

Estudio de Operadores de Cruce con Múltiples Descendientes para Algoritmos Genéticos con Codificación Real

Francisco Herrera,
Manuel Lozano

Dept. de Inteligencia Artificial y Ciencias de la
Computación
ETS Ingeniería Informática
Univ. de Granada
18071 Granada
{[herrera](mailto:herrera@decsai.ugr.es), lozano@decsai.ugr.es}

Ana M^a Sánchez,
Pedro Villar

Dept. de Lenguajes y Sistemas Informáticos
ETS Ingeniería Informática
Univ. de Granada
18071 Granada
{[amlopez](mailto:amlopez@ugr.es), pwillarc@ugr.es}

Resumen

En la mayoría de los casos, 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. Sin embargo, con el objetivo de incrementar la diversidad y efectividad del cruce, se han presentado operadores con *múltiples descendientes* 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 mayor número de posibles soluciones resultantes de su recombinación. 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.

En este trabajo haremos un estudio de los principales modelos existentes en la literatura que utilizan múltiples descendientes.

1. El operador de cruce

El operador de cruce es un mecanismo para compartir información entre los cromosomas. Generalmente se combinan las características (genes) de dos cromosomas padres para generar dos cromosomas hijos. El operador de cruce juega un papel central en los Algoritmos Genéticos

(AGs) [10]. De hecho puede considerarse como una de las características que definen y diferencian a los AGs de otros algoritmos basados en la evolución natural, siendo uno de los componentes que más se ha tenido en cuenta para mejorar el comportamiento de los AGs. En el caso de los Algoritmos Genéticos con Codificación Real (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 [4, 19]. 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 [8].

La aplicación del operador de cruce se realiza mediante tres mecanismos:

- Selección de padres: Determina como se emparejan los cromosomas en grupos para aplicarles el cruce.
- Generación de descendientes: Produce nuevos cromosomas a partir de los padres emparejados anteriormente. El parecido existente entre los padres y los descendientes determina el nivel de diversidad.
- Selección de descendientes: Escoge, entre todos los descendientes generados, los que formarán parte de la nueva población.

Según esto, el operador de cruce puede definirse en base a tres parámetros m , t y n : *Operador de cruce* (m , t , n), que indican el número de padres utilizados ($m \geq 2$), el número de descendientes generados ($t \geq 1$) y el número de descendientes seleccionados ($1 \leq n \leq t$) respectivamente.

Habitualmente, 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 [17]. Sin embargo, existen operadores de cruce con *múltiples padres* [11, 22, 24, 26], que combinan las características de más de dos padres para generar hijos. Su objetivo es introducir diversidad en la población, ya que mezclan información procedente de diversos padres. También se han presentado operadores de cruce con *múltiples descendientes* [14, 15, 18], 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 mayor número de posibles soluciones resultantes de su recombinación.

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. En este artículo se presentan distintas alternativas para conseguir este equilibrio en el que intervienen como factores primordiales el número de padres, el número de descendientes y la naturaleza de los mecanismos de generación y selección de descendientes.

El trabajo queda estructurado de la siguiente manera: En la Sección 2 se destaca la importancia de los múltiples descendientes. En la Sección 3 se presentan varios modelos de múltiples descendientes citados en la literatura. Por último, en la Sección 4 se presentan una serie de conclusiones y posibles trabajos futuros.

2. 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 cuales 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.

En la Figura 1 se representa la idea de los múltiples descendientes. En este caso, a partir de dos padres, se aplican N operaciones de cruce generando cada una de ellas dos descendientes de los cuales se seleccionarán los dos mejores.

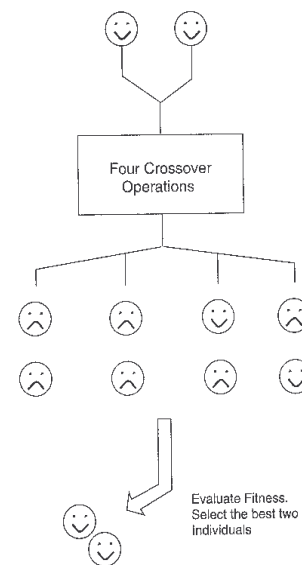


Figura 1. Múltiples descendientes

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 *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 estrategia se llama *pocos puntos muchos vecinos* [25].

Los métodos de múltiples descendientes implementan esta estrategia, ya que 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 muchos hijos alrededor de los padres y por tanto la búsqueda se centra en una amplia zona alrededor de un punto determinado. Con la estrategia muchos puntos pocos vecinos, la búsqueda se hace más dispersa en cuanto a las zonas en las que se busca pero se centra solo en algunos puntos de dichas zonas.

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 [28], 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.

3. Modelos de Múltiples Descendientes

En esta sección se presentan distintas propuestas sobre la utilización de múltiples descendientes presentes en la literatura.

3.1. Multiple Crossover per Couple (MCPC)

El modelo MCPC se desarrolló con codificación binaria usando un operador representativo de dicha codificación, Cruce Simple, generando múltiples descendientes [14, 15, 16]. Los pasos seguidos en la generación y selección de descendientes son los siguientes:

1. Se generan múltiples descendientes a partir de dos padres, utilizando más de una operación de cruce, introduciendo todos los descendientes en la nueva población
2. El número de hijos puede variar hasta un número máximo predefinido, controlando que no se exceda el tamaño de la población

Al compararlo con el mismo operador con solo dos descendientes se observó que los resultados eran igual de buenos pero se obtenían en menor tiempo.

3.2. Minimal Generation Gap (MGG)

Este método se desarrolló tanto con codificación real (cruce NDX) como con codificación binaria (Cruce Simple) [27]. Con este modelo, a partir de dos padres se obtienen múltiples descendientes siguiendo el proceso que se cita a continuación:

1. Seleccionar un par de padres de forma aleatoria.
2. Generar l hijos a partir de los padres usando cruce.
3. Los padres se reemplazan, uno por el mejor hijo, y otro por una solución obtenida aplicando la ruleta al conjunto de los dos padres y los l hijos

En este método, por una parte se está favoreciendo la explotación al asegurar la permanencia del mejor individuo pero, el hecho de que todos los descendientes compitan con los padres para formar parte de la nueva población evita que se produzca convergencia prematura.

Se realizaron distintas comparativas con otros modelos que utilizan distintas formas de llevar a cabo la actualización de la población y la selección para la reproducción y para la nueva población, tales como SS, CHC, IGS, SGA y ER. Todos estos modelos presentan una alta

explotación, siendo los modelos MGG y ER los que alcanzan un mejor equilibrio exploración-explotación. Además, el modelo MGG es el que obtiene mejores resultados.

3.3. Max-Min-Aritmético

Este modelo se basa en un operador de cruce que genera cuatro hijos, dos mediante cruce aritmético y dos mediante Max-Min [18]. Con este tipo de cruce se realiza un mecanismo de selección de descendientes consistente en que los dos cromosomas mejores son seleccionados como descendientes finales para la nueva población.

Al compararlo con operadores clásicos, como BLX- α , Lineal, Discreto y Simple, que generan solo dos descendientes, se obtienen mejores resultados.

3.4. Distance Dependent Alternation (DDA)

El modelo DDA [29], desarrollado para codificación real y con el operador de cruce UNDX-m, se planteó como una extensión del modelo MGG.

El modelo DDA está basado en la distancia existente entre los padres y los hijos generados, de tal forma que el número de padres y de hijos puede ser cualquiera y la decisión de quien pasa a formar parte de la nueva población viene determinada por dicha distancia. Los pasos seguidos se muestran a continuación:

1. Seleccionar m padres aleatoriamente
2. Generar l hijos a partir de los padres usando cruce
3. Si el mejor hijo es superior al padre más cercano, lo reemplaza
4. En otro caso, se busca el padre más lejano, y si el mejor hijo es superior, lo reemplaza

Este modelo es más robusto consiguiendo el óptimo que el modelo MGG. A partir de él se desarrolló una extensión, Distance Dependent Multiples Alternation [29], agrupando los hijos en torno a un padre, y realizándose el proceso en todos los grupos, de tal forma que en cada grupo se sustituirá un padre por el mejor hijo.

3.5. Family Competition

En [31, 32] se presenta un modelo de competición de familia en el que cada individuo de los N que componen la población, secuencialmente, llega a ser el "padre de familia". Este padre y otro escogido aleatoriamente del resto de la población inicial se cruzan generando un nuevo descendiente, al cual se le puede aplicar mutación. Este procedimiento se repite L veces para cada padre de familia, obteniendo al final L hijos de los cuales solamente sobrevivirá el que tenga mejor valor.

El hecho de que los L hijos se obtengan a partir del mismo padre de familia, llevándose a cabo una selección, hace que sea una estrategia de competición de familia. La nueva población se puede obtener de dos formas:

- Selección de familia: Para cada par formado por un padre y su mejor hijo, sobrevive el que tenga mejor valor.
- Selección de población: Se eligen los N mejores individuos entre los N padres y los N hijos.

Este modelo consigue resultados más estables y un comportamiento más robusto que otros algoritmos y estrategias de evolución como AGs simples, CHC y Genitor.

3.6. Generalized Generation Gap (G3)

El modelo G3 se desarrolló junto con el operador PCX [6, 7] y genera múltiples descendientes a partir del siguiente proceso:

1. Seleccionar el mejor padre y otros $m-l$ padres aleatoriamente
2. Generar l hijos a partir de los padres usando cruce
3. Escoger aleatoriamente dos de los padres
4. Reemplazar estos padres con los dos mejores elementos entre ellos y los l hijos generados.

Con este modelo se reemplaza la ruleta utilizada en el modelo MGG por una selección de las dos mejores soluciones. Preserva las soluciones elitistas de la iteración anterior, conservando para la nueva población los dos mejores individuos.

Al comparar este modelo con otros operadores y estrategias tales como Self-Adaptive Evolution Strategy, Differential Evolution Method y Quasi—Newton Method, resulta ser más consistente y obtiene mejores resultados.

3.7. Scaled Probabilistic Crossover (SPC)

Este modelo se desarrolló con los operadores de cruce SBX-0.01 (vSBX) y PNX-2.0 [3]. SPC trabaja según los pasos que se citan a continuación:

1. Seleccionar dos padres con selección aleatoria uniforme, sin reemplazamiento
2. Producir n hijos
3. Combinar los hijos y la población actual a fin de que el tamaño de la población permanezca constante

En los experimentos llevados a cabo para su estudio se comparó su comportamiento con el modelo G3, pudiendo comprobarse que, si bien para problemas con pocos óptimos locales no lo mejora, para problemas con muchos óptimos locales es mucho mejor.

3.8. Multiple Crossover per Couple with Selection of the Two Best Offspring

En [21] se presenta un nuevo modelo de múltiples descendientes. El proceso de generación y selección de descendientes se realiza con los siguientes pasos:

1. Seleccionar dos padres aleatoriamente
2. Generar n hijos a partir de los padres usando cruce
3. Reemplazar estos padres con los dos mejores hijos

Los operadores de cruce utilizados para realizar el estudio son: BLX-0.5 [13], FR-0.5 [30], SBX-1 [5], Aritmético [23] y en Dos Puntos [12]. Estos operadores se utilizan en la generación de 4, 6, 8, 16, 32, 64 y 128 descendientes, de los cuales se seleccionan los dos mejores (cruce 2-4-2, 2-6-2, 2-8-2, 2-16-2, 2-32-2, 2-64-2 y 2-128-2

respectivamente) 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.

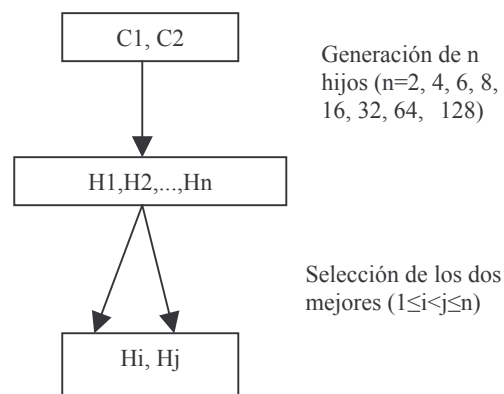


Figura 2. Esquema de funcionamiento de los operadores

En todos los experimentos desarrollados para este modelo se utilizó mutación no uniforme [19], 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 [1] asegurando la presencia del mejor individuo (elitismo) [9], y se utiliza el método de muestreo universal estocástico [2].

En la experimentación se utilizaron trece funciones de test: 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, Función de Bohachevsky, Función de Watson, y Función de Colville. La formulación de estos problemas se puede encontrar en [20]. 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. La dimensión del espacio de búsqueda es 25.

Los parámetros utilizados para llevar a cabo los experimentos figuran en la Tabla 1. Los algoritmos se ejecutaron 30 veces, con valor de semilla distinto para el generador de números aleatorios.

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

Tabla 1. Parámetros del Algoritmo Genético

En la Figura 3 se muestran los resultados obtenidos por cada uno de los operadores según el número de hijos generados. En ellas se muestran los porcentajes en que cada operador de cruce, en base al número de descendientes generados, obtiene los mejores valores medios (se ha aplicado

un t-test con nivel 0.05 de significancia) para cada una de las funciones de evaluación (trece consideradas).

El comportamiento de los operadores varía según el número de descendientes generados y las características asociadas a cada operador. Para los operadores BLX-0.5 y FR-0.5, se observa como los mejores resultados se consiguen con ocho y dieciseis descendientes.

Con el operador SBX-1 los mejores resultados se consiguen con cuatro descendientes, siendo también significativos los resultados alcanzados con seis y ocho descendientes. En general, para estos tres operadores, los mejores resultados se consiguen cuando el número de descendientes oscila entre cuatro y dieciseis.

Estos operadores presentan una alta diversidad en la generación de descendientes. El aumento del número de descendientes y selección de los dos mejores, que conlleva un aumento de la presión selectiva, mejora los resultados obtenidos por el operador (2,2,2) clásico. Sin embargo esto es así solamente hasta cierto número de descendientes.

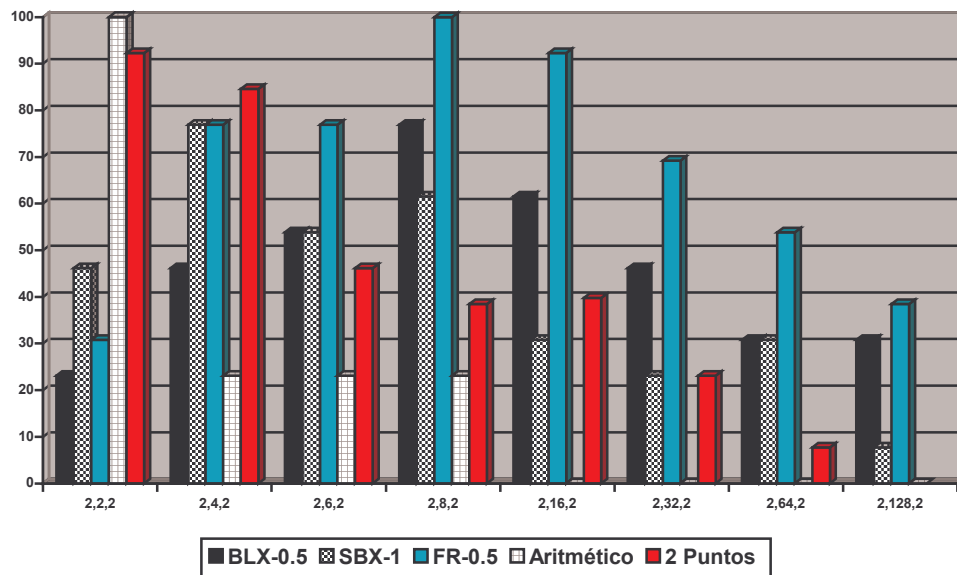


Figura 3. Resultados obtenidos por los distintos operadores

Hasta ocho descendientes se mantiene un buen equilibrio entre exploración y explotación, ya que la presión selectiva no es muy grande y hay bastante diversidad en la población. En cambio, al pasar de ocho descendientes, también se incrementa la presión selectiva, lo que provoca una pérdida de diversidad.

Para los operadores de cruce **aritmético** y en **2 puntos**, que presentan una mayor explotación y por tanto menor diversidad asociada, el aumento de descendientes y selección de los dos mejores no consigue buenos resultados debido a que se produce una explotación demasiado grande.

4. Conclusiones y Trabajo Futuro

En este trabajo hemos presentado varios modelos de múltiples descendientes que, mostrando distintas características, persiguen un mismo objetivo: el equilibrio entre la diversidad asociada a los mecanismos de generación de descendientes y la presión selectiva derivada del mecanismo de selección de descendientes.

Tras analizar los distintos modelos existentes se puede concluir que con estos modelos se consigue una mejora significativa del comportamiento del AG con respecto a operadores de cruce que solo generan dos descendientes.

Teniendo en cuenta las conclusiones obtenidas por los distintos autores citados, todos ellos coinciden en que estos modelos son más estables, robustos y consistentes, obteniendo mejores resultados y alcanzando un buen equilibrio entre exploración y explotación. Dada la importancia de estos modelos, cabe plantearse algunas líneas de trabajos futuros como el diseño de modelos de múltiples descendientes donde el número de hijos crezca según avance la evolución. Otra posible línea futura es estudiar modelos de selección de descendientes alternativos. También sería interesante realizar un estudio comparativo entre los distintos modelos citados.

Agradecimientos

Este trabajo ha sido realizado gracias al proyecto TIC2002-04036-C05-01.

Referencias

- [1] 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, pp. 14-21. 1987
- [2] 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.) L. Hillsdale, NJ: Lawrence Erlbaum, pp. 14-21. 1987
- [3] P. J. Ballester, J. N. Carter. An Effective Real-Parameter Genetic Algorithm with Parent Centric Normal Crossover for Multimodal Optimization. GECCO 2004, LNCS 3102, pp. 901-913. 2004
- [4] H.G. Beyer, K. Deb, On Self-Adaptive Features in Real Parameter Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation, vol 5, no. 3, pp. 250-270. 2001
- [5] K. Deb, R.B. Agrawal. Simulated Binary Crossover for Continuous Search Space. Complex Systems, 9, pp. 115-148. 1995
- [6] K. Deb, D. Joshi, A. Anand. Real Coded Evolutionary Algorithms with Parent-Centric Recombination. Proc. of the 2002 Congress on Evolutionary Computation, pp. 61-66. 2002
- [7] K. Deb, A. Anand, D. Joshi. A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization. Evolutionary Computation Journal 10(4), pp. 371-395. 2002
- [8] K. Deb, Multi-objective optimization using evolutionary algorithms. Chichester: Wiley, 2001.
- [9] K.A. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Tesis Doctoral, Universidad de Michigan, 1975
- [10] 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), pp. 1-26. 1992
- [11] A.E. Eiben, T. Bäck, An Empirical Investigation of Multi-Parent Recombination Operators in Evolution Strategies. Evolutionary Computation, 5(3), pp. 347-365. 1997
- [12] L.J. Eshelman, A. Caruana, J.D. Schaffer, Biases in the Crossover Landscape. Proc. of the Third Int. Conf. On Genetic Algorithms, pp. 86-91. 1989
- [13] L. J. Eshelman, J. D. Schaffer, Real-Coded Genetic Algorithms and Interval-Schemata. Foundations of Genetic Algorithms 2, pp. 187-202. 1993

- [14] S. Esquivel, R. Gallard, Z. Michalewicz. MPC: Another Approach to Crossover in Genetic Algorithms. *Actas del Primer Congreso de Ciencias de la Computación*, pp. 141-150. 1995
- [15] S. Esquivel, A. Leiva, R. Gallard, Multiple Crossover per Couple in Genetic Algorithms. *Proc. of the 4th IEEE Int. Conf. on Evolutionary Computation (ICEC'97)*, pp. 103-106. 1997
- [16] S. Esquivel, S. Ferrero, R. Gallard, H. Alfonso, M. Schütz. Enhanced Evolutionary Algorithms for Single and Multiobjective Optimization in the Job Shop Scheduling Problem. *Knowledge-Based Systems 15*, pp. 13-25. 2002
- [17] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York, 1989
- [18] F. Herrera, M. Lozano, J.L. Verdegay. Fuzzy Connectives Based Crossover Operators to Model Genetic Algorithms Population Diversity. *Fuzzy Sets and Systems 92:1*, pp. 21-30. 1997
- [19] F. Herrera, M. Lozano, J.L. Verdegay. Tackling Real-Coded Genetic Algorithms: Operators and tools for the Behaviour Analysis. *Artificial Intelligence Review 12*, pp. 265-319. 1998
- [20] F. Herrera, M. Lozano, A. Sánchez. A Taxonomy of Crossover Operators in Real Coded Genetic Algorithms: Classification and Analysis of their Behaviour. *International Journal of Intelligent Systems*, Vol. 18, pp. 309-338. 2003
- [21] F. Herrera, M. Lozano, A. Sánchez. Operadores de cruce con múltiples descendientes para algoritmos genéticos con codificación real: Estudio experimental. *III Congreso español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB03)*, pp- 292-299. 2003
- [22] H. Kita, I. Ono, S. Kobayashi. Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms. *Proc. of the Congress on Evolutionary Computation*, pp. 1581-1587. 1999
- [23] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York. 1992
- [24] Z. Michalewicz, G. Nazhiyath, and M. Michalewicz, A Note on Usefulness of Geometrical Crossover for Numerical Optimization Problems. En: *Proceedings of the 5th Annual Conference on Evolutionary Programming*, San Diego, CA, 29 February - 3 March. MIT Press, Cambridge, MA, pp. 305-312. 1996
- [25] U.M. O'Reilly, F. Oppacher. Hybridized Crossover-Based Search Techniques for Program Discovery. *IEEE International Conference on Evolutionary Computation*, pp. 573-578. 1995
- [26] D. Schlierkamp-Voosen, H. Mühlenbein, Strategy Adaptation by Competing Subpopulations. En: *Paralell Problem Solving from Nature 3*, Y. Davidor, H.-P. Schwefel, R. Mämmmer (Eds.), Berlin, Alemania, Springer-Verlag, pp. 199-208. 1994
- [27] H. Satoh, M. Yamamura, S. Kobayashi. Minimal Generation Gap Model for GAs Considering Both Exploration and Exploitation. *Proc. Methodologies for the Conception, Design and Application of Intelligent Systems (IIZUKA'96)*, pp. 494-497. 1996
- [28] W.A. Tackett, Recombination, Selection, and the Genetic Construction of Computer Programs. PhD thesis, University of Southern California, Department of Electrical Engineering Systems. 1994
- [29] O. Takahashi, H. Kita, S. Kobayashi, A distance alternation model on real-coded genetic algorithms. *IEEE SMC '99 Conference Proceedings*. *IEEE International Conference on*, Volume 1, pp. 104-111. 1999
- [30] H.M. Voigt, H. Mühlenbein, D. Cvetkovic, Fuzzy Recombination for the Breeder Genetic Algorithm. *Proc. Of the Sixth Int. Conf. on Genetic Algorithms*, pp. 104-111. 1995
- [31] J.-M. Yang, C.-Y. Kao. Integrating Adaptive Mutations and Family Competition into Genetic Algorithms as Function Optimizer. *Soft Computing 4*, pp. 89-102. 2000
- [32] J.-M. Yang, C.-Y. Kao. Flexible Ligand Docking Using a Robust Evolutionary Algorithm. *Journal of Computational Chemistry*, Vol. 21, N° 11, pp. 988-998. 2000