

# Algoritmos Genéticos con Codificación Real: Operadores de Cruce Híbridos Basados en Entornos con Múltiples Descendientes

Francisco Herrera<sup>1</sup>, Manuel Lozano<sup>1</sup>, Ana M<sup>a</sup> Sánchez<sup>2</sup>

**Resumen**—Cuando trabajamos con algoritmos genéticos con codificación real, algunos operadores de cruce son más efectivos que otros según se ajusten a las características del problema, incluso en distintas etapas del proceso en el mismo problema. Por esta razón se han propuesto técnicas que combinan distintos operadores de cruce como esquemas alternativos a la práctica común de aplicar un solo modelo de operador a todos los elementos de la población. También existen diversos modelos de operadores de cruce que generan múltiples descendientes y que presentan un mejor comportamiento que el mismo operador cuando solo genera dos hijos, logrando un mejor equilibrio entre exploración y explotación. En este trabajo presentamos un estudio de diversos operadores de cruce que generan múltiples descendientes a partir de dos padres, utilizando un operador distinto para cada par de hijos, lo que hemos denominado Operadores de cruce híbridos con múltiples descendientes

**Palabras clave**-- Algoritmos Genéticos, Codificación Real, Operador de Cruce Híbrido, Múltiples Descendientes

## I. INTRODUCCIÓN

En la formulación original de los algoritmos genéticos (AGs), las soluciones se han codificado usando el alfabeto binario. Sin embargo, se han aplicado codificaciones no binarias más naturales para los problemas particulares de aplicación. De entre ellas, destaca la codificación real, particularmente adecuada para problemas con parámetros en dominios continuos. Un cromosoma es un vector de números reales con un tamaño igual a la longitud del vector solución del problema. Los AGs basados en este tipo de codificación se denominan AGs con codificación real (AGCRs) [22]. En la actualidad, existe un creciente interés en la resolución de problemas de optimización reales usando AGCRs, en campos tan diversos como la

ingeniería industrial [2], biotecnología [32], economía [14], control [1], diseño aeroespacial [19], etc.

El operador de cruce juega un papel central en los AGs [12]. De hecho puede considerarse como una de las características que definen a estos algoritmos y es uno de los componentes a tener en cuenta para mejorar su eficacia. En el caso de los AGCRs, este operador influye decisivamente sobre el nivel de diversidad en la población, y por ello, es un factor determinante para evitar el problema de la convergencia prematura [6]. Esto explica que el principal esfuerzo en la investigación desarrollada para mejorar los AGCRs se centre en la propuesta y estudio de nuevos operadores de cruce [10].

Los operadores de cruce basados en entornos (OCEs) son una familia de operadores que actualmente reciben una especial atención [24]. Estos operadores generan los genes de los descendientes a partir de valores tomados del entorno asociado a los padres mediante distribuciones de probabilidad, normalmente a partir de un intervalo cuyos extremos dependen de los valores de los genes de los cromosomas padre. Ejemplos de estos operadores son BLX- $\alpha$  [7, 16], SBX- $\eta$  [9], FR-d [39] y PNX- $\eta$  [5] que están basados en distribuciones de probabilidad uniforme, polinomial, triangular y normal, respectivamente. El grado de diversidad de estos operadores puede variar dependiendo del valor del parámetro asociado al operador.

Normalmente, el operador de cruce se aplica sobre parejas de padres, generando dos hijos para cada una de ellas, los cuales se introducen en la población [18]. Sin embargo, también se han presentado operadores de cruce con *múltiples descendientes* [40, 38, 33, 17, 21], que generan más de dos descendientes por cada grupo de padres. En este caso, un mecanismo de selección de descendientes limita el número de hijos que entran a formar parte de la nueva población. Estos operadores, que están inspirados en la naturaleza, tratan de sacar más beneficio de los padres mediante el muestreo de un

<sup>1</sup>Dpto de Ciencias de la Computación e Inteligencia Artificial, Escuela Técnica Superior de Ingeniería Informática, Universidad de Granada, [herrera@decsai.ugr.es](mailto:herrera@decsai.ugr.es), [lozano@decsai.ugr.es](mailto:lozano@decsai.ugr.es)

<sup>2</sup>Dpto. de Lenguajes y Sistemas Informáticos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Granada. [amlopez@ugr.es](mailto:amlopez@ugr.es)

mayor número de posibles soluciones resultantes de su recombinación.

Por otra parte, a la hora de generar descendientes, dependiendo del problema con el que se esté trabajando, unos mecanismos de generación de descendientes serán más efectivos que otros según se ajusten a las características del problema. En este contexto, se hace especialmente atractiva la idea de aplicar conjuntamente diversos mecanismos de generación de descendientes sobre la población [28,30], con el propósito de asegurar que alguno de ellos se adecue a la forma concreta del problema a optimizar e incluso a la zona donde se centra actualmente la búsqueda. Diversos estudios de operadores de cruce híbridos contemplan esta idea [8, 23, 26, 27, 28, 36, 41]. Su objetivo es examinar la sinergia producida por la combinación de distintos operadores, y estudiar si el comportamiento de dicha combinación mejora el de los operadores simples que la componen.

La efectividad final del operador de cruce va a depender del equilibrio entre la diversidad asociada a los mecanismos de generación de descendientes y la presión selectiva derivada del mecanismo de selección de descendientes.

El propósito de este trabajo es analizar los efectos sinérgicos entre OCEs. Para ello, hemos diseñado operadores de cruce híbridos que generan múltiples descendientes para cada par de padres. Cada par de hijos se obtendrá mediante la aplicación de un OCE distinto. Esta propuesta va a ser analizada sobre 11 funciones/problemas que son utilizados usualmente para contrastar el rendimiento de los AGCRs.

El trabajo queda estructurado de la siguiente manera: En la Sección II describimos las principales características de los OCEs, así como distintos aspectos de las técnicas de hibridación y de múltiples descendientes. En la Sección III se presenta la propuesta de operadores de cruce híbridos con múltiples descendientes. En la Sección IV se muestra el estudio experimental, y el análisis de resultados. Por último, en la Sección V se presentan las conclusiones finales.

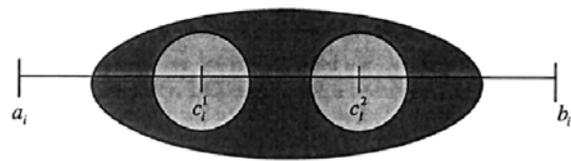
## II. PRELIMINARES

### A. Operadores de cruce basados en entornos

Estos operadores generan los genes de los descendientes a partir de valores tomados del entorno asociado a los padres,  $c_i^1, c_i^2$ , normalmente a partir de un intervalo  $[a_i, b_i]$  cuyos extremos dependen de los valores de los genes de los cromosomas padre. Este tipo de operadores tienen carácter aleatorio.

En la Figura 1 se observa el efecto de estos operadores.

Figura 1. Efecto de los OCE



Ejemplos de estos operadores son BLX- $\alpha$  [7, 16], SBX- $\eta$  [9], FR-d [39] y PNX- $\eta$  [5] que están basados en distribuciones de probabilidad uniforme, polinomial, triangular y normal, respectivamente.

Con SBX, FR y PNX cada gen del hijo se genera en el vecindario del correspondiente gen de uno de los padres. En cambio, BLX no muestra esta clara preferencia hacia zonas cercanas a los padres.

Todos ellos muestran dos importantes características:

- En su definición incluyen un parámetro que determina la extensión asociada con las distribuciones de probabilidad utilizadas para generar los hijos. El grado de diversidad inducido por estos operadores se puede ajustar variando el valor de estos parámetros.
- Definen una distribución de probabilidad de los hijos basada en la distancia de los padres. Si los padres están muy cerca el uno del otro, los hijos generados podrían distribuirse densamente alrededor de los padres. En cambio, si los padres están alejados entre sí, los hijos apenas se distribuirán a su alrededor.

### B. Operadores de cruce híbridos

Una idea interesante en el desarrollo de operadores de cruce consiste en la aplicación simultánea de diversos operadores sobre la población. De hecho, existen diversos estudios dirigidos al estudio de la sinergia producida por la combinación de distintos operadores [8, 23, 26, 27, 28, 36, 41].

Se han realizado distintos intentos para encontrar operadores sinérgicos:

1. Operadores de cruce híbridos. Estos operadores utilizan distintas clases de operadores de cruce para producir diversos hijos de los mismos padres. Por ejemplo, en [21], se presenta un híbrido que genera cuatro hijos por cada par de padres, aplicando dos operadores explorativos y dos explotativos. Los dos

hijos más prometedores sustituyen a los padres en la población.

2. AGs distribuidos heterogéneos. En [23], un AGCR mantiene, en paralelo, muchas subpoblaciones que son procesadas por AGs independientes que aplican diferentes operadores de cruce. Otros AGs distribuidos que hacen distinciones entre las subpoblaciones mediante la aplicación de diferentes operadores de cruce pueden encontrarse en [35] y [15].
3. Operadores de cruce adaptativos por probabilidad. Se dispone de un conjunto de operadores de cruce, cada uno de los cuales tiene una probabilidad de ser usado. Cada vez que se va a realizar el cruce, se selecciona un operador probabilísticamente. Además, un proceso adaptativo dinámico ajusta las probabilidades durante el proceso. Por ejemplo, en [8] y [29], los operadores que causan la generación de los mejores cromosomas tienen una mayor probabilidad, mientras que los operadores que generan hijos peores que los padres se usan menos frecuentemente. En [20], un AGCR aplica dos operadores de cruce diferentes, uno con propiedades de exploración y el otro de explotación. Un parámetro define la frecuencia de aplicación del operador explotativo. Cada cinco generaciones, un controlador lógico difuso evalúa dos medidas de diversidad de la población para ajustar este parámetro.

### C. Múltiples descendientes

Una de las formas en las que el operador de cruce proporciona diversidad es mediante la generación de múltiples descendientes. Evidentemente, cuanto más grande sea el número de descendientes mayor será la exploración que se haga del espacio de búsqueda, ya que las posibles soluciones aumentarán pudiendo estar localizadas en cualquier zona de dicho espacio. Este aumento de la exploración es el que favorece la diversidad.

Por otra parte, el hecho de generar múltiples descendientes conlleva la utilización de algún mecanismo de selección que determine cuáles de esos descendientes pasarán a formar parte de la nueva población. Este mecanismo de selección lleva asociada presión selectiva que puede derivar en convergencia prematura.

Utilizando estos modelos, el operador de cruce puede verse como un método de búsqueda local. Una vez que el AGCR ha encontrado zonas prometedoras del espacio de búsqueda, busca solo sobre una pequeña porción del vecindario alrededor de cada punto del espacio, con lo que se realizan múltiples exploraciones sobre puntos individuales en paralelo sobre sucesivas generaciones de una población. Esta estrategia denominada “muchos puntos pocos vecinos”, contrasta con la ascensión de colinas que potencialmente enfoca el esfuerzo sobre una gran fracción del vecindario de un punto pero solo alrededor de un punto cada vez. Esta última estrategia se denomina “pocos puntos muchos vecinos” [31].

Los métodos de múltiples descendientes llevan a cabo una búsqueda local sobre el vecindario de los padres envueltos en el cruce, con lo que el operador de cruce constituye un método de búsqueda local ya que produce hijos alrededor de los padres. Con la estrategia muchos puntos pocos vecinos, la búsqueda se hace más dispersa en cuanto a las zonas en las que se realiza la búsqueda, con poca profundidad en cada cromosoma padre.

La búsqueda local es una idea interesante ya que los operadores de cruce pueden generar hijos adaptativamente de acuerdo a la distribución de los padres. Al avanzar el número de generaciones, el AGCR pierde diversidad, lo que permite al cruce crear hijos distribuidos densamente alrededor de los padres, induciendo una efectiva búsqueda local.

Los múltiples descendientes se han utilizado tanto en AGs como en programación genética (PG). En el caso de la programación genética el primer autor que propuso este método fue Tackett [37], con la idea de evitar el efecto destructivo del cruce. Intentó modelar el hecho de que muchas especies animales producen muchos más hijos de los que realmente tienen esperanzas de sobrevivir, de tal forma que el exceso de hijos muere.

En general, tanto en AGs como en PG, el uso de múltiples descendientes promueve la idea de que explorando más combinaciones de una operación individual de cruce, hay más oportunidades de éxito, incluso aunque la evaluación sea menos eficiente, incrementándose la diversidad y efectividad del cruce.

En general, la potencia de este método reside en el hecho de que el cruce estándar elige los puntos de cruce de manera totalmente aleatoria, mientras que con múltiples descendientes se elige el mejor de un gran conjunto de tales recombinaciones aleatorias: Esto incrementa la probabilidad de que los hijos producidos mejoren a sus padres.

### III. PROPUESTA DE OPERADORES DE CRUCE HÍBRIDOS CON MÚLTIPLES DESCENDIENTES

Dado que, en AGCRs, los niveles de varianza en la población y por tanto de diversidad difieren del uso de unos operadores de cruce a otros, puede ser interesante combinar distintos operadores que aporten características complementarias en un intento de conseguir dicho objetivo. Los operadores de cruce híbridos combinan distintos operadores y facilitan el estudio de la sinergia entre dichos operadores.

Concretamente, la combinación de diferentes OCEs nos permite generar cada dos descendientes con una distribución de probabilidad diferente para crear los genes de los hijos en zonas restringidas del espacio de búsqueda alrededor de las regiones marcadas por los genes de los padres. Con SBX, FR y PNx cada gen del hijo se genera en el vecindario del correspondiente gen de uno de los padres. En cambio, BLX no muestra esta clara preferencia hacia zonas cercanas a los padres. Al combinar estos operadores se consigue una mayor diversidad.

El incremento de la diversidad también se consigue con la generación de múltiples descendientes, que hace posible una mayor exploración del espacio de búsqueda. Sin embargo, este aumento del número de descendientes conlleva la utilización de algún mecanismo de selección que determine cuales de esos descendientes pasarán a formar parte de la nueva población. Este mecanismo de selección lleva asociada presión selectiva que puede provocar una rápida convergencia que perjudique a la diversidad.

En esta sección, presentamos diversos modelos de operadores de cruce híbridos. Estos operadores generan múltiples descendientes (seis y ocho) para cada par de padres aplicando un OCE distinto para cada par de descendientes. Finalmente se seleccionan los dos mejores para formar parte de la nueva población.

En la Figura 2 se representa el esquema de funcionamiento de estos operadores en la generación y selección de múltiples descendientes.

Las diversas combinaciones que hemos realizado entre OCEs, se presentan en la Tabla I. Cinco de estas combinaciones consideran valores de los parámetros con alta diversidad. De ellas, el primer operador genera ocho descendientes, utilizando los cuatro OCEs, mientras que los otros cuatro operadores generan seis hijos, aplicando todas las posibles combinaciones entre los cuatro OCEs. El sexto operador genera seis hijos utilizando tres operadores con parámetros que ofrecen menor diversidad.

Figura 2. Esquema de funcionamiento de los operadores

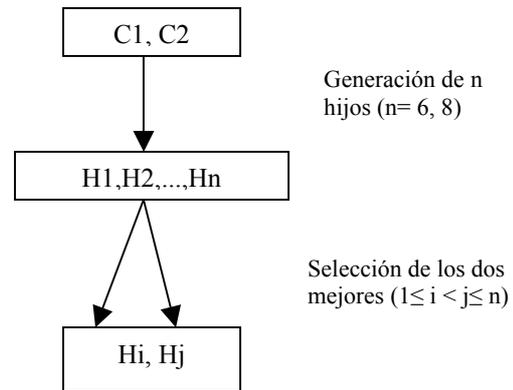


TABLA I. OPERADORES DE CRUCE UTILIZADOS

2BLX0.5-2FR0.5-2PNX3- 2SBX0.01
2BLX0.5-2FR0.5-2PNX3
2BLX0.5-2FR0.5- 2SBX0.01
2BLX0.5- 2PNX3- 2SBX0.01
2FR0.5-2PNX3-2SBX0.01
2BLX0.1-2FR0.3-2PNX4

### IV. ESTUDIO EXPERIMENTAL

En esta sección se estudia el comportamiento que presentan los operadores mencionados en la Tabla I. Para ello en la primera subsección se muestran los experimentos realizados y en la segunda se realiza un análisis de los resultados obtenidos.

#### A. Experimentación

En todos los experimentos se ha utilizado mutación no uniforme [22], con lo que la probabilidad de que un gen se mute disminuye según avanza la ejecución del algoritmo. De esta manera los cambios producidos en los genes son menores en las últimas generaciones, produciendo un ajuste local. El procedimiento de selección de la nueva población es el de ordenación lineal [3] asegurando la presencia del mejor individuo (elitismo) [11], y se ha utilizado el método de muestreo universal estocástico [4].

Hemos considerado once funciones de test frecuentemente utilizadas: Modelo Esférico, Función de Schwefel, Función de Rastrigin Generalizada, Función de Griewangk, Expansión de F10, Función de Rosenbrock Generalizada, Sistema de Ecuaciones Lineales, Problema de modulación de la frecuencia de sonido, Problema de aproximación polinomial, Función de Ackley y Función de Bohachevsky. La formulación de estos problemas se puede encontrar en [24]. Estas funciones varían con respecto a algunas características como continuidad, modalidad

o dimensiones, consiguiéndose de esta forma un amplio intervalo de posibles situaciones para la experimentación.

Los parámetros utilizados para llevar a cabo los experimentos figuran en la Tabla II. Los algoritmos se han ejecutado 30 veces, con valor de semilla distinto para el generador de números aleatorios.

TABLA II. PARÁMETROS DEL ALGORITMO GENÉTICO

Tamaño de la población	61
Probabilidad de cruce	0.6
Probabilidad de mutación	0.1
Número de evaluaciones	100000

Hemos utilizado cuatro tests no paramétricos, Bonferroni-Dunn, Friedman, Holm y Wilcoxon, para realizar un análisis estadístico de los resultados de los diferentes operadores [12, 25, 34].

### B. Análisis del comportamiento de los algoritmos

Para aplicar los tests, para cada función de evaluación, se han ordenado los operadores en base al valor medio de cada uno, para posteriormente, obtener la posición media de cada operador para el total de las funciones de evaluación. En la Tabla III se muestra la posición media de cada operador para las once funciones de evaluación.

TABLA III. RANKING DE LOS OPERADORES

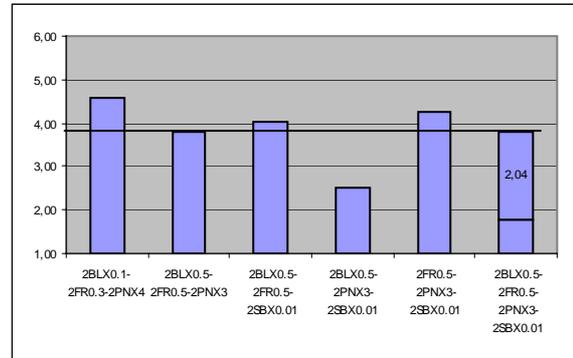
	Posición media
2BLX0.1-2FR0.3-2PNX4	4.59
2BLX0.5-2FR0.5-2PNX3	3.82
2BLX0.5-2FR0.5-2SBX0.01	4.05
2BLX0.5-2PNX3-2SBX0.01	2.50
2FR0.5-2PNX3-2SBX0.01	4.27
2BLX0.5-2FR0.5-2PNX3-2SBX0.01	1.77

El test de Friedman compara el rango de los algoritmos. Bajo la hipótesis nula, que establece que todos los algoritmos son equivalentes y por tanto sus rangos deberían ser iguales, el estadístico se distribuye de acuerdo a un valor obtenido en base al número de funciones y de algoritmos, con un grado de libertad igual al número de algoritmos menos uno. En nuestro caso, para seis operadores y once funciones, el valor obtenido, 17.41, es mayor que el valor crítico para  $\alpha = 0.05$ , 11.070, lo cual indica que existen diferencias significativas entre los operadores.

En la Figura 4, se muestran los resultados para el Test de Bonferroni-Dunn. Este test compara el algoritmo de mejor posición media con cada uno de los algoritmos restantes utilizando un valor de diferencia crítica que en nuestro caso es  $CD = 2.04$ . Podemos observar que el operador que genera ocho

descendientes, que es el que tiene el mejor valor medio, presenta diferencias con respecto a tres operadores.

Figura 4. ( $q_{0.05} = 2.576$ ,  $CD = 2.04$ )



Por último, el test de Holm para un error estándar  $SE = 0.7937$ , compara el algoritmo de mejor posición media con cada uno de los cinco restantes, encontrando diferencias significativas con cuatro de ellos ya que los correspondientes valores  $P$  son menores que el valor ajustado de  $\alpha$ . La Tabla IV muestra el resultado del test de Holm, con los algoritmos ordenados de mejor (1) a peor (5) rango de orden.

TABLA IV. TEST DE HOLM

i	$Z=(R_0 - R_i) / SE$	P	$\alpha/i$
5	$(1.77-4.59) / 0.7937 = 3.55$	0.0004	0.01
4	$(1.77-4.27) / 0.7937 = 3.14$	0.0016	0.0125
3	$(1.77-4.05) / 0.7937 = 2.87$	0.0042	0.17
2	$(1.77-3.82) / 0.7937 = 2.58$	0.0098	0.0125
1	$(1.77-2.5) / 0.7937 = 0.91$	0.3628	0.05

Así pues, podemos concluir que los operadores 2BLX0.5-2FR0.5-2PNX3-2SBX0.01 y 2BLX0.5-2PNX3-2SBX0.01, tienen un comportamiento similar entre sí, mejorando significativamente el del resto de operadores.

### C. Análisis del operador 2BLX0.5-2FR0.5-2PNX3-2SBX0.01 frente a los operadores simples

Con el objetivo de estudiar la sinergia entre los operadores, hemos comparado el operador 2BLX0.5-2FR0.5-2PNX3-2SBX0.01 con cada uno de los cuatro operadores clásicos que lo componen.

En la Tabla V se muestra la posición media de cada operador para las once funciones de evaluación, siendo el operador híbrido el que ocupa la mejor posición.

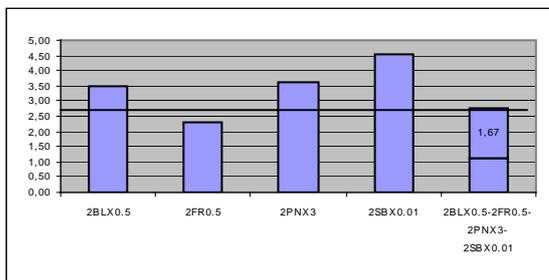
Tras aplicar el test de Friedman el valor obtenido, 31.66, es mayor que el valor crítico para  $\alpha = 0.05$ , 11.070, lo cual indica que existen diferencias significativas entre los operadores.

TABLA V. RANKING DE LOS OPERADORES

	Posición media
2BLX0.5	3,45
2FR0.5	2,27
2PNX3	3,64
2SBX0.01	4,55
2BLX0.5-2FR0.5-2PNX3- 2SBX0.01	1,09

En la Figura 5, se muestran los resultados para el Test de Bonferroni-Dunn, con  $CD = 1.67$ , poniéndose de manifiesto que, salvo con el operador FR-0.5, existen diferencias significativas con los demás.

Figura 5. ( $q_{0.05} = 2.498$ ,  $CD = 1.67$ )



El resultado del test de Holm para un error estándar  $SE = 0.6708$ , se muestra en la Tabla VI.

TABLA VI. TEST DE HOLM

i	$Z=(R_0- R_i /SE)$	P	$\alpha/i$
4	$(1,09-4,55)/0.6708 = 5,15$	$< 0.0002$	0.0125
3	$(1,09-3,64)/0.6708 = 3.80$	0.0002	0.17
2	$(1,09-3,45)/0.6708 = 3.51$	0.0004	0.0125
1	$(1,09-2,27)/0.6708 = 1.75$	0.0802	0.05

Dado que con el test de Holm, el operador híbrido tampoco presenta diferencias con respecto al operador FR-0.5, ya que el valor P es mayor que el valor crítico, hemos aplicado el test de Wilcoxon para compararlos entre sí. Este test ordena las diferencias de las medias de los dos operadores para cada función de evaluación, ignorando los signos, y comparando las posiciones para las diferencias positivas y negativas. Tras la aplicación del test, el valor obtenido 6, es menor que el valor crítico para  $\alpha = 0.05$ , 11. Puede concluirse por tanto que el operador es híbrido es mejor que el operador FR-0.5.

Este estudio pone de manifiesto que se produce sinergia entre los operadores que componen el operador híbrido, lo que hace que dicho operador ayude a mejorar el rendimiento de los AGCRs. Así, el operador híbrido reúne las propiedades necesarias en un operador de cruce efectivo, que son difíciles de conseguir con el uso de un operador de cruce clásico.

#### D. Análisis del operador 2BLX0.5-2FR0.5-2PNX3- 2SBX0.01 frente a los operadores con ocho descendientes homogéneos

También hemos comparado el operador híbrido con los operadores que lo componen cuando éstos generan ocho descendientes. En la Tabla VII se muestra la posición media de cada operador para las once funciones de evaluación.

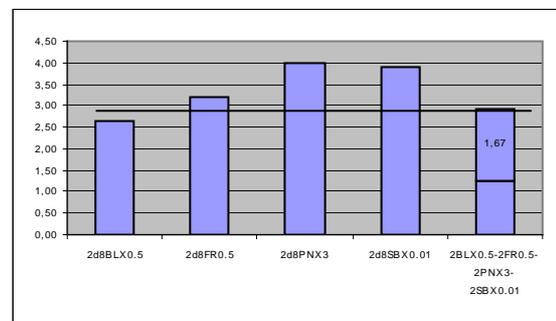
TABLA VII. RANKING DE LOS OPERADORES

	Posición media
2BLX0.5-2BLX0.5-2BLX0.5-2BLX0.5	2,64
2FR0.5-2FR0.5-2FR0.5-2FR0.5	3,18
2PNX3-2PNX3-2PNX3-2PNX3	4,00
2SBX0.01-2SBX0.01-2SBX0.01-2SBX0.01	3,91
2BLX0.5-2FR0.5-2PNX3- 2SBX0.01	1,27

Tras aplicar el test de Friedman el valor obtenido, 21.92, es mayor que el valor crítico para  $\alpha = 0.05$ , 11.070, lo cual indica que existen diferencias significativas entre los operadores.

En la Figura 6, se muestran los resultados para el Test de Bonferroni-Dunn, poniéndose de manifiesto que, salvo con el operador BLX-0.5, existen diferencias significativas con los demás.

Figura 6. ( $q_{0.05} = 2.498$ ,  $CD = 1.67$ )



El resultado del test de Holm para un error estándar  $SE = 0.6708$ , se muestra en la Tabla VIII, en la que se puede observar que el operador híbrido es mejor que el resto ya que los valores de P son menores que los valores críticos.

TABLA VIII. TEST DE HOLM

i	$Z=(R_0- R_i /SE)$	P	$\alpha/i$
4	$(1,27-4)/0.6708 = 4,06$	$< 0.0002$	0.0125
3	$(1,27-3,91)/0.6708 = 3.93$	$< 0.0002$	0.17
2	$(1,27-3,18)/0.6708 = 2.84$	0.0046	0.0125
1	$(1,27-2,64)/0.6708 = 2.04$	0.0414	0.05

Así pues, podemos concluir que cuando se generan múltiples descendientes, es interesante

utilizar operadores híbridos que combinen las características de los operadores clásicos para mejorar su eficiencia.

## VI. CONCLUSIONES

El hecho de generar múltiples descendientes hace posible una mayor exploración del espacio de búsqueda lo que también influye en el aumento de la diversidad, que se ve equilibrado con la presión selectiva asociada al mecanismo de selección de los dos mejores descendientes.

La combinación de diferentes OCEs con distintas distribuciones de probabilidad para crear los genes de los hijos, hace que se consiga una mayor diversidad, ya que se aportan características complementarias de los distintos operadores que forman parte del híbrido.

El operador que presenta un mejor comportamiento es el operador que genera ocho descendientes utilizando los cuatro OCEs: 2BLX0.5-2FR0.5-2PNX3-2SBX0.01.

Este operador se ha comparado con todos los OCEs clásicos que lo componen y que generan solo dos descendientes, así como con todas las posibles combinaciones de OCEs generando cuatro (por motivos de espacio no se presentan los resultados) y seis descendientes, cada dos con un operador distinto. También se ha comparado con los operadores que lo componen cuando éstos generan ocho descendientes. En todos los casos se pone de manifiesto que se produce una sinergia positiva entre los operadores que componen el operador híbrido, lo que hace que dicho operador ayude a mejorar el rendimiento de los AGCRs.

Dentro de esta línea, y como trabajo futuro, podría ser interesante realizar otras combinaciones con operadores que aporten nuevas características, como operadores basados en agregación y operadores discretos, así como ampliar el grupo de funciones de evaluación para hacer un análisis más exhaustivo.

## REFERENCIAS

[1] Y. Arfiadi, M. N. S. Hadi. Optimal direct (static) output feedback controller using real coded genetic algorithms. *Computers & Structures*, 79(17), 2001, 1625-1634.  
[2] P. N. Azariadis, A. C. Nearchou, N. A. Aspragathos. An evolutionary algorithm for generating planar developments of arbitrarily curved surfaces. *Computers in Industry*, Volume 47, Issue 3, March 2002, 357-368.

[3] J.E. Baker, Adaptive Selection Methods for Genetic Algorithms. En: *Proc. Of the First Int. Conf. On Genetic Algorithms and their Applications*, J.J. Grefenstette (Ed.) L. Erlbaum Associates, Hillsdale, MA, 1987, 14-21.  
[4] J.E. Baker, Reducing bias and inefficiency in the selection algorithm. En: *Proc. Of the Second Int. Conf. On Genetic Algorithms and their Applications*, J.J. Grefenstette, (Ed.), Hillsdale, NJ: Lawrence Erlbaum, 1987, 14-21.  
[5] P.J. Ballester, J.N. Carter. An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimisation. In: *Proc. of the Genetic and Evolutionary Computation Conference 2004*. Springer, LNCS 3102; 2004. 901-913.  
[6] H.-G. Beyer, K. Deb, On Self-Adaptive Features in Real Parameter Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, vol 5, no. 3, 2001, 250-270.  
[7] H.J. Bremermann, M. Rogson, S. Salaff. Global properties of evolution processes. In: Pattee HH, Edsack EA, Frin L, Callahan AB. Editors. *Natural Automata and Useful Simulations*. Washington, DC: Spartan; 1966. 3-41.  
[8] L. Davis, Adapting Operator Probabilities in Genetic Algorithms. In: *Proc. of the Third Int. Conf. on Genetic Algorithms*, J. David Schaffer (Ed.) (Morgan-Kaufmann Publishers, San Mateo, 1989), pp. 61-69.  
[9] K. Deb, R.B. Agrawal, Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9, 1995, 115-148.  
[10] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Chichester: Wiley, 2001.  
[11] K.A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Tesis Doctoral, Universidad de Michigan, 1975.  
[12] K.A. De Jong, W.M. Spears, A Formal Analysis of the Role of Multi-Point Crossover in Genetic Algorithms. *Annals of Mathematics and Artificial Intelligence*, 5(1), 1992, 1-26.  
[13] J. Demsar, Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 2006.  
[14] J. Duffy, P. D. McNelis, Approximating and simulating the stochastic growth model: Parameterized expectations, neural networks, and the genetic algorithm. *Journal of Economic Dynamics and Control*, Volume 25, Issue 9, September 2001, 1273-1303.  
[15] A. E. Eiben, I. G. Sprinkhuizen-Kuyper, B.A. Thijssen, Competing Crossovers in an Adaptive GA Framework. In: *Proc. IEEE Conf. Evolutionary Computation*, (IEEE Press, Piscataway, New Jersey, 1998), 787-792.  
[16] L.J. Eshelman, J.D. Schaffer. Real-coded genetic algorithms and interval-schemata. In: Whitley LD, editor. *Foundations of Genetic Algorithms 2*. San

- Mateo, CA: Morgan Kaufmann Publishers; 1993. 187-202.
- [17] S. Esquivel, A. Leiva, R. Gallard. Multiple crossover per couple in genetic algorithms. In: Proc. Of the 4th IEEE Int. Conf. on Evolutionary Computation (ICEC'97). Piscataway, NJ: IEEE Press; 1997. 103-106.
- [18] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, New York, 1989.
- [19] P. Hajela. Soft computing in multidisciplinary aerospace design. New directions for research, Progress in Aerospace Sciences, Volume 38, Issue 1, January 2002, 1-21.
- [20] F. Herrera, M. Lozano, Adaptation of Genetic Algorithm Parameters Based on Fuzzy Logic Controllers. In: Genetic Algorithms and Soft Computing, F. Herrera, J. L. Verdegay (Eds.) Physica-Verlag, 1996, 95-125.
- [21] F. Herrera, M. Lozano, J.L. Verdegay. Fuzzy connectives based crossover operators to model genetic algorithms population diversity. Fuzzy set and Systems 1997; 92(1): 21-30.
- [22] F. Herrera, M. Lozano, J.L. Verdegay, Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. Artificial Intelligence Review 12: 1998, 265-319 .
- [23] F. Herrera, M. Lozano, Gradual Distributed Real-Coded Genetic Algorithms. IEEE Transactions on Evolutionary Computation 4(1), 2000, 43-63.
- [24] F. Herrera, M. Lozano, A. Sánchez. A Taxonomy for the Crossover Operator for Real Coded Genetic Algorithms: An Experimental Study. International Journal of Intelligent Systems, Vol. 18, 2003, 309-338.
- [25] S. Holm, A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics, 6, 1979, 65-70.
- [26] I. Hong, A.B. Kahng, B.R. Moon, Exploiting Synergies of Multiple Crossovers: Initial Studies. In: Proc. of the Second IEEE Conference on Evolutionary Computation, (IEEE Press, Piscataway, New Jersey, 1995), pp. 245-250.
- [27] T.-P. Hong, H.-S. Wang, Automatically Adjusting Crossover Ratios of Multiple Crossover Operators. Journal of Information Science and Engineering 14(2), 1998, 369-390.
- [28] T.-P. Hong, H.-S. Wang, W.-Y. Lin, W.-Y. Lee, Evolution of Appropriate Crossover and Mutation Operators in a Genetic Process. Applied Intelligence 16, 2002, 7-17.
- [29] B. A. Julstrom, What Have you Done for me Lately? Adapting Operator Probabilities in a Steady-State Genetic Algorithm. In: Proc. of the Sixth Int. Conf. on Genetic Algorithms, L. Eshelman, (Ed.) (Morgan Kaufmann Publishers, San Francisco, 1995), 81-87.
- [30] I. Ono, H. Kita, S. Kobayashi. A Robust Real-Coded Genetic Algorithm using Unimodal Normal Distribution Crossover Augmented by Uniform Crossover: Effects of Self-Adaptation of Crossover Probabilities. Proc. of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann, CA, 1999, 496-503.
- [31] U.M. O'Reilly, F. Oppacher. Hybridized Crossover-Based Search Techniques for Program Discovery. IEEE International Conference on Evolutionary Computation, 1995. 573-578.
- [32] J. A. Roubos, G. van Straten, A. J. B. van Boxtel. An evolutionary strategy for fed-batch bioreactor optimization; concepts and performance, Journal of Biotechnology, Volume 67, Issues 2-3, 22 January 1999, 173-187.
- [33] H. Satoh, M. Yamamura, S. Kobayashi. Minimal generation gap model for GAs considering both exploration and exploitation. In: Proc. Methodologies for the Conception, Design and Application of Intelligent Systems (IIZUKA'96). 1996. 494-497.
- [34] D.J. Sheskin, Handbook of parametric and non parametric statistical procedures. Chapman & Hall/CRC. 2004.
- [35] D. Schlierkamp-Voosen, H. Mühlenbein. Strategy adaptation by competing subpopulations. In: Paralell Problem Solving from Nature 3, Y. Davidor, H.P. Schwefel, R. Männer, (Eds.) (Springer-Verlag, Berlin, Germany, 1994), 199-208.
- [36] W.M. Spears, Adapting Crossover in a Genetic Algorithm. In Proc. of the Fourth Annual Conference on Evolutionary Programming, J.R. McDonnell, R.G. Reynolds, D. Fogel (Eds.) (The MIT Press, 1995), 367-384.
- [37] W.A. Tackett, Recombination, Selection, and the Genetic Construction of Computer Programs. PhD thesis, University of Southern California, Department of Electrical Engineering Systems. 1994
- [38] W.A. Tackett. Greedy recombination and genetic search on the space of computer programs. In: Foundations of Genetic Algorithms 3. San Francisco: Morgan Kaufmann; 1994. 271-297.
- [39] H.M. Voigt, H. Mühlenbein, D. Cvetkovic. Fuzzy recombination for the breeder genetic algorithm. In: Eshelman L, editor. Proc. of the Sixth Int. Conf. on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann Publishers; 1995. 104-111.
- [40] A. Wright. Genetic algorithms for real parameter optimization. In: Rawlin GJE, editor. Foundations of Genetic Algorithms 1. San Mateo, CA: Morgan Kaufmann; 1991. 205-218.
- [41] H.-S. Yoon, B.-R. Moon, An Empirical Study on the Synergy of Multiple Crossover Operators. IEEE Transactions on Evolutionary Computation 6(2), 2002, 212-223.