

# Algoritmo Evolutivo Multi-Objetivo para la Toma de Decisiones Interactiva en Optimización Dinámica

Cristóbal Barba-González

Lenguajes y Ciencias de la Computación  
Universidad de Málaga  
Málaga, España  
cbarba@lcc.uma.es

José M. García-Nieto

Lenguajes y Ciencias de la Computación  
Universidad de Málaga  
Málaga, España  
jnieto@lcc.uma.es

Ana B. Ruiz

Economía Aplicada (Matemáticas)  
Universidad de Málaga  
Málaga, España  
abruiz@uma.es

Antonio J. Nebro

Lenguajes y Ciencias de la Computación  
Universidad de Málaga  
Málaga, España  
antonio@lcc.uma.es

Mariano Luque

Economía Aplicada (Matemáticas)  
Universidad de Málaga  
Málaga, España  
mluque@uma.es

José F. Aldana-Montes

Lenguajes y Ciencias de la Computación  
Universidad de Málaga  
Málaga, España  
jfam@lcc.uma.es

**Resumen**—Debido al creciente interés en el análisis de datos en streaming en entornos Big Data para la toma de decisiones, cada vez es más común la aparición de problemas de optimización dinámica que involucran dos o más objetivos en conflicto. Sin embargo, los enfoques que combinan optimización dinámica multi-objetivo con la articulación de preferencias para la toma de decisiones son todavía escasos. En este artículo, proponemos un nuevo algoritmo de optimización dinámica multi-objetivo llamado InDM2, que permite incorporar preferencias del experto (humano) de cara a la toma de decisiones para guiar el proceso de búsqueda. Con InDM2, el decisor no solo puede expresar sus preferencias mediante uno o más puntos de referencia (que definen la región de interés deseada), sino que estos puntos también se pueden modificar de manera interactiva. La propuesta incorpora métodos para mostrar gráficamente las diferentes aproximaciones de la región de interés obtenidas durante el proceso de optimización. El decisor puede así inspeccionar y cambiar, en tiempo de optimización, la región de interés de acuerdo con la información mostrada. Las principales características de InDM2 son descritas y se analiza su funcionamiento mediante casos de uso académicos.

**Index Terms**—Optimización Multi-Objetivo Dinámica, Toma de Decisiones, Selección de Preferencias, Algoritmos Interactivos

## I. INTRODUCCIÓN

En la actualidad, aunque la mayoría de los esfuerzos se centran en abordar problemas de optimización multi-objetivo (MOP) estáticos, existe un creciente interés en aquellos escenarios en los que el problema cambia de alguna manera con el tiempo, es decir, MOP dinámicos o DMOPs [1].

Recientemente, han ido apareciendo varias propuestas de metaheurísticas para la resolución de DMOPs [2]–[5]. Al tratar con un DMOP, siempre que existan cambios en el entorno que afecten a las soluciones del problema (es decir, en la población o en el conjunto de soluciones en el frente de Pareto), el algoritmo de optimización debe reaccionar para adaptar la búsqueda a las nuevas características del problema [6]. Esto significa que las metaheurísticas de optimización dinámica

multi-objetivo deben poder detectar cuándo cambia el problema y aplicar una estrategia para adaptarse a los cambios.

Sin embargo, analizar soluciones preferidas en el frente de Pareto aproximado y compararlas al mismo tiempo puede ser una tarea difícil en el proceso de toma de decisiones, especialmente en presencia de muchos objetivos. De hecho, puede ser computacionalmente costoso generar una gran cantidad de soluciones que aproximen todo el frente de Pareto, particularmente cuando se tratan problemas reales de la industria. Incluso, puede suponer un esfuerzo innecesario si el decisor está únicamente interesado en un subconjunto de soluciones ubicadas en un región particular.

Una forma de aliviar estos problemas es incorporar información sobre las preferencias del decisor en la metaheurística multi-objetivo, para enfocar progresivamente el proceso de búsqueda sobre aquel subconjunto de soluciones que corresponden a estas preferencias, conocido como la *región de interés*. Por lo tanto, en lugar de aproximar todo el frente de Pareto, la idea es aproximar sólo la región de interés.

Mientras que el manejo de preferencias ha sido ampliamente estudiado en su aplicación para MOPs estáticos en el campo de la toma de decisiones multi-criterio (*Multiple Criteria Decision Making - MCDM*) [7]–[9], apenas se ha explorado en la optimización dinámica de múltiples objetivos. En ese sentido, en un contexto donde el frente de Pareto puede cambiar con el tiempo, la especificación de preferencias que determinan la región de interés puede ser difícil debido al hecho de que el problema en sí cambia constantemente. Por lo tanto, una metaheurística de optimización dinámica multi-objetivo que incorpore preferencias, debería prestar atención especial a estas dos características simultáneamente: (1) visualización, para proporcionar al decisor una imagen gráfica de las aproximaciones de la región o regiones de interés de los frentes de Pareto que se encuentran durante el proceso de optimización; y (2) interactividad, para permitir que el decisor pueda indicar nueva información si quiere redefinir sus preferencias (es decir,

indicar una nueva región de interés) en base a su experiencia previa y al conocimiento que progresivamente adquiere sobre el problema al interactuar con el proceso de optimización.

Con esta motivación, proponemos en este artículo InDM2 (*Interactive Dynamic Multi-Objective Decision Making*), una nueva propuesta algorítmica que cumple con estos requisitos. InDM2 está diseñado para actuar como framework algorítmico bajo el que se puede incorporar cualquier metaheurística de optimización multi-objetivo basada en puntos de referencia, para la articulación de preferencias, delegando así en él el proceso de resolución del DMOP. Un punto de referencia está formado por valores de aspiración que el decisor desea alcanzar para cada una de las funciones objetivo.

En particular, hemos considerado una versión dinámica de dos algoritmos evolutivos basados en puntos de referencia: WASF-GA [10] y R-NSGA-II [11]. El propósito de WASF-GA es aproximar la región de interés determinada por el punto de referencia mediante el uso de una función escalarizada de logro [12] y un conjunto de vectores de pesos. Para ello, se clasifican los individuos en fronteras en base a los valores que alcanzan para dicha función. El algoritmo R-NSGA-II considera como información preferencial uno o varios puntos de referencia y modifica la distancia crowding y el operador niching de NSGA-II [13] para enfatizar las soluciones cercanas al/los punto(s) de referencia dado(s) por el decisor.

InDM2 incorpora un mecanismo que permite al decisor (si así lo desea) actualizar interactivamente el punto de referencia al resolver un DMOP. Además, ofrece un mecanismo para visualizar “sobre la marcha” las aproximaciones de la región de interés que se generan a lo largo del proceso de solución. Además, InDM2 se ha implementado con jMetalSP [14], por lo que todas las características de esta librería algorítmica están incorporadas en la propuesta, incluida la posibilidad de utilizar tecnologías Big Data, como Apache Spark. Esto permite, por ejemplo, procesar en streaming los datos almacenados en HDFS (el sistema de archivos Hadoop [15]), además de usar toda la potencia de los clústeres Spark.

Así, las principales contribuciones de este artículo son:

- Se presenta InDM2, un novedoso algoritmo que combina optimización dinámica multi-objetivo, toma de decisiones multi-criterio e interactividad. Una característica clave en InDM2 es el mecanismo para visualizar, en tiempo de optimización, las aproximaciones de la región de interés que se generan, junto con la posibilidad de indicar puntos de referencia para guiar estas aproximaciones.
- La propuesta se ha implementado dentro del framework jMetalSP, por lo que se puede utilizar para optimización en contextos Big Data, aprovechando las características del procesamiento de datos en streaming distribuido de Apache Spark. Esta implementación está disponible como código abierto en repositorio GitHub de jMetalSP<sup>1</sup>.
- Se presentan dos casos de estudio para ilustrar el funcionamiento de InDM2 en la resolución de problemas dinámicos del benchmark FDA para DMOPs [6]. En con-

creto, se utilizan las metaheurísticas interactivas WASF-GA y R-NSGA-II como optimizador base para los dos ejemplos sobre el problema FDA2.

El resto del artículo se organiza de la siguiente forma. La Sección II está dedicada a describir conceptos previos necesarios para la explicación de la propuesta. En la Sección III, se describen la estructura y los principales componentes de InDM2. La Sección IV explica los experimentos realizados a modo de caso de uso. Finalmente, la Sección V recoge las principales conclusiones y trabajo futuro.

## II. CONCEPTOS PREVIOS

Una forma común de expresar preferencias consiste en indicar valores deseables para las funciones objetivo, que constituyen las componentes de lo que se conoce como punto de referencia. Este concepto se introdujo en [12] para MOP estáticos, aunque se puede adaptar fácilmente a DMOP.

Para un DMOP con  $k$  funciones objetivo, en un instante de tiempo  $t$ , un *punto de referencia* viene dado por  $\mathbf{q}_t = (q_{t,1}, \dots, q_{t,k})^T$ , donde  $q_{t,i}$  es un nivel de aspiración para la función objetivo  $f_i(\cdot, t)$  proporcionada por el decisor ( $i = 1, \dots, k$ ). Por lo general, se dice que  $\mathbf{q}_t$  es *alcanzable* si existe, al menos, una solución factible del problema  $t$  cuyo vector objetivo iguala o domina  $\mathbf{q}_t$ . De lo contrario, se dice que el punto de referencia es *inalcanzable*, es decir, no todos los niveles de aspiración se pueden alcanzar o dominar simultáneamente.

Para un punto de referencia  $\mathbf{q}_t$ , se puede minimizar la *función escalarizada de logro* propuesta en [12] sobre el conjunto factible, para encontrar la solución óptima de Pareto que mejor satisfaga las expectativas del decisor en el instante  $t$ . Dado un vector de pesos en el instante  $t$ ,  $\mu_t = (\mu_{t,1}, \dots, \mu_{t,k})^T$  con  $\mu_{t,i} > 0$  ( $i = 1, \dots, k$ ), y un valor real pequeño  $\rho \geq 0$ , la función de logro propuesta en [12] viene dada por:

$$s(\mathbf{q}_t, \mathbf{f}(\mathbf{x}, t), \mu_t) = \max_{i=1, \dots, k} \{ \mu_{t,i} (f_i(\mathbf{x}, t) - q_{t,i}) \} + \rho \sum_{i=1}^k \mu_{t,i} (f_i(\mathbf{x}, t) - q_{t,i}).$$

## III. PROPUESTA ALGORÍTMICA: INDM2

El componente principal de InDM2 es una metaheurística multi-objetivo dinámica basada en preferencias, que actúa como motor de optimización. Una descripción más detallada de la implementación se puede encontrar en [16]. Para este trabajo, se han implementado nuevas versiones dinámicas de los algoritmos WASF-GA y R-NSGA-II de la siguiente manera.

La arquitectura orientada a objetos de jMetalSP permite reutilizar la plantilla de algoritmos evolutivos proporcionada por jMetal<sup>2</sup>. Esta plantilla tiene un método `run()` que codifica el pseudo-código de un algoritmo evolutivo genérico. Las implementaciones de las versiones dinámicas de WASF-GA y R-NSGA-II siguen esta plantilla, por lo

<sup>1</sup><https://github.com/jMetal/jMetalSP>

<sup>2</sup><https://jmetal.github.io/jMetal/>

que el desarrollo de InDM2 sobre WASF-GA y R-NSGA-II requiere simplemente la redefinición de dos métodos. En primer lugar, se debe adaptar el comportamiento del método `isStoppingConditionReached()`, ya que en lugar de simplemente terminar el algoritmo, este método debe hacer que la aproximación de la región de interés esté disponible para los métodos consumidores de datos de InDM2. De esta forma, el algoritmo de optimización subyacente (basado en WASF-GA o R-NSGA-II) pueda comenzar de nuevo teniendo en cuenta esta información. En segundo lugar, al final de cada iteración, el método `updateProgress()` incrementa un contador de evaluación.

Para permitir que el decisor pueda actualizar el(los) punto(s) de referencia “sobre la marcha”, InDM2 lee el(los) nuevo(s) punto(s) de referencia desde teclado (aunque podrían usarse otros métodos más sofisticados). Este evento se detecta mediante el método `updateProgress()`. Para ayudar al decisor a dar un nuevo punto de referencia de acuerdo con sus preferencias y teniendo en cuenta los resultados obtenidos, se ha incluido un consumidor de datos gráfico. Por lo tanto, las aproximaciones de la región de interés generadas por InDM2 se muestran cuando se producen, por lo que el decisor puede visualizarlas y cambiar los puntos de referencia como desee.

Por defecto, InDM2 siempre está funcionando. Esto significa que, si el DMOP no cambia durante un período de tiempo, el algoritmo lo resolverá una y otra vez, utilizando exactamente la misma configuración del problema. Como consecuencia, si el algoritmo converge la mayoría de las veces, la misma aproximación de la región de interés (o una muy similar, dado que estamos usando enfoques estocásticos) se mostrará constantemente. Esto puede ser confuso para el decisor, ya no percibirá cambios evidentes por un período de tiempo. Para evitar esta situación, el consumidor de datos gráficos solo muestra una nueva aproximación en caso de que ésta haya cambiado significativamente, en comparación con la anterior mostrada. Para verificar esto, se calcula internamente el indicador distancia generacional invertida ( $I_{IGD}$ ) [17] desde la aproximación de la región de interés actual, hasta la anterior. La nueva aproximación se mostrará únicamente si el valor de este indicador es inferior a un umbral  $\tau$ .

Todos los DMOP disponibles en jMetalSP tienen un método para averiguar si el problema ha cambiado o no. Por lo tanto, InDM2 solo tiene que llamar a este método al final de cada iteración (es decir, en el método `updateProgress()`), para saber si el parámetro del problema ha cambiado.

Una vez que se ha detectado un cambio en el problema y/o punto(s) de referencia, un aspecto clave en InDM2 es la estrategia de reinicio. Hemos adoptado un enfoque flexible que consiste en definir el reinicio de acuerdo con otras dos subestrategias: una para eliminar soluciones de la población y otra para llenar la población con nuevas soluciones. Para llenar la población, dado un valor  $N' > 0$ , la estrategia seguida es crear al azar  $N'$  nuevas soluciones. Para eliminar las soluciones, se pueden configurar las siguientes estrategias:

- Eliminar las primeras  $N'$  soluciones.
- Eliminar  $N'$  soluciones aleatorias.

- Eliminar las peores  $N'$  soluciones de acuerdo a la distancia crowding.
- Eliminar las peores  $N'$  soluciones de acuerdo a su contribución en hypervolumen con respecto a la última aproximación de la región de interés.

El pseudo-código del InDM2 se muestra en el Algoritmo 1. Después de la fase de inicialización (líneas 1-14), el algoritmo inicia el ciclo infinito (línea 15) que lleva a cabo múltiples rondas consecutivas de optimización dinámica y consumo de datos resultado (visualización). Cada ronda de optimización (líneas 16-25) implica un número máximo de iteraciones ( $G_{max}$ ) en las que se ejecuta el algoritmo evolutivo basado en preferencias (es decir, WASF-GA o R-NSGA-II, línea 17), y se invocan los procedimientos de reinicio cada vez que se detecta un cambio en el punto de referencia (línea 18) o en el estado del problema (línea 20). Al final de cada iteración, la aproximación de la región de interés encontrada se envía a los consumidores de datos (línea 25) y el ciclo comienza de nuevo.

---

#### Algorithm 1 Pseudo-código InDM2

---

```

1:  $N$  ; // Tamaño de población
2:  $N'$  ; // Número de soluciones a reemplazar
3:  $G_{max}$  ; // Número máximo de generaciones
4:  $c, m$  ; // Operadores genéticos (cruce y mutación)
5:  $t \leftarrow 0$  ; // Contador de generación
6:  $A_t$  ; // Estado del problema de optimización
7:  $\mathbf{q}_t$  ; // Punto(s) de referencia inicial(es)
8:  $P$  ; // Algoritmo de optimización basado en preferencias
9:  $\varphi_q$  ; // Estrategia de reinicio ante cambios del punto de referencia
10:  $\varphi_p$  ; // Estrategia de reinicio ante cambios del problema
11:  $M \leftarrow \{\text{WASF-GA, R-NSGA-II}\}$  ; // Optimizadores base
12:  $P_t \leftarrow \text{initializePopulation}(N)$  ;
13:  $evaluate(P_t, A_t)$  ;
14:  $E_t \leftarrow \text{initializeParetoSet}(P_t)$  ;
15: while true do
16:   while  $t < G_{max}$  do
17:      $(P_{t+1}, E_{t+1}) \leftarrow \text{compute}(M, \mathbf{q}_t, c, m, P_t, A_t)$  ;
18:     if  $\mathbf{q}_{t+1} \neq \mathbf{q}_t$  then
19:        $P'_{t+1} \leftarrow \text{restart}(P_{t+1}, \varphi_q, \mathbf{q}_{t+1}, N')$  ;
20:     else if  $A_{t+1} \neq A_t$  then
21:        $P'_{t+1} \leftarrow \text{restart}(P_{t+1}, \varphi_p, A_{t+1}, N')$  ;
22:     end if
23:      $t \leftarrow t + 1$  ;
24:   end while
25:    $C : (E_{t+1}, \tau)$  ; // Notifica aproximación a consumidores
26:    $t \leftarrow 0$  ;
27: end while

```

---

## IV. EXPERIMENTOS

El objetivo de este estudio no es tanto la realización de comparativas exhaustivas de rendimiento algorítmico, sino el de ilustrar cómo funciona InDM2 en la práctica, cuando interactúa con un decisor, para así mostrar los beneficios potenciales de la propuesta desde el punto de vista de la toma de decisiones.

En este sentido, se ha probado InDM2 con el problema FD2, introduciendo manualmente las preferencias (es decir, los puntos de referencia) durante el tiempo de ejecución. La

Tabla I: Configuración de InDM2.

Parámetros	
Tamaño de población	50
Probabilidad de cruce	0.9
Probabilidad de mutación	$1/L$ ( $L$ = número de variables)
Número máximo de evaluaciones	25,000
Reinicio por punto de referencia	Eliminar 100 % de las soluciones Generar 100 % de soluciones aleatoriamente
Reinicio por cambio de problema	Eliminar 50 % de soluciones (hypervolumen) Generar 50 % de soluciones aleatoriamente
Consumidores de datos	
Visualizador gráfico	Muestra frentes y puntos de referencia
Front writer	Carga los frentes generados en ficheros
Streaming runtime	
Mecanismo	Java threads

idea es observar cómo InDM2 puede adaptar el proceso de optimización a los cambios (tanto en la configuración del problema como en las preferencias), mostrando así el efecto de estos cambios en las aproximaciones de los frentes de Pareto obtenidas. Nos referimos a las dos configuraciones de InDM2 para los optimizadores base WASF-GA y R-NSGA-II como InDM2<sub>W</sub> y InDM2<sub>R</sub>, respectivamente.

El problema FD2 es un DMOPs de representación continua perteneciente al benchmark FDA [6]. Cada problema de esta familia tiene dependencia con el tiempo  $t$ , que se define mediante la ecuación  $t = \frac{1}{n_t} \lfloor \frac{\sigma}{\sigma_T} \rfloor$ , en la que  $\sigma$  es un contador de generaciones,  $\sigma_T$  es el número de generaciones en las que  $t$  permanece fijo y  $n_t$  es la cantidad de pasos distintos en  $t$ . Para un número real  $a$ ,  $\lfloor a \rfloor$  denota el entero más grande no mayor a  $a$ . En [6], los autores recomiendan establecer  $\sigma_T = 5$  y  $n_t = 10$ .

La formulación de FDA2 se puede encontrar en [6], por lo que siguiendo lo recomendado en este trabajo, se han utilizado soluciones de 31 variables con  $x_I = x_1$ ,  $|x_{II}| = |x_{III}| = 15$ . Este problema se ha seleccionado debido a que las aproximaciones de frentes de Pareto y sus conjuntos de Pareto cambian en el tiempo (tipo II [6]). Por lo tanto, los cambios dinámicos en las aproximaciones causadas por cambios en el punto de referencia se pueden visualizar fácilmente.

La Tabla I muestra la configuración de parámetros utilizada para InDM2. La condición de parada utilizada para informar de los resultados a los procesos consumidores es alcanzar un máximo de 25,000 evaluaciones. Los operadores de variación son SBX como operador de cruce y el de mutación polinómica. El operador de selección es torneo binario. Para la estrategia de reinicio al detectar un cambio en el problema, en primer lugar, se elimina el 50 % de las soluciones de acuerdo con la contribución de hipervolumen; en segundo lugar, se llena la población con nuevas soluciones creadas aleatoriamente. La estrategia de reinicio cuando se proporciona un nuevo punto de referencia es similar, pero en este caso se eliminan todas las soluciones de la población (100 %).

Para generar los eventos temporales que conducirán a cambios en el problema FDA2, hemos utilizado una fuente simple de transmisión de datos que cuenta cada segundo, lo que implica que, con un valor predeterminado de  $\sigma_T$ , el problema se actualiza con una frecuencia de 5 segundos. Además, hemos utilizado dos consumidores de datos, uno para

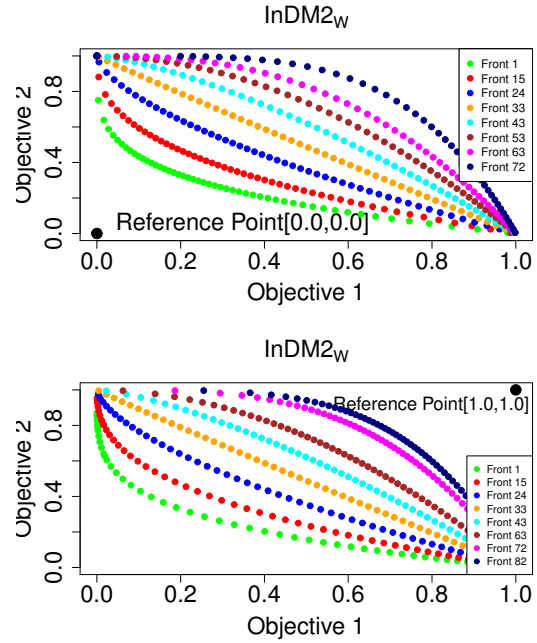


Figura 1: Aproximaciones de región de interés encontradas por InDM2<sub>W</sub> para el problema FD2, utilizando los puntos de referencia (0.0, 0.0) (arriba) y (1.0, 1.0) (abajo).

mostrar los frentes de Pareto aproximados que se generan y otra para guardar estas aproximaciones en archivos. En este estudio, solo se considera una fuente de datos, es decir, no hay ninguna ventaja de utilizar Apache Spark, por lo que hemos configurado el tiempo de ejecución en hebras de manera predeterminada.

El problema de optimización dinámica FDA2 se define con dos objetivos y frentes de Pareto que cambian de forma convexa a cóncava y vice-versa, con conjuntos de Pareto que también cambian con el tiempo.

Para tener una idea de los posibles valores de aspiración (deseables) que cada objetivo puede tener durante el proceso de búsqueda, se ha ejecutado inicialmente InDM2<sub>W</sub> utilizando dos puntos de referencia: el punto ideal (0.0, 0.0) (inalcanzable) y el punto nadir (1.0, 1.0) (alcanzable). Las aproximaciones producidas por InDM2<sub>W</sub> para las regiones de interés de ambos puntos de referencia se muestran en la Figura 1. En esta figura, las leyendas “Front<sub>*i*</sub>” marcan las aproximaciones de la región de interés que se muestran. Cada una de estas leyendas indica implícitamente en qué iteración  $i$  se detectó una mejora significativa en la aproximación generada, en comparación con la anterior mostrada (como se explicó en la Sección III, una nueva aproximación se muestra gráficamente solo en caso de que se haya detectado una mejora significativa con respecto al indicador  $I_{IGD}$ ).

En la Figura 1, podemos observar que cuando la región de interés aproximada es convexa, la distribución de las soluciones en las aproximaciones encontradas es ligeramente más densa en las regiones centrales que en las extremas,

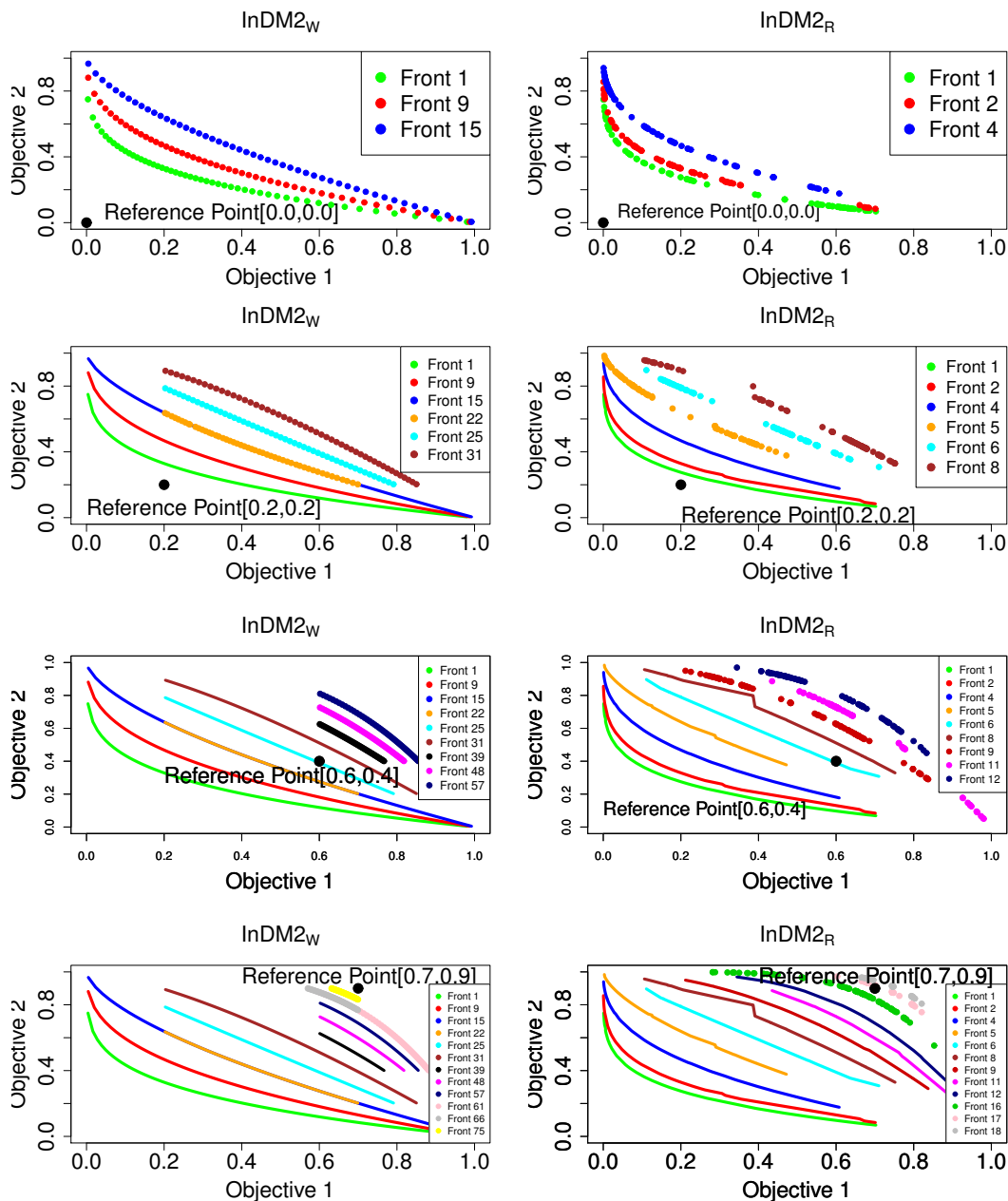


Figura 2: Aproximaciones de la región de interés encontradas por  $\text{InDM2}_W$  (izquierda) e  $\text{InDM2}_R$  (derecha) para el problema FDA2, utilizando diferentes puntos de referencia.

extendiéndose de manera uniforme cuando la región aproximada tiende a ser más lineal. Esto viene determinado por la distribución de las direcciones de proyección (es decir, vectores de peso) utilizadas internamente en WASF-GA, lo que influye en la distribución de las soluciones encontradas. Para ambos puntos de referencia,  $\text{InDM2}_W$  ha podido aproximar la región de interés (que, como se puede ver, es el frente de Pareto completo en ambos casos) al adaptar el proceso de búsqueda a los cambios en el problema.

A continuación, se realizaron pruebas con  $\text{InDM2}_W$  e  $\text{InDM2}_R$  para FDA2, pero en este caso, se simuló un escenario

realista en el que el decisor modifica sobre la marcha el punto de referencia para indicar la región deseada, varias veces durante el proceso de optimización. La Figura 2 muestra las aproximaciones generadas para los puntos de referencia indicados a lo largo del proceso de optimización de  $\text{InDM2}$ , para los dos algoritmos base:  $\text{InDM2}_W$  y  $\text{InDM2}_R$ .

En el proceso interactivo de optimización seguido por  $\text{InDM2}_W$ , suponiendo que los rangos del frente de Pareto para problema FDA2 son desconocidos, el decisor seleccionó inicialmente  $(0.0, 0.0)$  como primer punto de referencia. Después de observar las tres primeras aproximaciones devueltas por

InDM2<sub>W</sub> en las iteraciones 5, 9 y 15 (véase la gráfica superior izquierda en la Figura 2), el decisor decidió dar un nuevo punto de referencia para reducir la región de interés aproximada y establecerlo en (0.2, 0.2). Podemos observar las nuevas aproximaciones generadas en el segundo gráfico a la izquierda en la Figura 2. En este gráfico, las primeras aproximaciones generadas para el punto de referencia inicial (0.0, 0.0) se muestran con líneas continuas, para poder distinguirlas de las nuevas aproximaciones encontradas para el nuevo punto de referencia (0.2, 0.2), que se muestran con líneas punteadas. Más tarde, el decisor volvió a ajustar la región de interés cambiando el punto de referencia dos veces, indicando (0.6, 0.4) como tercer punto de referencia y (0.7, 0.9) como el cuarto. Respectivamente, los resultados obtenidos para estos dos nuevos puntos de referencia se muestran en las gráficas tercera y cuarta a la izquierda en la misma figura.

Respecto a los frentes obtenidos por InDM2<sub>R</sub> en las gráficas de la derecha en la Figura 2, se puede observar cómo la aproximación de las regiones de interés producidas por esta variante tienen peor diversidad que las generadas por InDM2<sub>W</sub>. Nótese que las soluciones generadas por InDM2<sub>W</sub> siempre pertenecen a la región de interés delimitada por los puntos de referencia, mientras que algunas de las encontradas por InDM2<sub>R</sub> también aproximan zonas fuera de esta región. Esto es consecuencia de las capacidades de búsqueda (convergencia y diversidad con respecto a la región de interés) de WASFGA y R-NSGA-II, lo que evidencia el impacto que puede tener el algoritmo evolutivo basado en preferencias utilizado como base en InDM2.

Según la Figura 2, podemos ver que tres de los puntos de referencia dados en el proceso interactivo de optimización son inalcanzables: (0.0, 0.0), (0.2, 0.2) y (0.6, 0.4); y uno de ellos es alcanzable (0.7, 0.9). Cabe señalar que, en la práctica, la alcanzabilidad de los puntos de referencia no puede conocerse de antemano, y solo se puede saber si son alcanzables o no cuando el algoritmo genera soluciones y se verifica su relación de dominancia con respecto al punto de referencia utilizado.

## V. CONCLUSIONES

En este trabajo se ha descrito InDM2, un nuevo framework algorítmico de optimización dinámica que permite incorporar cualquier metaheurística multi-objetivo basada en puntos de referencia para la articulación de preferencias. Se ha mostrado su funcionamiento mediante la resolución de un problema de optimización dinámica multi-objetivo de benchmarking (FD2). A partir de una serie de visualizaciones, se ha demostrado cómo InDM2 es capaz de generar aproximaciones ajustadas a las preferencias del decisor en tiempo real, tanto si cambia la configuración del problema como si se indican nuevos puntos de referencia (es decir, cambia la región de interés).

Una característica importante de InDM2 viene heredada de su implementación en el framework de jMetalSP, por lo que dispone de procesamiento en streaming mediante Spark de manera transparente. Esto nos permite incorporar múltiples fuentes de datos mediante el uso de proveedores, como sockets, archivos en un directorio, Kafka, Flume o Kinesis. Otra

consecuencia del uso de Spark es la ventaja de trabajar con tecnologías Big Data, lo que nos permite desplegar InDM2 en clusters Hadoop y acceder a datos en HDFS.

Como trabajo futuro, se pretende estudiar el rendimiento de InDM2 en problemas complejos del mundo real, usando preferencias y cambiándolas interactivamente por un decisor experto en el dominio de aplicación.

## AGRADECIMIENTOS

Este trabajo ha sido financiado por los proyectos TIN2017-86049-R, TIN2014-58304-R, ECO2017-88883-R (MECD) y P12-TIC-1519 (PAIDI). J. M. García-Nieto y A. B. Ruiz disfrutaron una beca Post-Doctoral "Captación de Talento para la Investigación", Universidad de Málaga. C. Barba-González dispone de una beca pre-doctoral BES-2015-072209.

## REFERENCIAS

- [1] C. Raquel and X. Yao, "Dynamic multi-objective optimization: A survey of the state-of-the-art," in *Evolutionary Computation for Dynamic Optimization Problems (2013)*, S. Yang and X. Yao, Eds. Springer, 2013, pp. 85–106.
- [2] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 65–82, 2017.
- [3] X. Chen, D. Zhang, and X. Zeng, "A stable matching-based selection and memory enhanced MOEA/D for evolutionary dynamic multiobjective optimization," in *Int. Conf. on Tools with AI*, 2015, pp. 478–485.
- [4] Y. Jin, C. Yang, J. Ding, and T. Chai, "Reference point based prediction for evolutionary dynamic multiobjective optimization," in *Proc. of the IEEE Cong. on Evol. Comp.*, 2016, pp. 3769–3776.
- [5] A. Muruganatham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via kalman filter prediction," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2862–2873, 2016.
- [6] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, 2004.
- [7] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic Publishers, 1999.
- [8] A. Jaszkiewicz and J. Branke, "Interactive multiobjective evolutionary algorithms," in *Multiobjective Optimization, Interactive and Evolutionary Approaches*, ser. LNCS, J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds., vol. 5252. Springer, 2008, pp. 179–193.
- [9] M. Helbig, K. Deb, and A. Engelbrecht, "Key challenges and future directions of dynamic multi-objective optimisation," in *Proc. of the IEEE Cong. on Evol. Comp.*, 2016, pp. 1256–1261.
- [10] A. B. Ruiz, R. Saborido, and M. Luque, "A preference-based evolutionary algorithm for multiobjective optimization: The weighting achievement scalarizing function genetic algorithm," *Journal of Global Optimization*, vol. 62, no. 1, pp. 101–129, 2015.
- [11] K. Deb, J. Sundar, B. Ubay, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithm," *Int. Jour. of Comp. Intel. Res.*, vol. 2, no. 6, pp. 273–286, 2006.
- [12] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization," in *Multiple Criteria Decision Making, Theory and Applications*, G. Fandel and T. Gal, Eds. Springer, 1980, pp. 468–486.
- [13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE TEVC*, vol. 6, no. 2, pp. 182–197, 2002.
- [14] C. Barba-Gonzalez, J. M. Garcia-Nieto, A. J. Nebro, J. A. Cordero, J. J. Durillo, I. Navas-Delgado, and J. F. Aldana-Montes, "jMetalSP: A framework for dynamic multi-objective big data optimization," *Applied Soft Computing*, vol. to appear, 2017.
- [15] T. White, *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 2009.
- [16] A. J. Nebro, A. B. Ruiz, C. Barba-González, J. García-Nieto, M. Luque, and J. F. Aldana-Montes, "InDM2: Interactive dynamic multi-objective decision making using evolutionary algorithms," *Swarm and Evolutionary Computation*, vol. 40, pp. 184 – 195, 2018.
- [17] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, *Modified Distance Calculation in Generational Distance and Inverted Generational Distance*. Springer International Publishing, 2015, pp. 110–125.