

# Análisis y diseño de operadores de cruce basados en el cruce binario simulado

J. Chacón, C. Segura, A. Hernández Aguirre

Centro de Investigación en Matemáticas,  
Guanajuato, México

{joel.chacon, carlos.segura, artha}@cimat.mx

**Resumen.** El operador de Cruce Binario Simulado (SBX) es uno de los operadores más utilizados para codificaciones con números reales. En este artículo se presenta un análisis novedoso del operador SBX, que involucra aspectos relacionados con la diversidad, similitud y reflexiones que se dan entre los padres y los hijos. En base a este análisis, se propone un nuevo operador denominado el operador Dinámico sin Reflexiones basado en el Cruce Binario Simulado (NRD-SBX). NRD-SBX aporta un mayor grado de intensificación y es capaz de adaptar el grado de exploración en el proceso de optimización. Esto se consigue modificando el operador SBX mediante la implementación de un modelo dinámico para el control de diversidad y las reflexiones. La validación experimental, la cual considera los tests WFG y varios Algoritmos Evolutivos Multi-Objetivo muy populares, muestra de forma clara los beneficios aportados por el nuevo operador.

**Palabras clave:** Optimización multi-objetivo, algoritmos evolutivos, operadores de cruce.

## Analysis and Design of Crossover Operators based on the Simulated Binary Crossover

**Abstract.** The Simulated Binary Crossover (SBX) is a popular operator used in continuous domains. In this work, an analysis of the SBX is presented, which involves the study of several diversity related issues, similarities among solutions and reflections that appear in the reproduction phase. Taking the analysis into account, the operator Non-Reflection Dynamic Simulated Binary Crossover (NRD-SBX) is proposed. This new operator extends SBX by implementing a dynamic model to control the diversity and reflections. NRD-SBX provides a higher intensification level than SBX and is able to adapt the exploration degree through the optimization process. The experimental validation is carried out with the WFG benchmark and several multi-objective EAs. The NRD-SBX operator shows a better performance than the original SBX.

**Keywords:** Optimization multi-objective, evolutionary algorithms, crossover operators.

## 1. Introducción

En las últimas décadas, la implementación de metaheurísticas para resolver problemas de optimización complejos ha ganado una gran popularidad [18]. Por ello, se han propuesto diversos tipos de metaheurísticas, destacando entre las más utilizadas los Algoritmos Genéticos (Genetic Algorithm - GA), Evolución Diferencial (Differential Evolution - DE), Estrategias Evolutivas (Evolution Strategies - ES), y Optimización por Enjambres de Partículas (Particle Swarm Optimization - PSO). Dadas las similitudes que existen entre algunas de las metaheurísticas poblacionales más populares, se propuso el término Algoritmo Evolutivo (Evolutionary Algorithm - EA) para agrupar a varias de ellas [16].

Debido al comportamiento prometedor de los EAs, estos han sido ampliados en múltiples formas. Por ejemplo, han sido usados para resolver Problemas de Optimización Multi-objetivo (Multi-objective Optimization Problems - MOPs). Un MOP continuo basado en minimización puede ser definido como se indica en la ecuación (1):

$$\begin{aligned} & \text{Minimizar } f_m(\mathbf{x}), \quad m = 1, 2, \dots, M; \\ & \text{Sujeto a } x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, n. \\ & \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

donde  $\mathbf{n}$  es la dimensión correspondiente al espacio de las variables,  $\Omega$  es el espacio factible,  $\mathbf{x}$  es un vector compuesto por  $\mathbf{n}$  variables de decisión  $x = (x_1, \dots, x_n) \in R^n$ , y  $x_i^{(L)}$  y  $x_i^{(U)}$  son los límites inferior y superior de cada variable. Cada solución es evaluada mediante el uso de la función  $F : \Omega \rightarrow R^m$ , la cual consiste de  $m$  funciones objetivo y  $R^m$  es conocido como el espacio objetivo. Dadas dos soluciones  $\mathbf{x}, \mathbf{y} \in \Omega$ ,  $\mathbf{x}$  domina a  $\mathbf{y}$ , denotado por  $\mathbf{x} \prec \mathbf{y}$ , si y solo si  $\forall m \in 1, 2, \dots, M : f_m(x_i) \leq f_m(y_i)$  y  $\exists m \in 1, 2, \dots, M : f_m(x_i) < f_m(y_i)$ . Una solución  $\mathbf{x}^* \in \Omega$  es conocida como solución óptima de Pareto si no existe otra solución  $\mathbf{x} \in \Omega$  que domine a  $\mathbf{x}^*$ . El conjunto de Pareto es el conjunto de todas las soluciones óptimas de Pareto y el frente de Pareto está formado por las imágenes del conjunto de Pareto. El propósito de los optimizadores multi-objetivo es, esencialmente, obtener un conjunto de soluciones bien distribuidas y cercanas a las soluciones del frente de Pareto.

Este trabajo se centra en el análisis y diseño de Algoritmos Evolutivos Multi-objetivo (Multi-objective Evolutionary Algorithm - MOEAs) basados en codificación real. Se ha mostrado que una de las claves para el diseño apropiado de los EAs consiste en inducir un balance apropiado entre exploración e intensificación [5]. En el caso de optimización mono-objetivo, se puede inducir

este balance controlando la diversidad en la fase de reemplazo [3]. En el caso de los MOEAs, dado que se manejan varios objetivos de forma simultánea, se mantiene un cierto grado de diversidad en la población de forma intrínseca. Sin embargo, esto no siempre es suficiente para asegurar los resultados deseados, pues los operadores genéticos deberían utilizar dicha diversidad para generar nuevas soluciones prometedoras, siendo este el tema en el que se centra este artículo.

En los EAs, los operadores de cruce tienen un efecto importante en la exploración e intensificación. Usualmente, los operadores de cruce tienden a promover mayor diversidad cuando se consideran padres distantes que cuando se consideran padres cercanos. De esta forma, los operadores de cruce son métodos adaptativos que dependen de la diversidad mantenida dentro de la población. En el caso de la codificación real, se han propuesto varios operadores de cruce [11], que pueden ser clasificados como centrados en los padres o en la media, en base a la región donde tienden a crear a los hijos.

El operador de Cruce Binario Simulado (Simulated Binary Crossover - SBX) [8] es probablemente uno de los más utilizados. En la versión inicial del SBX, se definió la forma de intercambiar la información para una variable. Tomando en cuenta las características de las distribuciones involucradas, el operador SBX se clasifica como un operador de cruce centrado en los padres. Sin embargo, en su versión inicial no se discutió como extender el mismo a casos de múltiples variables. Como resultado, se han propuesto diferentes implementaciones para hacer frente a problemas de múltiples variables. Las principales diferencias entre las diferentes propuestas tienen que ver con el número de variables que se heredan sin modificarse.

Además, en los casos de múltiples variables se dan algunos efectos como las reflexiones, que no se producen en los casos de una variable. Este tipo de transformaciones incrementan considerablemente la distancia entre los padres y los hijos cuando son utilizados en problemas de alta dimensionalidad. Dado que en comparación con los EAs mono-objetivo, los MOEAs internamente mantienen un grado más elevado de diversidad, los MOEAs que utilizan el operador SBX podrían crear individuos muy alejados de las regiones promisorias identificadas en fases anteriores de la búsqueda. Basados en el análisis del operador SBX presentado en este artículo, se propone el operador Dinámico sin Reflexiones basado en el Cruce Binario Simulado “Non-Reflection Dynamic Simulated Binary Crossover Operator” (NRD-SBX). El operador NRD-SBX adapta la probabilidad de heredar las variables sin modificarlas con el objetivo de inducir un mayor grado de exploración en las fases iniciales del proceso de optimización. Adicionalmente, el operador propuesto es modificado para no realizar las reflexiones propias de varias de las implementaciones más populares del SBX, con el objetivo de proveer un grado de intensificación adicional.

El resto de este artículo está organizado de la siguiente forma. La sección 2 provee una breve descripción de operadores de cruce y MOEAs más populares. La sección 3 analiza algunas de las dificultades que podrían estar presentes en la aplicación del operador SBX. La sección 4 describe el operador NRD-SBX. La

validación experimental se describe en la sección 5. Finalmente, las conclusiones y trabajos futuros se detallan en la sección 6.

## 2. Estado del arte

En esta sección se revisan algunos de los trabajos más importantes que están relacionados con la investigación que se presenta en este artículo, disponiendo de una sección dedicada a operadores de cruce y de otra dedicada a MOEAs.

### 2.1. Operadores de cruce

Los operadores de cruce se diseñan con el objetivo de generar soluciones candidatas utilizando la información de dos o más soluciones padres. Dado que existen múltiples operadores, se han propuesto varias taxonomías para clasificarlos. Una de las clasificaciones más populares distingue entre los operadores de cruce “Centrados en los Padres” y los “Centrados en la media” [14]. En los operadores centrados en los padres, las soluciones hijas son creadas alrededor de cada solución padre, mientras que en los operadores centrados en la media, las soluciones hijas son creadas alrededor de la media de las soluciones padres. Entre los operadores de cruce, el operador SBX es probablemente el más utilizado.

**Cruce Binario Simulado (SBX)** Uno de los operadores de cruce que se ha utilizado más comúnmente en los MOEAs es el SBX. El SBX se clasifica como un operador centrado en los padres y se basa en una función de distribución [11] que usa el parámetro  $\eta$  (llamado índice de distribución y que debe ser fijado por el usuario) para determinar la apertura de la distribución, y por tanto, su capacidad de exploración. Específicamente, un índice de distribución pequeño induce una mayor probabilidad de construir soluciones hijas alejadas de los padres, mientras que con un índice de distribución elevado, la probabilidad de crear soluciones hijas más similares a los padres se incrementa. Esto es ilustrado en la Figura 1, donde se muestran las funciones de densidad asociadas a los hijos para dos índices de distribución distintos. En la misma los círculos representan a los padres, y se puede apreciar que con  $\eta$  igual a 5, la probabilidad de crear soluciones más cercanas a los padres es mayor que con  $\eta = 2$ .

Algunas características interesantes del operador SBX son las siguientes:

- Los valores de las soluciones hijas son equidistantes de los padres.
- Existe una probabilidad no nula de generar una solución hija en cualquier parte en el espacio independientemente de donde se localicen los padres.
- La probabilidad de crear un par de soluciones hijas dentro del rango de las soluciones padres es idéntica a la probabilidad de crear dos soluciones hijas fuera de dicho rango.

Debido a la popularidad y al buen rendimiento del SBX, se han propuesto varias extensiones de este operador. En las primeras implementaciones del operador SBX [1], la forma de la distribución no estaba afectada por los límites

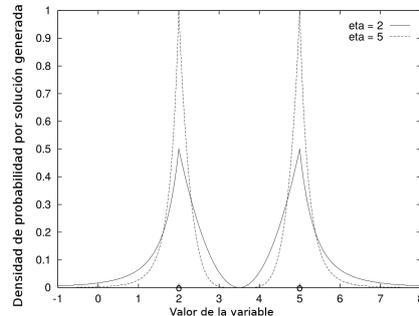


Fig. 1. Función de densidad del operador SBX con índices de distribución 2 y 5.

de cada variable, y todas las variables se modificaban en cada operación de cruce. Posteriormente, se propusieron variantes que sí consideran dichos límites así como una variante auto-adaptativa [8], que controla la diversidad mediante una variable conocida como el factor de dispersión. Dado que cambiar todas las variables de forma simultánea puede inducir un operador muy disruptivo, en las implementaciones actuales cada variable se cambia con una probabilidad igual a 0.5, mientras que en el resto de casos los valores se heredan sin alterar [7,15]. Además, por la forma en que se heredan las variables, pueden aparecer ciertas reflexiones que se analizan posteriormente. Por último, cabe destacar que no existen versiones del operador SBX en las que en lugar de usar dicha probabilidad fija en 0.5, esta sea cambiada a lo largo de la ejecución.

## 2.2. Algoritmos evolutivos multi-objetivo

Actualmente, existe una gran cantidad de MOEAs que se han diseñado siguiendo distintos principios. Para tener una mejor clasificación de los mismos, se han propuesto varias taxonomías [17]. Así, de acuerdo a los principios de diseño, los MOEAs pueden estar basados en la dominancia de Pareto, en indicadores y en descomposición [6]. Dado que todos ellos han reportado resultados de alta calidad, para la validación experimental del operador de cruce diseñado en este artículo, se ha seleccionado un método de cada tipo. Específicamente, la validación se ha llevado a cabo incluyendo el Algoritmo Genético de Ordenación de No Dominados II (Non-Dominated Sorting Genetic Algorithm - NSGA-II) [7], el MOEA Basado en Descomposición (MOEA Based on Decomposition -MOEA/D) [19], y Algoritmo de Optimización Multi-objetivo basado en la Selección con métrica S (S-Metric Selection Evolutionary Multi-objective Optimization Algorithm - SMS-EMOA) [2]. Los siguientes apartados describen brevemente cada uno de los paradigmas, así como los métodos seleccionados.

**MOEAs basados en dominancia - NSGA-II** Estos MOEAs se basan en considerar la noción de dominancia de Pareto para diseñar los distintos componentes

de los EAs. Dado que la relación de dominancia no promueve la preservación de diversidad en el espacio objetivo de forma intrínseca, es necesario utilizar técnicas auxiliares, como las basadas en nichos, muchedumbre o clústers, para promover la preservación de dicha diversidad. Probablemente, la técnica más popular de esta categoría es el NSGA-II [7]. Este algoritmo implementa un operador especial para la selección de padres e incorpora el uso de elitismo en la fase de reemplazo.

**MOEAs basados en indicadores - SMS-EMOA** Con el objetivo de comparar el rendimiento de los MOEAs, se han ideado varios indicadores de calidad que mapean las aproximaciones del conjunto de Pareto a valores reales. Dado que estos indicadores miden la calidad de las aproximaciones obtenidas por un determinado MOEA, se ha propuesto un paradigma basado en la aplicación de estos indicadores. En estos casos, en lugar de utilizar la dominancia de Pareto, se usan dichos indicadores en el MOEA para guiar el proceso de optimización. Entre los distintos indicadores, el hipervolumen es de lo más aceptados, habiendo sido usado en varios esquemas [13]. Una de las ventajas de los esquemas basados en indicadores es que estos usualmente toman en cuenta tanto la calidad de las soluciones como la diversidad en el espacio objetivo, por lo que no es necesario incluir mecanismos adicionales para preservar la diversidad. Entre los distintos MOEAs basados en indicadores, el SMS-EMOA [2] es de los más utilizados, probablemente debido a su simplicidad y superioridad sobre varios otros esquemas [10].

**MOEAs basados en descomposición - MOEA/D** Los MOEAs basados en descomposición transforman un problema multi-objetivo en un conjunto de problemas de optimización de un solo objetivo que son considerados de forma simultánea [4]. Esta transformación puede ser realizada de varias formas, por ejemplo, utilizando una suma lineal con pesos o con una función Tchebycheff basada en la agregación con pesos. Dado un conjunto de pesos que definen distintas funciones objetivo, el MOEA trata de encontrar buenas soluciones para cada una de ellas. Dentro de este grupo, el MOEA/D [19] es el algoritmo más popular. En este algoritmo se asocia un individuo de la población a cada subproblema. Entre sus principales características, además de que está basado en descomposición, cabe destacar que esto la hace con funciones basadas en peso, y que aplica restricciones de emparejamiento a través de la definición de un conjunto de vecindarios. En el MOEA/D se han propuesto distintos enfoques para la implementación de los pesos. Entre estos enfoques, el de Tchebycheff es uno de los más populares y es el que se usa en este artículo.

### **3. Análisis del operador SBX**

La implementación más utilizada del operador SBX se encuentra integrada en el NSGA-II publicada por Deb et al.[7]. En esta implementación cabe destacar dos puntos clave. El primero está relacionado con la similitud. En dicha

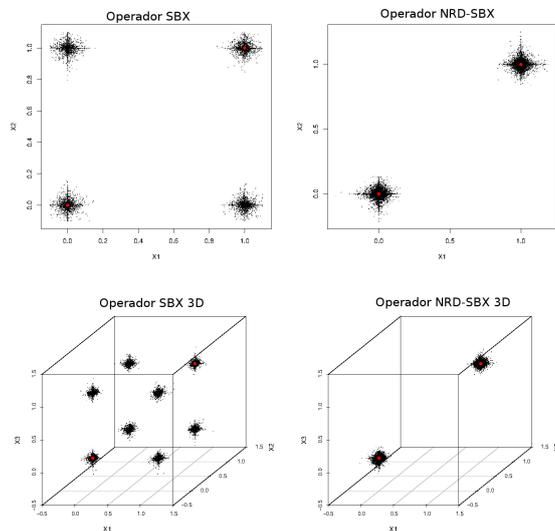
implementación los valores de las soluciones hijas son heredados directamente de las soluciones padre con una probabilidad fija igual a 0.5, mientras que el resto de las variables son modificadas mediante la distribución de probabilidad propia del SBX. En consecuencia la similitud que existe entre los padres y los hijos depende altamente del número de variables que se consideran en el problema de optimización, pues el incremento de la dimensionalidad involucra la creación de soluciones más distantes.

El segundo punto clave, que nunca ha sido analizado en detalle, está relacionado con el conjunto de reflexiones que se realizan. Después de generar los dos valores que deben ser heredados en los dos hijos, dichos valores son intercambiados entre sí con una probabilidad fija del 0.5. En consecuencia, cada vez que las variables son intercambiadas se realiza una reflexión, que puede inducir grandes distancias entre los padres y los hijos, a pesar de que el SBX es considerado como un cruce centrado en los padres.

La parte izquierda de la Figura 2 ilustra este comportamiento en el operador SBX para dos y tres variables. En esta figura, los padres están identificados con dos puntos rojos, y se ejecutó el operador SBX diez mil veces. Cada uno de los puntos de color negro es una solución hija, con lo que esta figura ilustra las zonas en las que se tienden a generar a los hijos. Se puede ver que los valores de cada variable siempre están cercanos a uno de los valores asociados a las variables de los padres. Sin embargo, debido al proceso de intercambio de valores, las soluciones hijas no siempre se encuentran cercanas a los padres.

Particularmente, en el caso de dos dimensiones mostrado, se crean soluciones en la esquina superior izquierda y en la esquina inferior derecha, mientras que los padres están en las esquinas opuestas. A medida que aumenta el número de dimensiones  $d$ , la probabilidad de que siempre o nunca haya intercambios, y que por tanto, la nueva solución esté cercana a uno de los padres es  $k^d + (1-k)^d$ , con lo que se produce una disminución exponencial respecto al número de dimensiones. En consecuencia, las reflexiones provocan un alto grado de exploración. En algunos MOPs con alta dimensionalidad en el espacio de las variables, esto podría representar un inconveniente porque las reflexiones localizan soluciones hijas en cada esquina del hipercubo mínimo que contiene a las soluciones padre, lo que significa que el operador original podría inducir un nivel muy bajo de intensificación.

El inconveniente relacionado con las reflexiones puede ser manejado parcialmente implementando restricciones de emparejamiento que traten de emparejar exclusivamente a soluciones similares. Bajo estas condiciones, el hipercubo sería de menor tamaño, con lo que se induciría un mayor grado de intensificación. Esta puede ser una de las razones por las que MOEA/D, que incorpora restricciones de apareamiento, ha sido capaz de resolver muchos problemas de forma exitosa. En este artículo, se trata de resolver esta problemática eliminando el intercambio de variables, así como realizando otras modificaciones en el SBX que se describen en la siguiente sección. La ventaja de esta segunda alternativa es que se puede incorporar de forma sencilla en cualquier MOEA.

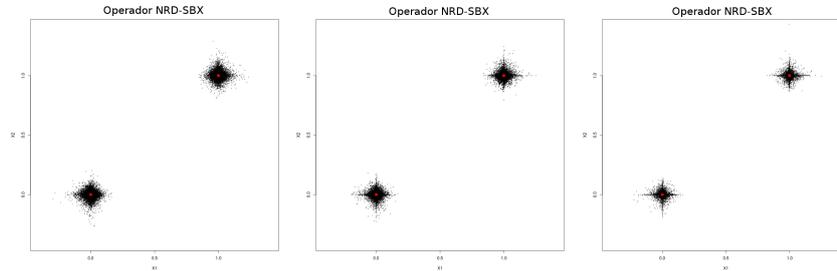


**Fig. 2.** En la columna izquierda se presenta la simulación del operador SBX y en la columna de la derecha se presenta el operador NRD-SBX, ambas distribuciones con un índice de distribución de 20.

#### 4. Propuesta algorítmica

Tomando en cuenta el análisis previo, así como el deseo de adaptar el operador a los requerimientos de las diferentes fases de optimización, se diseñó e implementó un nuevo operador de cruce, que se denominada operador Dinámico sin Reflexiones basado en el Cruce Binario Simulado (NRD-SBX). NRD-SBX realiza dos modificaciones al operador SBX descrito anteriormente. En primer lugar, en el NRD-SBX, la probabilidad de intercambiar variables se fija a 0, con lo que nunca aparecen las reflexiones propias del SBX original. El efecto de este cambio se muestra en la parte derecha de la Figura 2, en la que podemos ver que ahora todos los hijos quedaron localizados en regiones cercanas a los padres. Esto implica un mayor grado de intensificación. Es importante hacer notar la diferencia entre el operador modificado SBX y un operador de mutación. La distancia entre padres e hijos en el NRD-SBX es proporcional a la distancia entre los padres, mientras que los operadores de mutación sólo consideran la información de una solución padre.

La segunda modificación del NRD-SBX está relacionada con la cantidad de variables que se heredan sin modificarse. El operador SBX hereda directamente los valores que corresponden a los padres con una probabilidad igual a 0.5. Esto podría limitar el grado de exploración, especialmente en las primeras generaciones. Para evitar esta problemática, el operador NRD-SBX altera la probabilidad de heredar sin modificar las variables de los padres durante la ejecución del algoritmo. Particularmente, se implementa un modelo dinámico lineal, en el que



**Fig. 3.** Simulación del operador SBX con la probabilidad de heredar directamente las variables con 0%, 25% y 50% respectivamente. Los puntos rojos representan a los individuos padres.

en la primera generación, esta probabilidad es asignada a cero y con el transcurso de las generaciones, la probabilidad se incrementa de manera lineal, de forma que cuando han transcurrido la mitad de las generaciones la probabilidad es igual a 0.5. A partir de este momento, y hasta el final de la ejecución, se mantiene fija la probabilidad a 0.5. Esta regla tiende a ir disminuyendo paulatinamente el número de variables que se heredan sin realizar modificaciones, con lo que en las primeras fases se induce un mayor grado de exploración y las últimas fases, se induce un mayor grado de intensificación. La Figura 3, muestra para el caso de dos dimensiones, una simulación heredando directamente las variables con una probabilidad de 0, 0.25 y 0.50. Se puede apreciar que con probabilidades bajas (fases iniciales de la optimización) se exploran más zonas, mientras que en las fases finales se tienden a crear muchas soluciones que comparten valores con los padres, siendo este un paso mucho más intensificador. Cabe destacar que el comportamiento final en el que se mantienen bastantes variables sin cambiar, es deseable con MOPs en los que las variables son independientes. Sin embargo, en los problemas que involucran dependencia entre las variables, utilizar modificaciones de este tipo puede llevar a que el algoritmo se quede estancado en óptimos locales de baja calidad.

## 5. Validación experimental

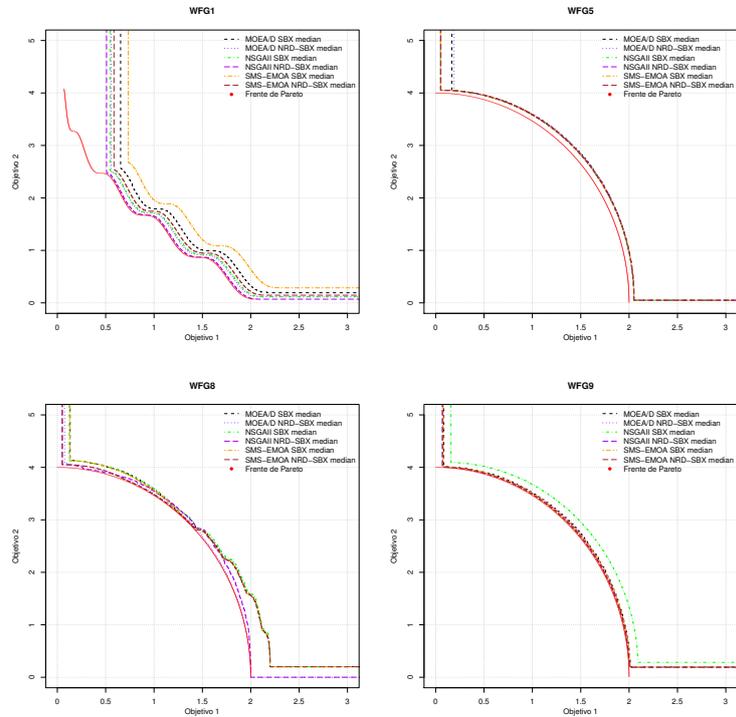
Esta sección está dedicada a mostrar el conjunto de experimentos que se realizaron para validar el rendimiento del operador NRD-SBX. Se muestran estudios con los tres algoritmos previamente descritos, considerando la implementación presente en jMetalcpp [15]. El único cambio que hubo que realizar en cada caso fue la inclusión del operador de cruce. La versión de SBX que viene incorporada en jMetalcpp difiere ligeramente de la incorporada en la versión oficial del NSGA-II, pues la variante de jMetalcpp incluye un paso de reflexión adicional. En nuestra validación se consideró la versión incluida en el NSGA-II oficial, así como el nuevo operador NRD-SBX. En lo referente a las funciones multi-objetivo, se consideran las 9 funciones WFG que se propusieron en [12].

Dado que todos los optimizadores son estocásticos, cada ejecución se repitió 35 veces con diferentes semillas en el generador de números aleatorios. Los siguientes aspectos fueron comunes para todos los algoritmos: el criterio de parada se fijó a 250,000 generaciones, el tamaño de población a 100, el operador de cruce SBX o NRD-SBX se configuró con un índice de distribución igual a 20 y la probabilidad de su uso fue de 0.9, se utilizó la mutación polinomial con un índice de distribución igual a 50 y su probabilidad de aplicación se fijó a 1/24. Además, los problemas WFG fueron configurados con 24 parámetros, siendo 20 de ellos parámetros de distancia, y 4 de ellos parámetros de posición. Por otro lado, (a excepción del NSGA-II) hay algunas configuraciones adicionales que son propias de cada uno de los algoritmos utilizados:

- **SMS-EMOA**: el desplazamiento se fijó a 100.
- **MOEA/D**: el tamaño de la vecindad se fijó a 20, el número máximo de actualización de subproblemas se fijó a 2 y la probabilidad de emparejar soluciones únicamente de la vecindad es de 0.9 .

El análisis experimental se realizó teniendo en cuenta las superficies de cubrimiento y el hipervolumen. En el caso de las comparativas en base al hipervolumen, y teniendo en cuenta el comportamiento estocástico de los optimizadores, se realizaron un conjunto de tests estadísticos siguiendo los criterios establecidos en [9]. Concretamente, en primer lugar se utilizó el test Shapiro-Wilk para comprobar si los resultados se ajustaban a una distribución Gaussiana. En los casos en que sí se ajustaban, se utilizó el test de Levene para comprobar la homogeneidad de las varianzas, procediendo con el test de ANOVA en caso positivo o con el de Welch en caso negativo. Por otro lado, para los casos que no se ajustaban a distribuciones Gaussianas, se utilizó el test de Kruskal-Wallis. En todos los casos se fijó el nivel de confianza al 95 %. Se considera que un algoritmo  $X$  es superior a un algoritmo  $Y$ , si el procedimiento anterior reporta diferencias significativas y si la media y mediana del hipervolumen obtenido por el método  $X$  son superiores a las obtenidas por el método  $Y$ .

En primer lugar se van a analizar los resultados en base a las superficies de cubrimiento. Para ello, de los 9 problemas se seleccionaron los 4 casos en los que se detectaron mayores diferencias. La Figura 4 muestra las superficies de cubrimiento al 50 % del WFG1, WFG5, WFG8 y WFG9. Se puede apreciar que para cada algoritmo, el uso del operador NRD-SBX produce resultados que dominan o igualan a los alcanzados por SBX en la mayor parte de los casos. De hecho, la única excepción se da en el WFG5 con el NSGA-II, en el que la utilización de NRD-SBX produce resultados ligeramente inferiores. En cualquier caso, los beneficios ofrecidos por NRD-SBX en el resto de casos son mucho mayores que la penalización que aparece en el caso del WFG5. Particularmente, los beneficios ofrecidos en el caso del WFG1 y WFG8 son especialmente claros. En el caso del WFG1, se ha mostrado en otros artículos [3] que la pérdida rápida de la diversidad puede ser un grave problema, con lo que el mayor grado de exploración que induce NRD-SBX en las fases iniciales es de gran ayuda. Por otro lado, dado que es un problema unimodal y separable, evitar las reflexiones



**Fig. 4.** Superficies de cubrimiento al 50 % obtenidas por los diferentes MOEAs con SBX y NRD-SBX.

y centrarse en crear soluciones cercanas a los padres también contribuye al buen rendimiento. En el caso del WFG8, hay un alto grado de dependencia entre las variables, por lo que ir cambiando de forma dinámica la cantidad de variables que se heredan sin realizar modificaciones y por tanto no centrarse tanto en soluciones que comparten muchos valores con los ya encontrados anteriormente es de gran ayuda. En el WFG9 se da una situación similar, aunque por ser este problema más sencillo, los beneficios sólo se aprecian de forma clara en el caso del NSGA-II.

Con el objetivo de confirmar y complementar los hallazgos anteriores, se realizaron análisis del hipervolumen alcanzado por los diferentes algoritmos, con cada uno de los operadores de cruce. La Tabla 1 muestra para cada algoritmo y operador de cruce el mínimo, máximo, media y desviación estándar de los valores del hipervolumen alcanzados. Tal y como es habitual [20], para el cálculo de este valor, se fijó el punto de referencia en (3.0, 5.0). Para cada algoritmo y problema, se realizaron comparativas estadísticas entre el operador SBX y el operador NRD-SBX. En los casos en los que se pudo confirmar la superioridad de alguno de los operadores, se resaltan los datos en la tabla. Para el caso del MOEA/D

**Tabla 1.** Valores de hipervolumen alcanzados por los diferentes MOEAs con los operadores de cruce SBX y NRD-SBX.

MOEA/D								
	NRD-SBX				SBX			
	Min.	Max.	Media	Desv.	Min.	Max.	Media	Desv.
WFG1	<b>9.61</b>	<b>10.72</b>	<b>10.26</b>	<b>3.1E-01</b>	8.39	10.29	9.62	4.3E-01
WFG2	<b>10.62</b>	<b>11.45</b>	<b>10.66</b>	<b>1.9E-01</b>	9.28	10.62	10.50	3.7E-01
WFG3	10.95	10.95	10.95	2.5E-04	10.95	10.95	10.95	3.0E-04
WFG4	8.68	8.68	8.68	1.0E-04	8.67	8.68	8.68	1.0E-04
WFG5	8.13	8.34	8.16	4.7E-02	<b>8.13</b>	<b>8.24</b>	<b>8.19</b>	<b>3.0E-02</b>
WFG6	<b>8.39</b>	<b>8.57</b>	<b>8.49</b>	<b>4.8E-02</b>	7.76	8.54	8.30	1.2E-01
WFG7	8.67	8.68	8.67	1.5E-04	8.67	8.68	8.67	2.3E-04
WFG8	<b>8.20</b>	<b>8.60</b>	<b>8.54</b>	<b>8.5E-02</b>	7.80	7.87	7.83	1.8E-02
WFG9	<b>7.70</b>	<b>8.56</b>	<b>8.41</b>	<b>1.9E-01</b>	7.70	8.54	8.20	3.1E-01

NSGA-II								
	NRD-SBX				SBX			
	Min.	Max.	Media	Desv.	Min.	Max.	Media	Desv.
WFG1	<b>9.91</b>	<b>10.70</b>	<b>10.52</b>	<b>1.8E-01</b>	9.51	10.76	10.21	3.1E-01
WFG2	<b>10.59</b>	<b>11.44</b>	<b>10.66</b>	<b>1.9E-01</b>	9.28	10.62	10.58	2.2E-01
WFG3	10.91	10.93	10.92	4.0E-03	<b>10.92</b>	<b>10.93</b>	<b>10.93</b>	<b>2.7E-03</b>
WFG4	8.65	8.66	8.66	2.3E-03	<b>8.66</b>	<b>8.67</b>	<b>8.67</b>	<b>2.2E-03</b>
WFG5	8.26	8.27	8.27	2.3E-03	8.27	8.28	8.27	1.6E-03
WFG6	<b>8.33</b>	<b>8.59</b>	<b>8.47</b>	<b>5.6E-02</b>	8.12	8.46	8.34	6.5E-02
WFG7	8.64	8.66	8.65	3.9E-03	<b>8.66</b>	<b>8.67</b>	<b>8.66</b>	<b>2.6E-03</b>
WFG8	<b>7.88</b>	<b>8.50</b>	<b>8.26</b>	<b>2.7E-01</b>	7.71	7.83	7.78	3.2E-02
WFG9	<b>8.34</b>	<b>8.61</b>	<b>8.43</b>	<b>7.9E-02</b>	7.69	8.51	7.85	2.8E-01

SMS-EMOA								
	NRD-SBX				SBX			
	Min.	Max.	Media	Desv.	Min.	Max.	Media	Desv.
WFG1	<b>8.60</b>	<b>10.70</b>	<b>10.03</b>	<b>4.6E-01</b>	8.00	9.89	8.92	5.0E-01
WFG2	<b>10.63</b>	<b>11.46</b>	<b>10.65</b>	<b>1.4E-01</b>	9.29	11.46	10.61	2.7E-01
WFG3	10.96	10.96	10.96	2.8E-04	10.96	10.96	10.96	3.6E-04
WFG4	8.69	8.69	8.69	7.7E-06	8.69	8.69	8.69	6.8E-06
WFG5	8.22	8.29	8.27	2.1E-02	<b>8.25</b>	<b>8.29</b>	<b>8.29</b>	<b>8.6E-03</b>
WFG6	<b>8.36</b>	<b>8.58</b>	<b>8.50</b>	<b>4.7E-02</b>	8.22	8.47	8.34	5.8E-02
WFG7	8.69	8.69	8.69	4.6E-06	8.69	8.69	8.69	6.5E-06
WFG8	<b>7.98</b>	<b>8.61</b>	<b>8.12</b>	<b>2.2E-01</b>	7.81	7.89	7.85	2.2E-02
WFG9	<b>8.33</b>	<b>8.66</b>	<b>8.47</b>	<b>8.7E-02</b>	7.71	8.56	8.20	3.1E-01

la superioridad del NRD-SBX se pudo confirmar en cinco casos, mientras que sólo fue inferior en uno; en el NSGA-II, NRD-SBX fue superior en 5 casos e inferior en 3; finalmente en el SMS-EMOA, NRD-SBX fue superior en 5 casos e inferior sólo en 1. Los resultados anteriores confirman la superioridad de NRD-SBX, pero además es importante recalcar que en los casos en que el operador SBX fue superior, la diferencia entre los valores de hipervolumen alcanzados fue muy pequeña. Por ejemplo, en los 3 problemas en los que NSGA-II con SBX fue superior a NSGA-II con NRD-SBX la diferencia de la media de los valores alcanzados fue sólo 0.01, mientras que en los casos en que NRD-SBX fue superior la diferencia es mayor en más de un orden de magnitud.

Finalmente, con el objetivo de recalcar la superioridad del NRD-SBX, se realizó un test adicional que cuantifica la mejora o empeoramiento aportado por cada algoritmo. En concreto, en todos los problemas, cada pareja de algoritmo y operador de cruce fue comparado estadísticamente frente al resto para obtener una puntuación global. En los casos en los que se pudo comprobar

**Tabla 2.** Puntuación obtenida por los diferentes algoritmos en base a los test estadísticos.

	NRD-SBX									SBX								
	MOEA/D			NSGA-II			SMS-EMOA			MOEA/D			NSGA-II			SMS-EMOA		
	↑	↓	Punt															
WFG1	2.2	0.3	1.9	3.6	0.0	3.6	1.5	0.7	0.8	0.7	2.5	-1.8	1.9	0.3	1.5	0.0	6.0	-6.0
WFG2	0.2	0.0	0.2	0.1	0.0	0.1	0.3	0.0	0.2	0.0	0.4	-0.4	0.0	0.1	-0.1	0.1	0.1	0.0
WFG3	0.1	0.0	0.0	0.0	0.2	-0.2	0.1	0.0	0.1	0.1	0.0	0.0	0.0	0.1	-0.1	0.1	0.0	0.1
WFG4	0.0	0.0	0.0	0.0	0.1	-0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.1	-0.1	0.1	0.0	0.1
WFG5	0.0	0.5	-0.5	0.2	0.0	0.2	0.2	0.0	0.2	0.0	0.4	-0.3	0.2	0.0	0.2	0.3	0.0	0.3
WFG6	0.5	0.0	0.5	0.4	0.0	0.4	0.5	0.0	0.5	0.0	0.5	-0.5	0.0	0.5	-0.5	0.0	0.4	-0.4
WFG7	0.0	0.0	0.0	0.0	0.1	-0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.1	-0.1	0.1	0.0	0.1
WFG8	2.9	0.0	2.9	1.3	0.3	1.0	0.9	0.4	0.5	0.1	1.4	-1.4	0.0	1.7	-1.7	0.1	1.4	-1.3
WFG9	1.0	0.1	0.9	1.0	0.0	1.0	1.3	0.0	1.3	0.4	0.7	-0.4	0.0	2.5	-2.5	0.4	0.7	-0.4
Total	<b>6.8</b>	0.9	6.0	<b>6.6</b>	0.8	5.9	<b>4.9</b>	1.2	3.8	<b>1.2</b>	6.1	-4.8	<b>2.1</b>	5.3	-3.2	<b>1.1</b>	8.7	-7.6

su superioridad, se sumó a su puntuación el valor absoluto de la diferencia entre la media de los valores de hipervolumen alcanzados, mientras que en los casos en que se pudo comprobar la inferioridad se restó dicho valor. La Tabla 2 muestra la puntuación obtenida en cada problema para cada par de algoritmo y operador de cruce. En las columnas etiquetadas con ↑ se muestra el acumulado de las puntuaciones positivas, mientras que en las columnas etiquetadas con la columna ↓ se muestra el acumulado de puntuaciones negativas. Finalmente, en la columna *Punt* se muestra la puntuación final. Se puede apreciar que los algoritmos que consideraron el operador NRD-SBX fueron los tres superiores. Entre ellos, el NSGA-II fue el superior, aunque con un rendimiento muy similar al MOEA/D. Por su parte, el rendimiento de los algoritmos con el operador SBX fue claramente inferior independientemente del MOEA aplicado.

## 6. Conclusiones y trabajos futuros

Los MOEAs son uno de los esquemas más populares para lidiar con problemas de optimización multi-objetivo complejos. En el caso de los entornos continuos, el operador de cruce SBX es uno de los más populares, habiendo mostrado un gran rendimiento tanto en optimización mono-objetivo como multi-objetivo. En este artículo se realiza un análisis del operador SBX y en base a este análisis se propone un nuevo operador, el NRD-SBX. El análisis realizado, muestra que las implementaciones actuales realizan un conjunto de reflexiones que provocan que el operador pueda crear individuos muy alejados de los padres, reduciendo así su capacidad de intensificación. Además, se destaca que el SBX cambia cada variable con una probabilidad igual a 0.5 durante toda la optimización. El NRD-SBX introduce dos cambios principales. Por un lado, evita la utilización de reflexiones con el fin de aumentar el poder de intensificación. Por otro lado, adapta la probabilidad de alterar las variables a lo largo de la ejecución, induciendo un mayor grado de exploración en las fases iniciales que en las fases finales de la optimización. Con el objetivo de validar el NRD-SBX, se realizó un análisis experimental con tres MOEAs muy populares, considerándose el conjunto de

tests WFG. El análisis de las superficies de cubrimiento e hipervolumen muestran las ventajas del NRD-SBX frente al SBX.

Actualmente se están abordando varias líneas de trabajo futuro. Por un lado se está trabajando en operadores de cruce que adapten la probabilidad de realizar reflexiones, así como la apertura de las distribuciones utilizadas. Por otro lado, se están diseñando algoritmos que controlan la diversidad de forma explícita para complementar los cambios realizados en el operador de cruce.

## Referencias

1. Agrawal, R.B., Deb, K., Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Tech. rep. (1994)
2. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181(3), 1653 – 1669 (2007)
3. Chacón, J., Segura, C., Hernández-Aguirre, A., Miranda, G., León, C.: In: *Proceedings of the 2017 on Genetic and Evolutionary Computation Conference Companion*. p. In Press. GECCO '17 Companion, ACM (2017)
4. Chiang, T.C., Lai, Y.P.: MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism. In: *2011 IEEE Congress of Evolutionary Computation (CEC)*. pp. 1473–1480 (June 2011)
5. Crepinšek, M., Liu, S.H., Mernik, M.: Exploration and Exploitation in Evolutionary Algorithms: A Survey. *ACM Computing Surveys* 45(3), 35:1–35:33 (Jul 2013)
6. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation* 15(1), 4–31 (Feb 2011)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (Apr 2002)
8. Deb, K., Beyer, H.g.: Self-adaptive genetic algorithms with simulated binary crossover. *Evol. Comput.* 9(2), 197–221 (Jun 2001), <http://dx.doi.org/10.1162/106365601750190406>
9. Durillo, J.J., Nebro, A.J., Coello, C.A.C., Garcia-Nieto, J., Luna, F., Alba, E.: A Study of Multiobjective Metaheuristics When Solving Parameter Scalable Problems. *IEEE Transactions on Evolutionary Computation* 14(4), 618–635 (Aug 2010)
10. Hernández Gómez, R., Coello Coello, C.A., Alba, E.: A Parallel Version of SMS-EMOA for Many-Objective Optimization Problems, pp. 568–577. Springer International Publishing, Cham (2016)
11. Herrera, F., Lozano, M., Sánchez, A.M.: A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems* 18(3), 309–338 (2003), <http://dx.doi.org/10.1002/int.10091>
12. Huband, S., Barone, L., While, L., Hingston, P.: A Scalable Multi-objective Test Problem Toolkit, pp. 280–295. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
13. Ishibuchi, H., Masuda, H., Tanigaki, Y., Nojima, Y.: Modified Distance Calculation in Generational Distance and Inverted Generational Distance, pp. 110–125. Springer International Publishing, Cham (2015), [http://dx.doi.org/10.1007/978-3-319-15892-1\\_8](http://dx.doi.org/10.1007/978-3-319-15892-1_8)

14. Jain, H., Deb, K.: Parent to mean-centric self-adaptation in sbx operator for real-parameter optimization. In: Proceedings of the Second International Conference on Swarm, Evolutionary, and Memetic Computing - Volume Part I. pp. 299–306. SEMCCO'11, Springer-Verlag, Berlin, Heidelberg (2011), [http://dx.doi.org/10.1007/978-3-642-27172-4\\_37](http://dx.doi.org/10.1007/978-3-642-27172-4_37)
15. jMetalcpp: <http://jmetalcpp.sourceforge.net/>
16. Jong, K.A.D.: Evolutionary computation - a unified approach. MIT Press (2006)
17. Slim Bechikh, Rituparna Datta, A.G.: Recent Advances in Evolutionary Multi-objective Optimization. springer
18. Talbi, E.G.: Metaheuristics - From Design to Implementation. Wiley (2009)
19. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (Dec 2007)
20. Zhu, Q., Lin, Q., Du, Z., Liang, Z., Wang, W., Zhu, Z., Chen, J., Huang, P., Ming, Z.: A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm. Information Sciences 345, 177 – 198 (2016)