiente.

El resto de este artículo es organizado de la siguiente manera: en la sección 2 se presenta brevemente los antecedentes requeridos y se discute el trabajo relacionado. En la sección 3 se proponen dos métodos para realizar pequeños ajustes a la solución dada de un MaOP. En la sección 4, se presentan algunos resultados numéricos y, finalmente, se presentan las conclusiones y las futuras rutas de investigación en la sección 5.

2. Antecedentes y trabajos relacionados

2.1. Optimización de múltiples y de muchos objetivos

Matemáticamente, el problema de optimización multiobjetivo (MOP) puede expresarse de la siguiente manera:

$$\begin{aligned} & \min_{x \in \Omega} F(x), \\ & \text{s.a. } g(x) \leq 0 \\ & \quad h(x) = 0. \end{aligned} \quad (1)$$

$F : \Omega \rightarrow \mathbb{R}^k$ se define como el vector de funciones objetivo:

$$F(x) = (f_1(x), \dots, f_k(x))^T, \quad (2)$$

donde $f_i : \Omega \rightarrow \mathbb{R}$ denota la i -ésima función objetivo. El dominio de F se define como

$$D := \{x \in \Omega : g(x) \leq 0 \text{ y } h(x) = 0\}. \quad (3)$$

La optimalidad de un MOP es definida mediante el concepto de *dominancia*.

Definición 1

- (a) Sean $v, w \in \mathbb{R}^k$. El vector v es menor que w ($v <_p w$), si $v_i < w_i$ para todo $i \in \{1, \dots, k\}$. La relación \leq_p se define análogamente.
- (b) Un vector $y \in D$ es dominado por un vector $x \in D$ ($x < y$) con respecto a (1) si $F(x) \leq_p F(y)$ y $F(x) \neq F(y)$, en otro caso se dice que y es no dominado por x .
- (c) Un punto $x \in D$ es nombrado óptimo de (Pareto) o punto de Pareto si no existe otro $y \in D$ que domine a x .
- (d) El conjunto P_D de todas las soluciones óptimas de Pareto es llamado conjunto de Pareto y su imagen $F(P_D)$ recibe el nombre de frente de Pareto.

Para MOPs continuos (i.e., $\Omega = \mathbb{R}^n$) se puede demostrar que tanto el conjunto como el frente de Pareto forman típicamente—i.e., bajo ciertas condiciones de regularidad—una superficie de dimensión $k - 1$ [26]. Análogamente, la magnitud del conjunto de Pareto crece conforme el valor de k se incrementa, incluso de manera exponencial en algunos casos. Es por esta razón que MOPs con $k > 3$ son denominados problemas de optimización con muchos objetivos (MaOPs), notación que se usará a partir de ahora.

La optimización evolutiva multiobjetivo (EMO) ha dado origen a una gran variedad de EMOAs (algoritmos para la optimización evolutiva multi-objetivo) [9, 11], los cuales no requieren que las funciones objetivo cumplan con determinadas características o propiedades, y se basan en procedimientos estocásticos de búsqueda. Estos algoritmos simulan los principios básicos del proceso evolutivo sobre una población (un conjunto de individuos o soluciones), por medio de los llamados operadores evolutivos. Existe una gran variedad de EMOAs

que consideran diversos mecanismos para asignar la aptitud a un individuo, la mayoría de ellos utiliza el concepto de dominancia de Pareto como su principal forma de discriminar las soluciones y guiar la búsqueda [9, 11]. Recientemente, los EMOAs se han vuelto muy populares en una gran variedad de aplicaciones y han demostrado ser muy adecuados para MOPs complejos con dos o tres funciones objetivo. Dos de los EMOAs más utilizados son el algoritmo genético elitista de clasificación no dominada (NSGA-II, por sus siglas en inglés), propuesto por K. Deb et al. [12] y el algoritmo evolutivo multiobjetivo basado en descomposición (MOEA/D, por sus siglas en inglés), propuesto por Zhang et al. [51].

Un enfoque distinto se da cuando el tomador de decisiones tiene una idea aproximada sobre el problema, o bien, es capaz de proporcionar información sobre sus preferencias. En este caso es posible utilizar EMOAs que manejen las preferencias del usuario (ver [10, 33] y sus referencias). Este tipo de métodos busca “ayudar a un tomador de decisiones a considerar los múltiples objetivos simultáneamente y encontrar una solución óptima de Pareto que lo satisfaga” [6]. Se presenta a continuación una clasificación típica de los métodos basados en las preferencias, en función de cómo son expresadas por el TD [4]: (i) *métodos a priori*, en donde las preferencias se expresan antes de calcular soluciones eficientes, por ejemplo a través de una función de utilidad [23] o mediante un punto de referencia [13]; (ii) *métodos a posteriori*, aquí el TD elige la solución de su preferencia después de que se ha calculado un conjunto de soluciones eficientes (por ejemplo, [42, 40]); (iii) *métodos interactivos*, en este enfoque, el DM guía la búsqueda con cierta función de utilidad y dicha función puede cambiar durante el proceso de optimización debido a la nueva información adquirida, por ejemplo [38, 7, 37, 48]; y finalmente (iv) *los métodos no basados en las preferencias*, en donde no está disponible la información adicional sobre las preferencias, la idea es encontrar un equilibrio entre los objetivos [4].

Sin embargo, varios investigadores han señalado algunos inconvenientes con el uso de EMOAs para resolver problemas que tienen 4 o

más funciones objetivo [17, 28, 30, 39]. Estas dificultades se deben al hecho de que conforme el número de funciones objetivo se incrementa, también crece la proporción de individuos no dominados en la población. Una expresión de esta proporción e para un problema de dimensión M , tal que dos soluciones sean clasificadas bajo el criterio de dominancia como equivalentes, está dada por $e = \frac{2^M - 2}{2^M}$ [17].

Recientemente, los métodos que buscan mejorar el desempeño de los EMOAs para los MaOPs han sido de mucho interés. Es posible clasificar estos métodos en dos grupos principales: (i) métodos que utilizan una relación de preferencia alternativa, y (ii) métodos que transforman el MaOP original en un SOP.

En el primer grupo se pueden encontrar ejemplos de diferentes relaciones de preferencia, como el orden de preferencias [15], una generalización de la optimalidad de Pareto la cual utiliza un par de definiciones de optimalidad más estrictas; o las relaciones difusas [17] basadas en el número de componentes mayores, menores e iguales entre dos vectores.

Por otra parte, el segundo grupo presenta su propia clasificación. Basados en funciones de logro, como el MOEA/D [51]; basados en indicadores, como el algoritmo evolutivo multiobjetivo de selección con la métrica S (SMS-EMOA, por sus siglas en inglés), el cual trata de minizar el Hipervolumen (HV) y el algoritmo de estimación del hipervolumen para optimización multiobjetivo (Hype, por sus siglas en inglés) [2] con el mismo propósito; basados en técnicas de reducción de la dimensión, como el algoritmo evolutivo de búsqueda de esquinas de Pareto (PCSEA, por sus siglas en inglés) [45], el cual hace una reducción de la dimensionalidad buscando las esquinas del frente de Pareto; y finalmente, basados en la partición del espacio, como el optimizador evolutivo multiobjetivo de clasificación ϵ (ϵ R-EMO, por sus siglas en inglés) [1].

El tratamiento de MaOPs con la incorporación de las preferencias del usuario ha sido recientemente abordado en [24], aquí se propone un método que no requiere parámetros adicionales para aproximar una región de interés. En este

trabajo los autores asumen que el TD expresa sus preferencias en la forma de un único punto de referencia. Sin embargo, el uso de una dirección, como medio para indicar las preferencias del TD, es una idea novedosa aplicada a MaOPs. Esta forma de definir las preferencias permite explorar el frente de Pareto y el resultado esperado es un camino de soluciones óptimas, que puede proporcionar información adicional en el proceso de la toma de decisiones.

2.2. Problema de enrutamiento de vehículos con ventanas de tiempo

El problema de enrutamiento de vehículos con ventanas de tiempo (VRPTW, por sus siglas en inglés) consiste en determinar el conjunto óptimo de rutas de vehículos idénticos (con la misma capacidad limitada) de manera de que todos los clientes (cuyas demandas son conocidas) sean atendidos una vez dentro de una ventana de tiempo dada para cada uno [16]. La formulación original de este problema está dada como

$$\text{mín} \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij}^k, \quad (4)$$

s.a.

$$x_{ij}^k = 0, \quad \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\}, \quad (4a)$$

$$\sum_{k=1}^K \sum_{i=1}^N x_{ij}^k = 1, \quad \forall j \in \{2, \dots, N\}, \quad (4b)$$

$$\sum_{i=1}^N x_{i0}^k = K, \quad (4c)$$

$$\sum_{j=1}^N x_{0j}^k = K, \quad (4d)$$

$$\sum_{i=1}^N \sum_{j=2}^N x_{ij}^k d_{ij} \leq q, \quad \forall k \in \{1, \dots, K\}, \quad (4e)$$

$$ot_j \leq s_{kj} \leq ct_j, \quad \forall j \in \{0, \dots, N\}, \forall k \in \{1, \dots, K\}, \quad (4f)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in \{0, \dots, N\}. \quad (4g)$$

Las restricciones (4a) y (4b) plantean que exactamente un vehículo debe entrar y exactamente uno debe partir, respectivamente. Las

restricciones (4c) y (4d) indican que el número de los vehículos que dejan el depósito es igual al número que entra. La restricción (4e) es la restricción de capacidad, la cual establece que las rutas tienen que ser conectadas y que la demanda en cada ruta no tiene que superar la capacidad del vehículo y la restricción (4f) asegura que las ventanas de tiempo sean tomadas en cuenta. Finalmente, la restricción (4g) define el dominio de las variables.

Para resolver este problema en el campo de muchos objetivos se utiliza como demostrador el VRPTW con cinco objetivos, los cuales son: minimización de la distancia, minimización del tiempo total, maximización de la eficiencia, minimización del número de vehículos y minimización del balance de ruta.

En primer lugar, la distancia es considerada como la suma de las distancias recorridas por cada vehículo, incluyendo su regreso hacia el almacén. El tiempo total es aquél que lleva realizar todas las entregas, esto incluye el tiempo del recorrido, de servicio y, de existir, el tiempo de retraso o de espera. La eficiencia está basada en la puntualidad de la entrega, dentro de las ventanas de tiempo se encuentra una hora la cual es considerada como la mejor para realizar la entrega por lo que la eficiencia es del 100 %, conforme la entrega se aleje de esta hora la eficiencia baja. Si no se cumple con la entrega dentro de las ventanas de tiempo se tendrá una eficiencia del 0%. El número de vehículos son el total necesario para realizar todas las entregas, cabe mencionar que estos vehículos necesitan utilizar el máximo posible de su capacidad. Finalmente, el balance de ruta es el equilibrio entre la carga de trabajo de cada vehículo, se espera encontrar un conjunto de rutas en el cual cada vehículo viaje aproximadamente la misma distancia.

En la literatura estos objetivos son frecuentemente utilizados, los más comunes han sido la distancia y el número de vehículos los cuales están presentes como objetivos en varios estudios [3, 49, 27, 32, 47]. El tiempo total es empleado en [18], como uno de los tres objetivos a optimizar. La eficiencia se considera en [20] y [25] como la satisfacción del cliente, la cual depende del

instante de llegada de la entrega y el balance de ruta aparece en [35] y [32].

El VRPTW ha sido sujeto de estudio en diversos trabajos; sin embargo, la mayoría de estos solamente resuelven problemas con dos y tres objetivos [32, 5, 36]. Uno de los estudios con el mayor número de objetivos es [21], con cuatro objetivos considerados que son: minimización de distancia, minimización de números de vehículos, minimización de tiempos de espera y maximización de satisfacción del cliente. Sin embargo, los resultados obtenidos son enfocados en encontrar las mejores soluciones conocidas en la literatura y no enfocadas en algún punto en específico como se plantea en este estudio.

3. Métodos

A continuación se proponen dos métodos de ajuste fino para una solución dada x_0 , seleccionada por el tomador de decisiones (TD) que posiblemente fue obtenida a través de técnicas de programación matemática o de un algoritmo evolutivo. Iniciando de x_0 , una secuencia de soluciones candidato x_i , $i = 1, \dots, N$ es generada a través de una dirección especificada por el usuario.

En este trabajo una dirección es la forma de indicar las preferencias del usuario. Al hacerlo de esta forma es posible especificar las funciones objetivo que desean mejorarse, las que serán sacrificadas e incluso en qué medida se verán afectadas durante el proceso de búsqueda. Esta idea se detalla a lo largo de este trabajo con diversos ejemplos.

3.1. Cambio en el espacio objetivo

El primer método de ajuste que se propone está basado en la suposición de que el TD tiene una dirección d preferida en el espacio objetivo. Es decir, dado x_0 , el TD está interesado en una solución x_1 del MaOP que se encuentra en el entorno de x_0 y tal que el valor en el espacio objetivo posee

$$F(x_1) \approx F(x_0) + td, \quad (5)$$

donde $t > 0$ es un tamaño de paso (pequeño) dado. Por ejemplo, si el TD quiere reducir el valor del primer objetivo él/ella podría establecer

$$d^{(1)} := \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad (6)$$

(suponiendo un problema bi-objetivo por razones de simplicidad y para no perder la generalización). Como otro ejemplo, si el TD quiere reducir el valor de f_1 a costa de f_2 donde f_1 es considerada más importante que f_2 (con un factor de 2), él/ella puede establecer

$$d^{(2)} := \begin{pmatrix} -2 \\ 1 \end{pmatrix} \quad (7)$$

y se puede continuar del mismo modo. Ya que es muy poco probable que exista un punto x_1 que sea exactamente como la Ecuación (5) (tenga en cuenta que el frente de Pareto alrededor de $F(x_0)$ es desconocido), se debe calcular la 'mejor aproximación' la cual se ha hecho de la siguiente manera

$$Z_1 := F(x_0) + \bar{t}d, \quad (8)$$

donde $\bar{t} > 0$ es un tamaño de paso fijo dado (dependiendo del problema), se desea calcular la solución óptima de Pareto para el MaOP dado que sea más cercana al punto de referencia Z_1 . Para ello, se ha utilizado la función de escalarización aumentada de Tchebycheff [46] la cual se define de la siguiente manera:

$$\begin{aligned} \min_{x \in \Omega} \max_{i=1, \dots, k} \{ \lambda_i |f_i(x) - Z_{1,i}| \} + \rho \sum_{i=1}^k \lambda_i (f_i(x) - Z_{1,i}) \\ \text{s.a. } g(x) \leq 0 \\ h(x) = 0. \end{aligned} \quad (9)$$

En el cual, $Z_{1,i}$ denota el i -ésimo componente del punto de referencia Z_1 , $\lambda = (\lambda_1, \dots, \lambda_k)^T$ es un vector de pesos con $\lambda_i \geq 0$, $i = 1, \dots, k$ y $\lambda \neq 0$.

Por lo tanto, al proceder de esta manera se obtiene una secuencia de soluciones candidato tales que la respectiva secuencia de imágenes realice (idealmente) un movimiento en la dirección d . El Algoritmo 1 muestra el pseudocódigo del método propuesto. La elección de $\|\bar{t}_{i-1}d\|$

define la distancia de dos soluciones consecutivas $\|F(x_{i-1}) - F(x_i)\|$ en el espacio objetivo y por lo tanto depende del problema.

Algoritmo 1 Reajuste I: Cambio en espacio objetivo

Require: punto inicial x_0 , dirección d en el espacio objetivo

Ensure: : sucesión $\{x_i\}$ de soluciones candidatas

for $i = 1, 2, \dots$ **do**

 elegir $\bar{t}_{i-1} \in \mathbb{R}_+$

 asignar $Z_i := F(x_{i-1}) + \bar{t}_{i-1}d$

 resolver (9) iniciando con x_{i-1} para obtener

x_i

end for

La Figura 1 muestra dos escenarios hipotéticos, de nuevo para un problema bi-objetivo debido a que se tiene una mejor visualización. Para x_0 , el punto de referencia Z_1 es factible cuando se elige una dirección $d = [1, -1]$. Es decir, existe un punto \tilde{x} tal que $F(\tilde{x}) = Z_1$. El segundo término en (9) impide que \tilde{x} sea elegido. En lugar de eso, la solución x_1 es una solución óptima de Pareto tal que $F(x_1)$ es el elemento más cercano a Z_1 en el frente de Pareto. El segundo escenario es para un punto dado \bar{x} donde \bar{Z}_1 es no factible. Una vez más, la solución \bar{x}_1 de (9) es una solución óptima de Pareto cuyo $F(\bar{x}_1)$ es la más cercana al punto de referencia dado.

3.2. Cambio en el espacio de pesos

El segundo algoritmo de ajuste que se propone se basa en la suposición de que el TD tiene una idea acerca de la importancia de preferencia de cada uno de los objetivos y le gustaría hacer un cambio conforme a esto en el espacio de pesos. Este método está basado en el método de suma ponderada [19] el cual probablemente sea el método de escalarización más antiguo para MOPs. Más precisamente, dado un vector α de pesos tal que

$$\alpha_i \geq 0, \quad i = 1, \dots, k, \quad \text{y} \quad \sum_{i=1}^k \alpha_i = 1, \quad (10)$$

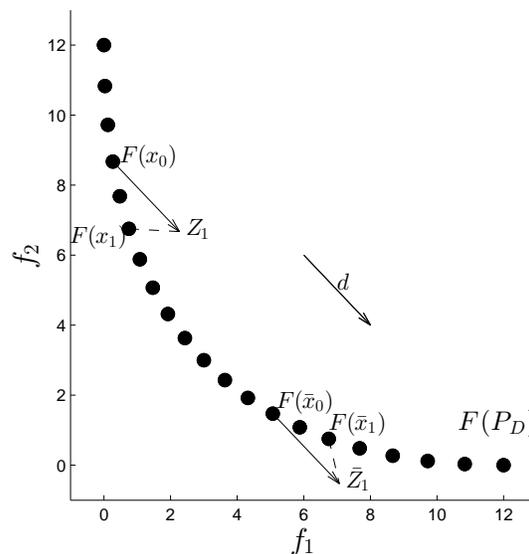


Fig. 1. Dos escenarios hipotéticos para el cambio en el espacio objetivo en un problema bi-objetivo

(α es también llamada una combinación convexa), entonces, el objetivo a optimizar es minimizar la suma ponderada de los objetivos

$$\min_{x \in D} \sum_{i=1}^k \alpha_i f_i(x). \quad (11)$$

El cambio en el espacio de pesos puede ser realizado como se muestra a continuación: asumiendo que una combinación convexa $\alpha^{(0)}$ es dada y un vector $\Delta\alpha \in \mathbb{R}^k$ el cual representa los cambios deseados con la propiedad

$$\sum_{i=1}^k \Delta\alpha_i = 0, \quad (12)$$

entonces, el nuevo vector de pesos puede ser calculado a través de

$$\alpha^{(1)} := \alpha^{(0)} + t\Delta\alpha, \quad (13)$$

donde $t > 0$ es un tamaño de paso dado. Es fácil de ver que si t es lo suficientemente pequeño, entonces $\alpha^{(1)}$ es también una combinación convexa. Como ejemplo se puede considerar que

los pesos iniciales están dado por

$$\alpha^{(0)} = \begin{pmatrix} 0.4 \\ 0.3 \\ 0.3 \end{pmatrix} \quad (14)$$

para un problema hipotético de tres objetivos, y que el TD quisiera dejar la importancia en f_1 mientras incrementa la importancia en f_2 y reduce f_3 . Por lo tanto, se obtiene

$$\Delta\alpha = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} \quad (15)$$

y por consiguiente

$$\alpha^{(1)} = \begin{pmatrix} 0.4 \\ 0.3 \\ 0.3 \end{pmatrix} + 0.1 \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0.4 \\ 0.4 \\ 0.2 \end{pmatrix}. \quad (16)$$

Un problema que permanece es que el vector inicial de pesos $\alpha^{(0)}$ no se conoce exactamente a menos que la solución inicial x_0 sea obtenida a través del método de suma ponderada. Por lo tanto, si $\alpha^{(0)}$ es solo una aproximación, se puede llegar a tener un gran cambio no deseado en la solución de (11) cuando se utiliza el nuevo peso. Para evitar esto, fue modificado el método de suma ponderada como se muestra:

$$\begin{aligned} \min_{x \in D} \sum_{i=1}^k \alpha_i f_i(x) \\ \text{s.a. } \|F(x) - F(x_0)\| \leq \epsilon, \end{aligned} \quad (17)$$

donde $\epsilon > 0$ es un valor (que depende del problema) el cual tiene en principio la misma influencia en la solución $\|\bar{i}d\|$ que para el primer enfoque de ajuste. El Algoritmo 2 muestra el pseudo código para el segundo método de ajuste propuesto. El algoritmo tiene que ser detenido si uno de los pesos se vuelve no positivo.

4. Resultados numéricos

A continuación se presentan algunos resultados hipotéticos para los métodos de ajuste propuestos. Se considera cada método en un MaOP discreto

Algoritmo 2 Reajuste II: Cambio en el espacio de pesos

Require: punto inicial x_0 , peso inicial $\alpha^{(0)}$, vector de cambio $\Delta\alpha$, tolerancia ϵ para el cambio máximo en el espacio objetivo

Ensure: : sucesión $\{x_i\}$ de soluciones candidatas
for $i = 1, 2, \dots$ **do**

elegir $\bar{t}_{i-1} \in \mathbb{R}_+$

asignar $\alpha^{(i)} := \alpha^{(i-1)} + \bar{t}_{i-1} \Delta\alpha$

resolver (17) utilizando $\alpha^{(i)}$ como vector de pesos y ϵ como tolerancia para obtener x_i

end for

con 5 funciones objetivo y en un MaOP continuo con 10 funciones objetivo.

Como MaOP discreto, se ha optado por tomar el VRPTW con 5 los objetivos explicados anteriormente con fines ilustrativos. Fueron utilizados datos de referencia de [8] y fue seleccionado un problema con 100 variables que dan origen a los cinco objetivos que se consideraron: distancia f_1 , tiempo total f_2 , número de vehículos f_3 , eficiencia f_4 y el balance de ruta f_5 . Con el fin de resolver los SOPs definidos en este trabajo, fue utilizado un algoritmo genético, donde el VRPTW se representa como una lista de $n + 1$ nodos, aquí n es el número de clientes, el nodo $i \in \{1, n\}$ representa un cliente, y el nodo 0 representa el depósito. Cada nodo también se asocia con la demanda del cliente, tiempo de espera, los límites de tiempo y el tiempo de servicio. La solución inicial se genera mediante la elección de una permutación aleatoria que representa los nodos, formando una lista de longitud n , donde cada nodo aparece una sola vez y los rendimientos de los vehículos no se consideran. El resultado se basa en los siguientes parámetros: tamaño de la población = 200, número de generaciones = 3000, la probabilidad de cruza = 0.9 y probabilidad de mutación = 0.25.

En primer lugar, se considera el ajuste a través del cambio en el espacio objetivo. Para ello, se considera el siguiente escenario: con base en el conocimiento previo del problema, se ha elegido el conjunto de referencia inicial como $Z_0 = [2200, 530000, 19, 0.45, 77]$ y se calculó la solución óptima inicial x_0 mediante la resolución de (9).

El correspondiente vector objetivo es $F(x_0) = [2298.4, 489780, 19, 0.42129, 85.6]$. Además, se asume que el TD desea explorar las posibles soluciones óptimas alrededor de x_0 donde f_2 se mejora con un sacrificio de los valores de f_1 y f_5 , lo que induce la dirección $d = [100, -7500, 0, 0, 1.65]$. Los 20 nuevos puntos de referencia se establecen como

$$Z_i := Z_0 + i \begin{pmatrix} 100 \\ -7500 \\ 0 \\ 0 \\ 1.65 \end{pmatrix}, \quad i = 1 \dots 20. \quad (18)$$

Para resolver cada SOP se establecieron 200 individuos y 30,000 evaluaciones de la función como criterio de paro para cada punto de referencia en el conjunto. Se realizaron 30 ejecuciones independientes bajo las condiciones antes descritas. La Figura 2 muestra los valores resultantes de una ejecución aleatoria utilizando la gráfica de radar [31] a cada iteración para ilustrar los valores de los objetivos. En este tipo de gráfica un mejor valor de la función objetivo se representa con un vértice del polígono más cercano al círculo.

La Figura 3 muestra el mismo resultado, pero esta vez se representa la iteración i contra los valores de cada función objetivo (se denomina gráfica de línea [31]). En esta gráfica, se muestran también los valores de los puntos de referencia. Como se puede observar, se obtiene una muy buena aproximación (note que un resultado perfecto sólo puede obtenerse si Z_i coincide con el frente de Pareto, lo cual no es el caso aquí).

Por su parte, la Tabla 1 muestra los valores de las funciones para tres soluciones seleccionadas las cuales siguen la dirección d especificada por el TD.

Tabla 1. Valores seleccionados de las funciones objetivo de una ejecución aleatoria para el primer escenario

Punto	f_1	f_2	f_3	f_4	f_5
$F(x_5)$	2434.2	465600	19	0.431103	91.1
$F(x_{10})$	3188.2	454080	19	0.449417	93.3
$F(x_{20})$	3960.3	393780	19	0.436728	106

Con el fin de comparar nuestro método, se utilizó también el enfoque de conjunto de aspiración que

se propone en [41]. En este trabajo se extiende la idea de métodos de *puntos* de referencia, ya que el objetivo es minimizar una relación de la imagen de una población hacia un *conjunto* de referencia dado. Más precisamente, los autores proponen resolver el siguiente SOP:

$$\begin{aligned} \min_{\substack{A \subset \mathbb{R}^n \\ |A| = N}} \Delta_p(F(A), Z), \end{aligned} \quad (19)$$

donde Δ_p denota la distancia de Hausdorff promedio (en los resultados que aquí se presentan se utiliza el valor $p = 2$), y $Z := \{Z_0, Z_1, \dots, Z_{20}\}$ es la unión de los puntos de referencia considerados. Para resolver este problema, se ha considerado un tamaño de población de 200 y se ha fijado un límite de 65,000 evaluaciones de función para el problema (19), con el fin de obtener un desempeño general similar (note que una evaluación de la función en (19) es equivalente a $n \cdot N$ evaluaciones de función de F). Se realizaron también 30 ejecuciones independientes con este enfoque. La Figura 4 muestra la gráfica de línea del resultado obtenido para una ejecución aleatoria. Como puede verse, el comportamiento no es tan bueno como en el obtenido en el resultado anterior. Para tener una mejor comparación de los resultados, se consideran los siguientes indicadores: en primer lugar, el indicador de cobertura de conjunto (CS, por sus siglas en inglés) [52] el cual calcula la dominancia relativa que hay entre dos conjuntos. Sean $A, B \in \Omega$ dos conjuntos, el CS se define como la relación de la pareja ordenada (A, B) al intervalo $[0, 1]$:

$$CS(A, B) := \frac{|\{a \in A; \exists b \in B : b \succeq a\}|}{|A|}. \quad (20)$$

En segundo lugar, se propone el uso de la *distancia total (DT)* entre el conjunto de referencia $Z \in \mathbb{R}^k$ y la imagen de la aproximación obtenida A . i.e.,

$$DT(Z, A) := \sum_{i=1}^{|Z|} \|Z_i - F(A_i)\|. \quad (21)$$

Con el fin de hacer comparaciones justas, se utilizaron las 20 soluciones de la población final del

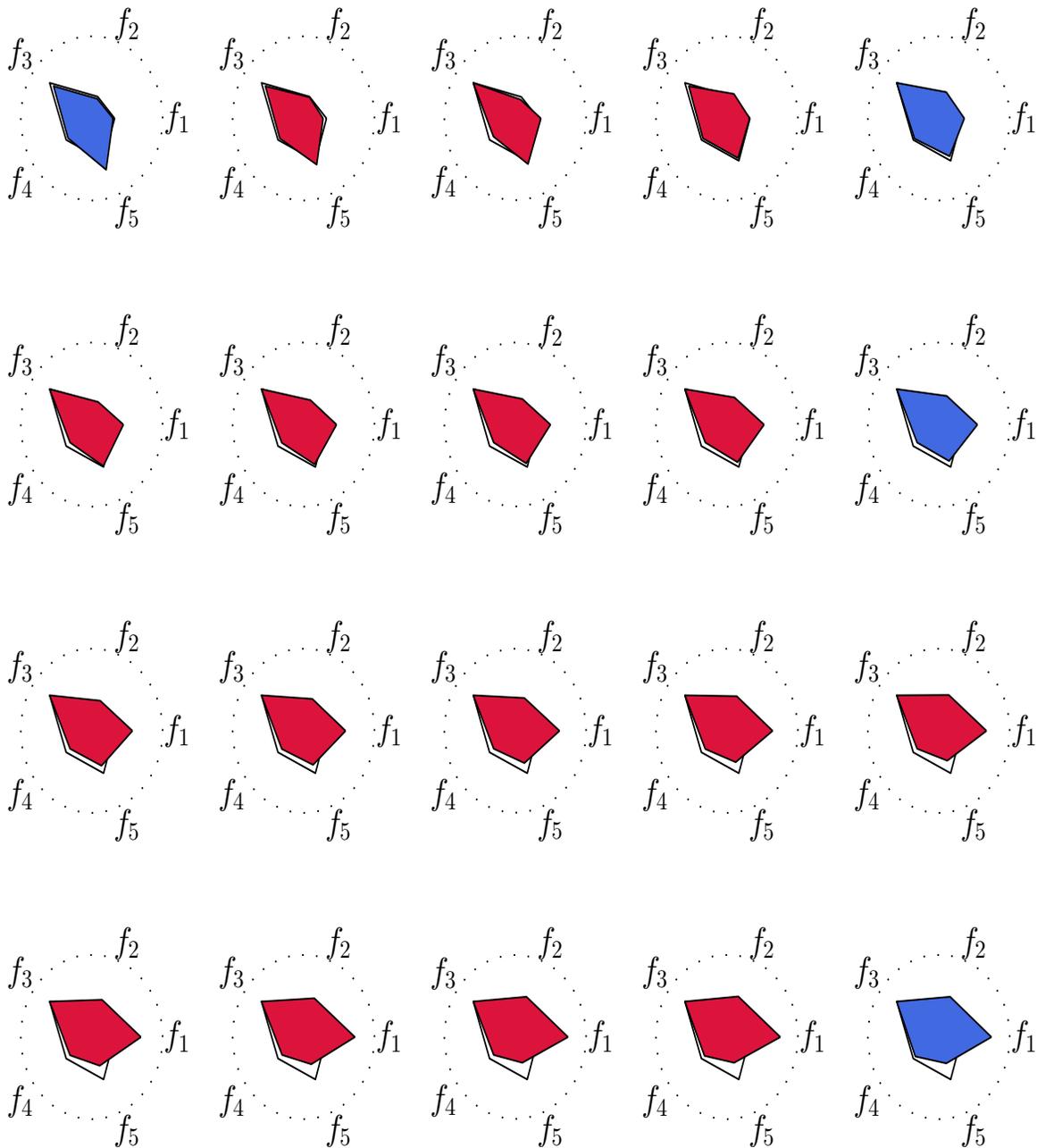


Fig. 2. Representación con la gráfica de radar del resultado del primer método de ajuste para una ejecución aleatoria del algoritmo. Las soluciones se muestran desde x_0 en la esquina superior izquierda hasta x_{20} en la esquina inferior derecha

enfoque de aspiración (19) que más contribuyen al valor de Δ_p [43]. Los resultados considerando las 30 ejecuciones de cada método se muestran en la Tabla 2.

La Tabla 2 revela que el enfoque de ajuste es mejor de acuerdo con ambos indicadores. Los resultados muestran una significancia estadística bajo la prueba de Wilcoxon en todos los casos.

Tabla 2. Valores de los indicadores de los dos métodos elegidos para el primer escenario. Se indican el valor mínimo, la media junto con la desviación estándar y el valor máximo en cada caso. Las pruebas presentan una significancia estadística bajo la prueba de Wilcoxon

Indicador	Método de Ajuste	Método de Aspiración
CS	0.9500	0.0000
	0.9975 (0.0112)	0.000 (0.0000)
	1.0000	0.0000
DT	3.3169e+04	3.4846e+05
	4.3045e+04 (5.2092e+03)	3.8645e+05 (2.5070e+04)
	5.2426e+04	4.1134e+05
Δ_p	3.2517e+03	4.0561e+04
	4.3355e+03 (3.3590e+02)	4.6899e+04 ((3.6755e+03)
	4.8641e+03	5.7172e+04

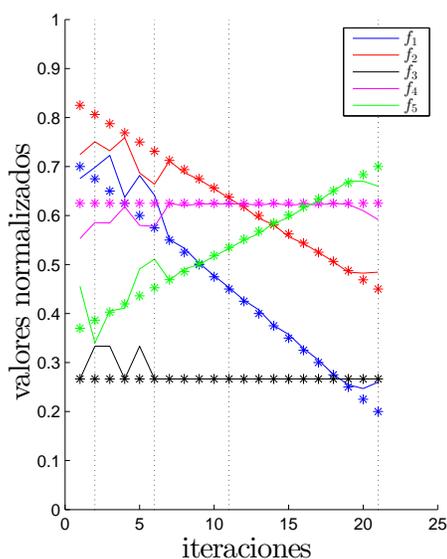


Fig. 3. Representación con la gráfica de línea del resultado del primer método de ajuste para una ejecución aleatoria

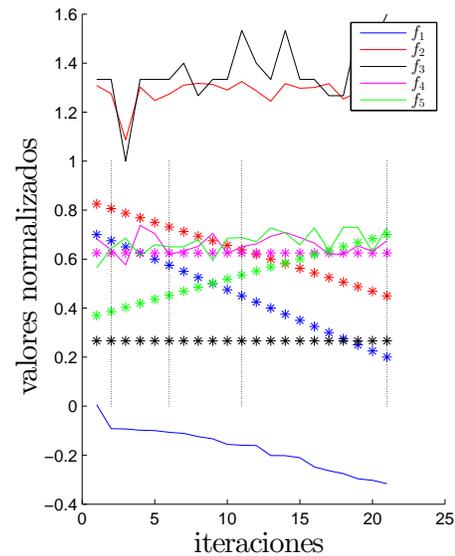


Fig. 4. Representación con la gráfica de línea del resultado del enfoque de logro usado para la comparación

Ahora, se considera un escenario hipotético para el segundo enfoque de ajuste. Se toma el mismo problema de 5 objetivos y se elige un peso inicial $\alpha^{(0)} := [0.2, 0.2, 0.2, 0.2, 0.2] \in \mathbb{R}^5$. La solución al problema de suma ponderada para $\alpha^{(0)}$ está dado por el vector x_0 con $F(x_0) = [2679.5, 363900, 13, 0.348953, 95.7]$. Además, se ha elegido $\Delta\alpha = [0.3, -0.2, -0.2, -0.2, 0.3]$ y $\epsilon = 10,000$. $\Delta\alpha$, lo que corresponde a una reducción de f_1 y f_5 , mientras que el valor de los objetivos f_2, f_3 y f_4 son sacrificados. Como secuencia de los pesos se tiene entonces

$$\alpha^{(i)} := \alpha_0 + i \begin{pmatrix} 0.3 \\ -0.2 \\ 0.2 \\ 0.2 \\ 0.3 \end{pmatrix}, i = 1 \dots 20. \quad (22)$$

Se considera una población de 200 individuos y se establecen 30,000 evaluaciones de función como condición de paro para obtener la solución de cada problema del tipo (17). Aunque este método no será comparado, se realizaron también 30 ejecuciones independientes bajo las condiciones

antes descritas. Los resultados se muestran en las Figuras 5 y 6 para una ejecución aleatoria. La Tabla 3 muestra los valores de las funciones objetivo para las iteraciones elegidas $i = 5, 10$ y 20 de la misma ejecución seleccionada previamente. Estas iteraciones (así como la inicial) se indican en la gráfica de línea mediante una línea vertical punteada y en la gráfica de radar mediante un color diferente del polígono. Los valores de cada función objetivo han sido normalizados, en ambos casos, con base en los resultados obtenidos por los métodos de ajuste. Se conserva la misma normalización en la comparación contra el enfoque de logro.

Se puede notar que los saltos en el espacio objetivo para este método no son tan suaves como los obtenidos para el caso del enfoque de cambio en el espacio objetivo. Esto se debe probablemente al hecho de que no existe una relación directa entre los pesos y el control de tamaño de paso. Sin embargo, el uso de ϵ , ayuda a preservar una cierta distancia. De acuerdo con los resultados, la distancia máxima entre dos soluciones consecutivas fue de 9540, mientras que la mínima fue de 900, i.e., se tiene una diferencia de un orden de magnitud. No obstante, el método elegido puede ser útil en el proceso de toma de decisiones ya que presenta una serie de soluciones candidato a lo largo del conjunto de Pareto de acuerdo a la dirección de preferencia dada. Cabe señalar que para este enfoque no es posible presentar una comparación ya que no existe un método suficientemente parecido para este fin.

Tabla 3. Valores seleccionados de las funciones objetivo para el segundo escenario

Punto	f_1	f_2	f_3	f_4	f_5
$F(x_5)$	2689.8	405480	15	0.359921	72.5
$F(x_{10})$	2291.9	387780	14	0.36190	81.1
$F(x_{20})$	1737.0	398880	15	0.400363	78.7

Se destaca que se ha realizado la demostración en un problema ampliamente estudiado con 5 objetivos. Nótese que en ambos casos se considera la solución de una serie de SOPs y que el número k de los objetivos no tiene una influencia crucial en la complejidad de los problemas. Por

lo tanto, los métodos presentados pueden ser, en principio, aplicados a los MaOPs para cualquier número k de objetivos, algo que no es posible con otros métodos; por ejemplo, aquellos que buscan calcular todo el conjunto de soluciones.

Es importante señalar que los métodos de reajuste propuestos en este trabajo pueden de igual forma ser utilizados en problemas continuos. Al utilizar un algoritmo evolutivo para resolver los problemas (9) o (17), según sea el caso, se tiene la posibilidad de extender el dominio de los problemas que pueden resolverse con estos enfoques.

En este trabajo se considera el MaOP continuo DTLZ2 [14] únicamente con el fin de ilustrar el funcionamiento general de los métodos discutidos anteriormente. Dicho problema está dado por

$$\begin{aligned}
 f_1(x) &= (1 + g(X_k)) \prod_{i=1}^{k-1} \cos(0.5x_i\pi), \\
 f_2(x) &= (1 + g(X_k)) \sin(0.5x_{k-1}\pi) \prod_{i=1}^{k-2} \cos(0.5x_i\pi), \\
 &\vdots \\
 f_k(x) &= (1 + g(X_k)) \sin(0.5x_1\pi),
 \end{aligned} \tag{23}$$

donde

$$g(X_k) = \sum_{x_i \in X_k} (x_i - 0.5)^2, \tag{24}$$

y X_k es el vector (x_k, \dots, x_n) para $n > k$. En los ejemplos siguientes se utiliza $n = 15$ y $k = 10$.

Para ejemplo del método de reajuste I se tiene $F(x_0) = (0.3164, \dots, 0.3164)^T \in \mathbb{R}^{10}$, la dirección $d = (1, -1, \dots, 1, -1)^T \in \mathbb{R}^{10}$ y un tamaño de paso $t = 0.05$. Es decir, partiendo de una solución en la que todos los objetivos valen los mismo, se desea mejorar el valor de las funciones pares sacrificando las funciones impares. Se calcularon 50 iteraciones del algoritmo.

El resultado obtenido se presenta en forma de gráfica de radar en la Figura 7a. Se presentan cuatro momentos del método: el inicial, luego de un cuarto del total de las iteraciones, a la mitad del proceso y la iteración final, respectivamente.

Por otro lado, para el método de reajuste II se consideró: $x_0 = (0.5, \dots, 0.5)^T \in \mathbb{R}^{15}$ como

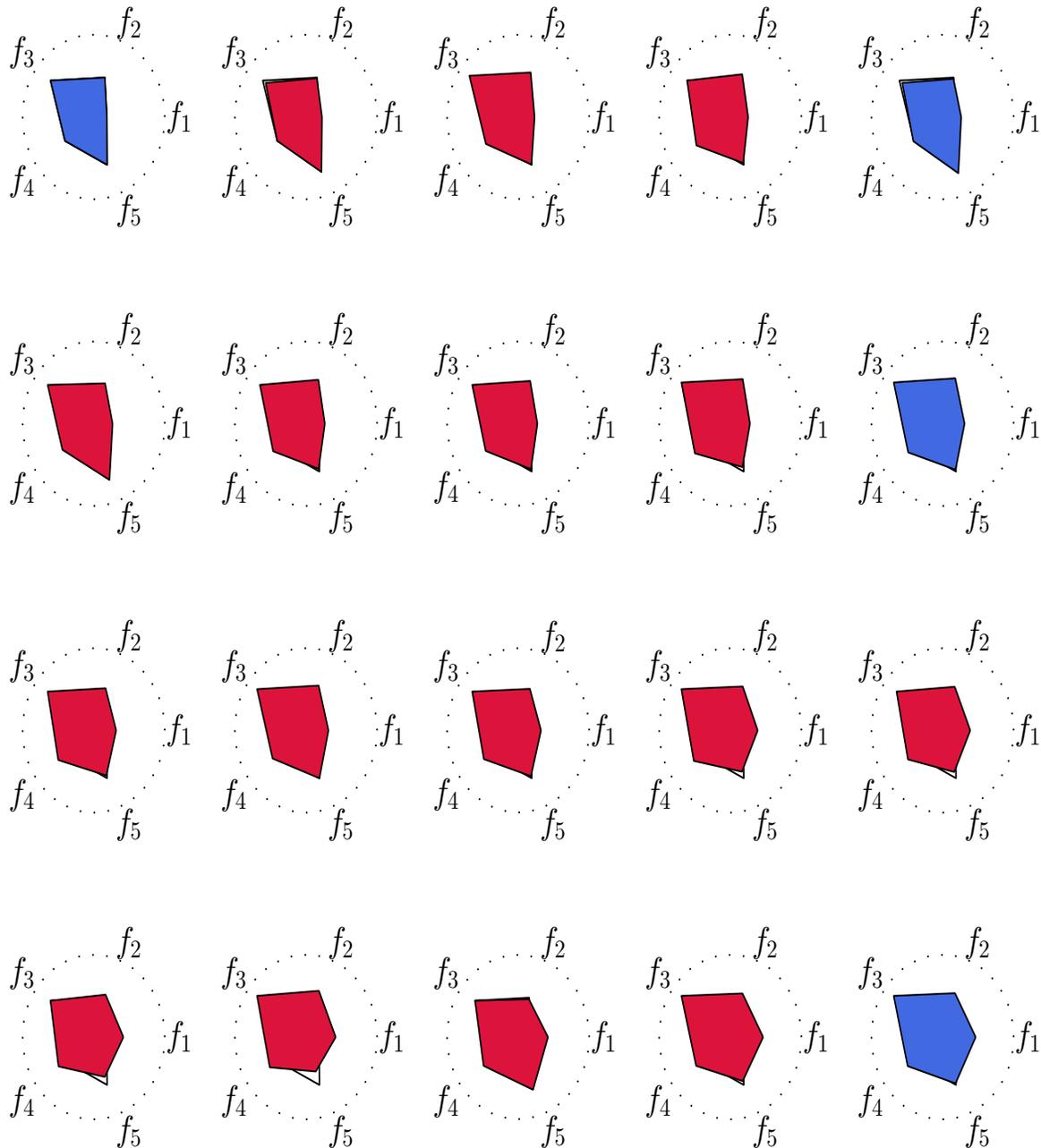


Fig. 5. Representación con la gráfica de radar del resultado del segundo método de ajuste. Las soluciones se muestran desde x_0 en la esquina superior izquierda hasta x_{20} en la esquina inferior derecha

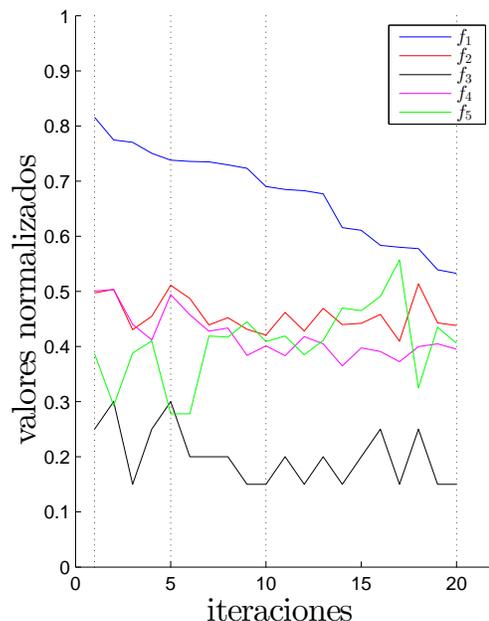


Fig. 6. Representación con la gráfica de línea del resultado del segundo método de ajuste

solución óptima inicial, $\Delta\alpha = (1, 1, \dots, 1, -9, 1) \in \mathbb{R}^{10}$ y $t = 0.05$. En este caso el objetivo es mejorar el valor de todas las funciones a excepción de f_9 , la función sacrificada.

En este ejemplo se calcularon 22 iteraciones del método antes que el peso asignado a f_9 fuera negativo. La gráfica de radar para este ejemplo puede verse en la Figura 7b, se consideran los mismos momentos que en ejemplo anterior.

Se resalta de las instancias para el caso continuo la aplicabilidad de los enfoques propuestos en diferentes dominios. Además, se obtuvieron trayectorias que siguen la dirección propuesta para un problema con 10 objetivos, se consideraron direcciones ilustrativas para este fin.

5. Conclusiones y trabajo futuro

En este trabajo hemos abordado el procesamiento numérico de Problemas de Optimización de Muchos Objetivos (MaOPs) discretos por medio de estrategias evolutivas. Debido a los recientes

avances en este campo, los algoritmos evolutivos modernos son capaces de encontrar rápidamente un conjunto de soluciones esparcidas a lo largo del conjunto/frente de Pareto en un determinado MaOP.

Sin embargo, como las magnitudes de los frentes de Pareto pueden ser muy grandes con el incremento del número de objetivos, estos conjuntos ya no se pueden calcular completamente ni ser visualizados de manera eficiente. Por lo tanto, es de esperarse que cierto conjunto S obtenido por un algoritmo evolutivo (o cualquier otro tipo de algoritmo) revele grandes vacíos. Por consecuencia, no se puede esperar que S contenga la solución 'ideal' para el problema dado. En cambio, parece lógico encontrar una solución que se adapte mejor a satisfacer las necesidades de quien toma las decisiones.

En este artículo, se han propuesto dos métodos que son capaces de obtener una secuencia de soluciones partiendo de un punto inicial y siguiendo la trayectoria de una dirección especificada por el usuario. Concretamente, se propuso un método que dirige la búsqueda a lo largo del frente de Pareto acorde a una dirección d dada en el espacio objetivo.

El segundo método utiliza las preferencias del usuario entre los diferentes objetivos para lo cual se ha modificado el método de suma ponderada. Los resultados numéricos, en dos escenarios, de un caso particular del problema de enrutamiento de vehículos con 5 objetivos, así como en un MaOP continuo con 10 objetivos, han demostrado la aplicabilidad del nuevo enfoque.

Aunque, estos primeros resultados son prometedores, todavía hay varias cuestiones que deben abordarse en trabajos futuros. Por ejemplo, uno de ellos sería el considerar aún más características en el movimiento que puedan interesar al tomador de decisiones. Además, la consideración de las comparaciones debe ser fortalecida. Finalmente, se desea atacar ciertos problemas del mundo real con el fin de demostrar la fortaleza de este nuevo enfoque.

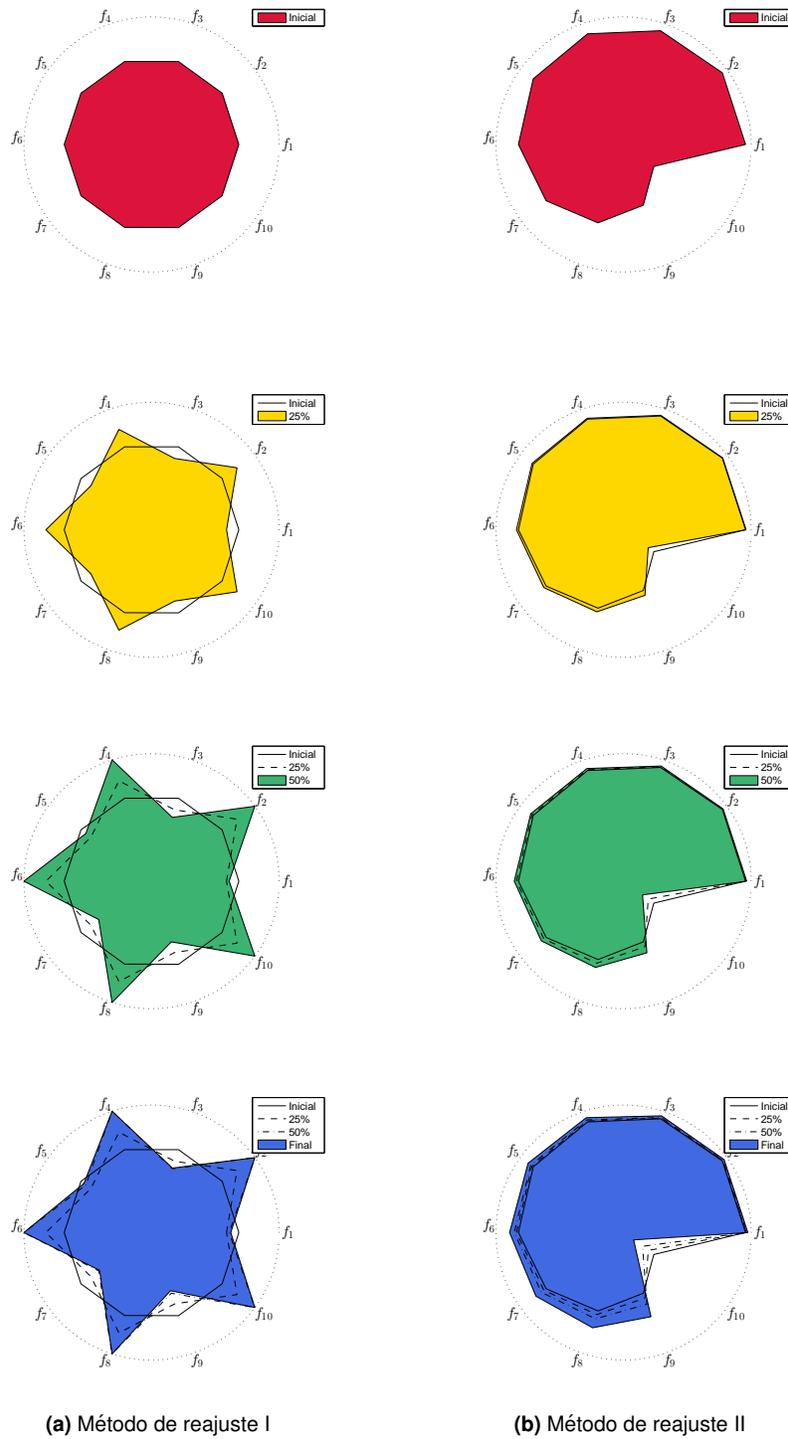


Fig. 7. Ejemplos de los enfoques propuestos para la función DTLZ2

Agradecimientos

Nancy Pérez agradece el apoyo recibido por parte del CONACyT por la beca que se le otorgó para la realización de una estancia de investigación en el CINESTAV-IPN, donde nació este proyecto, así mismo por el apoyo recibido durante sus estudios de maestría en la UACJ. Oliver Cuate agradece el apoyo del CONACyT para proseguir sus estudios de doctorado en el CINESTAV-IPN.

Referencias

1. **Aguirre, H. & Tanaka, K. (2009).** Many-objective optimization by space partitioning and adaptive ε -ranking on mnk-landscapes. *Evolutionary Multi-Criterion Optimization*, Springer, pp. 407–422.
2. **Bader, J. & Zitzler, E. (2011).** Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, Vol. 19, No. 1, pp. 45–76.
3. **Baños, R., Ortega, J., Gil, C., Márquez, A. L., & De Toro, F. (2013).** A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. *Computers & Industrial Engineering*, Vol. 65, No. 2, pp. 286–296.
4. **Batista, L. S., Campelo, F., Guimarães, F. G., & Ramírez, J. A. (2011).** A comparison of dominance criteria in many-objective optimization problems. *2011 IEEE Congress of Evolutionary Computation (CEC)*, IEEE, pp. 2359–2366.
5. **Braekers, K., Caris, A., & Janssens, G. K. (2011).** A deterministic annealing algorithm for a bi-objective full truckload vehicle routing problem in drayage operations. *Procedia-Social and Behavioral Sciences*, Vol. 20, pp. 344–353.
6. **Branke, J., Deb, K., Miettinen, K., & Slowiński, R. (2008).** *Multiobjective optimization: Interactive and evolutionary approaches*, volume 5252. Springer.
7. **Buchanan, J. T. (1997).** A naive approach for solving MCDM problems: The guess method. *Journal of the Operational Research Society*, Vol. 48, No. 2, pp. 202–206.
8. **Castro-Gutierrez, J., Landa-Silva, D., & Pérez, J. M. (2011).** Nature of real-world multi-objective vehicle routing with evolutionary algorithms. *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, IEEE, pp. 257–264.
9. **Coello, C. A. C., Van Veldhuizen, D. A., & Lamont, G. B. (2002).** *Evolutionary algorithms for solving multi-objective problems*, volume 242. Springer.
10. **Coello, C. C. (2000).** Handling preferences in evolutionary multiobjective optimization: A survey. *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, IEEE, pp. 30–37.
11. **Deb, K. (2001).** *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons.
12. **Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002).** A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, Vol. 6, No. 2, pp. 182–197.
13. **Deb, K., Sundar, J., Udaya Bhaskara Rao, N., & Chaudhuri, S. (2006).** Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research*, Vol. 2, No. 3, pp. 273–286.
14. **Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002).** Scalable multi-objective optimization test problems. *Proceedings of the Congress on Evolutionary Computation (CEC-2002)*, pp. 825–830.
15. **Di Pierro, F., Khu, S.-T., Savic, D., et al. (2007).** An investigation on preference order ranking scheme for multiobjective evolutionary optimization. *Evolutionary Computation, IEEE Transactions on*, Vol. 11, No. 1, pp. 17–45.
16. **El-Sherbeny, N. A. (2010).** Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University-Science*, Vol. 22, No. 3, pp. 123–131.
17. **Farina, M. & Amato, P. (2002).** On the optimal solution definition for many-criteria optimization problems. *Proceedings of the NAFIPS-FLINT international conference*, pp. 233–238.
18. **García-Najera, A. & Bullinaria, J. A. (2011).** An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, Vol. 38, No. 1, pp. 287–300.
19. **Gass, S. & Saaty, T. (1955).** The computational algorithm for the parametric objective function. *Naval research logistics quarterly*, Vol. 2, No. 1-2, pp. 39–45.
20. **Ghannadpour, S. F., Noori, S., & Tavakkoli-Moghaddam, R. (2013).** Multiobjective dynamic vehicle routing problem with fuzzy travel times and

- customers' satisfaction in supply chain management. *Engineering Management, IEEE Transactions on*, Vol. 60, No. 4, pp. 777–790.
21. Ghannadpour, S. F., Noori, S., Tavakkoli-Moghaddam, R., & Ghoseiri, K. (2014). A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. *Applied Soft Computing*, Vol. 14, pp. 504–527.
 22. Ghosh, A. & Tsutsui, S. (2012). *Advances in evolutionary computing: theory and applications*. Springer Science & Business Media.
 23. Gong, M., Liu, F., Zhang, W., Jiao, L., & Zhang, Q. (2011). Interactive moea/d for multi-objective decision making. *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, ACM, pp. 721–728.
 24. Goulart, F. & Campelo, F. (2016). Preference-guided evolutionary algorithms for many-objective optimization. *Information Sciences*, Vol. 329, pp. 236–255.
 25. Guerriero, F., Surace, R., Loscri, V., & Natalizio, E. (2014). A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows constraints. *Applied Mathematical Modelling*, Vol. 38, No. 3, pp. 839–852.
 26. Hillermeier, C. (2001). *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser.
 27. Hsu, W.-H. & Chiang, T.-C. (2012). A multiobjective evolutionary algorithm with enhanced reproduction operators for the vehicle routing problem with time windows. *Evolutionary Computation (CEC), 2012 IEEE Congress on*, IEEE, pp. 1–8.
 28. Hughes, E. J. (2005). Evolutionary many-objective optimisation: many once or one many? *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, IEEE, pp. 222–227.
 29. Ishibuchi, H., Sakane, Y., Tsukamoto, N., & Nojima, Y. (2009). Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations. *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, IEEE, pp. 1758–1763.
 30. Knowles, J. & Corne, D. (2007). Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. *Evolutionary multi-criterion optimization*, Springer, pp. 757–771.
 31. Korhonen, P. & Wallenius, J. (2008). Visualization in the multiple objective decision-making framework. In *Multiobjective optimization*. Springer, pp. 195–212.
 32. Kumar, V. S., Thansekhar, M., Saravanan, R., & Amali, S. M. J. (2014). Solving multi-objective vehicle routing problem with time windows by FAGA. *Procedia Engineering*, Vol. 97, pp. 2176–2185.
 33. Luque, M., Miettinen, K., Eskelinen, P., & Ruiz, F. (2009). Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, Vol. 37, No. 2, pp. 450–462.
 34. Martín, A. & Schütze, O. (2014). A new predictor corrector variant for unconstrained bi-objective optimization problems. In *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*. Springer, pp. 165–179.
 35. Martínez-Salazar, I. A., Molina, J., Ángel-Bello, F., Gómez, T., & Caballero, R. (2014). Solving a bi-objective transportation location routing problem by metaheuristic algorithms. *European Journal of Operational Research*, Vol. 234, No. 1, pp. 25–36.
 36. Melián-Batista, B., De Santiago, A., AngelBello, F., & Alvarez, A. (2014). A bi-objective vehicle routing problem with time windows: A real case in tenerife. *Applied Soft Computing*, Vol. 17, pp. 140–152.
 37. Miettinen, K., Eskelinen, P., Ruiz, F., & Luque, M. (2010). Nautilus method: An interactive technique in multiobjective optimization based on the nadir point. *European Journal of Operational Research*, Vol. 206, No. 2, pp. 426–434.
 38. Miettinen, K. & Mäkelä, M. M. (2000). Interactive multiobjective optimization system www-nimbus on the internet. *Computers & Operations Research*, Vol. 27, No. 7, pp. 709–723.
 39. Purshouse, R. C. & Fleming, P. J. (2007). On the evolutionary optimization of many conflicting objectives. *Evolutionary Computation, IEEE Transactions on*, Vol. 11, No. 6, pp. 770–784.
 40. Ranjithan, S. R., Chetan, S. K., & Dakshina, H. K. (2001). Constraint method-based evolutionary algorithm (cmea) for multiobjective optimization. *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, pp. 299–313.
 41. Rudolph, G., Schütze, O., Grimme, C., & Trautmann, H. (2014). A multiobjective evolutionary algorithm guided by averaged hausdorff distance to

- aspiration sets. In *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*. Springer, pp. 261–273.
42. **Schaffer, J. D. (1985)**. Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of the 1st international Conference on Genetic Algorithms*, L. Erlbaum Associates Inc., pp. 93–100.
 43. **Schütze, O., Esquivel, X., Lara, A., & Coello, C. A. C. (2012)**. Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. *Evolutionary Computation, IEEE Transactions on*, Vol. 16, No. 4, pp. 504–522.
 44. **Schütze, O., Martín, A., Lara, A., Alvarado, S., Salinas, E., & Coello, C. A. (2016)**. The directed search method for multiobjective memetic algorithms. *Computational Optimization and Applications*, Vol. 63, pp. 305–332.
 45. **Singh, H. K., Isaacs, A., & Ray, T. (2011)**. A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *Evolutionary Computation, IEEE Transactions on*, Vol. 15, No. 4, pp. 539–556.
 46. **Steuer, R. E. & Choo, E.-U. (1983)**. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical programming*, Vol. 26, No. 3, pp. 326–344.
 47. **Tan, K. C., Cheong, C. Y., & Goh, C. K. (2007)**. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of operational research*, Vol. 177, No. 2, pp. 813–839.
 48. **Thiele, L., Miettinen, K., Korhonen, P. J., & Molina, J. (2009)**. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary computation*, Vol. 17, No. 3, pp. 411–436.
 49. **Yalcin, G. D. & Erginel, N. (2015)**. Fuzzy multi-objective programming algorithm for vehicle routing problems with backhauls. *Expert Systems with Applications*, Vol. 42, No. 13, pp. 5632–5644.
 50. **Yang, S., Li, M., Liu, X., & Zheng, J. (2013)**. A grid-based evolutionary algorithm for many-objective optimization. *Evolutionary Computation, IEEE Transactions on*, Vol. 17, No. 5, pp. 721–736.
 51. **Zhang, Q. & Li, H. (2007)**. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, Vol. 11, No. 6, pp. 712–731.
 52. **Zitzler, E., Deb, K., & Thiele, L. (2000)**. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, Vol. 8, No. 2, pp. 173–195.
 53. **Zou, X., Chen, Y., Liu, M., & Kang, L. (2008)**. A new evolutionary algorithm for solving many-objective optimization problems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, Vol. 38, No. 5, pp. 1402–1412.
- Nancy Pérez** es Ingeniero Industrial por la Universidad Autónoma de Ciudad Juárez y acaba de concluir sus estudios como Maestro en Ingeniería Industrial con especialidad en Calidad. Sus principales áreas de interés son la optimización multiobjetivo enfocada en su aplicación en el campo industrial, así como la optimización numérica en general.
- Oliver Cuate** es Ingeniero en Matemáticas por la Escuela Superior de Física y Matemáticas del IPN en 2013 y Maestro Ciencias de la Computación por CINVESTAV-IPN en 2015. Actualmente es estudiante de doctorado del departamento de computación del CINVESTAV-IPN. Sus áreas de interés son la optimización multiobjetivo, así como la optimización numérica en general.
- Oliver Schütze** actualmente es profesor de tiempo completo en el Departamento de Computación del CINVESTAV-IPN. Es Doctor en Matemáticas Aplicadas por la Universidad Paderborn, Alemania en 2004; Licenciado en Matemáticas por la Universidad Bayreuth, Alemania en 1999. Sus áreas de investigación son la optimización multiobjetivo (con técnicas de programación matemática y con metaheurísticas), así como la optimización numérica en general. Es miembro del Sistema Nacional de Investigadores (Nivel 2).
- Alejandro Alvarado** es profesor de tiempo completo en el Departamento de Ingeniería Industrial y Manufactura de la Universidad Autónoma de Ciudad Juárez. Recibió sus grados de Doctor y Maestro en Ciencias en Ingeniería Industrial de la Universidad Estatal de Nuevo México (EUA), y su título de Licenciatura en Ingeniería Electrónica por el Instituto Tecnológico de Chihuahua. Sus

principales áreas de investigación se enfocan en la optimización multi-objetivo (con técnicas de programación matemática y con metaheurísticas), así como la optimización numérica en general y la toma de decisiones mediante el uso de modelación matemática, aplicadas principalmente en el sector

industrial. Es miembro del Sistema Nacional de Investigadores (Nivel 1).

*Article received on 21/07/2016; accepted on 28/10/2016.
Corresponding author is Nancy Pérez.*