Noname manuscript No.
(will be inserted by the editor)

Diversidad en los Algoritmos Evolutivos

Carlos Segura · Joel Chacón Castillo

Received: date / Accepted: date

Resumen Keywords Diversidad · Multi-objetivo · Convergencia Prematura · Evolución Diferencial

1. Introducción

Los Algoritmos Evolutivos (Evolutionary Algorithms EA) son considerados como uno de los enfoques con mayor aficacia para resolver distintas categorías de optimización. Particularmente, se han aplicado en problemas tanto de dominio continuo ref 3ANDBRS como de dominio discreto. Especialmente, los EAs son aplicados para resolver problemas complejos cuyo enfoque determinístico es complicado o imposible [1]. Además, diversas variantes se han utilizado y aplicado en muchos campos, como es en ciencia, economía e ingeniería. Actualmente, los EAs son plenamente conocidos como metaheurísticas ref 3-ANDBRS. A pesar del éxito que tienen los EAs, existe una dificultad en su adaptación o configuración ante nuevos problemas. Una dificultad popular en el diseño de un EA es obtener un balanceo propio entre exploración y explotación ref5-ANDBRS. Sin embargo, no siempre son comprendidas las implicaciones al mantener un grado de diversidad en este balanceo y como son promovidas las exploración y explotación ref6-ANDBRS. Desde su inicio los EAs han presentado problemas de convergencia siendo como una desventaja importante ref6-ANDBRS. La convergencia prematura es originada cuando todos los miembros de la población están ubicados en una parte reducida del espacio de búsqueda, esta región es distinta a la región óptima y además los

Carlos Segura
Centro de Investigación en Matemáticas, Guanajuato, Mexico
E-mail: carlos.segura@cimat.mx

Joel Chacón Castillo
Centro de Investigación en Matemáticas, Guanajuato, Mexico
E-mail: joel.chacon@cimat.mx

componentes seleccionados no son suficientes para escapar de esta región. Basado en esto se han desarrollado varias estrategias para aliviar este problems. Además en através de varios estudios se ha revelado que mantener una población diversa en un requisito previo para evitar la convergencia prematura ref6-ANDBRS. Sin embargo, si la población es muy diversa, entonces un grado adecuado de explotación podría ser prevenido, resultando en un convergencia lenta y por lo tanto soluciones de baja calidad. Por esta razón, Mahfoud ref10-ANDBRS utilizó el concepto de diversidad util, con cual se refiere a la cantidad de diversidad que resulta en soluciones de alta calidad. En la literatura existen distintas forma para aliviar el problema de convergencia prematura ref40-TIPDMEALCO. En los 90s, la mayoría de estrategias para aliviar la convergencia prematura se centraron en modificar el esquema de selección de padres. La principal razón es que en esa época la mayoría de esquemas eran generacionales, por lo tanto la presión de selección estaba definida principalmente en la selección de padres. Sin embargo, se descubrió que tratando de aliviar la convergencia prematura donde únicamente sea considerada la selección de padres no fue suficiente ref4-TIPDMEALCO. Posteriormente, la una gran cantidad de EAs incorporaron una fase de reemplazo que abandonaron al menos parcialmente a los métodos generacionales iniciales. Basado en esto muchos autores descubrieron la posibilidad de incorporar métodos para aliviar el problema de convergencia prematura ref10-TIPDMEALCO. Es importante considerar que aún cuando los métodos de reemplazamiento generacionales ref11-ANDBRS fueron suficientemente populares, algunos autores ya habían tomado en cuenta esta estrategia ref-30-TIPDMEALCO. Sin embargo, con la efectividad de elitismo y otras estrategias de reemplazo, el número de esquemas que adoptaron estos principios crecieron de forma considerable ref-27-TIPDMEALCO.

Un aspecto importante de la convergencia prematura es que depende completamente de la cantidad de tiempo y/o generaciones asignados a las ejecuciones del EA, es decir el criterio de paro. En su lugar, un EA debería ser ejecutado para resolver un problema dado por un tiempo definido y éste debería proporcionar soluciones prometedoras. A pesar de esto, es sorprendente que la mayoría de métodos que se han propuesto para aliviar la desventaja de la convergencia prematura no consideran el criterio de paro el cual es asignado por el usuario para alterar su comportamiento interno. Esto significa que, dependiendo en el criterio de paro, distintas parametrizaciones podrían ser requeridas. Como resultado, para criterio de paro distinto, el usuario debería estudiar el efecto de distintos parámetros. Un ejemplo popular de esto es *El Torneo de Selección Restringida (TSR)* ref14-TIPDMEALCO, este método retrasa la convergencia de los EAs. Específicamente, este método incorpora un parámetro que puede ser utilizado para alterar el balanceo entre exploración e intensificación. Sin embargo, la pérdida de diversidad y posteriormente el balanceo entre exploración e intensificación no dependen únicamente en este parámetro, por lo tanto distintos valores deberían ser utilizados para cada problema y además para cada criterio de paro. El principio básico de las técnicas para la preservación de la diversidad y que afectan a la fase de reemplazo se

basa en que el efecto de diversificar a los sobrevivientes inducide un mayor grado de exploración. Esto se debe a varios aspectos importantes, principalmente una población grande mantiene varias regiones del espacio de búsqueda. Además los operadores de cruce tienden a se más explorativos cuando están involucrados individuos distantes ref13-TIPDMEALCO.

Entre las distintas categorías de EAs, Evolución Diferencial (Differential Evolution - DE) es una de las estrategias más efectivas para lidiar con problemas de optimización contínua [2]. De hecho, esta categoría de algoritmos han sido los ganadores en varias competencias de optimización [3]. Similarmente a otros EAs. DE está inspirado en el proceso de evolución natural, y además involucra la aplicaciónde mutación, recombinación y selección. La principal característica de DE es que éste considera las diferencias de los vectores que están presentes en la población con el motivo de explorar el espacio de búsqueda. En este sentido DE es similar a los optimizadores *Nelder-Mead* [4] y a la *Búsqueda Aleatoria Controlada (BAC)* [5]. A pesar de la efectividad de DE, existen varias debilidades que han sido detectadas y resueltas de forma parcial que por lo tanto a generado extensiones a la variante estándar de DE [3]. Algunos de los problemas más conocidos es la sensibilidad de la parametrización [6], la apariencia de estancamiento debido a las capacidades de exploración reducidas [7, 8] y la convergencia prematura [9]. Desde la aparición de DE, varias críticas se hicieron debido a su falta de capacidad para mantener un grado de diversidad suficiente dado a la elevada presión de selección [7]. Por lo tanto, se han generado varias extensiones de DE para aliviar la convergencia prematura, como la adaptiación de parámetros [9], auto-adaptación de la diversidad en la población [10] y estrategias de selección con una menor presión de selección [7]. Algúnos de los últimos estudios en el diseño metaheurísticas poblacionales [11] mostraron que controlando explícitamente la diversidad es particularmente útil para obtener un propio balanceo entre el grado de exploración y de intensificación. Nuestra hipótesis es que introduciendo un mecanismo para la preservación de la diversidad results en un balanceo adecuado entre exploración e intensidifación, que a su vez propociona soluciones de alta calidad en ejecuciones a largo plazo.

Principalmente en este capítulo se explican dos propuestas, primeramente DE con Mantenimiento de Diversidad Mejorado (DEwith Enhanced Diversity MaintenanceDE-EDM), el cual integra un principio similar en DE y el operador Dinámico basado en el Cruce Binario Simulado (Dynamic Simulated Binary Crossover DSBX).

Our novel proposal, which is called DE with Enhanced Diversity Maintenance (DE-EDM), integrates a similar principle into DE. El resto de este capítulo está organizado de la siguiente forma.

2. Diseño de Evolución Diferencial basado en diversidad

2.1. Evolución Diferencia: Conceptos Básicos

Esta sección esta dedicada para repasar la variante clásica de DE y para introducir algunos de los mas importantes términos utilizados en el campo de DE. El clásico esquema DE es identificado como DE/rand/1/bin el cual ha sido extensamente utilizado para generar más variantes complejas [3]. De hecho, nuestra propuesta también extiende a la clásica versión DE. Originalmente DE fue propuesta como un método de búsqueda directo para optimización contínua mono-objetivo. El conjunto de variables involucradas en el planteamiento de un problema son dados como un vector de la forma $\mathbf{X} = [x_1, x_2, ..., x_D]$, donde D es la dimensión del problema. En optimización contínua, cada x_i es un número real, además son proporcionadas restricciones de caja, es decir, existe un límite inferior (a_i) y un límite superior (b_i) para cada variable. El objetivo de un proceso de optimización es obtener un vector $\mathbf{X}^*$ el cual minimiza una función objetivo dada, esto matemáticamente es definido por $f : \Omega \subseteq \Re^D \rightarrow \Re$. En el caso de la restricción de caja $\Omega = \prod_{j=1}^{D}[a_j, b_j]$.

DE es un algoritmo estocástico basado en una población, por lo tanto éste involucra iterativamente a un conjunto de soluciones candidatas. En DE dichas soluciones candidatas son usualmente conocidas como vectores. En la variante básica de DE, para cada miembro de la población conocidos como *vectores objetivo* es generado un nuevo vector conocido como *vector mutado*. Entonces, el vector mutado es combinado con el vector objetivo para generar al *vector de prueba*. Finalmente, una fase de selección es aplicada para seleccionar a los vectores sobrevivientes. De esta forma, transcurren las generaciones hasta cumplir el criterio de paro. El i-ésimo vector de la población en la generación G es definido como$\mathbf{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, ..., X_{D,i,G}]$. A continuación se explica en más detalle cada componente de DE.

2.1.1. Inicialización

DE usually starts the optimization process with a randomly initiated population of NP vectors. Since there is commonly no information about the performance of different regions, uniform random generators are usually applied. Hence, the jth component of the ith vector is initialized as $x_{j,i,0} = a_j + rand_{i,j}[0,1](b_j - a_j)$, where $rand_{i,j}[0,1]$ is an uniformly distributed random number lying between 0 and 1.

2.1.2. Mutación

Para cada vector objetivo un vector mutado es creado, varias estratigias para realizar este procedimiento han sido propuestas. En la variante clásica de DE se aplica la estrategia rand/1. En este caso, es creado el vector mutado $V_{i,G}$ de la siguiente forma:

$$\mathbf{V}_{i,G} = \mathbf{X}_{r1,G} + F \times (\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G}) \quad r1 \neq r2 \neq r3 \tag{1}$$

Los índices $r1, r2, r3 \in [1, NP]$ distintos enteros seleccionados de forma aleatoria en el rango $[1, NP]$. Además, estos índices son distintos al índice i. Es imporante tomar en cuenta que la diferencia entre los vectores es escalada por medio del parámetro F, el cual se define usualmente en el intérvalo [0,4, 1]. La diferencia escalada es agregada al tercer vector, por lo tanto los vectores mutados son similares a los vectores objetivo cuando el grado de diversidad es poco y las diferencias son pequeñas. Como resultado, es importante mantener un grado de diversidad mínimo en DE.

2.1.3. Cruza

En orden con el objetivo de combinar la información de distintas soluciones candiadtas y con el propósito de incrementar la diversidad es aplicado el operador de cruce. Específicamente, cada vector objetivo $\mathbf{X}_{i,G}$ es mezclado con su correspondiente vector mutado $V_{i,G}$ para general un vector de prueba $\mathbf{U_{i,G}} = [u_{1,i,G}, u_{2,i,G}, ..., u_{D,i,G}]$. La estrategia de cruza mas típica es conocida como cruza *binomial*, el cual funciona de la siguiente forma:

$$\mathbf{U}_{j,i,G} = \begin{cases} \mathbf{V}_{j,i,G}, & \text{si}(rand_{i,j}[0,1] \leq CR \quad or \quad j = j_{rand}) \\ \mathbf{X}_{j,i,G}, & \text{de otra forma} \end{cases} \tag{2}$$

donde $rand_{i,j}[0, 1]$ es un número uniformemente distribuido, j_{rand} es un índice aleatoriamente seleccionado el cual asegura que $\mathbf{U}_{i,G}$ genera al menos un componente de $\mathbf{V}_{i,G}$ y $CR \in [0, 1]$ es la razón de cruce.

2.1.4. Selección

Finalmente, se aplica una selección glotona para determinar a los sobreviviente de la siguiente generación. Cada vector de prueba es comparado con su correspondiente vector objetivo y el mejor es el que sobrevive:

$$\mathbf{X}_{j,i,G+1} = \begin{cases} \mathbf{U}_{i,G}, & \text{si} \quad f(\mathbf{U}_{i,G}) \leq f(\mathbf{X}_{i,G}) \\ \mathbf{X}_{i,G}, & \text{de otra forma} \end{cases} \tag{3}$$

Hence, each population member either gets better or remains with the same objective value in each generation. Since members never deteriorate, it is considered to be a selection with high pressure. Note that in case of a tie, the trial vector survives.

2.2. Diversidad en Evolución Diferencial

Los algoritmos basados en DE son altamente suceptibles a la pérdida de diversidad devido a la estrategia de selección agresiva. Sin embargo, se han desarrollado varios análisis para lidiar con este problema. Desde que se conocen las implicaciones de cada parámetro en la diversidad, una alternativa es la estimación teórica de los valores adecuados en DE [9]. Alternativamente, se han desarrollado algunos análisis donde es considerado el efecto de los vectores de diferencia en la mutación [12]. Estos análisis y otros estudios empíricos basados en la cruza permitieron concluir que ciertos tipos de movimientos deberían ser deshabilidatos para retrasar la convergencia [13]. En este último estudio varía el tipo de movimientos aceptados a lo largo de la ejecución. Específicamente, esto descarta movimientos menores a un umbral el cual es decrementado conforme transcurren las generaciones. Se han propuesto otras formas de alterar el procedimiento en que se aceptan los movimientos [14]. Es importante notar que este tipo de métodos tienen similitudes con nuestra propuesta en el sentido de que las decisiones están basadas por el número de generaciones transcurridas. Sin embargo, nuestro método opera en la estrategia de reemplazo y no en la fase de mutación. Mas aún, estos métodos no consideran de forma explícita las diferencias que aparecen en la población entera. En su lugar, las restricciones son aplicadas a las diferencias que aparecen en la fase de reemplazo.

Una alternativa distinta reside en alterar el operador de selección [7]. Particularmente, se relaja la presión de selección a través de una selección probabilística con el propósito de mantener la diversidad en la población y consecuentemente permitir escapar de la base de atracción de un óptimo local. Sin embargo este método es muy sensible a transformaciones desde que esta estrategia considera la aptitud para definir las probabilidades para aceptar un individuo mutado. En este caso las decisiones no se basan en las generaciones transcurridas.

Finalmente, en el algoritmo *Diversidad de la Población Auto-Mejorado* (*Auto-Enhanced Population Diversity* - AEPD), la diversidad es explícitamente medida y esto es un dispara un mecanismo para diversificar a la población cuando se detecta poca diversidad en la población [10]. También ya se han propuesto estrategias con principios similares pero con esquemas de perturbación distintos.

Es importante notar que las mejores variantes-DE de las competencias no utilizan estas modificaciones y que la mayoría de estas extensiones no han sido implementadas en los herramientas de optimización más utilizadas. Como resultado, estas extensiones no son ampliamente utilizadas por la comunidad a pesar de sus beneficios en ciertos casos.

2.3. Propuesta

Nuestra propuesta está motivada por dos trabajos significativos de ésta área cuyo propósito es el control de la diversidad en los EAs. Por su parte el

primero es un estudio empírico desarrollado por Montgomery y otros [13], este trabajo presenta varios análisis empíricos los cuales confirman los problemas relacionados con la convergencia prematura.

Por otro lado, el segundo trabajo propuesto por Segura [15] y otros, proporciona mejoras significativas en el campo optimización combinatoria, en esta propuesta se desarrolla una novel estrategia de reemplazo nombrada *Reemplazo con Control de Diversidad Dinámico Basado en Varios Objetivos* (Replacement with Multi-objective based Dynamic Diversity Control - RMDDC) donde se controla el grado de diversidad con el criterio de paro y las generaciones transcurridas. Se obtuvieron beneficios por los métodos que incluyeron el RMDDC, por lo tanto y basados en las conclusiones de los trabajos previos, la propuesta de esta sección es una novel variante de DE que incluye un mecanismo explícito el cual sigue uno de los principios del RMDDC. Este novel optimizador es nombrado *Evolución Diferencial con Mantinimiento Mejorado de Diversidad* (Differential Evolution with Enhanced Diversity Maintenance - DE-EDM) y su código fuente está disponible de forma gratuita [1]

La escencia de DE-EDM (ver algoritmo 1) es suficiente similar a la versión estándar de DE. De hecho la forma en que se crean los vectores de prueba no es modificado de forma significativa (líneas 5 y 6). La novedad de la propuesta es que incorpora una población elite (E) y una novel estrategia de reemplazo basada en la diversidad. En orden, para seleccionar a los miembros de la población, se aplica el reemplazo agresivo (glotón) de la versión original de DE(línea 7). Por otra parte, se considera otra estrategia de reemplazo (línea 8), la cual realiza la selección de los miembros que participarán en el siguiente procedimiento de selección, esto se realiza siguiendo el mismo principio que el RMDDC, es decir los individuos que contribuyen muy poco a la diversidad no deberán ser aceptados como miembros en la siguiente generación. En este sentido, no se utiliza la misma estrategia de selección agresiva que pertenece a DE para mantener a la población padre (X). En este orden para establecer una contribución aceptable de diversidad mínima para realizar la selección, son tomados en cuenta el criterio de paro y las generaciones transcurridas. Una de las principales debilidades del RMDDC es que su convergencia se retrasa de forma significativa. Por lo tanto, se realizan dos modificaciones al RMDDC para promover una convergencia acelerada. Primero, no son considerados los conceptos multi-objetivo, en su lugar se considera una selección mas agresiva. Segundo, también es considerada la población elite en la estrategia de reemplazo.

Nuestra estrategia de reemplazo (ver Algoritmo 2) funciona de la siguiente forma. Éste recive a la población padre (vectores objetivo), la población de hijos (vectores de prueba) y a los vectores elite. En cada generación son seleccionados NP vectores para la siguiente población de padres. Primero, en base al número de evaluaciones a funcion (línea 2) es calculada una distancia mínima D_t deseada para mantener la diversidad. Entonces, son juntadas las tres poblaciones en un conjunto de miembros candidatos (línea 3). El conjunto

[1] El código en C++ puede ser descargado en la siguiente dirección `https://github.com/joelchaconcastillo/Diversity_DE_Research.git`

Algorithm 1 Esquema general del DE-EDM

1: Inicializar de forma aleatoria a la población con NP individuos, donde cada uno es distribuído de forma uniforme.
2: $G = 0$
3: **while** El criterio de paro no sea alcanzado **do**
4: **for** $i = 1$ to NP **do**
5: Mutación: Generar al vector mutado ($V_{i,G}$) de acuerdo a la ecuación (1).
6: Cruza: Utilizar la recombinación para general al vector de prueba ($U_{i,G}$) de acuerdo a la ecuación (2).
7: Selección: Actualizar al vector elite ($E_{i,G}$ en lugar de $X_{i,G}$) de acuerdo a la ecuación (3).
8: Reemplazo: Seleccionar a los vectores objetivo (X_{G+1}) de acuerdo a la ecuación (2).
9: $G = G + 1$

de miembros candidatos continen vectores que podrían ser seleccionados para sobrevivir. Entonces, el conjunto de individuos sobrevivientes y penalizados son inicializados por el conjunto vacío (línea 4). En orden para seleccionar a los NP sobreviviente (población padre de la siguiente generación) se repite un procedo iterativo (líneas 5 - 13). En cada paso es seleccionado el mejor individuo para sobrevivir del *Conjunto de Candidatos*, es decir al individuo que tiene la mejor aptitud, entonces es movido al *Conjunto de Sobrevivientes*. Entonces, los individuos que pertenecen al *Conjunto de Candidatos* cuya métrica de mínima distancia sea menor que D_t son transferidos *Conjunto de Penalizados* (línea 9).

La forma para calcular la distancia entre dos indivíduos es en base a la distancia Euclideana normalizada descrita en la ecuación 4, donde D es la dimensión del problema, y a_d, b_d son los límites menores y mayores de cada dimensión (d). En los casos donde *El conjunto de Candidatos* está vacío previamente a la selcción de los NP individuos, el *Conjunto de Sobrevivientes* se llena seleccionando en cada iteración al individuo *Penalizado* con la mayor distancia al individuo más cercano al *Conjunto de Sobrevivientes* (líneas 10 - 13).

$$distance(x_i, x_j) = \frac{\sqrt{\sum_{d=1}^{D} \left(\frac{x_i^d - x_j^d}{b_d - a_d}\right)^2}}{\sqrt{D}} \tag{4}$$

En este orden con el propósito de completar la descripción es importante especificar la forma en que se calcula D_t y el para actualizar a los individuos elite. El resto del algortimo es mantenido de igual forma que la clásica variante de DE. El valor de D_t es utilizado para alterar el grado entre exploración y explotación, por lo tanto éste parámetro debería depender en la etapa de optimización. Específicamente, este valor debería se reducido conforme se alcanza el criterio de paro con el objetivo de promover un grado de intensificación. En nuestro esquema se requiere asignar un valor inicial para D_t (D_I). Así, similarmente que en [15], se calcula una reducción lineal de D_t considerando las evoluciones a función y el criterio de paro, Particularmente, en este trabajo, el criterio de paro es asigna en base a las evaluaciones a función. La reducción es calculada de tal forma que en el 95 % del máximo número de evaluaciones

Algorithm 2 Fase de Reemplazo

1: Entrada: *Población* (*Vectores Objetivo*), *Hijos* (*Vectores de prueba*), y *Elite*
2: Actulizar $D_t = D_I - D_I * (nfes/(0{,}95 * max\_nfes))$
3: *Candidatos* = *Población* ∪ *Hijos* ∪ *Elite*.
4: *Sobrevivientes* = *Penalizados* ∅.
5: **while** *Sobrevivientes* $< NP$ y $|Candidatos| > 0$ **do**
6: *Seleccionados* = Seleccionar al mejor individuo de *Candidatos*.
7: Eliminar *Seleccionado* de *Candidatos*.
8: Copias *Seleccionado* a *Sobrevivientes*.
9: Encontrar los individuos de *Candidatos* cuya distancia a *Seleccionados* sea menor que D_t y moverlos al *Penalizados*. En esta parte se considera la distancia normalizada (Ecuación 4).
10: **while** $|Sobrevivientes| < NP$ **do**
11: *Seleccionado* = Seleccionar al individuo de *Penalizados* con la mayor distancia al individuo mas cercano a *Sobrevivientes*.
12: Eliminar *Seleccionado* de *Penalizados*.
13: Copiar *Seleccionado* a *Sobrevivientes*
14: **return** *Survivors*

a función el valor de D_t es 0. Por lo tanto, la diversidad no es considerada del todo en el restante 5 %. Entonces, si max_nfes es el máximo número de evaluaciones y $nfes$ es el número de evaluciones trascurridas $nfes$, entonces D_t puede ser calculado de la siguiente forma $D_t = D_I - D_I * (nfes/(0{,}95 * max\_nfes))$.

La distancia inicial (D_I) afecta de forma considerable al rendimiento del DE-EDM. Si este parámetro es elevado, entonces el algoritmo tiene como objetivo maximizar la diversidad de la población en las primeras etapas de optimización, por lo tanto se genera una exploración adecuada la cual es muy importante, particularmente en varios tipos de problemas tales como altamente multi-modales y deceptivos. Entonces, se podría aliviar el efecto de la convergencia prematura. Un valor muy elevado de D_I podría inducir exploración excesivamente y por lo tanto una fase de intensificación podría no ser no es efectuada. Por otra parte, un valor muy pequeño de D_I podría evitar la fase de exploración, por lo tanto será más difícil evitar óptimos locales. El optimo D_I podría variar dependiendo en el tipo problema y el criterio de paro. En su lugar, los problemas deceptivos y altamente multi-modales usualmente requieren valores más elevados que en los problemas unimodales. Sin embargo, en nuestra propuesta no se adapta un valor D_I para cada problema, por lo tanto con el propósito de analizar la estabilidad de este parámetro se realiza un análisis con diferentes valores de D_I en la sección de validación experimental.

Al igual que a la versión estándar DE, en nuestra propuesta DE-EDM se debe asignar una probabilidad de cruce (CR) y un factor de mutación (F).

El primero es quizás es más importante de acuerdo a varios estudios desarrollados por Montgomery y otros [16]. Estos autores probaron de forma empírica que valores extremos de CR resultan en un comportamiento muy distinto. Ellos explicaron que bajos valores de CR resultan en una búsqueda que es alineada con un pequeño número de ejes y además induce pequeños desplazamientos. Esto provoca una convergencia lenta y gradual que en algunos escenarios podría provocar un comportamiento robusto. Adicionalmente, valores elevados de CR podrían generar soluciones de mayor calidad con una menor probabilidad. Sin embargo, estas transformaciones provocan largos desplaza-

mientos que podrían mejorar de forma significativa. De acuerdo a esto, en nuestra propuesta se emplean los dos principios, es decir, valores elevados y pequeños de CR como es mostrado en la ecuación 5.

$$CR = \begin{cases} Normal(0{,}2, 0{,}1), & \text{si} \quad rand[0,1] \leq 0{,}5 \\ Normal(0{,}9, 0{,}1), & \text{de otra forma} \end{cases} \tag{5}$$

Siguiendo los principios de distintas variantes del SHADE [17, 18], se consideran las evaluaciones a función en el proceso de generación aleatorio del factor de mutación F. Particularmente, cada valor F es una muestra de una distribución Cauchy (Ecuación6).

$$Cauchy(0{,}5, 0{,}5 * nfes/max\_nfes) \tag{6}$$

Por lo tanto, en las primeras etapas de optimización los valores de F son generados de forma cercana a 0,5. Conforme la execución transcurre, la función de densidad sufre una transformación gradual donde la varianza se incremente, esto implica que son generados valores fuera del intérvalo [0,0, 1,0] con una probabilidad alta. En los casos cuando los valores son mayores a 1,0, es utilizado un valor de 1,0. Si se genera un valor negativo, entonces este valor se vuelve a generar. Uno de los efectos de este enfoque es incrementar la probabilidad de generar valores elevados de F conforme transcurren las generaciones con el objetivo de evitar una convergencia en las últimas etapas de optimización.

2.4. Resultados

En esta sección se presenta la validación experimental. Especialmente, demostramos que los resultados de los algoritmos que pertenecen al estado-del-arte pueden ser mejorados controlando explícitamente la diversidad en el clásico DE. Particularmente, se consideradon los conjuntos de prueba del CEC 2016 y CEC 2017. Cada uno está compuesto de treinta distintos problemas. El estado-del-arte está compuesto por los algoritmos que alcanzaros los primeros lugares en cada año. Adicionalmente, se incluyó la versión estándar DE. Por lo tanto, los algoritmos considerados del CEC 2016 son el UMOEAs-II [19] y L-SHADE-EpSin [17], los cuales alcanzaron el primero y el segundo lugar respectivamente. Similarmente, los mejores algoritmos del CEC 2017 son el EBOwithCMAR [20] y el jSO [21].

Es imporante destacar que el EBOwithCMAR es considerado como una mejora del UMOEAs-II. Adicionalmente, el jSO y el L-SHADE-EpSin pertenecen al a la familia de SHADE. Todos estos algoritmos fueron probados con los dos conjuntos de prueba como es sugeriodo en [22]. Debido a que todos los algoritmos son estocásticos se realizaron 51 ejecuciones con distintas semillas.

En cada caso, el criterio de paso fue asignado a $25,000,000$ evaluaciones a función. La evaluación de los algoritmos se realizó siguiendo los lineamientos de las competencias del CEC. Entonces, se asignó un error de 0 si la diferencia entre la mejor solución encontrada y la solución óptima era menor que 10^{-8}.

Para cada algoritmo Se utilizó la parametrización indicada por los respectivos autores, que son definidos a continuación:

- **EBOwithCMAR**: Para la parte EBO, el tamaño máximo de la población de $S_1 = 18D$, el tamaño mínimo de la población de $S_1 = 4$, el tamaño máximo de la población de $S_2 = 146{,}8D$, el tamaño mínimo de la población de $S_2 = 10$, el tamaño de la memoria histórica H=6. Para la parte de CMAR el tamaño de la población $S_3 = 4 + 3log(D)$, $\sigma = 0{,}3$, CS = 50, la probabilidad de búsqueda local $pl = 0{,}1$ y $cfe_{ls} = 0{,}4 * FE_{max}$.
- **UMOEAs-II**: Para la parte de MODE, el tamaño máxio de la población de $S_1 = 18D$, el tamaño mínimo de la població de $S_1 = 4$, el tamaño de la memoria histórica H=6. Para la parte del CMA-ES el tamaño de la población $S_2 = 4 + \lfloor 3log(D) \rfloor$, $\mu = \frac{PS}{2}$, $\sigma = 0{,}3$, CS = 50. Para la búsqueda local, $cfe_{ls} = 0{,}2 * FE_{max}$.
- **jSO**: El tamaño máximo de la población $= 25log(D)\sqrt{D}$, el tamaño de la memoria histórica H= 5, valor de mutación inicial de la memoria $M_F = 0{,}5$, probabilidad inicial de la memoria $M_{CR} = 0{,}8$, tamaño mínimo de la población = 4, valor inicial p-best $= 0{,}25 * N$, valor final p-best = 2.
- **L-SHADE-EpSin**: Tamaño máximo de la población $= 25log(D)\sqrt{D}$, tamaño de la memoria histórica H= 5, valor de la mutación inicial de la memoria $M_F = 0{,}5$, probabilidad inicial de la memoria $M_{CR} = 0{,}5$, frecuencia inicial de la memoria $\mu_F = 0{,}5$, tamaño mínimo de la población = 4, valor inicial p-best $= 0{,}25 * N$, valor final p-best = 2, generaciones de la búsqueda local $G_{LS} = 250$.
- **DE-EDM**: $D_I = 0{,}3$, tamaño de la población = 250.
- **Standard-DE**: tamaño de la población = 250 (mismos operadores que en DE-EDM).

Nuestro análisis experimental se desarrolló en base a la diferencia entre la solución óptima y la mejor solución obtenida. En orden para comparar los resultados estadísticamente, se siguió un procedimiento similar que el propuesto en [23].

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 Guassianas, 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 las tablas 1 y 2 se presenta un resúmen de los resultados obtenidos para el CEC 2016 y el CEC 2017 respectivamente. La columna etiquetada con "Simpre Resuelto" muestra el número de funciones en que se obtuvo un error de cero en las 51 ejecuciones. Adicionalmente, la columna etiquetada con "Al menos una vez resuelto" muestra el número de soluciones que se resolvieron

en al menos una ejecución. Practicamente nuestra propuesta resolvió al menos una vez todas las funciones (28 funciones) que pertenecen al conjunto de problemas del CEC 2017. Adicionalmente, fueron resueltas al menos una vez 21 funciones que pertenecen al CEC 2016. Esta es una diferencia sustancial con los resultados obtenidos por los algoritmos que pertenecen al el estado-del-arte. Estos algoritmos obtuvieron los valores óptimos significativamente en menos funciones. En orden para confirmar la superioridad del DE-EDM, se implementaron las pruebas estadísticas por pares. La columna etiquetada con el símbolo $\uparrow$ muestra que el número de veces en que cada método fue superiror, mientras que la columna etiquetada con $\downarrow$ cuenta el número de casos donde el método fue inferior. Finalmente, la columna etiquetada con $\longleftrightarrow$ muestra el número de comparaciones cuyas diferencias no fueron significativas. Las pruebas estadísticas indican que el DE-EDM alcanzó los mejores resultados en los dos años. De hecho el número de veces en que nuestra propuesta ganó en el CEC 2016 y el CEC 2017 fue de 77 y 88 respectivemente. Además el número de veces en que perdió fueron de 25 y 6 respectivamente. Adicionalmente, el último lugar alcanzado en los dos años fue por el algoritmo L-SHADE-Epsilon con 20 comparaciones positivas en el 2016 y 7 comparaciones positivas en el 2017. La última columna etiquetada con "Puntaje" muestra un análisis que fue propuesto en las competencias del CEC. Particularmente, este método de evaluación combina dos puntajes como se indica en la ecuación (7). Por lo tanto el puntaje final está compuesto por la suma $Score = Score_1 + Score_2$.

$$\begin{aligned} Score_1 &= \left(1 - \frac{SE - SE_{min}}{SE}\right) \times 50, \\ Score_2 &= \left(1 - \frac{SR - SR_{min}}{SR}\right) \times 50, \end{aligned} \tag{7}$$

donde, SE_{min} es la suma mínima de errres entre todos los algoritmos, y SE es la suma de errores dado un algoritmo $SE = \sum_{i=1}^{30} error\_f_i$. Similarmente, SR_{min} es la suma mínima de los rangos entre todos los algoritmos, especificamente es la suma de cada rango en cada función para los algoritmos considerados $SE = \sum_{i=1}^{30} error\_f_i$. Principalmente, nuestra propuesta alcanzó los mejores puntajes (100,00) en los dos años, demostrando su superioridad. Adicionalmente, la versión estándar de DE alcanzó resultados suficientemente buenos, de hecho obtuvo el tercer y el segundo lugar en los años 2016 y 2017 respectivamente. Esto muestra que el rendimiento de los algortimos en el-estado-del-arte es distinto al considerar ejecuciones a largo plazo. El algoritmo L-SHADE-Epsilon obtuvo un puntaje competitivo a pesar de que en el CEC del 2017 alcanzó el menor número de comparaciones positivas dentro de las pruebas estadísticas. Esto podría ocurrir desde que los puntajes estadísticos consideran la media y mediana de los errores. Es más el puntaje considera el rango y la media basado en el error.

Dado que nuestra propuesta está basada en el control explícito de la diversidad y con el objetivo de entender mejor su comportamiento, en la figura 1 se muestra la evolución de la diversidad a través de las evaluaciones a función.

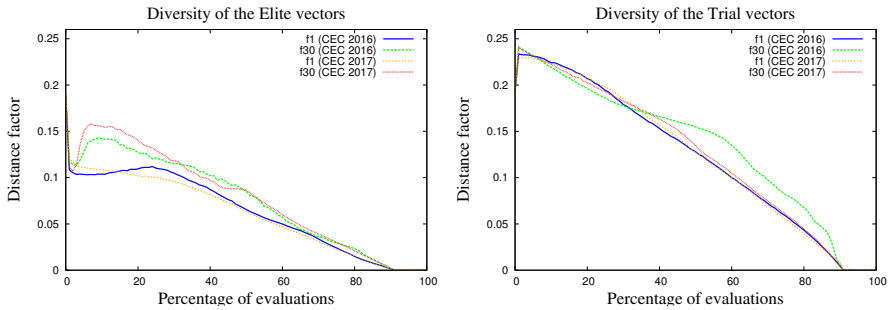

Figura 1 Promedio del DCN de las 51 ejecuciones con los problemas f_1 y f_{30} (CEC 2016 y CEC 2017). El factor de distancia inicial corresponde a $D_I = 0{,}3$.

Cuadro 1 Resúmen de los resultados - CEC 2016

Algorithm	Siempre Resuelto	Resuelto al menos una vez	Pruebas Estadísticas			Puntaje
			↑	↓	⟷	
EBOwithCMAR	8	14	35	56	59	50.28
jSO	9	17	47	51	52	55.43
UMOEAs-II	9	14	51	31	68	62.45
L-SHADE-Epsilon	7	13	20	71	59	50.12
DE-EDM	13	21	77	25	48	100.00
Standard-DE	11	19	50	46	54	56.29

Cuadro 2 Resúmen de los resultados - CEC 2017

Algorithm	Siempre Resuelto	Resuelto al menos una vez	Pruebas Estadísticas			Puntaje
			↑	↓	⟷	
EBOwithCMAR	9	18	34	46	70	37.14
jSO	8	15	29	55	66	29.30
UMOEAs-II	11	15	43	40	67	26.89
L-SHADE-Epsilon	8	19	7	81	62	32.78
DE-EDM	21	28	88	6	56	100.00
Standard-DE	12	21	56	29	65	42.91

Particularmente, se ejecutó el DE-EDM con las funciones F_1 y f_{30}. Basado en sus propiedades la primer función se resuelve de forma sencilla (unimodal) y la segunda función es considerada como una de las más difíciles (híbrida). En la parte izquierda se muestra la diversidad que se mantienen en la población Elite. A pesar de que existen mecanismos para evitar la pérdida de diversidad en la población Elite, se puede observar que se mantiene un grado de diversidad de forma implícito en las dos funciones. Similarmente, la parte derecha corresponde a la diversidad de los vectores de prueba. Esto demuestra que se demuestra un grado de diversidad de forma explícito, es decir hasta el 95 % del total de evaluaciones a función.

En orden, con el fin de proporcionar resultados comparables, en las tablas 3 y 4 se reporta el mejor, peor, mediana, media, desviación estándar y razón de éxito. Particularmente, en estas tablas se observa que nuestra propuesta resuelve a todos los problemas unimodales. Además, varias funciones

Cuadro 3 Resultados del DE-EDM con los problemas del CEC 2016

	Mejor	**Peor**	**Mediana**	**Media**	**Sd**	**Razón de éxito**
f_1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_6	0.00E+00	3.60E-02	4.00E-03	7.39E-03	1.15E-02	3.92E-01
f_7	2.00E-02	1.02E-01	5.90E-02	5.77E-02	4.93E-02	0.00E+00
f_8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{10}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{11}	0.00E+00	6.00E-02	0.00E+00	5.88E-03	1.90E-02	9.02E-01
f_{12}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{13}	1.00E-02	8.00E-02	5.00E-02	4.67E-02	2.60E-02	0.00E+00
f_{14}	1.00E-02	5.00E-02	3.00E-02	2.82E-02	2.13E-02	0.00E+00
f_{15}	0.00E+00	4.70E-01	2.20E-01	1.99E-01	1.55E-01	1.96E-02
f_{16}	4.00E-02	1.50E-01	8.00E-02	8.47E-02	4.96E-02	0.00E+00
f_{17}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{18}	0.00E+00	2.00E-02	1.00E-02	7.65E-03	6.32E-03	3.14E-01
f_{19}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{20}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{21}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{22}	0.00E+00	3.00E-02	0.00E+00	3.73E-03	2.76E-02	7.65E-01
f_{23}	0.00E+00	1.00E+02	0.00E+00	2.55E+01	5.10E+01	7.45E-01
f_{24}	0.00E+00	6.90E-01	0.00E+00	2.61E-02	1.33E-01	9.61E-01
f_{25}	1.00E+02	1.00E+02	1.00E+02	1.00E+02	0.00E+00	0.00E+00
f_{26}	8.00E-02	1.00E+02	5.29E+01	5.20E+01	3.19E+01	0.00E+00
f_{27}	2.50E-01	9.10E-01	5.40E-01	5.60E-01	2.92E-01	0.00E+00
f_{28}	0.00E+00	3.57E+02	3.43E+02	2.76E+02	1.60E+02	1.96E-01
f_{29}	1.00E+02	1.00E+02	1.00E+02	1.00E+02	0.00E+00	0.00E+00
f_{30}	1.84E+02	1.84E+02	1.84E+02	1.84E+02	3.25E-02	0.00E+00

multimodales son aproximadas de forma aceptable. Principalmente, nuestra propuesta resolvió y mejoro significativamente varias funciones complejas (por ejemplo funciones computestas), que por otra parte no fueron resueltas por los algoritmos del estado-del-arte.

2.5. Análisis Empirico del factor de distancia inicial

En nuestra propuesta la diversidad es explícitamente promovida a través de varias etapas, y son promovidas por medio del factor de distancia inicial D_I. Por lo tanto, se analiza en detalle el efecto de este parámetro. Paricularmente, se considera la configuración general de la validación experimental. Entonces, se consideraron varios factores de distancia inicial ($D_I = \{0{,}0, 0{,}1, 0{,}2, 0{,}3, 0{,}4, 0{,}5, 0{,}6, 0{,}7, 0{,}8, 0{,}9, 1{,}0, 1{,}1\}$).

En la figura 2 se muestra la razón de éxito promedio vs. el factor de distancia inicial (D_I). Principalmente se puede observar lo siguiente:

- Si la diversidad no es promovida ($D_I = 0{,}0$) entonces el rendimiento del algoritmo está comprometido.
- En este escenario la configuración ideal es de $D_I = 0{,}3$, a pesar de que aún existen soluciones de calidad en el rango $[0{,}1, 0{,}4]$.

Cuadro 4 Resultados del DE-EDM con los problemas del CEC 2017

	Mejor	**Peor**	**Mediana**	**Media**	**Sd**	**Razón de éxito**
f_1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{10}	0.00E+00	1.20E-01	0.00E+00	1.65E-02	3.39E-02	7.45E-01
f_{11}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{12}	0.00E+00	2.20E-01	0.00E+00	6.37E-02	1.76E-01	6.67E-01
f_{13}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{14}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{15}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{16}	0.00E+00	2.10E-01	0.00E+00	2.47E-02	7.27E-02	8.82E-01
f_{17}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{18}	0.00E+00	1.00E-02	0.00E+00	1.96E-03	4.47E-03	8.04E-01
f_{19}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{20}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{21}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{22}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{23}	0.00E+00	3.00E+02	0.00E+00	3.49E+01	1.03E+02	8.82E-01
f_{24}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{25}	0.00E+00	1.00E+02	0.00E+00	3.92E+00	2.00E+01	9.61E-01
f_{26}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{27}	0.00E+00	3.87E+02	3.87E+02	2.05E+02	2.68E+02	1.96E-02
f_{28}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{29}	1.45E+02	2.26E+02	2.18E+02	1.99E+02	4.21E+01	0.00E+00
f_{30}	3.95E+02	3.95E+02	3.95E+02	3.95E+02	2.10E-01	0.00E+00

- Si se incrementa la diversidad inicial entonces se observa un deteriodo en la calidad de las soluciones.

Finalmente, es importante aclarar que en base a varios estudios la calidad de las soluciones es afectado en un menor grado por el tamaño de la población que con el parámetro D_I.

3. Diseño de operadores de cruce basados en diversidad

3.1. Conceptos Básicos

Es uno de los operadores de cruce más utilizados en los MOEAs y es clasificado como centrado en los padres, particularmente tiene la propiedad de poder crear cualquier valor en el espacio de búsqueda combinando dos individuos padres. En el SBX en base a dos individuos padres (p_1, p_2) se generan dos hijos (c_1, c_2) considerando a una distribución de probabilidad. Para tener más control la forma de la distribución de la probabilidad se puede controlar con un índice de distribución η_c que determina la apertura de la distribución, y por tanto, la 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

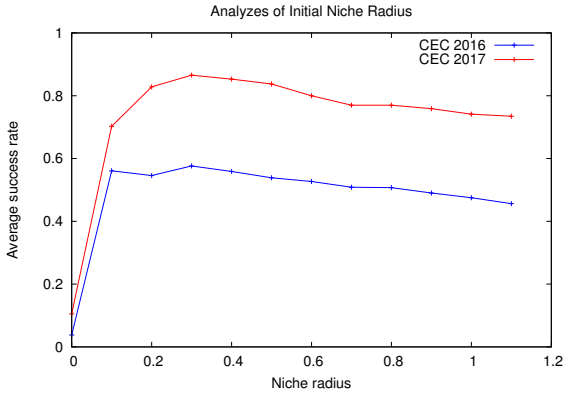

Figura 2 Razón de éxito promedio con distintos factores de distancia inicial con los problemas de prueba del CEC 2016 y CEC 2017, específicamente se considera una población de 250 individuos y $25,000,000$ evaluaciones a función.

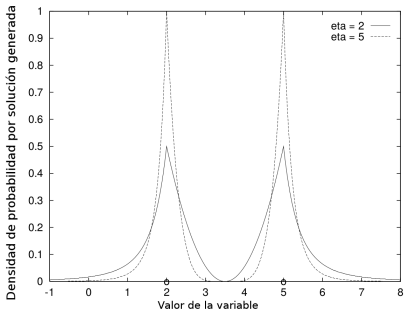

Figura 3 Función de densidad del operador SBX con índices de distribución 2 y 5.

probabilidad de crear soluciones hijas más similares a los padres se incrementa. El efecto de η_c es ilustrado en la Figura **??**, donde se muestran las funciones de densidad de dos índices de distribución distintos. Específicamente los círculos representan a los padres, y se puede apreciar que con $\eta_c = 5$, la probabilidad de crear soluciones más cercanas a los padres es mayor que con $\eta_c = 2$.

El SBX tiene la propiedad de preservar una relación entre la media de los valores padres e hijos ($c_1 + c_2 = p_1 + p_2$), con lo que se define el factor de dispersión $\beta = |c_1 - c_2|/|p_1 - p_2|$, y en base al valor de β, podemos determinar donde quedarían localizados los hijos. En el SBX la probabilidad de distribución para $\beta \in [0, \infty]$ viene dada por:

$$P(\beta) = \begin{cases} 0{,}5(\eta_c + 1)\beta^{\eta_c}, & \text{si} \quad \beta \leq 1 \\ 0{,}5(\eta_c + 1)\frac{1}{\beta^{\eta_c+2}}, & \text{de otra forma} \end{cases} \tag{8}$$

En base a esto, se cumplen las siguientes propiedades:

- 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.

A la hora de implementar esta distribución para dos valores de los individuos padres (p_1, p_2), se crean los valores de los hijos (c_1, c_2) como combinación lineal de los valores padres con un número aleatorio $u \in [0, 1]$ de la forma ([**?**]):

$$\begin{aligned} c_1 &= 0{,}5(1 + \beta(u))p_1 + 0{,}5(1 - \beta(u))p_2 \\ c_2 &= 0{,}5(1 - \beta(u))p_1 + 0{,}5(1 + \beta(u))p_2 \end{aligned} \tag{9}$$

Así, para realiza la simulación del parámetro $\beta(u)$, primero se genera un número aleatorio $u \in [0, 1]$, y se utiliza en la siguiente fórmula:

$$\beta(u) = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{si} \quad u \leq 0{,}5, \\ (\frac{1}{2(1-u)})^{\frac{1}{\eta+1}}, & \text{de otra forma} \end{cases} \tag{10}$$

Se puede observar que la forma de la distribución no está afectada por los límites de cada variable, en base a este inconveniente, los autores [**?**] modificaron la ecuación (**??**) de forma que exista una probabilidad cero de crear individuos fuera del dominio de la variable, por lo tanto se puede calcular $\beta(u, a)$ en función de un número aleatorio $u \in [0, 1]$ de la forma:

$$\beta(u, a) = \begin{cases} (2u(1 - \rho_a))^{\frac{1}{\eta_c+1}}, & \text{si} \quad u \leq 0{,}5/(1 - \rho_a), \\ (\frac{1}{2(1-u(1-\rho_a))})^{\frac{1}{\eta+1}}, & \text{de otra forma} \end{cases} \tag{11}$$

donde $a = x_i^{(L)}$ es el límite inferior y $b = x_i^{(U)}$ es el límite superior, y $\rho_a = 1/(2\beta_a^{\eta_c+1})$, donde $\beta_a = 1 + (p_1 - a)/(p_2 - p_1)$. Similarmente, ρ_b es calculado remplazando β_a por $\beta_b = 1 + (b - p_2)/(p_2 - p_1)$, así $\beta(u, b)$ es calculado con la ecuación (**??**), generando así dos individuos hijo en base a sus respectivas funciones de distribución $\beta(u, a)$ y $\beta(u, b)$.

En la práctica, se trabajan con funciones de múltiples variables, con lo que hay que generalizar lo anterior. Una forma sencilla sería aplicar lo mismo a cada variable, pero se ha visto que eso es muy disruptivo, con lo que habitualmente no se cambian todas las variables. Por ejemplo, en la implementación que más se usa en la práctica (jMETAL, NSGA-II, etc.) cuando se aplica el SBX cada variable sufre el proceso de cruce con una probabilidad igual a 0.5, mientras que en el resto de casos los valores se heredaran sin ser alterados ([**?**, **?**]). Además, cuando se produce un cruce también aplican intercambios entre las valores generadas con probabilidad 0.5, lo que resulta en reflexiones que son analizados posteriormente. Por último, cabe destacar que no se ha podido encontrar variantes del operador SBX en las que en lugar de usar dichas probabilidades fijas en 0.5, estas sean cambiadas a lo largo de la ejecución.

Algorithm 3 Operador de Cruce Binario Simulado (SBX)

1: Entrada: Individuos Padre (P_1, P_2), Indice de distribución (η_c), Probabilidad de cruza (P_c).
2: Salida: Individuos hijo(C_1, C_2).
3: $r_1 \leftarrow U[0,1]$.
4: **if** $r_1 \leq P_c$ **then**
5: **for** cada variable d **do**
6: **if** $U[0,1] \leq 0{,}5$ **then**
7: $a = LowBound(d)$.
8: $b = UpperBound(d)$.
9: $r_2 \leftarrow U[0,1]$.
10: $\beta_a = 1 + (p_1 - a)/(p_2 - p_1)$.
11: $\rho_a = 1/(2\beta_a^{\eta_c+1})$.
12: Utilizar r_2 y ρ_a en la ecuación **??** para generar $\beta(u,a)$.
13: Generar a $C_1(d)$ utilizando $\beta(u,a)$ en la ecuación **??**.
14: $\beta_b = 1 + (b - p_2)/(p_2 - p_1)$.
15: $\rho_b = 1/(2\beta_b^{\eta_c+1})$.
16: Utilizar r_2 y ρ_b en la ecuación **??** para generar $\beta(u,b)$.
17: Generar a $C_2(d)$ utilizando $\beta(u,b)$ en la ecuación **??**.
18: **if** $U[0,1] \leq 0{,}5$ **then**
19: Intercambiar los valores de $C_1(d)$ con $C_2(d)$.
20: **else**
21: $C_1(d) = P_1(d)$.
22: $C_2(d) = P_2(d)$.
23: **else**
24: $C_1 = P_1$
25: $C_2 = P_2$

3.2. Análisis operador de cruce SBX

La implementación más utilizada del operador SBX se encuentra integrada en el NSGA-II publicada en ([**?**]). Este procedimiento está descrito en el algoritmo **??**, donde destacan dos aspectos clave. El primero está relacionado con la similitud entre los individuos padre y los hijos (líneas 22 y 23). En dicha implementación los valores de las soluciones hijas son heredadas 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 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 en las líneas 18 - 20. 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 **??** 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 es 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 $k^d+(1-k)^d$, donde k es la probabilidad de realizar un intercambio, con lo que se produce un decremento 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 cruzar exclusivamente a soluciones similares. Bajo esta 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 el MOEA/D, que incorpora restricciones de apareamiento, ha sido capaz de resolver muchos problemas de forma exitosa. En este trabajo, 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.

This section is devoted to review some of the most important works that are highly related to the research presented in this paper. First, the most important paradigms are defined. Thereafter, some relevant classifications of crossover operators are introduced. Finally, the popular operator, which is used extensively in this paper, is discussed.

3.3. Multi-objective Evolutionary Algorithms

In the last years, a large number of following different design principles have been devised. In order to better classify them, several taxonomies have been proposed [**?**]. Attending to the principles of design, can be based on Pareto dominance, indicators and/or decomposition [**?**]. All of them have quite competitive representatives, so in this paper belonging to the different groups are taken into account. Particularly, the experimental validation has been carried out by including the Non-Dominated Sorting Genetic Algorithm (NSGA-II) [**?**], the MOEA based on Decomposition () [**?**], and the S-Metric Selection Evolutionary Multi-objective Optimization Algorithm () [**?**]. They are representative methods of the domination-based, decomposition-based and indicator-based paradigms, respectively. The following subsections briefly describe each one of these paradigms and introduce the selected methods.

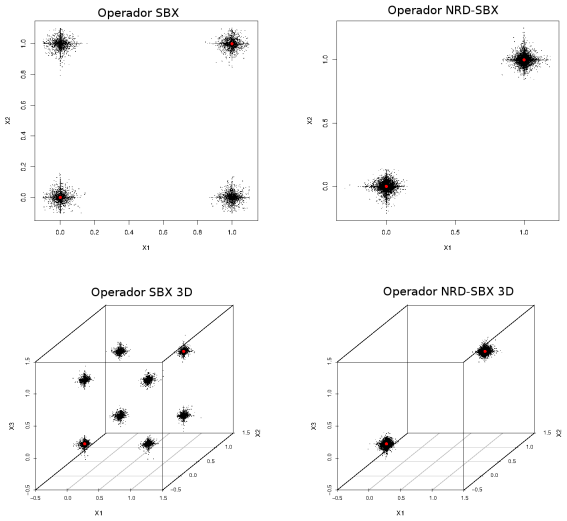

Figura 4 En la columna izquierda se presenta la simulación del operador SBX y en la columna de la derecha se presenta el operador NRD-SBX, ambas distribuciones con un índice de distribución de 20.

3.3.1. Domination Based MOEAs - NSGA-II

One of the most recognized paradigms is the domination based approach. belonging to this category are based on the application of the dominance relation to design different components of the EAs, specially the selection phase. Given that the dominance relation does not inherently promotes diversity in the objective space, auxiliary techniques such as niching, crowding and/or clustering are usually integrated to obtain an acceptable spread and diversity in the objective space. A critical drawback of methods based on the dominance relation is its scalability in terms of the dimensionality of the objective space. In fact the selection pressure is substantially reduced as the number of objectives increases. Although some strategies have been developed to deal with this issue [**?**] it remains as an important drawback for this kind of algorithms.

One of the most popular techniques of this group is the . This algorithm [**?**] considers a special selection operator based on non-dominated sorting and crowding. Non-dominated sorting is used to provide convergence to the Pareto front whereas crowding promotes the preservation of diversity in the objective space.

3.3.2. Decomposition Based MOEAs - MOEA/D

Decomposition-based [**?**] transform a in a set of single-objective optimization problems that are tackled simultaneously. This transformation can be achieved through several approaches. The most popular of them is applying a

weighted Tchebycheff function, thus requiring a set of well distributed weights to attain well-spread solutions. An important drawback of this kind of approaches is related to the dependency between the Pareto front geometry and the weights required to attain proper solutions.

[?] is a recently designed decomposition-based . It includes several features such as problem decomposition, weighted aggregation of objectives and mating restrictions based on neighborhood definitions. Particularly, the neighborhoods are considered in the mating selection. A popular variant of is the , which uses the DE operators [?] and the polynomial mutation operator [?] in the reproduction phase. Additionally, it has two extra mechanisms for maintaining the population diversity [?].

3.3.3. Indicator Based MOEAs - SMS-EMOA

In multi-objective optimization several quality indicators have been developed to compare the performance of MOEAs. Since these indicators measure the quality of the approximations attained by , a paradigm based on the application of these indicators was proposed. Particularly, the indicators replace the Pareto dominance relation with the aim of guiding the optimization process. Among the different indicators, hypervolume is a widely accepted Pareto-compliance quality indicator [?]. One of the main advantages of these algorithms is that indicators usually take into account both the quality and diversity of the solutions, so no additional mechanisms to preserve diversity are required.

A popular and extensively used indicator-based algorithm is the [?]. This algorithm might be considered as hybrid, since it involves both indicators and Pareto dominance concepts. Essentially, it integrates the non-dominated sorting method with the use of the hypervolume metric. Thus, uses the hypervolume as a density estimator which results in a computationally expensive task. Particularly, the replacement phase erases the individual of the worst ranked front with the minimum contribution to the hypervolume. Taking into account the promising behavior of , it has been used in our experimental validation.

3.4. Crossover operators

The crossover operators are designed to generate offspring solutions using information of the parent solutions. They combine features of two or more parent solutions to generate new candidate solutions. Since several crossover operators have been proposed, some taxonomies have also been provided. The taxonomies are based on features such as the location of new generated solutions or the kinds of relations among the variables.

A popular taxonomy classifies crossover operators into variable-wise operators and vector-wise operators. In the variable wise category, each variable from parent solutions is recombined independently with a certain pre-specified

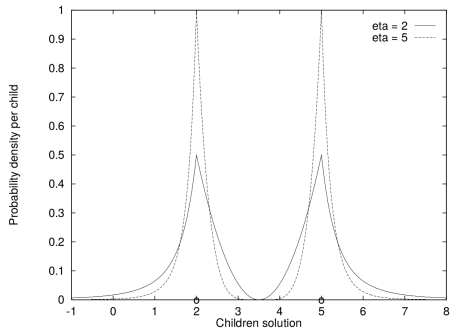

Figura 5 Probability density function of the operator with indexes of distribution 2 and 5. The parents are located in 2 and 5 respectively.

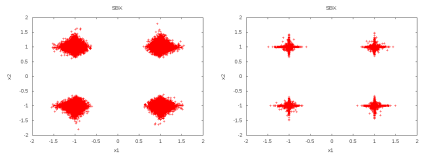

Figura 6 Simulations of the operator with a distribution index equal to 20. Parents are located in $P_1 = (-1{,}0, -1{,}0)$ and $P_2 = (1{,}0, 1{,}0)$. The left simulation corresponds to a probability of altering a variable (δ_1 in Algorithm **??**) equal to 1,0 and in the right it corresponds to 0,1.

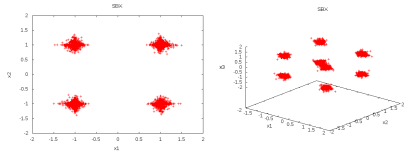

Figura 7 Simulations of the operator with a distribution index equal to 20. Parents are located in $P_1 = (-1{,}0, -1{,}0)$ and $P_2 = (1{,}0, 1{,}0)$ and $P_1 = (-1{,}0, -1{,}0, -1{,}0)$ and $P_2 = (1{,}0, 1{,}0, 1{,}0)$ for two and three variables respectively.

probability to create new values. These operators are specially suitable to deal with separable problems. Some operators belonging to this category are the Blend Crossover () [**?**], and the [**?**]. Alternatively, the vector-wise recombination operators are designed to take into account the linkage among variables. They usually perform a linear combination of the variable vectors. Some operators belonging to this category are the Unimodal Normally Distributed Crossover () [**?**], and the simplex crossover () [**?**]. Additionally, crossover operators can be classified as Parent-Centric and Mean-Centric [**?**]. In Parent-Centric operators, children solutions are created around one of the parent solutions, whereas in Mean-Centric operators, children solutions tend to be created mostly around the mean of the participating parent solutions. Among

the crossover operators, is probably the most frequently used operator, so this research focuses on this crossover.

3.4.1. Simulated Binary Crossover - SBX

The reproduction operators are one of the most relevant components that influence the search process of EAs. Specifically, the crossover and mutation operators are highly related with the diversity of solutions. Hence, the quality of solutions are highly affected by the applied operators.

Simulated Binary Crossover () [**?**] is probably the most popular operator for continuous domains and most have been extensively tested with such an operator [**?**,**?**]. is classified as Parent-Centric, meaning that two children values (c_1 and c_2) are created around the parent values (p_1 and p_2). The process of generating the children values is based on a probability distribution. This distribution controls the spread factor $\beta = |c_1 - c_2|/|p_1 - p_2|$ defined as the ratio between the spread of the children values and parent values. In order to define this density function a distribution index η_c (a user-defined control parameter) alters the exploration capability of the operator. Specifically, a small index induces a larger probability of building children values distant to the parent values, whereas high indexes tend to create solutions very similar to the parents as is shown in Figure **??**.

The probability distribution to create an offspring value is defined as a function of $\beta \in [0, \infty]$ as follows:

$$P(\beta) = \begin{cases} 0{,}5(\eta_c + 1)\beta^{\eta_c}, & \text{if} \quad \beta \leq 1 \\ 0{,}5(\eta_c + 1)\frac{1}{\beta^{\eta_c+2}}, & \text{otherwise} \end{cases} \tag{12}$$

Based in the mean-preserving property of children values and parent values, has the following properties:

- Both offspring values are equi-distant from parent values.
- There exist a non-zero probability to create offspring solutions in the entire feasible space from any two parent values.
- The overall probability of creating a pair of offspring values within the range of parent values is identical to the overall probability of creating two offspring values outside the range of parent values.

Therefore, considering two participating parent values (p_1 and p_2), two offspring values (c_1 and c_2) can be created as linear combination of parent values with a uniform random number $u \in [0, 1]$, as follows:

$$\begin{aligned} c_1 &= 0{,}5(1 + \beta(u))p_1 + 0{,}5(1 - \beta(u))p_2 \\ c_2 &= 0{,}5(1 - \beta(u))p_1 + 0{,}5(1 + \beta(u))p_2 \end{aligned} \tag{13}$$

The parameter $\beta(u)$ depends on the random number u, as follows:

$$\beta(u) = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{if} \quad u \leq 0{,}5, \\ (\frac{1}{2(1-u)})^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \tag{14}$$

Algorithm 4 Simulated Binary Crossover ()

```
1: Input: Parents (P_1, P_2), Distribution index (η_c), Crossover probability (P_c).
2: Output: Children (C_1, C_2).
3: if U[0,1] ≤ P_c then
4:    for each variable d do
5:       if U[0,1] ≤ δ_1 then
6:          Generate C_{1,d} with Equations (??) and (??).
7:          Generate C_{2,d} with Equations (??) and (??).
8:          if U[0,1] ≤ (1 − δ_2) then
9:             Swap C_{1,d} with C_{2,d}.
10:      else
11:         C_{1,d} = P_{1,d}.
12:         C_{2,d} = P_{2,d}.
13: else
14:    C_1 = P_1.
15:    C_2 = P_2.
```

The above equation considers an optimization problem with no variable bounds. In most practical problems, each variable is bounded within a lower and upper bound. Thus, the modification of the probability distribution shown in Equation (**??**) was proposed [**?**] with the aim of taking into account such bounds. This last variant is extensively used nowadays.

$$\beta(u) = \begin{cases} (2u(1-\gamma))^{\frac{1}{\eta_c+1}}, & \text{if} \quad u \leq 0{,}5/(1-\gamma), \\ (\frac{1}{2(1-u(1-\gamma))})^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \tag{15}$$

$$c_1 = 0{,}5(1+\beta(u))p_1 + 0{,}5(1-\beta(u))p_2 \tag{16}$$

$$c_2 = 0{,}5(1+\beta(u))p_1 + 0{,}5(1-\beta(u))p_2 \tag{17}$$

In this case, the child c_1 which is nearest to p_1 is calculated according to the Equation (**??**). Considering that $p_1 < p_2$ and with a lower bound equal to a, $\gamma = 1/(\alpha^{\eta_c+1})$, where $\alpha = 1 + (p_1 - a)/(p_2 - p_1)$. Similarly, the second child c_2 is computed with $\alpha = 1 + (b - p_2)/(p_2 - p_1)$, where b correspond to the upper bound. Then, the second child is computed as is indicated in Equation (**??**).

Note that as reported in [**?**] several extensions of the to problems with multiple variables might be provided. Authors considered a simple strategy for choosing the variables to cross [**?**]. Specifically, each variable is crossed with probability 0.5, following the principles of uniform crossover. Authors recognized the important implications on the linkage among variables of such decisions. In any case, this is the most typical way of applying in problems with multiple variables nowadays.

3.4.2. Implementation and analyses of SBX operator

This section discusses some of the main characteristics of the most currently used implementation of the operator for problems with multiple variables. Essentially, three key components that might affect its performance are discussed. Firstly, as already mentioned it alters each variable with a fixed probability equal to 0,5. If this probability value is increased, the children tend

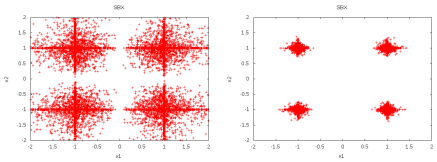

Figura 8 Simulation of the operator sampling 10,000 children values, the parents are located in $P_1 = (-1{,}0, -1{,}0)$ and $P_2 = (1{,}0, 1{,}0)$. The left and right are with a distribution index of 2 and 20 respectively.

to be more distant to the parents, since in average more variables are modified simultaneously. In separable problems, altering only one variable might be adequate. However, for non-separable problems altering several variables simultaneously seems more promising. The implications of varying this probability is illustrated in Figure **??**, where a problem with two variables is taken into account. In the right side a low probability is used and it provokes a bias to explore by keeping some values intact creating a figure similar to a cross in the two-dimensional case. This feature might be suitable for separable problems. Alternatively, the left side shows that when using a high probability this bias disappears, which could be more suitable for non-separable problems. Note that this probability is somewhat related with the distribution index in the sense that both have a direct effect on the similarity between parents and children.

The second key issue is that after generating the two child values with the distribution, such values are interchanged with a fixed probability that is usually set to 0,5, i.e. the value closer to parent p_1 is not always inherited by c_1. This is a feature that is not usually discussed but it is important for the obtained performance. In some contexts this probability is known as "Variable uniform crossover probability" [**?**] or "Discrete Recombination" [**?**]. Since in multi-objective optimization more diversity is maintained these swaps might produce a high disruptive operator. In fact, in some sense due to this action it is not so clear that can be categorized as a parent-centric operator. These interchanges between the children has the effect of performing multiple "reflections" in the search space. When increasing the dimensions of the decision variables the number of regions covered increases exponentially as is illustrated in Figure **??** where cases with two and three decision variables are taken into account. Note also that this feature has a considerable effect on the distance between parents and offspring.

Finally, the last component is the distribution index, which is probably the most well known feature of the . A low index results in greater exploration levels. In fact, a distribution index equal to one has a similar effect to the Fuzzy Recombination Operator [**?**]. The effect of applying different indexes is illustrated in Figure **??** where the left side considers a low index value whereas the right side takes into account a higher index value, which creates new candidate solutions that are more similar to the parents.

The implementation is shown in Algorithm **??**. This pseudocode is based on the implementation that is integrated in the NSGA-II code published by Deb et al. [**?**] and which is the most popular variant nowadays. As an input it requires two parents (P_1 and P_2) and it creates two children (C_1 and C_2). The first and second key components commented previously correspond to the lines 5 and 8, respectively. As is usual, for the basic case, is configured with $\delta_1 = \delta_2 = 0{,}5$ and $\eta_c = 20$. It is important take into account that this implementation does not consider the dimension of the decision variables or the stopping criteria to set any of its internal parameters.

3.5. Propuesta

Based on the previous analyses and with the aim of inducing an appropriate balance between exploration and intensification, the following modifications are proposed. First, the probability to modify a variable (δ_1) is dynamically modified during the execution. The rationality behind this modification is to increase the exploration capability in the initial stages by altering simultaneously several variables and then, as the evolution proceeds reduce the number of variables that are modified. The value of δ_1 is changed in base of a linear decreasing model, where initially it is fixed to 1,0 and then it is decreased so that at the half of total generations is equal to 0,5. This last value is maintained until the end of the execution, i.e. from the half of the execution it behaves as the traditional implementation. Equation (**??**) is the one used to set the value of δ_1, where $G_{Elapsed}$ is the current generation and G_{End} is the total number of generations.

In a similar way, the second change is related to the probability of performing reflections ($1 - \delta_2$). In this case δ_2 is also updated as in Equation (**??**), meaning that the probability of performing a reflection increases from 0,0 to 0,5 during the execution. This modification is performed with the aim of avoiding the disruptive behavior of interchanging the variables at the first generations because this might result in very drastic modifications. Once that the individuals converge to certain degree it might make more sense to perform such reflections. Thus, this probability is increased to 0,5 which is the value used in the standard implementation of .

$$\delta_1 = \delta_2 = max\left(0{,}5, 1{,}0 - \frac{G_{Elapsed}}{G_{End}}\right) \tag{18}$$

Finally, the distribution index is also changed during the execution. At the first stages a low distribution index is induced with the aim of increasing the exploration capabilities of . Then, it is linearly incremented which has the effect of closing the distribution curve, meaning that more intensification is promoted. The linear increment is governed by Equation (**??**), meaning that the distribution index is altered from 2 to 22. Note that modifications similar to this last one have been explored previously [**?**], [**?**].

Cuadro 5 References points for the HV indicator

Instances	**Reference Point**
WFG1-WFG9	[2,1, ..., $2m + 0$,1]
DTLZ 1, 2, 4	[1,1, ..., 1,1]
DTLZ 3, 5, 6	[3, ..., 3]
DTLZ7	[1,1, ..., 1,1, $2m$]
UF 1-10	[2, ..., 2]

Cuadro 6 Statistical Information of Metrics with two objectives

	NSGA-II						MOEA/D						SMS-EMOA					
	1	2	3	4	5	DE	1	2	3	4	5	DE	1	2	3	4	5	DE
Average HV	0.88	0.90	0.90	0.91	0.93	**0.94**	0.87	0.87	0.87	0.90	**0.91**	**0.91**	0.88	0.89	0.87	0.91	0.92	**0.93**
Average IGD+	0.12	0.09	0.11	0.07	0.06	**0.05**	0.14	0.12	0.14	0.09	0.08	**0.07**	0.13	0.11	0.14	0.08	0.07	**0.05**

Cuadro 7 Statistical Information of Metrics with three objectives

	NSGA-II						MOEA/D						SMS-EMOA					
	1	2	3	4	5	DE	1	2	3	4	5	DE	1	2	3	4	5	DE
Average HV	**0.87**	0.84	**0.87**	**0.87**	**0.87**	0.85	0.84	0.84	0.84	**0.86**	**0.86**	0.85	0.90	0.89	0.88	**0.91**	**0.91**	**0.91**
Average IGD+	0.13	0.16	0.13	**0.12**	**0.12**	0.13	0.15	0.14	0.15	**0.11**	**0.11**	0.13	0.11	0.11	0.13	**0.09**	**0.09**	0.13

Cuadro 8 Summary of Statistical Tests

NSGA-II															
	1			2			3			4			5		
	↑	↓	⟷	↑	↓	⟷	↑	↓	⟷	↑	↓	⟷	↑	↓	⟷
HV-2obj	16	29	47	6	61	25	28	19	45	31	23	38	**54**	3	35
HV-3obj	15	19	42	12	50	14	17	15	44	**33**	10	33	26	9	41
IGD-2obj	14	30	48	4	60	28	25	17	50	33	19	40	**52**	2	38
IGD-3obj	14	18	44	13	44	19	18	15	43	**33**	15	28	23	9	44
MOEA/D															
	1			2			3			4			5		
	↑	↓	⟷	↑	↓	⟷	↑	↓	⟷	↑	↓	⟷	↑	↓	⟷
HV-2obj	15	33	44	10	60	22	25	26	41	39	18	35	**57**	9	26
HV-3obj	10	22	44	12	39	25	11	19	46	24	10	42	**38**	5	33
IGD-2obj	16	31	45	9	60	23	23	27	42	37	17	38	**57**	7	28
IGD-3obj	12	22	42	13	43	20	13	24	39	30	9	37	**40**	10	26
SMS-EMOA															
	1			2			3			4			5		
	↑	↓	⟷	↑	↓	⟷	↑	↓	⟷	↑	↓	⟷	↑	↓	⟷
HV-2obj	9	35	48	7	43	42	16	31	45	41	9	42	**53**	8	31
HV-3obj	7	21	48	9	35	32	13	21	42	27	6	43	**31**	4	41
IGD-2obj	10	34	48	15	48	29	12	33	47	41	12	39	**55**	6	31
IGD-3obj	8	20	48	13	30	33	9	19	48	22	5	49	**27**	5	44

$$\eta_c = 2 + 20 \times \left(\frac{G_{Elapsed}}{G_{End}} \right) \tag{19}$$

3.6. Resultados

This section is devoted to analyze the results obtained with the dynamic variants of (DSBX). The novel crossover operator was integrated with , and . First, three variants that alter only one of each of the components previously

discussed are analyzed. Then, a case that alters two of them simultaneously is taken into account. The WFG [**?**], DTLZ [**?**] and UF [**?**] test problems have been used for our purpose. Our experimental validation also includes the variant of Differential Evolution known as DEMO [**?**] with the aim of comparing our extension of with other well-known operators.

Given that all the methods are stochastic algorithms, each execution was repeated 35 times with different seeds. The common configuration in all of them was the following: the stopping criterion was set to $25,000$ generations, the population size was fixed to 100, WFG test problems were configured with two and three objectives, and 24 variables were considered, where 20 of them are distance parameters and 4 of them are position parameters. In the case of the DTLZ test instances, the number of decision variables were set to $n = M+r-1$, where $r = \{5, 10, 20\}$ for DTLZ1, DTLZ2 to DTLZ6 and DTLZ7 respectively, as is suggested in [**?**]. In the UF benchmark set the number of decision variables were set to 10. Finally, the polynomial mutation was used with a mutation probability equal to $1/n$ and with a distribution index equal to 50, whereas for the cases that used the , the crossover probability was set to 0,9 and the distribution index was set to 20. The additional parameterization of each algorithm was as follows:

- **DEMO**: CR = 0.3 and F = 0.5.
- **SMS-EMOA**: offset = 100.
- **MOEA/D**: size of neighborhood = 10, max updates by sub-problem (nr) = 2 and $\delta = 0{,}9$.

In order to compare the fronts obtained by the different methods the normalized hypervolume (HV) and IGD+ was taken into account. The reference points used for the hypervolume indicator are shown in the Table **??** and are similar to the ones used in [**?**, **?**].

In order to statistically compare the results (IGD+ and HV values), the following statistical tests were performed. First a Shapiro-Wilk test was performed to check whatever or not the values of the results followed a Gaussian distribution. If, so, the Levene test was used to check for the homogeneity of the variances. If samples had equal variance, an ANOVA test was done; if not, a Welch test was performed. For non-Gaussian distributions, the non-parametric Kruskal-Wallis test was used to test whether samples are drawn from the same distribution. An algorithm X is said to win algorithm Y when the differences between them are statistically significant, and the mean and median obtained by X are higher (in HV) or lower (in IGD+) than the mean and median achieved by Y.

3.7. Analysis of isolated components

In this section we discuss about the independent effect of each component that is dynamically modified. The effect of each component is analyzed through four cases, based in the Algorithm **??**. Each case is described as follows:

- **Case 1**: The standard SBX operator where $\delta_1 = \delta_2 = 0{,}5$ and $\eta_c = 20$.
- **Case 2**: The value δ_1 is updated according to Equation (**??**), $\delta_2 = 0{,}5$ and $\eta_c = 20$.
- **Case 3**: The value δ_2 is updated according to Equation (**??**), $\delta_1 = 0{,}5$ and $\eta_c = 20$.
- **Case 4**: The distribution index is updated according to Equation (**??**), $\delta_1 = \delta_2 = 0{,}5$.

In order to analyze the performance of each Case (Case 5 is discussed later), Tables **??** and **??** shows information about the Normalized Hyper-volume (HV) [**?**] and about the Inverted Generational Distance Plus (IGD+) [**?**]. Specifically, the mean of the HV and IGD+ for all considered problems are shown for two and three objectives. It is clear that case 4 outperforms case 1, case 2 and case 3 both with two and three objectives in all the tested algorithms. Therefore, increasing the distribution index during the execution seems to be the most beneficial action. This occurs because the initially open distribution curve leads to a higher degree of exploration, whereas as the evolution proceeds more intensification is promoted. On the other hand, case 2 presented a lower performance than case 1 when taking into account three objectives. Thus, it seems that altering almost all the variables convert the new approach into a too disruptive operator. Perhaps, altering δ_1 in a different way might provide better results, but this is left as a future work.

Previous analyses are only based on the mean obtained for all the problems. However, depending on the problem the performance might vary. This is analyzed in the following section. Additionally, more detailed results are available[2].

3.8. Simultaneous modification of several components

Based on the previously discussed results, a variant of the is proposed where the case 3 and case 4 are mixed, i.e. both δ_2 and the distribution index are updated dynamically. Since case 2 did not report significant benefits, the updating mechanism for δ_1 was discarded. Specifically in our case 5, Algorithm **??** is configured as follows. The parameter δ_1 is fixed to 0,5, i.e. in a similar way than the standard . Following the case 3, δ_2 is updated according to Equation (**??**). Finally, according to case 4 η_c is updated in base of Equation (**??**).

Attending to the mean HV and IGD+ obtained by the case 5 (see Tables **??** and **??**) it is clear that integrating case 3 and case 4 is beneficial. The advantages are clearer in the case of two objectives, whereas in the case of three objectives, case 4 and case 5 are similar in terms of mean performance. Moreover, results attained with case 5 are superior to the ones obtained with DE in three objectives, whereas when using the traditional results deteriorate.

[2] https://github.com/joelchaconcastillo/SBX_CEC2018.

Thus, when properly configuring a DSBX results similar or superior to DEMO could be obtained.

Finally, since previous analyses only consider the mean among all the benchmark problems, an additional analyses was developed to better understand the contributions of the different cases. Particularly, pair-wise statistical tests among all the five cases that consider and DSBX were carried out. This was performed independently for , and . Results of these statistical tests are shown in Table **??**. For each algorithm and case, the column "↑" reports the number of comparisons where the statistical tests confirmed the superiority of the corresponding case, whereas the column "↓" reports the number of cases where it was inferior and "⟷" indicates the number of comparisons where differences were not statistically significant. The advantages of case 5 are quite clear. Only in the case of with three objectives, case 4 could outperform the results obtained by case 5. Thus, by properly combining several dynamic modification, results can be improved further. Moreover, results confirm the advantages of our proposals when compared to the standard (case 1). The only case that is not clearly superior to the standard is the case number 2, as it was previously discussed.

Referencias

1. U.K. Chakraborty, *Advances in differential evolution*, vol. 143 (Springer, 2008)
2. R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, Journal of global optimization **11**(4), 341 (1997)
3. S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE transactions on evolutionary computation **15**(1), 4 (2011)
4. J.A. Nelder, R. Mead, A simplex method for function minimization, The computer journal **7**(4), 308 (1965)
5. W. Price, Global optimization by controlled random search, Journal of Optimization Theory and Applications **40**(3), 333 (1983)
6. J. Zhang, A.C. Sanderson, Jade: adaptive differential evolution with optional external archive, IEEE Transactions on evolutionary computation **13**(5), 945 (2009)
7. Â.A. Sá, A.O. Andrade, A.B. Soares, S.J. Nasuto, in *AISB 2008 Convention Communication, Interaction and Social Intelligence*, vol. 1 (2008), vol. 1, p. 57
8. J. Lampinen, I. Zelinka, et al., in *Proceedings of MENDEL* (2000), pp. 76–83
9. D. Zaharie, in *Proc. of MENDEL*, vol. 9 (2003), vol. 9, pp. 41–46
10. M. Yang, C. Li, Z. Cai, J. Guan, Differential evolution with auto-enhanced population diversity, IEEE transactions on cybernetics **45**(2), 302 (2015)
11. M. Črepinšek, S.H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, ACM Computing Surveys **45**(3), 35:1 (2013)
12. J. Montgomery, in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on* (IEEE, 2009), pp. 2833–2840
13. J. Montgomery, S. Chen, in *Evolutionary Computation (CEC), 2012 IEEE Congress on* (IEEE, 2012), pp. 1–8
14. A. Bolufé-Röhler, S. Estévez-Velarde, A. Piad-Morffis, S. Chen, J. Montgomery, in *Evolutionary Computation (CEC), 2013 IEEE Congress on* (IEEE, 2013), pp. 40–47
15. C. Segura, C.A.C. Coello, E. Segredo, A.H. Aguirre, A novel diversity-based replacement strategy for evolutionary algorithms, IEEE transactions on cybernetics **46**(12), 3233 (2016)
16. J. Montgomery, S. Chen, in *Evolutionary Computation (CEC), 2010 IEEE Congress on* (IEEE, 2010), pp. 1–8

17. N.H. Awad, M.Z. Ali, P.N. Suganthan, R.G. Reynolds, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 2958–2965
18. J. Brest, M.S. Maučec, B. Bošković, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 1188–1195
19. S. Elsayed, N. Hamza, R. Sarker, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 2966–2973
20. A. Kumar, R.K. Misra, D. Singh, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 1835–1842
21. J. Brest, M.S. Maučec, B. Bošković, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 1311–1318
22. D. Molina, F. Moreno-García, F. Herrera, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 805–812
23. J.J. Durillo, A.J. Nebro, C.A.C. Coello, J. Garcia-Nieto, F. Luna, E. Alba, A Study of Multiobjective Metaheuristics When Solving Parameter Scalable Problems, IEEE Transactions on Evolutionary Computation **14**(4), 618 (2010)