Importancie de la Diversidad en el Diseño de Algoritmos Evolutivos

Carlos Segura[a,*], Joel Chacón Castillo[a]

[a] *Area of Computer Science, Centre for Research in Mathematics (CIMAT), Callejón Jalisco s/n, Mineral de Valenciana, Guanajuato, Guanajuato 36240, Mexico*

Abstract

La convergencia prematura es uno de las mayores desventajas que afectan el rendimiento de los algoritmos evolutivos. Mantener un grado de diversidad de forma explícito es una alternativa para lidiar con este problema. En este capítulo se realizan dos aportaciones para promover la diversidad en el espacio de las variables. En la primera en el ámbito de evolución diferencial propone una estrategia de reemplazo que combina una población elite y un mecanismo para mantener la diversidad de forma explícito. La novedad de esta primera propuesta es el uso de un balanceo dinámico entre exploración e intensificación en las distintas etapas de optimización. La validación experimental es llevada a cabo con varios problemas de prueba propuestos en los problemas del Congreso de Cómputo Evolutivo. Por otra parte, en la segunda aportación enfocado a los operadores de cruce, se analiza el Operador de Cruce basado en Simulación Binaria (Simulated Binary Crossover - SBX). Además, se proponen extensiones del SBX donde se consideran modificaciones de forma dinámica donde es considerado el criterio de paro. Esto con el propósito de inducir un cambio gradual entre exploración a intensificación en el proceso de búsqueda. La validación experimental se realizó con los problemas de prueba mas populares del ámbito multi-objetivo, donde se demostró una mejora significativa.

Keywords: Diversidad, Multi-objetivo, Convergencia Prematura, Evolución Diferencial

1. Introducción

Los Algoritmos Evolutivos (Evolutionary Algorithms - EAs) son considerados como uno de los enfoques con mayor aficacia para resolver distintas categorías de optimización. Además, diversas variantes se han desarrollado y aplicado en muchos campos, como es en ciencia, economía e ingeniería. Particularmente, se han aplicado en problemas tanto de dominio continuo [28] como de dominio discreto [54]. Especialmente, los EAs son aplicados para resolver problemas complejos cuyo enfoque determinístico es complicado o imposible [11]. Un problema de optimización puede ser definido de forma general como se indica en la ecuación (1).

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

donde D es la dimensión correspondiente al espacio de las variables, Ω es el espacio factible cuyo límite inferior es $x_i^{(L)}$ y límite superior es $x_i^{(U)}$, $\vec{X}$ es un vector compuesto por D variables de decisión $\vec{X} = [x_1, x_2, ..., x_D]$, además 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, a no se que se indique lo contrario únicamente se considera una función objetivo. Actualmente, los EAs son probablemente conocidos como una de las metaheurísticas más conocidas [28], y a pesar de su éxito, presentan la desventaja de ser configurados

*Corresponding author. Tel.: + 52 473 732 7155

Email addresses: carlos.segura@cimat.mx (Carlos Segura), joel.chacon@cimat.mx (Joel Chacón Castillo)

ante problemas nuevos. Estos problemas implican la toma de varias decisiones difíciles. Particularmente, como parte del diseño de un EA, estan presentes varios aspectos relevantes con la meta de obtener soluciones de calidad. Esto es alcanzado al inducir un balance adecuado entre exploración e intensificación [31]. Sin embargo, no siempre se comprenden las implicaciones de mantener un grado de diversidad adecuado para alcanzar este balance, muchas veces esto se debe a que existen varios componentes específicos los cuales afectan a la exploración e intensificación [60]

Desde los inicios de los EAs se han observado problemas de convegencia, que por ende son considerados como una desventaja muy importante[60]. Particularmente, 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 componentes seleccionados no son suficientes para escapar de esta región. En base a esto, se han desarrollado varias estrategias para aliviar tales inconvenientes.

A través de varios estudios se ha revelado que mantener una población diversa es un requisito previo para evitar la convergencia prematura [60]. Sin embargo, si la población es muy diversa no se podría alcanzar un grado adecuado de intensificación y por lo tanto se tendría una convergencia lenta, esto en consecuencia generaría soluciones de baja calidad. Por esta razón, Mahfoud [13] utilizó el concepto de diversidad útil, donde se refiere a la cantidad de diversidad necesaria para generar soluciones de calidad.

En relación al diseño de los EAs, se observa que en sus inicios, la mayoría de enfoques fueron generacionales [15], es decir que las soluciones hijo reemplazaban a la población sin importar su respectiva aptitud. En estos esquemas iniciales se utilizó la selección de padres para influir al proceso de búsqueda hacia las regiones más prometedoras En resultado y con el propósito de alcanzar un balance entre exploración e intensificación, se desarrollaron muchas estrategias basadas en las selección de padres. Además se desarrollaron otras alternativas en las cuales se modificaba la estrategia de variación [32] y/o un modelo poblacional [2]. Sin embargo en los EAs más recientes se reemplaza la "reproducción con énfasis" o al menos es combinado con el principio de "el sobreviviente más apto" [25]. Específicamente, estos algoritmos utilizan una fase de selección adicional (en lugar de reemplazar a la población anterior), esto se realiza con el propósito de elegir a los individuos que sobrevivirán a la siguiente generación [23], particularmente a este procediemento se le conoce como la estrategia de reemplazo o selección del sobreviviente. De forma general, los trabajos que se presentan en este capítulo están basados en la hipótesis de que se puede inducir un balance entre exploración e intensificación al considerar un mecanismo para preservar la diversidad de forma explícita que en consecuencia se producirán soluciones de calidad al considerar ejecuciones a largo plazo [56]. Esto se fundamenta en las fases de variación y selección de padres, estas fases realizan decisiones que podrían afectar a las generaciones actuales ya que la elección de los sobrevivientes influye de forma significativa a todo proceso de optimización. Especialmente, este mecanismo se enfoca en seleccionar a las soluciones que sobreviven en la siguiente generación, por lo tanto la fase de reemplazo puede actuar adecuadamente evitando la selección de los individuos no deseados que provienen de las fases de variación y selección de padres.

Particularmente, en este capítulo se describen dos aportaciones, la primera está enfocada al área de *Evolución Diferencial* (Differential Evolution DE),esta primera aportación es nombrada DE *Mejorado con Mantenimiento de Diversidad* (DEwith Enhanced Diversity Maintenance DE-EDM) e integra una estrategia de reemplazo que mantiene un grado de diversidad de forma explícita considerando además una población elite. En la fase de reemplazo que incorpara el DE-EDM se promueve un balance entre exploración e intensificación de forma dinámica donde es considerando el criterio de paro. De esta forma, en las primeras etapas se induce un grado de exploración ya que los individuos sobrevivientes son diversificados, posteriormente conforme transcurren las generaciones y de forma gradual se induce un grado de intecificación. La segunda aportación están enfocada a los operadores de cruce, particularmente se analizan los componentes que conforman a *El Operador de Cruce basado en Simulación Binaria* (Simulated Binary Crossover - SBX), y además se propone una variante DSBX el cual es considerado como una modificación del operador SBX cuyo comportamiento interno es alterado de forma dinámica con el propósito de alcanzar un balance. El resto de este capítulo está organizado de la siguiente forma.

2. Preservación de diversidad en algoritmos evolutivos

La convergencia prematura en una desventaja muy conocida en el ámbito de los EAs, por lo tanto se han desarrollado una gran cantidad de técnicas para lidiar con este inconveniente [50]. Muchas de estas técnicas se basan en manejar la diversidad de forma directa o indirecta [61]. Estas estrategias varían desde técnicas generales hasta heurísticas las cuales dependen de cada problema. En este apartado se revisan algunas de las técnicas mas populares. De forma general se mencionan dos esquemas, los enfoque clásicos para administrar la diversidad y las estrategias de reemplazamiento basadas en diversidad. Finalmente, este estudio se acota al campo de Evolución Diferencial (Differential Evolution - DE) donde se revisan los trabajos más significativos y que están relacionado con diversidad en este campo. Sin embargo, para tener una detallada revisión de las estrategias relacionadas con diversidad se sugiere consultar [61]

2.1. Esquemas clásicos para administrar la diversidad

Los primeros EAs se basaron principalemente en esquemas generacionales, por lo tanto con la intención de obtener un balance propio entre exploración e intensificación se desarrollaron varias estrategias la cuales no afectan el mecanismo para la selección de los sobrevivientes. Específicamente, en los EAs generacionales se induce la presión de selección al considerar a la selección de padres. En los 90s se desarrollaron varios esquemas donde la presión de selección es controlada [23]. Además se diseñaron varias estrategias dinámicas con el fin de adaptar el balance entre exploración e intensificación. Sin embargo, en base a varios estudios se observó que no se puede mantener un grado adecuado de diversidad considerando únicamente los operadores de selección, esto se mantiene con poblaciones grandes.

Además, se han desarrollado modelos poblacionales con el propósito de mejorar la presevación de diversidad en los EAs. Por lo tanto, en los últimos años los EAs poblacionales han ganado reconocimiento [27]. En estos esquemas se imponen algunas restricciones de emparejamiento en base a la ubicación de los individuos en la población. De esta forma, se han ideado algunos esquemas con el propósito de reducir la interacción entre los individuos, lo cual facilita su paralelización. Sin embargo, estos esquemas tienen efectos importantes en la diversidad [2], que en consecuencia se utilizan como un mecanismo para promover la exploración. Particularmente, estos esquemas no son efectivos en ejecuciones a largo plazo ya que no mantinen un grado de diversidad de forma explícita. Además, no es sencillo controlar la reducción de la diversidad ya que existen varios componentes lo cuales influyen en la pérdida de diversidad [10]. Asimismo, estas estrategias usualmente introducen muchos parámetros y por lo tanto es necesario un proceso de ajuste.

Los esquemas puramente basados en restricciones de emparejamiento son similares a los previamente descritos en el sentido de que se evitan algunas interacciones entre los individuos. Sin embargo, en estas estrategias no se consideran las posiciones de los individuos de la población. En su lugar, se considera la distancia normalizada entre los individuos. Además, en algunos casos parece ser más prometedor promover el emparejamiento entre individuos no similares [25], sin embargo en algunos escenarios esto no se mantiene [16]. Es importante resaltar que estas estrategias no previenen a la convergencia prematura de forma total, es decir solo retrasan a la convergencia. Por lo tanto estas estrategias podrían introducir mecanismos adicionales. Otra alternativa es adaptar la fase de variación en distintas etapas. Por lo tanto se han desarrollado diversas técnicas para controlar los parámetros con el propósito de adaptar el balanceo entre exploración e intensificación. Particularmente, esto se realiza considerando distintos valores en los parámetros para distintas etapas a lo largo del proceso de optimización [64]. En otros casos se consideran varios operadores con distintas capacidades de búsqueda [40]. Usualmente, en estos esquemas no se considera la diversidad de forma directa. En su lugar, esto se maneja de forma implícita utilizando distintos operadores o parámetros, lo cual en algunas situaciones puede causar desventajas. Un enfoque muy prometedor es utilizar un procedimiento de forma directa para controlar la diversidad de forma explícita [55, 40]. Por otra parte se puede utilizar un mecanismo de memoria para adaptar la fase de variación [65]. Usualmente esto no es posible para ejecuciones de largo plazo, y por lo tanto se deben considerar otras alternativas.

Finalmente se encuentran los esquemas de reinicio los cuales son suficientemente populares. En estos esquemas, en lugar de evitar la convergencia acelerada, se aplica un reinicio total o parcial de la población. En base a esto se han propuesto varias estrategias para establecer los puntos de reinicio [37]. Además, estos esquemas se implementan de forma fácil y en algunos casos han proporcionado mejoras significativas [38].

Adicionalmente, estas estrategias puede se utilizas con esquemas para mantener la diversidad ya que estan enfocadas en recuperar la diversidad.

2.2. Esquemas de reemplazamiento basados en diversidad

Este tipo de mecanismos modifican la fase de reemplazo para preservar la diversidad. La idea principal de estos esquemas es inducir un grado de exploración adecuado diversificando a los individuos sobrevivientes. Esto se sucede ya son exploradas más regiones del espacio de búsqueda si se mantinen una población cuya diversidad sea elevada. Además, los operadores de cruce tienen un efecto de exploración al considerar individuos distantes [22]. Particularmente, el esquema de pre-selección propuesto por Cavicchio's [29] es uno de los primeros estudios que utilizan la fase de reemplazamiento para controlar la diversidad. Posteriormente, esta pre-selección se extendió para generar el *amontonamiento* (crowding) [14] el cual ha sido muy popular en los últioms años [42, 43]. El principio del crowing se basa en forzar el ingreso de nuevos individuos los cuales únicamente sustituyen a sus padres con los que son similares.

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 únicamente considerando la selección de padres no es suficiente para alivar la convergencia prematura [6]. Posteriormente, una gran cantidad de EAs incorporaron una fase de reemplazo y 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 [60]. Es importante hacer mención de que aún cuando los métodos de reemplazamiento y generacionales [15] fueron suficientemente populares, anteriormente algunos autores ya habían tomado en cuenta esta idea [42]. Sin embargo, con la efectividad del elitismo y otras estrategias de reemplazo, el número de esquemas que adoptaron estos principios crecieron de forma considerable [41].

2.3. Diversidad en Evolución Diferencial

Los algoritmos basados en DE son altamente suceptibles a la pérdida de diversidad, esto se debe a la estrategia de selección la cual es considerada muy 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 ante distintos problemasDE [66]. Alternativamente, se han desarrollado algunos análisis donde se considera el efecto de los vectores de diferencia en el operador de mutación [45]. Estos análisis y otros estudios empíricos están basados en el operador de cruce, donde se establece un mecanismo para retrasar la convergencia prematura, particularmente este deshabilita ciertos movimientos que pueden resultar perjudiciales para el rendimiento del algoritmo [47]. En este último estudio, el tipo de movimientos aceptados varía a lo largo de la ejecución. Particularmente, 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 [7]. 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 los vectores de diferencias que aparecen en la toda población. En su lugar, las restricciones son aplicadas a las diferencias que aparecen en la fase de reemplazo.

Una distinta alternativa reside en alterar el operador de selección [53]. Particularmente, con el propósito de mantener la diversidad de la población, se altera la presión de selección mediante una selección probabilística, posteriormente eso podría permitir escapar de las bases de atracción que pertenecen a los óptimos locales. Sin embargo este método es muy sensible al mapeo del espacio de búsqueda ya que considera la aptitud para definir las probabilidades para seleccionar a un individuo. En este caso las decisiones no se basan en las generaciones transcurridas.

Finalmente, el algoritmo *Diversidad de la Población Auto-Mejorado* (*Auto-Enhanced Population Diversity* - AEPD) mide la diversidad de forma explícita y cuando se detecta poca diversidad en la población se inicia un mecanismo de diversificación [63]. Es importante mencionar que ya se han propuesto estrategias con principios similares pero con esquemas de perturbación distintos [69].

Particularmente, es interesante notar que las variantes de DE que alcanzaron los primeros lugares en varias competencias de optimización no consideran estas modificaciones, y además estas variantes no han sido incorporadas en las herramientas de optimización más populares. Como resultado, estos algoritmo no son ampliamente utilizados.

3. Diseño de evolución diferencial basado en diversidad

3.1. Evolución diferencia: conceptos básicos

Esta sección esta dedicada para revisar a la variante clásica de DE y para introducir varios términos importantes que son utilizados en el campo de DE. El clásico esquema DE es identificado como DE/rand/1/bin el cual se utiliza de forma como base para el desarrollo de variantes complejas [12]. 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. Particularmente, en un problema de optimización mono-objetivo se desea resolver la ecuación 1, esto significa que únicamente se considera un objetivo.

DE es un algoritmo estocástico el cual es basado en una población, por lo tanto éste involucra a un conjunto de soluciones candidatas de forma iterativa. En DE dichas soluciones candidatas son usualmente conocidas como vectores. En la variante básica de DE, para cada miembro de la población (conocido 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, las generaciones transcurren de forma iterativa hasta cumplir el criterio de paro. El i-ésimo vector de la población en la generación G es definido como$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, ..., X_{D,i,G}]$. A continuación se explica en más detalle cada componente de DE.

3.1.1. Inicialización

Usualmente, DE inicia el proceso de optimización con una población de NP vectores los cuales son inicializados de forma aleatoria. Principalmente, los vectores de la población inicial son generados en base a una distribución uniforme, ya que usualmente no se posee información del espacio de búsqueda. Por lo tanto el j-ésimo componente del i-ésimo vector es inicializado de la forma $x_{j,i,0} = x_j^{(L)} + rand_{i,j}[0,1](x_j^{(U)} - x_j^{(L)})$, donde $rand_{i,j}[0,1]$ es un número aleatorio uniformemente distribuido que se encuentra entre 0 y 1.

3.1.2. Operador de Mutación

Por cada vector objetivo se genera un vector mutado, para realizar esto se han propuesto varias estrategias. Especialmente, en la variante clásica de DE se aplica la estrategia rand/1. En este caso, se crea un vector mutado $V_{i,G}$ de la siguiente forma:

$$\vec{V}_{i,G} = \vec{X}_{r1,G} + F \times (\vec{X}_{r2,G} - \vec{X}_{r3,G}) \quad r1 \neq r2 \neq r3 \tag{2}$$

Los índices $r1, r2, r3 \in [1, NP]$ deben ser enteros distintos y son generados de forma aleatoria en el rango $[1, NP]$. Además, estos índices son distintos al índice i. Es imporante notar que la diferencia entre los vectores es escalada por medio del parámetro F, el cual usualmente se define en el intrervalo [0,4, 1]. Posteriormente, el vector de diferencia (escalado) es agregado a un tercer vector. Por lo tanto los vectores mutados son similares a los vectores objetivo si el grado de diversidad es poco y los vectores de diferencias son pequeños. Como resultado, es crítico mantener un grado mínimo de diversidad en DE.

3.1.3. Operador de Cruce

Este operador es plicado con el objetivo de combinar la información de distintas soluciones candidatas y para incrementar la diversidad de los vectores. Específicamente, cada vector objetivo $\vec{X}_{i,G}$ es mezclado con su correspondiente vector mutado $V_{i,G}$ para generar un vector de prueba $\vec{U_{i,G}} = [u_{1,i,G}, u_{2,i,G}, ..., u_{D,i,G}]$. La estrategia de cruce mas típico es conocido como *cruce binomial*, el cual funciona de la siguiente forma:

$$\vec{U}_{j,i,G} = \begin{cases} \vec{V}_{j,i,G}, & \text{si}(rand_{i,j}[0,1] \leq CR \quad or \quad j = j_{rand}) \\ \vec{X}_{j,i,G}, & \text{de otra forma} \end{cases} \tag{3}$$

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

3.1.4. Operador de Selección

Finalmente, se aplica una selección agresiva (greedy), esto con el propósito de determinar a los vectores sobrevivientes que participarán en la siguiente generación. Cada vector de prueba es comparado con su correspondiente vector objetivo y el sobrevive el tiene la mejor aptitud:

$$\vec{X}_{j,i,G+1} = \begin{cases} \vec{U}_{i,G}, & \text{si} \quad f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}) \\ \vec{X}_{i,G}, & \text{de otra forma} \end{cases} \tag{4}$$

Por lo tanto, en cada generación los miembros de la población permanecen iguales o mejoran. En particular, esta selección se considera agresiva ya que siempre se selecciona al vector con mayor aptitud. Es importante notar que el vector de prueba sobrevive si existe un empate en la aptitud.

3.2. Propuesta

Principalmente, nuestra propuesta está motivada por dos trabajos lo cuales están relacionados con el propósito de controlar la diversidad y son diseñados en el campo de los EAs. El primero es un estudio empírico desarrollado por Montgomery et al. [47], este trabajo presenta varios análisis empíricos que confirman varios problemas relacionados con la convergencia prematura.

Por otro lado, el segundo trabajo propuesto por Segura et al. [56], proporciona mejoras significativas en el campo optimización combinatoria, en esta propuesta se desarrolla una estrategia novel basada en la fase de reemplazo conocida como *Reemplazo con Control de Diversidad Dinámico Basado en Varios Objetivos* (Replacement with Multi-objective based Dynamic Diversity Control - RMDDC). Particularmente el RMDDC controla el grado de diversidad en base al criterio de paro y a las generaciones transcurridas. Es importante resaltar que se obtuvieron beneficios en varias aplicaciones por el RMDDC, por lo tanto y basados en las conclusiones de los trabajos previos, la propuesta de esta sección es una variante novedosa de DE la cual incluye un mecanismo explícito que sigue un principio del RMDDC. Este optimizador novedoso es nombrado *Evolución Diferencial con Mantinimiento Mejorado de Diversidad* (Differential Evolution with Enhanced Diversity Maintenance - DE-EDM) y su código fuente está disponible [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 del DE-EDM es que incorpora una población elite (E) y una estrategia de reemplazo basada en la diversidad. Específicamente, con el propósito de seleccionar a los miembros de la población elite se considera el operador de selección agresivo que pertenece a DE (línea 7). Por otra parte, se aplica una 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, la cual se utiliza originalmente para mantener a la población padre (X). En su lugar, se consideran tanto el criterio de paro como las generaciones transcurridas en la selección. Así, se establecera un grado de diversidad mínimo dependiente al tiempo de ejecución. 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 con el propóstivo de promover una convergencia acelerada. Primero, no son considerados los conceptos multi-objetivo, en su lugar se considera una selección más agresiva. Segundo, también es considerada una población elite en la estrategia de reemplazo.

Nuestra estrategia de reemplazo (ver Algoritmo 2) funciona de la siguiente forma. Inicialmente, ésta recive a la población padre (vectores objetivo), la población de hijos (vectores de prueba) y a los vectores elite. Así,

[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 distribuido 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 (2).
6: Cruce: Utilizar la recombinación para general al vector de prueba ($U_{i,G}$) de acuerdo a la Ecuación (3).
7: Selección: Actualizar al vector elite ($E_{i,G}$ en lugar de $X_{i,G}$) de acuerdo a la Ecuación (4).
8: Reemplazo: Seleccionar a los vectores objetivo (X_{G+1}) de acuerdo a la Ecuación (2).
9: $G = G + 1$

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* $\emptyset$.
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 a 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 5).
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*

en cada generación se seleccionan NP vectores para la siguiente población de padres. Primero, en base al número de evaluaciones a funcion (línea 2) se calcula una distancia mínima D_t para mantener la diversidad en la población. Entonces, se juntan las tres poblaciones en un conjunto de miembros candidatos (línea 3). El conjunto de miembros candidatos continen vectores que podrían ser seleccionados para sobrevivir. Entonces, se inicializa tanto el conjunto de individuos sobrevivientes como los penalizados con el conjunto vacío (línea 4). De otra forma, para seleccionar a los NP sobrevivientes (población padre de la siguiente generación) se repite el siguiente proceso iterativo (líneas 5 - 13). En cada paso se selecciona al mejor individuo para sobrevivir del *Conjunto de Candidatos*, es decir al individuo que tiene la mejor aptitud, entonces éste se mueve al *Conjunto de Sobrevivientes*. Posteriormente, los individuos que pertenecen al *Conjunto de Candidatos* y cuya mínima distancia sea menor que D_t son transferidos *Conjunto de Penalizados* (línea 9).

La forma para calcular la distancia entre dos individuos es en base a la distancia Euclideana normalizada descrita en la ecuación 5, 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).

$$distance(x_i, x_j) = \frac{\sqrt{\sum_{d=1}^{D} \left(\frac{x_i^d - x_j^d}{b_d - a_d}\right)^2}}{\sqrt{D}} \tag{5}$$

Adicionalmente, con el propósito de completar la descripción del DE-EDM se especifica la forma en que se calcula D_t y el procedimiento con el cual se actualizan a los individuos elite. El resto del algortimo se mantiene igual que la clásica variante de DE. El valor de D_t se utiliza para alterar el grado entre exploración e intensificación, 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 las últimas etapas de optimización. En nuestro esquema se requiere asignar un valor inicial para D_t (D_I). Así, similarmente que en [56], se calcula una reducción lineal de D_t considerando tanto las evoluciones a función como el criterio de paro, Particularmente, en este trabajo, el criterio de paro se asigna en base a las evaluaciones a función, sin embargo podría ser asignado en base al tiempo. La reducción se calcula de tal forma que al 95 % del número máximo de evaluaciones el valor de diversidad mínimo requerido es nulo ($D_t = 0$). Por lo tanto, la diversidad no es considerada del todo en el resto de la ejecución (5 %).

Entonces, si max_nfes es el número máximo 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 en varios tipos de problemas tales como multi-modales y deceptivos. Por lo tanto, la convegencia prematura podría ser aliviada. Sin embargo, 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 los óptimos locales. El óptimo valor de 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 en la sección de validación experimental se realiza un análisis de estabilidad. Específicamente, se analiza el comportamiento del DE-EDM con distintos valores D_I.

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

De acuerdo a varios estudios desarrollados por Montgomery y otros [46] la probabilidad de cruce es el más imporante. Estos autores probaron de forma empírica que valores extremos de CR resultan en un comportamiento muy distinto. Ellos explicaron que valores pequeños de CR resultan en una búsqueda que es alineada con un pequeño número de ejes y además induce pequeños desplazamientos. Esto provoca una gradual y lenta convergencia que en algunos escenarios podría resultar en un comportamiento robusto. Por otra parte, valores elevados de CR podrían generar soluciones de mayor calidad con una menor probabilidad. Sin embargo, estas transformaciones provocan largos desplazamientos que podrían mejorar la aptitud 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 6. Es importante notar que si se aplica este procedimiento en la versión estándar de DE podría presentarse una convergencia prematura.

$$CR = \begin{cases} Normal(0{,}2, 0{,}1), & \text{si} \quad rand[0,1] \leq 0{,}5 \\ Normal(0{,}9, 0{,}1), & \text{de otra forma} \end{cases} \tag{6}$$

Siguiendo los principios de distintas variantes del SHADE [3, 8], se consideran las evaluaciones a función para generar el factor de mutación F. Particularmente, cada valor F es una muestra de una distribución Cauchy (Ecuación7).

$$Cauchy(0{,}5, 0{,}5 * nfes/max\_nfes) \tag{7}$$

Así, en las primeras etapas de optimización se generan valores de F similares a 0,5. Conforme la execución transcurre, la función de densidad sufre una transformación gradual ya que la varianza se incrementa, esto implica que se generan valores fuera del intervalo $[0{,}0, 1{,}0]$ con una probabilidad alta. En los casos cuando los valores son mayores a 1,0, se utiliza 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 el de incrementar la probabilidad de generar valores elevados de F conforme transcurren las generaciones, así se evita una convergencia acelerada en las últimas etapas de optimización.

3.3. Resultados DE-EDM

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 consideraron los conjuntos de prueba del CEC 2016 y CEC 2017. Cada uno está compuesto de treinta problemas distintos. El estado-del-arte está compuesto por los algoritmos que alcanzaron 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 [24] y L-SHADE-EpSin [3], que alcanzaron el primero y el segundo lugar respectivamente. Similarmente, los mejores algoritmos del CEC 2017 son el EBOwithCMAR [39] y el jSO [9].

Es imporante destacar que el EBOwithCMAR es considerado como una mejora del UMOEAs-II. Adicionalmente, el jSO y el L-SHADE-EpSin pertenecen a la familia de SHADE. Todos estos algoritmos fueron probados con los dos conjuntos de prueba como es sugeriodo en [44]. 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 sus autores y son descritos 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. Adicionalmente, para comparar los resultados estadísticamente, se siguió un procedimiento similar que el propuesto en [21].

Concretamente, en primer lugar se utilizó la prueba Shapiro-Wilk para comprobar si los resultados se ajustaban a una distribución Gaussiana. En los casos en que sí se ajustaban, se utilizó la prueba de Levene para comprobar la homogeneidad de las varianzas, procediendo con la prueba de ANOVA en caso positivo o con la prueba de Welch en caso negativo. Por otro lado, para los casos que no se ajustaban a distribuciones Guassianas, se utilizó la prueba 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 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, se resolvieron 21 funciones al menos una vez pertenecientes 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 en significativamente 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

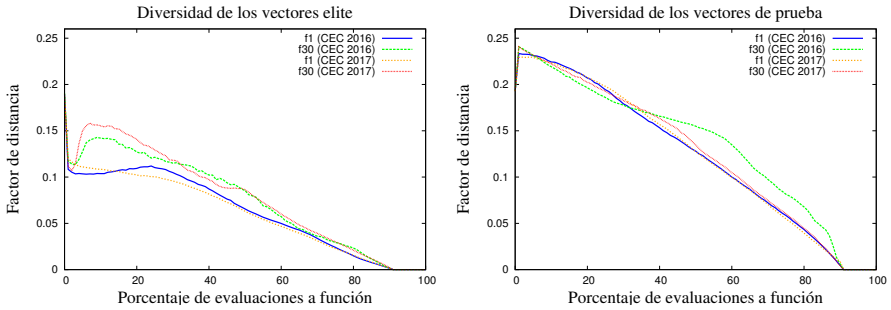

Figura 1: Promedio del DCN de las 51 ejecuciones con los problemas f_1 y f_{30} (CEC 2016 y CEC 2017). El factor de distancia inicial corresponde a $D_I = 0{,}3$.

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 algoritmo L-SHADE-Epsilon alcanzó el último lugar en los dos años 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 (8). Por lo tanto el puntaje final está compuesto por la suma $Score = Score_1 + Score_2$.

$$\begin{aligned} Score_1 &= \left(1 - \frac{SE - SE_{min}}{SE}\right) \times 50, \\ Score_2 &= \left(1 - \frac{SR - SR_{min}}{SR}\right) \times 50, \end{aligned} \tag{8}$$

donde, SE_{min} es la suma errores mínimo 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 se aplica la suma de cada rango en cada función para todos los algoritmos $SE = \sum_{i=1}^{30} error\_f_i$. 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. Además, el puntaje considera el rango y la media del 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. Particularmente, se ejecutó el DE-EDM con las funciones f_1 y f_{30}. Basado en sus propiedades, la primera 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 en las dos funciones se mantiene un grado de diversidad de forma implícito. Similarmente, la parte derecha corresponde a la diversidad de los vectores de prueba. Se puede apreciar el grado de diversidad mantenido explícitamenete hasta el 95 % del total de evaluaciones a función.

Además, 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

Tabla 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

Tabla 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

Tabla 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

que nuestra propuesta resuelve a todos los problemas unimodales. Además, varias funciones multi-modales son aproximadas de forma aceptable. Principalmente, nuestra propuesta resolvió y mejoró significativamente varias funciones complejas (por ejemplo funciones computestas), las cuales no fueron resueltas por los algoritmos del estado-del-arte.

3.4. Análisis Empirico del factor de distancia inicial

En nuestra propuesta, la diversidad es explícitamente promovida a través de varias etapas que son controlada mediante el factor de distancia inicial D_I. Por lo tanto, el efecto de este parámetro se analiza en detalle. En base a la configuración general que define con anterioridad se consideran los siguiente factores de distancia inicial $D_I = \{0{,}0, 0{,}1, 0{,}2, 0{,}3, 0{,}4, 0{,}5, 0{,}6, 0{,}7, 0{,}8, 0{,}9, 1{,}0, 1{,}1\}$. Para realizar esto se considera el promedio de la razón de éxito de cada año.

Tabla 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

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 y por en la calidad de las soluciones están comprometidos.
- En este escenario la configuración ideal es de $D_I = 0{,}3$, a pesar de que aún existen soluciones de calidad en el rango $[0{,}1, 0{,}4]$.
- Si se asigna excesivamente la diversidad inicial se observa un deteriodo en la calidad de las soluciones.

Finalmente, es importante notar que en base a otras experimentos se observa que la calidad de las soluciones es más afectado por el factor de distancia inicial que por el tamaño de la población.

4. Diseño de operadores de cruce basados en diversidad

En esta sección se revisan algunos de los trabajos más importantes y que están relacionados con los operadores de cruce. Inicialmente, se definen varios conceptos del campo multi-objetivo. Posteriormente, se introducen algunas clasificaciones más populares de los operadores de cruce. Además, se describe describe detalladamente el operador de cruce SBX Finalmente, se realiza un análisis del operador SBX y en base a esto se propune una variante dinámica *Operador de Cruce Dinámico basado en Simulación Binaria* (Dynamic Simulated Binray Crossover - DSBX).

4.1. Algoritmos Evolutivos Multi-objetivo

Los EAs son usualmente utilizados para para resolver Problemas de Optimización Multi-objetivo (Multi-objective Optimization Problems - MOPs), donde los objetivos usualmente están en conflicto. Particularmente, Un MOP continuo basado en minimización puede ser definido como se indica en la Ecuación (1) de la sección 1. Dadas dos soluciones $\vec{x}$, $\vec{y} \in \Omega$, $\vec{x}$ domina a $\vec{y}$, denotado por $\vec{x} \prec \vec{y}$, si y solo si

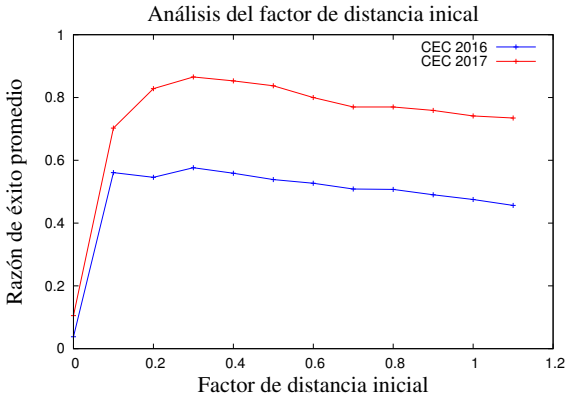

Figura 2: Razón de éxito promedio con distintos factores de distancia inicial con los problemas de prueba del CEC 2016 y CEC 2017, específicamente se considera una población de 250 individuos y $25,000,000$ evaluaciones a función.

$\forall m \in 1,2,...,M : f_m(x_i) \leq f_m(y_i)$ y $\exists m \in 1,2,...,M : f_m(x_i) < f_m(y_i)$. Una solución $\vec{x^*} \in \Omega$ es conocida como solución óptima de Pareto si no existe otra solución $\vec{x} \in \Omega$ que domine a $\vec{x^*}$. 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 un *Algoritmo Evolutivo Multi-Objetivo* (Multi-objective Evolutionary Algorithm - MOEA) es, esencialmente, obtener un conjunto de soluciones bien distribuidas y cercanas a las soluciones del frente de Pareto.

En los últimos años, se han ideado una gran cantidad de MOEAs los cuales siguen distintos principios. Así, para mejor tener una mejor clasificacón, se han propuesto varias taxonomías [57]. En base a sus principios de diseño, los MOEAs pueden ser basados en la dominancia de Pareto, indicadores y/o descomposición [51]. Todos estos algoritmos son suficientemente competitivos, por lo tanto en esta sección se consideraron MOEAs de distintos grupos. Particularmente, la validación experimental se desarrolló incluyendo a los algoritmos *Algoritmo Genético basado en Ordenación de los No-Dominados* (Non-Dominated Sorting Genetic Algorithm - NSGA-II) [19], *el MOEA basado en descomposición* (the MOEA based on Decomposition - MOEA/D) [67] y el *Algoritmo Evolutivo Multi-objetivo basado en la Métrica-S* (the S-Metric Selection Evolutionary Multi-objective Optimization Algorithm - SMS-EMOA) [5].) Estos algoritmos son representantivos de los basados en dominancia, basados en descomposición y basados en indicadores respectivamente. Las siguientes subsecciones describen cada uno de los paradigmas involucrados en los métodos seleccionados.

4.1.1. Algoritmos Basados en el concepto de Dominancia - NSGA-II

Uno de los paradigmas mas reconocidos son los enfoques basados en dominancia. Los MOEAs que pertenecen a esta categoría se basan en la aplicación de la relación de dominancia para diseñar los distintos componentes de los mismos, especialmente en la fase de selección. Dado que la relación de dominancia no promueve la diversidad de forma implícita en el espacio de los objetivos se han desarrollado técnicas para obtener una diversidad en el espacio de los objetivos como es el niching, crowding y/o clustering que usualmente se integran con el propósito de obtener una diversidad aceptable en el espacio de los objetivos. Una debilidad importante en los métodos que estan basados en la relación de dominancia, es la escalabilidad de la dimensionalidad en el espacio de los objetivos. De hecho, conforme se incrementa el número de objetivos, la presión de selección se reduce significativamente. A pesar de que se han desarrollado algunas estrategias para solventar este inconveniente [33], este parece ser la debilidad mas importante de este tipo de algoritmos.

El NSGA-II es uno de los algoritmos mas importantes de este grupo. Este algoritmo [19] considera un operador de selección especial el cual está basado en los procedimientos de la ordenación de las soluciones no dominadas y en el amontonamiento (crowding). Particularmente, el procedimiento que realiza la ordenación de las soluciones no dominadas se utiliza para proporcionar una convergencia hacia el frente de Pareto, mientras que el procedimiento de amontonamiento se utiliza para promover la diversidad en el espacio de los objetivos.

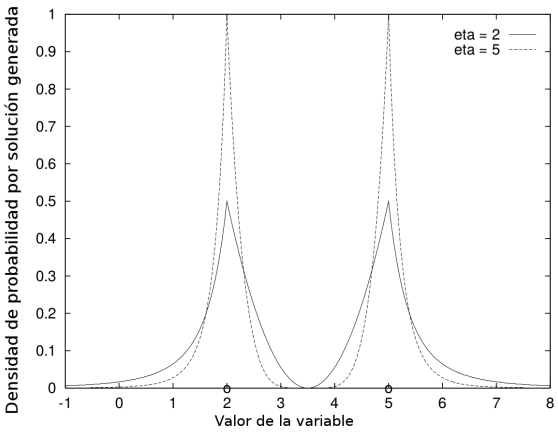

Figura 3: Función de densidad del operador de cruce SBX con índices de distribución 2 y 5. Las soluciones padre estan ubicadas en 2 y 5 respectivamente.

4.1.2. Algoritmos multi-objetivo basados en descomposición - MOEA/D

Los MOEAs basados en descomposición [67] transforman un MOP en un conjunto de problemas de optimización mono-objetivo los cuales son resueltos simultáneamente. Esta transformación se puede lograr a través de distintos enfoques. La función pesada de Tchebycheff es quizás uno de los enfoque más importantes. Específicamente, esta función requiere un conjunto de pesos bien distribuídos, esto con el propósito de alcanzar soluciones bien distribuídas a lo largo del frente de Pareto. Sin embargo, este tipo de algoritmos poseen una desventaja importante, es decir los vectores de pesos dependen de la geometría que posee el frente de Pareto. El MOEA [67] es un MOEA muy popular de los basados en descomposición. Este incluye varias características, tales como la descomposición de los problemas, la agregación de los pesos con los objetivos y las restricciones de emparejamiento que están basadas en la definición de vecindarios. El algoritmo MOEA/D-DE es considerado como una variante popular del MOEA/D, el cual utiliza operadores de DE [52] y el operador de mutación polinomial [30] en la fase de reemplazo. Adicionalmente, este algoritmo tiene dos mecanismos especiales para mantener la diversidad de la población [68].

4.1.3. Algoritmos multi-objetivo basados en indcadores - SMS-EMOA

En optimización multi-objetivo se han desarrollado varios indicadores de calidad con el propósito de comparar el rendimiento de los MOEAs. Desde que estos indicadores miden la calidad de las aproximaciones obtenidas por los MOEAs, se ha propuso un paradigma basado en en la aplicación de estos indicadores. Particularmente, los indicadores reemplazan a la relación de dominancia de Pareto con el propósito de guiar el proceso de optimización. Principalmente, el hypervolúmen es un indicador ampliamente aceptado por su relación completa de Pareto (Pareto-compliance) [35]. Una de la principales ventajas de estos algoritmos es que los indicadores normalmente consideran tanto la calidad de las soluciones como su diversidad, por lo tanto no se requieren mecanismos adicionales para preservar la diversidad.

El SMS-EMOA [5] es un MOEA popular basado en indicadores. Este MOEA es considerado como un algoritmo híbrido ya que utiliza tanto indicadores como el concepto de la dominancia de Pareto. Escencialmente, este algoritmo integra el procedimiento para ordenar a las soluciones no dominadas con la métrica del hipervolúmen. Por lo tanto el SMS-EMOA aplica el hipervolúmen como estimador de densidad el cual es computacionalmente complejo. Particularmente, la fase de reemplazo elimina al individuo que pertenece al frente con peor rango y cuya contribución al hipervolúmen sea mínima. Debido a su comportamiento prometedor, el SMS-EMOA se ha considerado como parte de nuestra validación experimental.

4.2. Operadores de cruce

Los operadores de cruce son diseñados para generar soluciones hijo utilizando la información de las soluciones padre. Estos combinan las características de dos o más soluciones padre con el propósito de

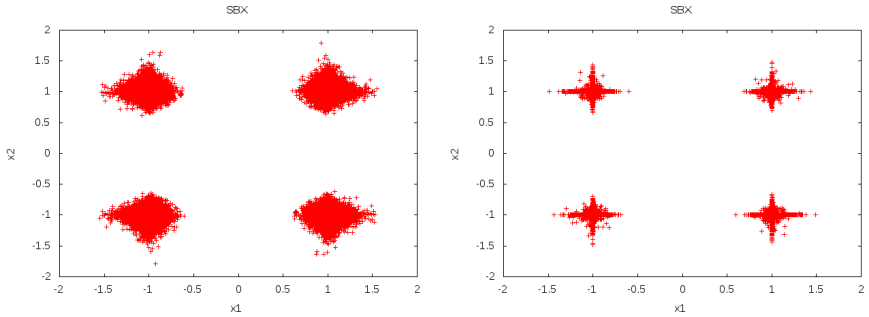

Figura 4: Simulaciones de operador SBX con un índice de distribución de 20. Las soluciones padre están ubicadas en $P_1 = (-1{,}0, -1{,}0)$ y $P_2 = (1{,}0, 1{,}0)$. En la parte izquierda la simulación consiste en alterar una variable con probabilidad de 1,0 y en la pare derecha con probabilidad de 0,1 (parámetro δ_1 en el Algoritmo 3).

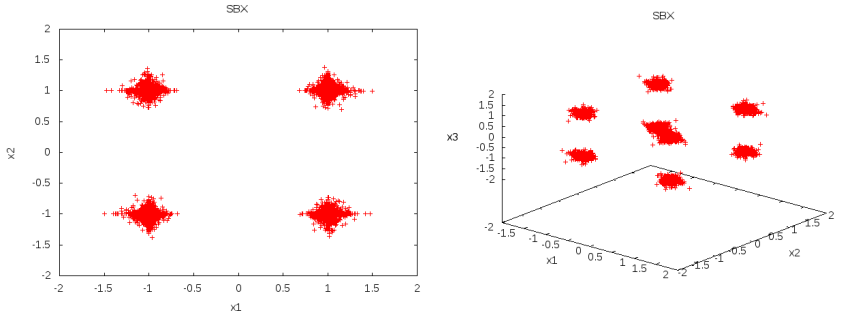

Figura 5: Simulaciones del operador SBX con un índice de distribución de 20. Las soluciones padre están ubicadas en $P_1 = (-1{,}0, -1{,}0)$, $P_2 = (1{,}0, 1{,}0)$ (parte izquierda) y $P_1 = (-1{,}0, -1{,}0, -1{,}0)$, $P_2 = (1{,}0, 1{,}0, 1{,}0)$ (parte derecha).

generar nuevas soluciones candidatas. En base a que en la literatura existen varios operadores de cruce, se han propuesto varias taxonomías para clasificarlos. Particularmente, las taxonomías se basan en varias características tales como la ubicación de las nuevas soluciones generadas o por el tipo de relaciones que existen en las variables.

Una taxonomía popular clasifica a los operadores de cruce en *basados en variables* y *basados en vectores*. En los *basados en las variables*, cada variable de las soluciones padre son combinadas para crear nuevos valores, esto se realiza de forma independiente y en base a una probabilidad especificada con anterioridad Este tipo de operadores son ideales para lidiar con problemas separables. Algunos operadores que pertenecen a esta categoría son el *Operador de Cruce Ciego* (the Blend Crossover - BLX) [26] y el SBX [1]. De otra forma, los operadores de recombinación *basados en vectores* son diseñados para considerar la dependencia que existe entre las variables. Este tipo de operadores regularmente realizan una combinación lineal de las soluciones involucradas. Algunos operadores que pertenecen a esta categoría son *El Operador de Cruce Unimodal Normalmente Distribuído* (The Unimodal Normally Distributed Crossover - UNDX) [49], y *El Operador de Cruce basado en el Simplex* (The simplex crossover - SPX) [58]. Adicionalmente, los operadores de cruce pueden ser clasificados como *basados en los padres* y *basados en la media* [36]. En los operadores basados en los padres, las soluciones hijo son creadas alrededor de cada solución padre, mientras que en los operadore basados en la media existen una tendencia de crear a las soluciones hijo alrededor de la media generada por las soluciones padres. Entre los operadores de cruce el SBX es probablemente uno de los más utilizados, por lo tanto esta sección se centra en este operador de cruce.

4.2.1. El Operador de Cruce Basado en Simulación Binaria - SBX

Los operadores de reproducción son uno de los componentes mas relevantes para influenciar el proceso de búsqueda en los EAs. Específicamente, los operadores de cruce y mutación están altamente relacionados con la diversidad de las soluciones. Por lo tanto, los operadores afectan significativamente la calidad de las soluciones.

Probablemente *El Operador de Cruce Basado en Simulación Binaria* (Simulated Binary Crossover - SBX) [17] es uno de los operadores más populares en dominios continuos y por lo tanto ha sido utilizado extensamente en muchos MOEAs [19, 5]. El operador SBX es clasificado con un operador basado en los padres, lo cual significa que las soluciones las cuales corresponden a los hijos (c_1 y c_2) serán creadas alrededor de los valores de las soluciones padre (p_1 y p_2). Específicamente, el proceso para generar los valores de las soluciones hijo se basa en una distribución de probabilidad. Esta distribución controla el factor de dispersión $\beta = |c_1 - c_2|/|p_1 - p_2|$ el cual es definido como la razón entre la dispersión de los valores de las soluciones hijo y los valores de las soluciones padre. Además, esta función de densidad se define en base a un índice de distribución η_c (es un parámetro de control especificado por el usuario) el cual altera la capacidad de exploración. Especificamente, un índice pequeño induce una probabilidad elevada de crear valores de las soluciones hijo distantes de los valores de las soluciones padre. Mientras que índices elevados tienden a crear soluciones muy similares a las soluciones padre, esto se demuestra en la Figura 3.

Particularmente, para crear una solución hijo se utiliza una distribución de probabilidad la cual está en función de $\beta \in [0, \infty]$ de la siguiente forma:

$$P(\beta) = \begin{cases} 0{,}5(\eta_c + 1)\beta^{\eta_c}, & \text{si} \quad \beta \leq 1 \\ 0{,}5(\eta_c + 1)\frac{1}{\beta^{\eta_c+2}}, & \text{de otra forma} \end{cases} \tag{9}$$

Basado en la propiedad de preservar la media en los valores que corresponden a las soluciones hijo y padre, el SBX tiene las siguientes características:

- Los valores de las soluciones hijo son equidistantes de los valores de las soluciones padre.
- Existe una probabilidad no nula de crear soluciones hijo en el espacio factible entero por cualquier par de soluciones padre.
- La probabilidad general de crear un par de soluciones hijo dentro del rango de las soluciones padre es idéntico a la probabilidad general de crear un par de soluciones hijo fuera del rango de las soluciones padre.

Por lo tanto, considerando dos valores de las soluciones padre (p_1 y p_2) se pueden crear dos valores de las soluciones hijo (c_1 y c_2) en base a una combinación lineal de los valores de las soluciones padre y con un número aleatorio uniforme $u \in [0, 1]$, especificado a continuación:

$$\begin{aligned} c_1 &= 0{,}5(1 + \beta(u))p_1 + 0{,}5(1 - \beta(u))p_2 \\ c_2 &= 0{,}5(1 - \beta(u))p_1 + 0{,}5(1 + \beta(u))p_2 \end{aligned} \tag{10}$$

El parámetro $\beta(u)$ depende en el número aleatorio u de la siguiente forma:

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

La ecuación anterior es formulada en base a un problema de optimización sin límites en las variables. Sin embargo, en problemas prácticos cada variable es limitada dentro de un límite inferior y superior. Por lo tanto, para considerar los límites del espacio de decisión [18] propusieron una modificación de la distribución de probabilidad en la ecuación (12). Es importante resaltar que esta última variante es una de las más utilizadas.

Algorithm 3 Operador de Cruce basado en Simulación Binaria (SBX)

```
Entrada: Soluciones padre (P_1, P_2), Indice de distribución (η_c), Probabilidad de cruce (P_c).
Salida: Soluciones hijo (C_1, C_2).
if U[0,1] ≤ P_c then
    for para cada variable d do
        if U[0,1] ≤ δ_1 then
            Generar C_{1,d} utilizando las ecuaciones (12) y (13).
            Generar C_{2,d} utilizando las ecuaciones (12) y (14).
            if U[0,1] ≤ (1 − δ_2) then
                Intercambiar C_{1,d} con C_{2,d}.
        else
            C_{1,d} = P_{1,d}.
            C_{2,d} = P_{2,d}.
else
    C_1 = P_1.
    C_2 = P_2.
```

$$\beta(u) = \begin{cases} (2u(1-\gamma))^{\frac{1}{\eta_c+1}}, & \text{si} \quad u \leq 0{,}5/(1-\gamma), \\ (\frac{1}{2(1-u(1-\gamma))})^{\frac{1}{\eta_c+1}}, & \text{de otra forma} \end{cases} \tag{12}$$

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

$$c_2 = 0{,}5(1+\beta(u))p_1 + 0{,}5(1-\beta(u))p_2 \tag{14}$$

En este caso, mediante la Ecuación (13) se calcula el valor de la solución hijo c_1 la cual está más cercana a p_1. Considerando que $p_1 < p_2$ y además con un límite inferior igual a a, se tiene que $\gamma = 1/(\alpha^{\eta_c+1})$, donde $\alpha = 1 + (p_1 - a)/(p_2 - p_1)$. Similarmente, el segundo valor de la solución hijo c_2 se calcula con $\alpha = 1 + (b - p_2)/(p_2 - p_1)$, donde b corresponde al límite superior. Entonces, el segundo valor de la solución hijo se calcula como se indica en la Ecuación (14).

Es importante aclarar que la primer versión del SBX fue diseñada en base a una sola variable, posteriormente los autores consideraron aplicarlo a problemas con múltiples variables [1]. Para esto los autores consideraron una estrategia simple para escoger las variables que se van a cruzar [49]. Específicamente, en base a los principios del operador de cruce unifome cada variable es cruzada con una probabilidad de 0,5. Sin embargo, los autores reconocieron las implicaciones que existen con los problemas con dependencia entre las variables. En cualquier caso, actualmente ésta es la forma mas común de aplicar el SBX en problemas con múltiple varibles.

4.2.2. Implementación y análisis del operador SBX

En este apartado se discuten algunas de las principales características de la implementación más utilizada del operador SBX. Escencialmente, se consideran tres componentes clave los cuales podrían afectar el rendimiento de los MOEAs. Primeramente, cada variable es alterada dada una probabilidad de 0,5. Si este valor de probabilidad se incrementa entonces existe una tendencia de generar valores de la solución hijo mas distantes de los padres debido a que mas variables son modificadas por cada operación. De acuerdo a esto, sería adecuado modificar una variable en los problemas separables. Por otra parte, parece ser mas conveniente modificar un conjunto de variables de forma simulánea en problemas no-separables. En la figura 4 se pueden observar las implicaciones al variar esta probabilidad considerando un problema con dos variables. Particularmente, en la parte derecha se muestra que una probabilidad pequeña provoca una tendencia de exploración ya algunas variables no son modificadas, es decir hay una tendencia de generar desplazamientos paralelos a los ejes. Esta característica es ideal para problemas separables. Por otra parte, en la parte izquierda se muestra que utilizando una probabilidad elevada existe un comportamiento de búsqueda distinto donde la tendencia anterior desaparece, lo cual podría ser ideal para problemas no separables. Es importante destacar que existe una relación entre esta probabilidad y con el índice de distribución, de hecho estos dos factores tienen un efecto directo en la similitud que existe entre las soluciones padre e hijo.

El segundo aspecto clave es después de generar dos valores de las soluciones hijo con la distribución SBX, ya que estos valores son intercambiados con una probabilidad fija que usualmente es 0,5, es decir el valor de la solución hijo c_1 no siempre es heradado por la solución padre mas cercana p_1. Esta es una característica no

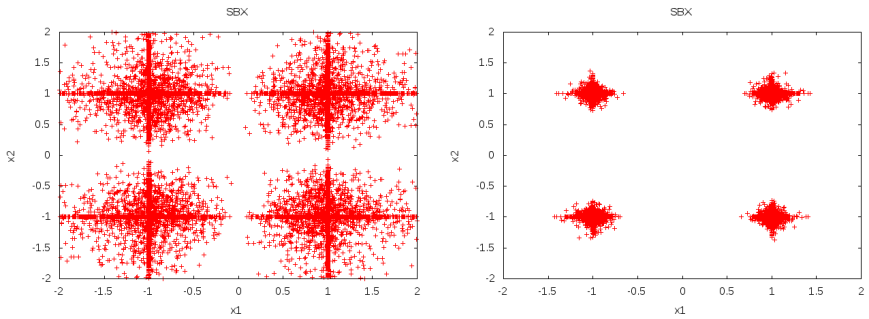

Figura 6: Simulación del operador SBX donde las soluciones padre están ubicadas en $P_1 = (-1{,}0, -1{,}0)$ y $P_2 = (1{,}0, 1{,}0)$. En la parte izquierda se consideró un índice de distribución de 2 y en la derecha de 20.

muy discutida, sin embargo es un aspecto muy relevante y afecta al rendimiento del algoritmo. En algunos contextos esta probabilidad se identifica como "Probabilidad de cruce uniforme por variable" (Variable uniform crossover probability) [59] o "Recombinación Discreta" (Discrete Recombination) [48].

Desde que en el ámbito multi-objetivo se promueve más diversidad en las variables de decision de forma implícita, estos intercambios podrían ser altamente disruptivos. De hecho, en algún sentido y dado a esto, no es totalmente claro que el SBX pueda ser categorizado como un operador totalmente basado en los padres. Estos intercambios que existen entre los valores de las soluciones hijo tienen un efecto de realizar múltiples "reflexiones" en el espacio de búsqueda. De hecho, conforme incrementa la dimensionalidad en el espacio de las variales, el número de regiones cubiertas incrementa de forma exponencial como se puede observar en los casos de dos y tres dimensiones en la figura 5. Es importante notar que esta característica tiene un efecto relevante en la distancia entre las soluciones padre y las soluciones hijo.

Finalmente, siendo quizás la característica mas conocida del operador SBX es el índice de distribución. Un índice de distribución pequeño provoca un grado de exploración elevado. De hecho, un índice de distribución de la unidad tiene un efecto similar al *Operador de Recombinación Difusa* (Fuzzy Recombination Operator) [62]. En la figura 6 se puede observar el efecto de aplicar distintos índices de distribución. Particularmente, en la parte izquierda se considera un índice de distribución pequeño, mientras en la parte derecha se considera un índice de distribución grande, se observa que este último genera soluciones candidatas similares a las soluciones padre.

En el algoritmo 3 se muestra la implementación del operador SBX. Este pseudocódigo está basado en la implementación que está integrado en el código NSGA-IIpropuesto por Deb y otros [19], el cual es considerada como la variante mas popular. Como parámetros de entrada se requieren dos soluciones padre (P_1 and P_2), y como salida se obtinen dos soluciones hijo (C_1 and C_2). El primero y el segundo componente que mencionados previamente corresponden a las líneas 5 y 9 respectivamente. Como es usual, el caso clásico del operador SBX es configurado asignando los parámetros $\delta_1 = \delta_2 = 0{,}5$ y $\eta_c = 20$. Es important notar que la implementación clásica no considera la dimensión de la variable o el criterio de paro en sus parámetros internos.

4.3. Propuesta

Basado en el análisis anterior y con el propósito de inducir un balance entre exploración e intensificación, se proponen las siguiente modificaciones. Primeramente, se modifica la probabilidad de alterar una variable (δ_1) durante la ejecución de forma dinámica. La intención de esta modificación es incrementar la capacidad de exploración en las primeras etapas, esto alterando de forma simultánea un conjunto de variables y conforme la execución procede se reduce el número de variables que son modificadas. El valor de δ_1 se cambia en base a un modelo lineal decreciente, donde inicialmente es fijado a 1,0 y entonces es decrementado hasta la mitad del total de generaciones con un valor de 0,5. Esta último valor es mantenido haste el final de la ejecución, es decir desde la mitad de la execución este parámetro se comporta similar a la implementación tradicional del SBX.

Tabla 5: Puntos de referencias para el indicador HV

Instancias	**Punto de referencia**
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]

Tabla 6: Información estadística de las métricas considerando dos objetivos

	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**

Para asignar el valor δ_1 se utiliza la ecuación (15), donde $G_{Transcurridas}$ corresponde a la generación actual y G_{Total} corresponde al número total de generaciones.

Similarmente, el segundo cambio está relacionado con la probabilidad de aplicar reflexiones $(1 - \delta_2)$. En este caso δ_2 es actualizado de acuerdo a la ecuación (15), por lo tanto la probabilidad de aplicar una reflexión incrementa de 0,0 a 0,5 durante la ejecución. Esta modificación se realiza con el propósito de evitar el comportamiento disruptivo de intercambiar variables en las primeras generaciones ya que esto provocaría modificaciones muy drásticas. De esta forma, sería mas sensato aplicar estas reflexiones una vez que los individuos convergen a cierto grado- Por lo tanto, esta probabilidad es incrementado a 0,5, siendo el valor utilizando en la implementación del SBX estándar.

$$\delta_1 = \delta_2 = max\left(0{,}5, 1{,}0 - \frac{G_{Transcurridas}}{G_{Total}}\right) \tag{15}$$

Finalmente, el índice de distribución también es modificado durante la ejecución. De esta forma, en la primeras etapas se promueve un íncide de distribución pequeño con el propósito de incrementar la capacidad de exploración del SBX. Posteriormente es decrementado de forma lineal, lo cual tiene implica que la curva de distribución se cierre, por lo tanto se promueve un mayor grado de intensificación en las etapas finales. El incremento lineal es indicado en la ecuación (16), por lo tanto el índice de distribución es alterado de 2 a 22. Es importante hacer mención que ya se han considerado modificaciones similares al índice de distribución [71], [30].

$$\eta_c = 2 + 20 \times \left(\frac{G_{Elapsed}}{G_{End}}\right) \tag{16}$$

4.4. Resultados

En este apartado se analizan los resultados obtenidos con las variantes dinámicas del SBX (DSBX). Cada uno de los casos estudiados y una propuesta final se integraron en los algoritmos NSGA-II, MOEA/D and SMS-EMOA. Primeramente, es analizado cada uno de los tres componentes previamente analizados. Posteriormente se construye un caso donde se consideran dos componentes de forma simultánea. Se consideran los problemas de prueba WFG [34], DTLZ [20] and UF [68]. Con el propósito de comparar nuestra extensión del SBX con otros operadores populares en la validación experimental incluímos una variante de evolución diferencial mejor conocida como DEMO [59]. Debido a que todos los algoritmos con estocástico cada ejecución se repitió 35 veces con distintas semilla La configuración global que es aplicada a todos los algoritmos es como es indica a continuación. Se asgina el criterio de paro a $25,000$ generaciones, el tamaño de la población a 100, se configuraron los problemas de prueba WFG con dos y tres objetivos, además se consideraron 24 variables, donde 20 variables eran considerados como parámetros de distancia y 4 se consideraron como parámetros de posición. Como es sugerido en [20] se consideran $n = M + r - 1$ variables de decisión en los problemas de prueba DTLZ, donde para los problemas DTLZ1, DTLZ2 - DTLZ6 y DTLZ7 se consideraron $r = \{5, 10, 20\}$ respectivamente. Para los problemas de prueba UF se asignaron 10 variable de decisión. Finalmente, se asignó al operador de mutación polinomial con una probabilidad de cruce de $1/n$ y

Tabla 7: Información estadística de las métricas considerando tres objetivos

	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

Tabla 8: Resúmen de las pruebas estadísticas

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

con un índice de distribución de 50, mientras que el operador de cruce SBX se asignó con una probabilidad de cruce de 0,9 y un índice de distribución de 20. A continuación se especifica la parametrización adicionel de cada algoritmo:

- **DEMO**: CR = 0.3 and F = 0.5.
- **SMS-EMOA**: desplazamiento para calcular el HV = 100.
- **MOEA/D**: tamaño de la vecindad = 10, el número de actualizaciones por subproblema (nr) = 2 y δ = 0,9. i

Para comparar los frentes obtenidos de cada método se utiliza el hipervolúmen normalizado (HV) y la *Distanca Generacional Invertida Modificada* (Inverted Generational Distance Plus - IGD+). En la tabla 5 se presentan los puntos de referencia utilizados para el indicador del hipervolúmen los cuales son similares a los utilizados en [4, 70]. Por otra parte para comparar los resultados estadísticamente (valores del IGD+ y HV), se siguió un procedimiento similar al que se propuso en [21]. En primer lugar se utilizó el test Shapiro-Wilk para comprobar si los resultados se ajustaban a una distribución Gaussiana. 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.

4.5. Análisis de cada compenente en el operador SBX

En este apartado se analiza el efecto que cada componente tiene de forma independente al ser dinámicamente modificado Basados en el algoritmo 3 el efecto de cada componente es analizado a través de cuatro casos Cada caso es descrito a continuación.

- **Caso 1**: Se aplica la versión estañdar del operador SBX donde $\delta_1 = \delta_2 = 0{,}5$ y $\eta_c = 20$.
- **Caso 2**: Se actualiza el valor δ_1 como se indica en la ecuación (15), $\delta_2 = 0{,}5$ y $\eta_c = 20$.
- **Caso 3**: Se actualiza el valor δ_2 como se indica en la ecuación (15), $\delta_1 = 0{,}5$ y $\eta_c = 20$.
- **Caso 4**: Se actualiza el índice de distribución de acuerdo a (16), $\delta_1 = \delta_2 = 0{,}5$.

En las tablas 6 y 7 se muestra información del HV normalizado [71] y del IGD+ [35] para cada caso (Posteriormente se analiza el Caso 5). Específicamente, se muestra la media del HV e IGD+ para todos los problemas en dos y tres objetivos. Se observa que considerando dos y tres objetivos el Caso 4 mejora al Caso 1, al Caso 2 y al Caso 3 en todos los algoritmos. Por lo tanto se aprecian los beneficios de incrementar el índice de distribución durante la ejecución. Esto sucede porque inicialmente la curva que corresponde a la distribución de probabilidad conduce a un grado elevado de exploración, mientras que al trancurrir las generaciones se cambia gradualmente a un procedo de intensificación. Por otra parte, al considerar tres objetivos el Caso 2 presentó un rendimiento menor que al Caso 1. Por lo tanto, se observa que alterar todas las variables provoca un comportamiento muy disruptivo. Quizás los resultados podrían mejorar si el parámetro δ_1 es alterado de una forma distinta, sin embargo esto se deja como trabajo futuro. Los análisis anteriores únicamente consideran la media obtenida en todos los problemas de prueba. Sin embargo, dependiendo en el tipo de problema el rendimiento del algoritmo podría cambiar. Esto se analiza en la siguiente sección. Adicionalmente, se anexan resultados detallados[2].

4.6. Modificación simiultánea de varios componentes

En base a los resultados obtenidos con anterioridad se propone una variante del operador SBX. En esta variante se considera el Caso 3 y el Caso 4 de forma simultánea, es decir existen un cambio dinámico en el parámetro δ_2 y en el índice de distribución. El Caso 2 donde se aplica un mecanismo para actualiza δ_1 no se considera debido que los beneficios con esta variante no fueron significativos. Particularmente, el Caso 5 es construido en base al Algoritmo 3 y es configurado como se indica a continuación. El parámetro δ_1 es asignado con 0,5 ya que es la forma de la versión estándar del SBX. En base al Caso 3, el parámetro δ_2 es actualizado de acuerdo a la ecuación (15). Finalmente, en base al Caso 4, el parámetro δ_2 es actualizado como se indica en la ecuación (16).

De acuerdo a la media del HV y del IGD+ que se obtuvieron en el Caso 5 (revisar tablas 6 and 7), se pueden observar los beneficios de integrando los Casos 3 y 4. En el caso de dos objetivos se observa una ventaja significativa, sin embargo en el caso de tres objetivo los Casos 4 y 5 son muy similares en base a la media. Además, al considerar tres objetivos los resultados que se obtuvieron en el Caso 5 son superior a los obtenidos con DE, mientras que al considerar la versión estándar de SBX se observa un deterioro. Por lo tanto, el operador SBX puede generar resultados similares o superiores que los algoritmos DEMO si éste primero es configurado de una forma adecuada.

Finalmente, debido a que los análisis previos únicamente consideran la media entre la problemas de prueba, a continuación se presenta un análisis adicional para comprender mejor las contribuciones de los distintos casos. Particularmente, se realizan comparaciones en pares en base a pruebas estadísticas entre los cinco casos donde es considerado el SBX y DSBX. Este procedimiento se realizó de forma independiente para cada el NSGA-II, MOEA/D y el SMS-EMOA. En la tabla 8, se muestra los resultados de las pruebas estadísticas. Para cada algoritmo y para cada caso, la columna "↑" reporta el número de comparaciones donde las pruebas estadísticas confirmaron la superioridad del caso correspondiente, mientras que en la

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

columna "↓" se reporta el número de veces donde este caso fue inferior y la columna "⟷" indica el número de comparaciones donde la diferencia estadística no fue significativamente distinto. Los beneficios obtenidos por el caso 5 son suficientemente claros. Únicamente el Caso 4 obtuvo mejores resultados que el Caso 5 al considerar tres objetivos y al algoritmo NSGA-II. Entonces, los resultados pueden mejorar aún mas al integrar varias modificaciones dinámicas. Mas aún, los resultados que confirman varias ventajas al comparar nuestras variantes con la versión estándar del operador SBX (Caso 1). El único caso que no es superiormente claro a la versión estándar del SBX es el Caso 2, el cual fue discutido con anterioridad.

5. Conclusiones y trabajo futuro

Referencias

[1] R. B. Agrawal, K. Deb, K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, Tech. rep. (1994).
[2] E. Alba, Parallel metaheuristics: a new class of algorithms, vol. 47, John Wiley & Sons, 2005.
[3] N. H. Awad, M. Z. Ali, P. N. Suganthan, R. G. Reynolds, An ensemble sinusoidal parameter adaptation incorporated with l-shade for solving cec2014 benchmark problems, in: Evolutionary Computation (CEC), 2016 IEEE Congress on, IEEE, 2016, pp. 2958–2965.
[4] J. A. M. Berenguer, C. A. C. Coello, Evolutionary many-objective optimization based on kuhn-munkres' algorithm, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2015, pp. 3–17.
[5] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume , European Journal of Operational Research 181 (3) (2007) 1653 – 1669.
[6] T. Blickle, L. Thiele, A comparison of selection schemes used in evolutionary algorithms, Evolutionary Computation 4 (4) (1996) 361–394.
[7] A. Bolufé-Röhler, S. Estévez-Velarde, A. Piad-Morffis, S. Chen, J. Montgomery, Differential evolution with thresheld convergence, in: Evolutionary Computation (CEC), 2013 IEEE Congress on, IEEE, 2013, pp. 40–47.
[8] J. Brest, M. S. Maučec, B. Bošković, il-shade: Improved l-shade algorithm for single objective real-parameter optimization, in: Evolutionary Computation (CEC), 2016 IEEE Congress on, IEEE, 2016, pp. 1188–1195.
[9] J. Brest, M. S. Maučec, B. Bošković, Single objective real-parameter optimization: Algorithm jso, in: Evolutionary Computation (CEC), 2017 IEEE Congress on, IEEE, 2017, pp. 1311–1318.
[10] E. Cantu-Paz, Efficient and accurate parallel genetic algorithms, vol. 1, Springer Science & Business Media, 2000.
[11] U. K. Chakraborty, Advances in differential evolution, vol. 143, Springer, 2008.
[12] S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE transactions on evolutionary computation 15 (1) (2011) 4–31.
[13] D. Dasgupta, Z. Michalewicz, Evolutionary algorithms in engineering applications, Springer Science & Business Media, 2013.
[14] K. A. De Jong, Analysis of the behavior of a class of genetic adaptive systems.
[15] K. A. De Jong, Evolutionary computation: a unified approach, MIT press, 2006.
[16] K. Deb, An investigation of niche and species formation in genetic function optimization, ICGA'89 (1989) 42–50.
[17] K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, Complex Systems 9 (3) (1994) 1–15.
[18] K. Deb, H.-G. Beyer, Self-adaptive genetic algorithms with simulated binary crossover, Secretary of the SFB 531, 1999.
[19] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[20] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, Springer, 2005.
[21] 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) (2010) 618–635.
[22] A. E. Eiben, C. A. Schippers, On evolutionary exploration and exploitation, Fundamenta Informaticae 35 (1-4) (1998) 35–50.
[23] A. E. Eiben, J. E. Smith, et al., Introduction to evolutionary computing, vol. 53, Springer, 2003.
[24] S. Elsayed, N. Hamza, R. Sarker, Testing united multi-operator evolutionary algorithms-ii on single objective optimization problems, in: Evolutionary Computation (CEC), 2016 IEEE Congress on, IEEE, 2016, pp. 2966–2973.
[25] L. J. Eshelman, The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination, in: G. J. Rawlins (ed.), Foundations of Genetic Algorithms, vol. 1, Morgan Kaufmann Publishers, 1991, pp. 265 – 283.
[26] L. J. Eshelman, Real coded genetic algorithms and interval-schemata, Foundations of genetic algorithms 2 (1993) 187–202.
[27] W. Gao, G. G. Yen, S. Liu, A cluster-based differential evolution with self-adaptive strategy for multimodal optimization, IEEE transactions on cybernetics 44 (8) (2014) 1314–1327.
[28] F. Glover, G. Kochenberger, Handbook of metaheuristics (international series in operations research and management science), JOURNAL-OPERATIONAL RESEARCH SOCIETY 56 (5) (2005) 614–614.
[29] J. J. Grefenstette, Optimization of control parameters for genetic algorithms, IEEE Transactions on systems, man, and cybernetics 16 (1) (1986) 122–128.
[30] M. Hamdan, The distribution index in polynomial mutation for evolutionary multiobjective optimisation algorithms: An experimental study, in: International Conference on Electronics Computer Technology, Kanyakumari, India, 2012.

[31] F. Herrera, M. Lozano, Adaptation of genetic algorithm parameters based on fuzzy logic controllers, Genetic Algorithms and Soft Computing 8 (1996) 95–125.
[32] F. Herrera, M. Lozano, Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions, Soft Computing 7 (8) (2003) 545–562.
[33] C. Horoba, F. Neumann, Benefits and drawbacks for the use of epsilon-dominance in evolutionary multi-objective optimization, in: Proceedings of the 10th annual conference on Genetic and evolutionary computation, ACM, 2008, pp. 641–648.
[34] S. Huband, L. Barone, L. While, P. Hingston, A Scalable Multi-objective Test Problem Toolkit, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 280–295.
[35] H. Ishibuchi, H. Masuda, Y. Tanigaki, Y. Nojima, Modified distance calculation in generational distance and inverted generational distance, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2015, pp. 110–125.
[36] H. Jain, K. Deb, Parent to mean-centric self-adaptation in sbx operator for real-parameter optimization, Swarm, Evolutionary, and Memetic Computing (2011) 299–306.
[37] T. Jansen, On the analysis of dynamic restart strategies for evolutionary algorithms, in: International conference on parallel problem solving from nature, Springer, 2002, pp. 33–43.
[38] V. K. Koumousis, C. P. Katsaras, A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance, IEEE Transactions on Evolutionary Computation 10 (1) (2006) 19–28.
[39] A. Kumar, R. K. Misra, D. Singh, Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase, in: Evolutionary Computation (CEC), 2017 IEEE Congress on, IEEE, 2017, pp. 1835–1842.
[40] F. Lobo, C. F. Lima, Z. Michalewicz, Parameter setting in evolutionary algorithms, vol. 54, Springer Science & Business Media, 2007.
[41] M. Lozano, F. Herrera, J. R. Cano, Replacement strategies to preserve useful diversity in steady-state genetic algorithms, Information Sciences 178 (23) (2008) 4421–4433.
[42] S. W. Mahfoud, Crowding and preselection revisited, Urbana 51 (1992) 61801.
[43] O. J. Mengshoel, S. F. Galán, A. De Dios, Adaptive generalized crowding for genetic algorithms, Information Sciences 258 (2014) 140–159.
[44] D. Molina, F. Moreno-García, F. Herrera, Analysis among winners of different ieee cec competitions on real-parameters optimization: Is there always improvement?, in: Evolutionary Computation (CEC), 2017 IEEE Congress on, IEEE, 2017, pp. 805–812.
[45] J. Montgomery, Differential evolution: Difference vectors and movement in solution space, in: Evolutionary Computation, 2009. CEC'09. IEEE Congress on, IEEE, 2009, pp. 2833–2840.
[46] J. Montgomery, S. Chen, An analysis of the operation of differential evolution at high and low crossover rates, in: Evolutionary Computation (CEC), 2010 IEEE Congress on, IEEE, 2010, pp. 1–8.
[47] J. Montgomery, S. Chen, A simple strategy for maintaining diversity and reducing crowding in differential evolution, in: Evolutionary Computation (CEC), 2012 IEEE Congress on, IEEE, 2012, pp. 1–8.
[48] H. Mühlenbein, D. Schlierkamp-Voosen, Predictive models for the breeder genetic algorithm i. continuous parameter optimization, Evolutionary computation 1 (1) (1993) 25–49.
[49] I. Ono, H. Kita, S. Kobayashi, Advances in evolutionary computing, chap. A Real-coded Genetic Algorithm Using the Unimodal Normal Distribution Crossover, Springer-Verlag New York, Inc., New York, NY, USA, 2003, pp. 213–237.
URL http://dl.acm.org/citation.cfm?id=903758.903767
[50] H. M. Pandey, A. Chaudhary, D. Mehrotra, A comparative review of approaches to prevent premature convergence in ga, Applied Soft Computing 24 (2014) 1047–1077.
[51] M. Pilát, Evolutionary multiobjective optimization: A short survey of the state-of-the-art, Proceedings of the Contributed Papers Part I-Mathematics and Computer Sciences, WDS, Prague, Czech (2010) 1–4.
[52] K. Price, R. M. Storn, J. A. Lampinen, Differential evolution: a practical approach to global optimization, Springer Science & Business Media, 2006.
[53] Â. A. Sá, A. O. Andrade, A. B. Soares, S. J. Nasuto, Exploration vs. exploitation in differential evolution, in: AISB 2008 Convention Communication, Interaction and Social Intelligence, vol. 1, 2008, p. 57.
[54] E. Segredo, C. Segura, C. León, A multiobjectivised memetic algorithm for the frequency assignment problem, in: Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE, 2011, pp. 1132–1139.
[55] E. Segredo, C. Segura, C. León, Fuzzy logic-controlled diversity-based multi-objective memetic algorithm applied to a frequency assignment problem, Engineering Applications of Artificial Intelligence 30 (2014) 199–212.
[56] 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) (2016) 3233–3246.
[57] A. G. Slim Bechikh, Rituparna Datta, Recent Advances in Evolutionary Multi-objective Optimization, springer.
[58] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. of Global Optimization 11 (4) (1997) 341–359.
URL http://dx.doi.org/10.1023/A:1008202821328
[59] T. Tušar, B. Filipič, Differential evolution versus genetic algorithms in multiobjective optimization, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 257–271.
[60] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, ACM Computing Surveys 45 (3) (2013) 35:1–35:33.
[61] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and Exploitation in Evolutionary Algorithms: A Survey, ACM Computing Surveys 45 (3) (2013) 35:1–35:33.

[62] H.-M. Voigt, H. Mühlenbein, D. Cvetkovic, Fuzzy recombination for the breeder genetic algorithm, in: Proc. Sixth Int. Conf. on Genetic Algorithms, Citeseer, 1995.
[63] M. Yang, C. Li, Z. Cai, J. Guan, Differential evolution with auto-enhanced population diversity, IEEE transactions on cybernetics 45 (2) (2015) 302–315.
[64] W.-J. Yu, M. Shen, W.-N. Chen, Z.-H. Zhan, Y.-J. Gong, Y. Lin, O. Liu, J. Zhang, Differential evolution with two-level parameter adaptation, IEEE Transactions on Cybernetics 44 (7) (2014) 1080–1099.
[65] S. Y. Yuen, C. K. Chow, A genetic algorithm that adaptively mutates and never revisits, IEEE transactions on evolutionary computation 13 (2) (2009) 454–472.
[66] D. Zaharie, Control of population diversity and adaptation in differential evolution algorithms, in: Proc. of MENDEL, vol. 9, 2003, pp. 41–46.
[67] Q. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.
[68] Q. Zhang, W. Liu, H. Li, The performance of a new version of moea/d on cec09 unconstrained mop test instances, in: Evolutionary Computation, 2009. CEC'09. IEEE Congress on, IEEE, 2009, pp. 203–208.
[69] L. Zhao, C.-j. Sun, X.-c. Huang, B.-x. Zhou, Differential evolution with strategy of improved population diversity, in: Control Conference (CCC), 2016 35th Chinese, IEEE, 2016, pp. 2784–2787.
[70] Q. Zhu, Q. Lin, Z. Du, Z. Liang, W. Wang, Z. Zhu, J. Chen, P. Huang, Z. Ming, A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm , Information Sciences 345 (2016) 177 – 198.
[71] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE transactions on Evolutionary Computation 3 (4) (1999) 257–271.