

Diversidad en los Algoritmos Evolutivos

Carlos Segura · Joel Chacón Castillo

Received: date / Accepted: date

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

1. Introducción

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

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

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

componentes seleccionados no son suficientes para escapar de esta región. Basado en esto se han desarrollado varias estrategias para aliviar este problema. Además en través de varios estudios se ha revelado que mantener una población diversa en un requisito previo para evitar la convergencia prematura ref6-ANDBRS. Sin embargo, si la población es muy diversa, entonces un grado adecuado de explotación podría ser prevenido, resultando en una convergencia lenta y por lo tanto soluciones de baja calidad. Por esta razón, Mahfoud ref10-ANDBRS utilizó el concepto de diversidad útil, con cual se refiere a la cantidad de diversidad que resulta en soluciones de alta calidad. En la literatura existen distintas formas para aliviar el problema de convergencia prematura ref40-TIPDMEALCO. En los 90s, la mayoría de estrategias para aliviar la convergencia prematura se centraron en modificar el esquema de selección de padres. La principal razón es que en esa época la mayoría de esquemas eran generacionales, por lo tanto la presión de selección estaba definida principalmente en la selección de padres. Sin embargo, se descubrió que tratando de aliviar la convergencia prematura donde únicamente sea considerada la selección de padres no fue suficiente ref4-TIPDMEALCO. Posteriormente, la una gran cantidad de EAs incorporaron una fase de reemplazo que abandonaron al menos parcialmente a los métodos generacionales iniciales. Basado en esto muchos autores descubrieron la posibilidad de incorporar métodos para aliviar el problema de convergencia prematura ref10-TIPDMEALCO. Es importante considerar que aún cuando los métodos de reemplazamiento generacionales ref11-ANDBRS fueron suficientemente populares, algunos autores ya habían tomado en cuenta esta estrategia ref30-TIPDMEALCO. Sin embargo, con la efectividad de elitismo y otras estrategias de reemplazo, el número de esquemas que adoptaron estos principios crecieron de forma considerable ref27-TIPDMEALCO.

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

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

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

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

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

2. Diseño de Evolución Diferencial basado en diversidad

2.1. Evolución Diferencia: Conceptos Básicos

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

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

2.1.1. Inicialización

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

2.1.2. Mutación

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

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

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

2.1.3. Cruza

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

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

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

2.1.4. Selección

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

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

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

2.2. Diversidad en Evolución Diferencial

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

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

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

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

2.3. Propuesta

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

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

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

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

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

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

Algorithm 1 Esquema general del DE-EDM

```

1: Inicializar de forma aleatoria a la población con  $NP$  individuos, donde cada uno es 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 (1).
6:     Cruza: Utilizar la recombinación para generar al vector de prueba ( $U_{i,G}$ ) de acuerdo a
       la ecuación (2).
7:     Selección: Actualizar al vector elite ( $E_{i,G}$  en lugar de  $X_{i,G}$ ) de acuerdo a la ecuación
       (3).
8:   Reemplazo: Seleccionar a los vectores objetivo ( $X_{G+1}$ ) de acuerdo a la ecuación (2).
9:    $G = G + 1$ 

```

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

La forma para calcular la distancia entre dos individuos es en base a la distancia Euclideana normalizada descrita en la ecuación 4, donde D es la dimensión del problema, y a_d, b_d son los límites menores y mayores de cada dimensión (d). En los casos donde *El conjunto de Candidatos* está vacío previamente a la selección 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).

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

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

Algorithm 2 Fase de Reemplazo

```

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

```

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

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

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

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

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

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

Siguiendo los principios de distintas variantes del SHADE [17,18], se consideran las evaluaciones a función en el proceso de generación aleatorio del factor de mutación F . Particularmente, cada valor F es una muestra de una distribución Cauchy (Ecuación 6).

$$Cauchy(0,5,0,5 * n_{fes}/max_n_{fes}) \quad (6)$$

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

2.4. Resultados

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

Es importante 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 sugerido en [22]. Debido a que todos los algoritmos son estocásticos se realizaron 51 ejecuciones con distintas semillas.

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

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

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

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

Concretamente, en primer lugar se utilizó el test Shapiro-Wilk para comprobar si los resultados se ajustaban a una distribución Gaussiana. En los casos en que sí se ajustaban, se utilizó el test de Levene para comprobar la homogeneidad de las varianzas, procediendo con el test de ANOVA en caso positivo o con el de Welch en caso negativo. Por otro lado, para los casos que no se ajustaban a distribuciones Gaussianas, se utilizó el test de Kruskal-Wallis. En todos los casos se fijó el nivel de confianza al 95 %. Se considera que un algoritmo X es superior a un algoritmo Y , si el procedimiento anterior reporta diferencias significativas y si la media y mediana del hipervolumen obtenido por el método X son superiores a las obtenidas por el método Y . En las tablas 1 y 2 se presenta un resumen de los resultados obtenidos para el CEC 2016 y el CEC 2017 respectivamente. La columna etiquetada con “Siempre 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. Prácticamente nuestra propuesta resolvió al menos una vez todas las funciones (28 funciones) que pertenecen al conjunto de problemas del CEC 2017. Adicionalmente, fueron resueltas al menos una vez 21 funciones que pertenecen al CEC 2016. Esta es una diferencia sustancial con los resultados obtenidos por los algoritmos que pertenecen al estado-del-arte. Estos algoritmos obtuvieron los valores óptimos significativamente en menos funciones. En orden para confirmar la superioridad del DE-EDM, se implementaron las pruebas estadísticas por pares. La columna etiquetada con el símbolo \uparrow muestra que el número de veces en que cada método fue superior, mientras que la columna etiquetada con \downarrow cuenta el número de casos donde el método fue inferior. Finalmente, la columna etiquetada con \longleftrightarrow muestra el número de comparaciones cuyas diferencias no fueron significativas. Las pruebas estadísticas indican que el DE-EDM alcanzó los mejores resultados en los dos años. De hecho el número de veces en que nuestra propuesta ganó en el CEC 2016 y el CEC 2017 fue de 77 y 88 respectivamente. Además el número de veces en que perdió fueron de 25 y 6 respectivamente. Adicionalmente, el último lugar alcanzado en los dos años fue por el algoritmo L-SHADE-Epsilon con 20 comparaciones positivas en el 2016 y 7 comparaciones positivas en el 2017. La última columna etiquetada con “Puntaje” muestra un análisis que fue propuesto en las competencias del CEC. Particularmente, este método de evaluación combina dos puntajes como se indica en la ecuación (7). Por lo tanto el puntaje final está compuesto por la suma $Score = Score_1 + Score_2$.

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

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

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

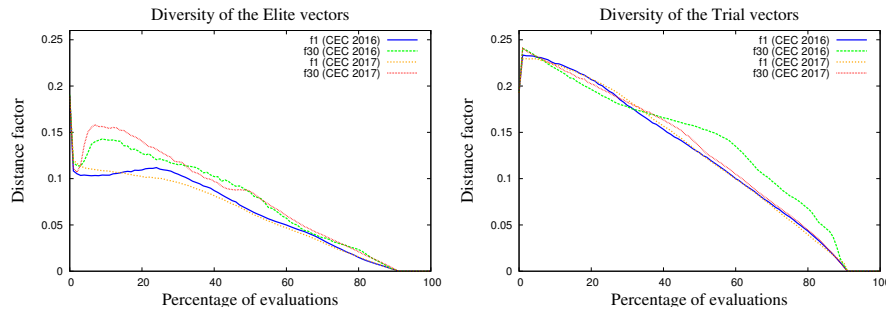


Figura 1 Promedio del DCN de las 51 ejecuciones con los problemas f_1 y f_{30} (CEC 2016 y CEC 2017). El factor de distancia inicial corresponde a $D_I = 0,3$.

Cuadro 1 Resumen de los resultados - CEC 2016

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

Cuadro 2 Resumen de los resultados - CEC 2017

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

Particularmente, se ejecutó el DE-EDM con las funciones F_1 y f_{30} . Basado en sus propiedades la 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 mantiene en la población Elite. A pesar de que existen mecanismos para evitar la pérdida de diversidad en la población Elite, se puede observar que se mantiene un grado de diversidad de forma implícito en las dos funciones. Similarmente, la parte derecha corresponde a la diversidad de los vectores de prueba. Esto demuestra que se demuestra un grado de diversidad de forma explícito, es decir hasta el 95 % del total de evaluaciones a función.

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

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

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

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

2.5. Análisis Empírico del factor de distancia inicial

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

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

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

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

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

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

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

Referencias

1. U.K. Chakraborty, *Advances in differential evolution*, vol. 143 (Springer, 2008)
2. R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *Journal of global optimization* **11**(4), 341 (1997)
3. S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE transactions on evolutionary computation* **15**(1), 4 (2011)
4. J.A. Nelder, R. Mead, A simplex method for function minimization, *The computer journal* **7**(4), 308 (1965)
5. W. Price, Global optimization by controlled random search, *Journal of Optimization Theory and Applications* **40**(3), 333 (1983)
6. J. Zhang, A.C. Sanderson, Jade: adaptive differential evolution with optional external archive, *IEEE Transactions on evolutionary computation* **13**(5), 945 (2009)
7. Â.A. Sá, A.O. Andrade, A.B. Soares, S.J. Nasuto, in *AISB 2008 Convention Communication, Interaction and Social Intelligence*, vol. 1 (2008), vol. 1, p. 57
8. J. Lampinen, I. Zelinka, et al., in *Proceedings of MENDEL* (2000), pp. 76–83
9. D. Zaharie, in *Proc. of MENDEL*, vol. 9 (2003), vol. 9, pp. 41–46

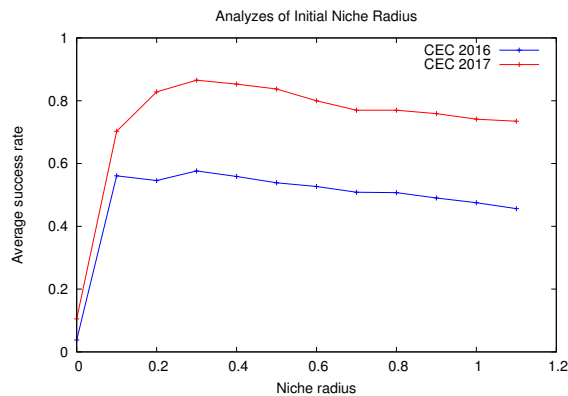


Figura 2 Razón de éxito promedio con distintos factores de distancia inicial con los problemas de prueba del CEC 2016 y CEC 2017, específicamente se considera una población de 250 individuos y 25,000,000 evaluaciones a función.

10. M. Yang, C. Li, Z. Cai, J. Guan, Differential evolution with auto-enhanced population diversity, *IEEE transactions on cybernetics* **45**(2), 302 (2015)
11. M. Črepinšek, S.H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, *ACM Computing Surveys* **45**(3), 35:1 (2013)
12. J. Montgomery, in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on* (IEEE, 2009), pp. 2833–2840
13. J. Montgomery, S. Chen, in *Evolutionary Computation (CEC), 2012 IEEE Congress on* (IEEE, 2012), pp. 1–8
14. A. Bolufé-Röhler, S. Estévez-Velarde, A. Piad-Morffis, S. Chen, J. Montgomery, in *Evolutionary Computation (CEC), 2013 IEEE Congress on* (IEEE, 2013), pp. 40–47
15. C. Segura, C.A.C. Coello, E. Segredo, A.H. Aguirre, A novel diversity-based replacement strategy for evolutionary algorithms, *IEEE transactions on cybernetics* **46**(12), 3233 (2016)
16. J. Montgomery, S. Chen, in *Evolutionary Computation (CEC), 2010 IEEE Congress on* (IEEE, 2010), pp. 1–8
17. N.H. Awad, M.Z. Ali, P.N. Suganthan, R.G. Reynolds, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 2958–2965
18. J. Brest, M.S. Maučec, B. Bošković, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 1188–1195
19. S. Elsayed, N. Hamza, R. Sarker, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 2966–2973
20. A. Kumar, R.K. Misra, D. Singh, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 1835–1842
21. J. Brest, M.S. Maučec, B. Bošković, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 1311–1318
22. D. Molina, F. Moreno-García, F. Herrera, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 805–812
23. J.J. Durillo, A.J. Nebro, C.A.C. Coello, J. Garcia-Nieto, F. Luna, E. Alba, A Study of Multiobjective Metaheuristics When Solving Parameter Scalable Problems, *IEEE Transactions on Evolutionary Computation* **14**(4), 618 (2010)