

Algoritmo Memético Basado en Encadenamiento de Búsquedas Locales para Problemas de Optimización Continua

Daniel Molina

Manuel Lozano

Francisco Herrera

25 de noviembre de 2008

Resumen

Este trabajo presenta un algoritmo memético para problemas de optimización continua, específicamente diseñado para métodos de búsqueda local que adaptan los parámetros utilizados para guiar la búsqueda y obtener así soluciones más efectivas. Este proceso de adaptación implica que el método de búsqueda local requiera evaluar un mayor número de soluciones (a dicho número se denomina intensidad de la búsqueda local) para poder adaptar los parámetros y dirigir mejor la búsqueda. Esta mayor intensidad dificulta su uso dentro de los algoritmos meméticos. Nuestra propuesta, al aplicar la búsqueda local sobre una solución, considera una intensidad en función de sus características, mediante el encadenamiento de consecutivas aplicaciones de la búsqueda local. Mediante esta técnica, en cada nueva aplicación de la búsqueda local, ésta continúa desde el estado resultante de su aplicación anterior. Siguiendo este proceso se propone un algoritmo memético que integra el algoritmo CMA-ES como su proceso de búsqueda local. Esta propuesta es evaluada siguiendo el conjunto de funciones de prueba propuesto por los organizadores de la Sesión Especial en Metaheurísticas, Algoritmos Evolutivos y Bioinspirados para Problemas de Optimización Continua del MAEB'09.

1. Introducción

Se ha demostrado que hibridar los *algoritmos evolutivos* (AEs) con otras técnicas puede incrementar sustancialmente su eficiencia de búsqueda [4, 5].

Los AEs obtenidos mediante la hibridación con técnicas de búsqueda local (BL) son denominados *algoritmos meméticos* (AMs) [18, 19, 16]. Un procedimiento usual de mejora es aplicar el método de BL a los nuevos miembros de la población, para explotar las mejores regiones de búsqueda obtenidas durante el muestreo global del EA. Esto permite diseñar AMs para optimización continua capaces de obtener soluciones precisas para este tipo de problemas [13].

La mayoría de los métodos de BL más relevantes hacen uso de parámetros estratégicos explícitos (como el tamaño de salto) para guiar la búsqueda. Es común adaptar estos parámetros con el propósito de incrementar la probabilidad de producir soluciones más efectivas. Debido a esta adaptación de parámetros, estos algoritmos pueden necesitar un número elevado de evaluaciones (son métodos de BL *intensos*) para poder conseguir un nivel de adaptación suficiente que conduzca finalmente a soluciones adecuadas. Este comportamiento hace que los modelos de hibridación usuales no sean lo suficientemente adecuados para estos algoritmos de BL intensos, ya que el número de evaluaciones requerido por el operador de BL podría ser demasiado alto, impidiendo una sinergia adecuada entre el algoritmo evolutivo (AE) y el algoritmo de BL.

En este trabajo, presentamos un modelo de AM para optimización continua específicamente diseñado para incorporar estos métodos continuos de BL intensos como su procedimiento de BL. Nuestra propuesta aplica la BL hasta alcanzar un número máximo de evaluaciones (intensidad) adaptado a cada solución, explotando así con mayor intensidad los individuos más prometedores. Para adaptar la intensidad de la

BL, nuestra propuesta puede aplicar el operador de BL más de una vez, con una intensidad determinada, sobre el mismo individuo, creando *cadena de BL*. Con esta técnica de *cadena de BL*, el individuo resultante de aplicar la BL puede ser posteriormente la solución inicial de una nueva aplicación de la BL, utilizando como valores iniciales de los parámetros de búsqueda sus valores finales tras la aplicación anterior. De esta forma, el método de BL puede de forma adaptable ajustar los parámetros a las características particulares de la zona de búsqueda. Siguiendo este modelo presentamos un AM que hace uso de CMA-ES [12] como algoritmo de BL intenso, el cual se ha confirmado como un excelente algoritmo de búsqueda local.

Este trabajo se estructura de la siguiente forma. Primero describimos el algoritmo propuesto, presentando los distintos componentes que lo constituyen. Luego, se aplica sobre el conjunto de funciones de prueba propuesto por el MAEB'09. Finalmente, se analizan los resultados obtenidos con algoritmos de comparación y se muestran las conclusiones obtenidas.

2. Algoritmo Memético Propuesto

Los algoritmos de BL continuos adaptativos, como el algoritmo CMA-ES que describimos posteriormente, requieren una alta intensidad [3]. Llamamos a estos algoritmos que requieren una alta intensidad *métodos de BL intensos*. Esta necesidad de alta intensidad les hace difíciles de utilizar dentro de un modelo clásico de hibridación como el propuesto por Hart[13]. Esta dificultad deriva de la inconveniencia de aplicar alta intensidad sobre soluciones poco prometedoras. Sin embargo, este problema se alivia mediante el empleo de una intensidad adaptativa. Para alcanzar dicha adaptación, proponemos un nuevo modelo de hibridación, bajo el que puede ser seleccionada la misma solución varias veces para ser mejorada mediante la aplicación de la BL. Además, emplea una memoria de parámetros de la BL que permite que cuando se seleccione para la BL una solución ya

mejorada anteriormente, el proceso de BL *continúe* del estado anterior. Este algoritmo fue especialmente diseñado para obtener buenos resultados en métodos de BL intensos [17].

2.1. Algoritmo Genético Estacionario

El algoritmo propuesto aplica como algoritmo evolutivo un algoritmo genético (AG) estacionario [14, 23], especialmente diseñado para promover altos niveles de diversidad en la población, como se recomienda en [15]. Esta diversidad es obtenida mediante el operador de cruce BLX- α con un alto valor de α para fomentar diversidad ($\alpha = 0,5$) y la *estrategia negativa inversa* [6], en combinación con la estrategia de *reemplazar el peor*. Esta combinación de mecanismo de selección de padres y estrategia de reemplazo permite alcanzar un adecuado equilibrio entre exploración y explotación. Adicionalmente, se introduce mayor diversidad mediante la aplicación del operador de mutación BGA [20] al 15% de los nuevos individuos.

2.2. La Estrategia Evolutiva mediante Adaptación de la Matriz de Covarianza (CMA-ES)

La *estrategia de evolución mediante adaptación de la matriz de covarianza (covariance matrix adaptation evolution strategy, CMA-ES)* [12, 11] es un algoritmo capaz de obtener muy buenos resultados en problemas de optimización continua. Aunque se propuso inicialmente como un algoritmo de búsqueda global muy competitivo [10], posee una gran habilidad para ajustarse localmente al espacio de búsqueda. Este comportamiento lo convierte en un algoritmo muy adecuado para optimización local [2].

En CMA-ES, se adapta el tamaño del operador de mutación, la dirección en el espacio multidimensional considerado, e incluso su forma, definida mediante una matriz de covarianza. En el denominado modelo (μ_W, λ) CMA-ES, en cada generación se genera una población de λ descendientes mediante la siguiente distribución normal multivariante:

$$x_i \sim N(m, \sigma^2 C) = m + \sigma N_i(0, C) \text{ for } i = 1, \dots, \lambda,$$

donde el vector media m representa la mejor solución encontrada, el denominado tamaño-paso σ controla la distancia de la distribución, y la matriz de covarianza C determina la forma de la distribución elipsoide.

Luego, se eligen las μ mejores soluciones encontradas en el paso anterior, y se utilizan para formar un nuevo valor medio mediante una media ponderada: $\sum_{i=1}^{\mu} w_i x_{i:\lambda}$, en donde el total de pesos suman uno. La matriz de covarianza y el tamaño de paso se actualizan mediante las ecuaciones indicadas en [12] y [10]. En dichas referencias se pueden encontrar también los valores por defecto de todos los parámetros, a excepción de los parámetros m y σ , que hay que asignar en cada caso.

2.3. Cadenas de BL

En AMs estacionarios, los individuos a los que se aplicó la BL pueden perdurar durante bastante tiempo en la población. En esta situación, consideramos que se produce un encadenamiento de la BL al aplicarse sucesivamente sobre una misma solución cuando el estado final (valores de los parámetros, variables internas, ...) alcanzado por una aplicación de la BL es empleado como la configuración inicial de la siguiente aplicación.

De esta forma, el algoritmo de BL continúa bajo las mismas condiciones alcanzadas cuando se detuvo la operación de BL, ofreciendo una *conexión entre sucesivas aplicaciones de la BL*, generando una *cadena de BL*.

Hay que considerar dos importantes cuestiones para llevar a cabo el encadenamiento:

- Cada vez que se aplica el algoritmo de BL sobre un cromosoma, se aplica con una intensidad fija denominada *tramo de intensidad de BL* (I_{str}).

De esta forma, aplicar n_{app} veces la BL sobre una misma solución con intensidad I_{str} produce la misma solución que aplicando el algoritmo de BL durante $n_{app} \cdot I_{str}$ evaluaciones.

- Después de la operación de BL, todos los parámetros que definen el estado actual del proceso de BL son almacenados asociados al individuo final. Esto permite que cuando se seleccione

posteriormente un individuo, se puedan recuperar los valores iniciales de los parámetros.

A la hora de aplicar CMA-ES, es necesario asignar los parámetros m y σ . En nuestro modelo se inician estos parámetros siguiendo los criterios siguientes:

- Se considera como la media inicial de la distribución m la solución inicial C_i .
- El valor inicial de σ es la mitad de la distancia de C_i al individuo de la población más cercano (este valor permite una exploración efectiva alrededor de C_i).

2.4. Modelo General propuesto

El modelo general es el mostrado en la Figura 1.

Este algoritmo presenta las siguientes importantes características:

1. Es un modelo de AM estacionario.
2. Mantiene un ratio predeterminado de evaluaciones locales/totales constante. Gracias a esto, se estabiliza fácilmente este ratio, que posee una fuerte influencia sobre el comportamiento final del AM, evitando una excesiva explotación.
3. Favorece el alargamiento de aquellas cadenas de BL que muestran mejoras prometedoras sobre las mejores zonas alcanzadas por la población del AG estacionario. Además, favorece la creación de nuevas cadenas para explorar nuevas regiones, cuando las mejores encontradas hasta ahora no ofrecen una mejora suficiente. El criterio de elección del individuo sobre el que aplicar la BL se diseña especialmente para este propósito (Pasos 3 y 4).

El algoritmo define la siguiente relación entre el AG estacionario y la intensidad de aplicación de la BL (paso 2): *cada n_{frec} evaluaciones del AG estacionario, se aplica el algoritmo de BL al cromosoma seleccionado c_{BL} con intensidad I_{str}* . El valor n_{frec} se calcula para mantener constante $r_{L/G}$, definido como el porcentaje de evaluaciones realizadas durante la BL entre el total de evaluaciones del algoritmo. Para

ello, se calcula automáticamente el valor n_{frec} mediante la expresión $n_{frec} = I_{str} \frac{1-r_{L/G}}{r_{L/G}}$, donde I_{str} es el tramo de intensidad (Sección 2.3).

Figura 1: Seudocódigo del algoritmo propuesto

1. Generar la población inicial.
2. Ejecutar el AG estacionario durante n_{frec} evaluaciones.
3. Construir el conjunto S_{BL} con los individuos que pueden ser mejorados mediante la BL.
4. Seleccionar el mejor individuo de S_{BL} (Denominamos c_{BL} a este individuo).
5. Si c_{BL} pertenece a una cadena de BL previa entonces
6. Inicializar el operador de BL con el estado de la BL previo almacenado junto con c_{BL} .
7. Sino
8. Inicializar el operador de BL con el estado por defecto.
9. Aplicar el algoritmo de la BL a c_{BL} con intensidad I_{str} (c_{BL}^r es el individuo resultante).
10. Reemplazar c_{BL} por c_{BL}^r en la población del AG estacionario.
11. Almacenar el estado final de la BL asociado a c_{BL}^r .
12. Si no se cumple la condición de terminación ir al paso 2.

Para seleccionar c_{BL} se aplica el siguiente procedimiento (Pasos 3 y 4):

1. Se construye el conjunto de individuos S_{BL} como los individuos de la población que cumple que:
 - a) Nunca han sido mejorados mediante el algoritmo de BL, o
 - b) Fueron previamente mejorados mediante la BL, obteniendo una mejora en *fitness* mayor que δ_{BL}^{min} (parámetro del algoritmo).

2. Si $|S_{BL}| \neq 0$, aplica el método de BL al mejor individuo de dicho conjunto. Si la condición no se cumple, se aplica sobre el mejor individuo de la población del AG.

Con este mecanismo, cuando el AG estacionario obtiene una solución mejor que la mejor encontrada hasta el momento, ésta será mejorada mediante la BL lo más pronto posible. Además, se aplica la BL sobre el mejor individuo de la población del AG, siempre y cuando la mejora en su última aplicación sea mayor que el valor umbral δ_{BL}^{min} (definido como 10^{-8} , ya que es el valor umbral de las funciones de evaluación).

3. Estudio Experimental

En este apartado vamos a presentar los resultados de aplicar el algoritmo propuesto al conjunto de 20 funciones de prueba propuesto para la sesión especial *Metaheurísticas, Algoritmos Evolutivos y Bioinspirados para Problemas de Optimización Continua* del MAEB'09. El uso de estas funciones permitirá posteriores comparaciones de los resultados obtenidos por cada algoritmo enviado a dicha sesión. La elección del conjunto de funciones empleado y las condiciones de la experimentación se han realizado según la guía indicada en dicha sesión especial ¹.

Todas las funciones son multimodales, y se compone de distintos grupos:

- Las primeras 7 son funciones básicas conocidas de la literatura: (*Rosenbrock*, *Griewank* rotada, *Ackley* rotada, *Rastrigin*, *Rastrigin* rotada, *Weierstrass* rotada, y *El problema Schwefel 2.13*).
- Dos funciones expandidas (inicializadas fuera de la región con el óptimo) de dos funciones (*Griewank* y *Scaffer*).
- Las últimas 11 son complejas, combinación de funciones anteriores y otras más simples.

¹<http://decsai.ugr.es/~lozano/AEBs-Continuo/AEBs.htm>

Todas las funciones están desplazadas. En [8] se puede obtener una descripción detallada de las distintas funciones.

El AM aplicado posee los siguientes parámetros:

- Una población de 60 individuos.
- Cruce $BLX - \alpha$, con $\alpha = 0,5$.
- Selección de padres *estrategia negativa inversa*, con $N_{NAM} = 3$, y estrategia de reemplazo *reemplazar el peor*.
- Operador de mutación BGA al 15% de los nuevos individuos.
- Aplica el algoritmo CMA-ES con un tramo de intensidad de 500 evaluaciones, y un ratio $r_{L/G}$ de 0,5. Es decir, cada 500 evaluaciones del AG estacionario aplica el CMA-ES otras tantas evaluaciones.

El algoritmo es ejecutado sobre las funciones con distintos valores de dimensión: 10, 30. Para cada función es ejecutado 25 veces con un número máximo de evaluaciones de $10000 \cdot dimension$. Cada ejecución termina o bien cuando el error obtenido es menor que 10^{-8} , o cuando se alcanza el número máximo de evaluaciones. Tras evaluar las 25 ejecuciones se calcula el error medio. Dicho valor medio es el valor utilizado en las comparaciones.

La Tabla 1 muestra el error acumulado para cada valor de dimensión.

Para confirmar la bondad de nuestra propuesta, hemos comparado nuestra propuesta con algoritmos de referencia, (G-CMA-ES[1], DE[21] y K-PCX[22]), que alcanzan muy buenos resultados para dicho conjunto de pruebas [9].

En las comparativas hemos hecho uso de los métodos de comparación no paramétricos detallados en [7] ya que, para las funciones de prueba consideradas, no pueden emplearse las funciones paramétricas al no cumplir las condiciones requeridas para tal. Puede consultar dicho trabajo para obtener mayor información.

Primero, hemos aplicado el test de Iman-Davenport (con un $p-value$ de 0.05) para comprobar si existe alguna diferencia significativa al comparar

Tabla 1: Error medio alcanzado por la propuesta, para cada valor de dimensión

| Función de Test | Dimensión 10 | Dimensión 30 |
|-----------------|--------------|--------------|
| F6 | 7.919168e-9 | 1.191003e+1 |
| F7 | 1.576340e-2 | 8.871392e-4 |
| F8 | 2.025390e+1 | 2.027016e+1 |
| F9 | 7.955018e-9 | 7.827714e-9 |
| F10 | 2.547095e+0 | 1.838684e+1 |
| F11 | 4.996535e-1 | 4.350834e+0 |
| F12 | 1.830865e+2 | 7.690185e+2 |
| F13 | 5.483822e-1 | 2.344814e+0 |
| F14 | 2.184448e+0 | 1.268192e+1 |
| F15 | 2.437411e+2 | 3.080000e+2 |
| F16 | 9.273844e+1 | 1.363134e+2 |
| F17 | 9.299357e+1 | 1.345630e+2 |
| F18 | 8.335419e+2 | 8.156512e+2 |
| F19 | 8.436303e+2 | 8.163714e+2 |
| F20 | 8.091376e+2 | 8.157765e+2 |
| F21 | 7.756537e+2 | 5.120000e+2 |
| F22 | 7.376647e+2 | 5.258481e+2 |
| F23 | 9.242833e+2 | 5.341643e+2 |
| F24 | 2.643189e+2 | 2.000000e+2 |
| F25 | 4.234632e+2 | 2.108472e+2 |

Tabla 2: Resultados del test de Iman-Davenport para cada valor de dimensión

| Dimensión | valor Iman-Davenport | Valor crítico | ¿Diferencia Significativa? |
|-----------|----------------------|---------------|----------------------------|
| 10 | 4.47 | 2.77 | Sí |
| 30 | 4.56 | 2.77 | Sí |

estos algoritmos junto con la propuesta presentada. La Tabla 2 muestra el resultado, de donde se observa claramente que existe una diferencia significativa para cada valor de dimensión.

Tabla 3: Resultados de comparar aplicando el test de Wilcoxon ($p-value = 0.05$), dimensión 10

| Algoritmo | R+ | R- | Valor Crítico | ¿Diferencia Significativa? |
|-----------|-------------|-------------|---------------|----------------------------|
| DE | 98.0 | 112.0 | 52 | No |
| G-CMA-ES | 61.5 | 148.5 | 52 | No |
| K-PCX | 185.0 | 25.0 | 52 | Sí |

Una vez confirmado que existe una diferencia significativa, aplicamos el test de Wilcoxon (con $p-value$ de 0.05) comparando cada algoritmo con nuestra propuesta. La Tabla 3 muestra los resultados para dimensión 10. Se puede observar que para dimensión

10 nuestra propuesta es estadísticamente mejor que K-PCX (ya que el valor $R-$ es menor que el de $R+$). Ofrece peores resultados que G-CMA-ES ó DE, pero la diferencia no es estadísticamente significativa.

Tabla 4: Resultados de comparar aplicando el test de Wilcoxon (p -value = 0.05), dimensión 30

| Algoritmo | $R+$ | $R-$ | Valor Crítico | ¿Diferencia Significativa? |
|-----------|-------|-------------|---------------|----------------------------|
| DE | 181.0 | 28.5 | 52 | Si |
| G-CMA-ES | 124.5 | 85.5 | 52 | No |
| K-PCX | 157.0 | 53.0 | 52 | No |

La Tabla 4 muestra los resultados del test de Wilcoxon para dimensión 30. Se puede observar que para dimensión 30 nuestra propuesta es mejor que cada uno de ellos (ya que el valor $R-$ es menor que el de $R+$ en todos los casos), y que es estadísticamente mejor que DE.

4. Conclusiones

En este trabajo hemos presentado un AM capaz de aplicarse de forma exitosa sobre métodos de BL intensos, como CMA-ES, que son capaces de ofrecer muy buenos resultados en optimización continua. Estos métodos requieren una alta intensidad, lo cual origina distintos problemas para utilizarse en el diseño de AMs. Nuestra propuesta consigue utilizar el algoritmo CMA-ES como método de BL mediante un proceso que le permite aplicar la BL con intensidad adaptativa. Hemos experimentado nuestro algoritmo con el conjunto de funciones de prueba propuesto por los organizadores de la sesión, y lo hemos comparado con algoritmos que ofrecen un buen comportamiento para dichas funciones, obteniendo que nuestra propuesta presenta buenos resultados, especialmente en dimensión 30.

Referencias

[1] A. Auger and N. Hansen. A Restart CMA Evolution Strategy with Increasing Population Size. In *2005 IEEE Congress on Evolutionary Computation*, pages 1769–1776, 2005.

[2] A. Auger and N. Hansen. Performance Evaluation of an Advanced Local Search Evolutionary Algorithm. In *2005 IEEE Congress on Evolutionary Computation*, pages 1777–1784, 2005.

[3] A. Auger, M. Schoenauer, and N. Vanhaecke. LS-CMAES: a second-order algorithm for covariance matrix adaptation. In *Proc. of the Parallel problems solving for Nature - PPSN VIII, Sept. 2004, Birmingham*, 2004.

[4] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[5] W. Banzhaf et al., editor. *Optimizing global-local search hybrids*. Morgan Kaufmann, San Mateo, California, 1999.

[6] C. Fernandes and A. Rosa. A Study of non-Random Matching and Varying Population Size in Genetic Algorithm using a Royal Road Function. *Proc. of the 2001 Congress on Evolutionary Computation*, pages 60–66, 2001.

[7] S. García, D. Molina, M. Lozano, and F. Herrera. A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms’ Behaviour: A Case Study on the CEC’2005 Special Session on Real Parameter Optimization. *Journal of Heuristics. In Press.*, 2008.

[8] N. Hansen. Compilation of Results on the CEC Benchmark Function Set. Technical report, Institute of Computational Science, ETH Zurich, Switerland, 2005. available as http://www.ntu.edu.sg/home/epnsugan/index_files/CEC-05/compareresults.pdf.

[9] N. Hansen. Compilation of Results on the CEC Benchmark Function Set. In *2005 IEEE Congress on Evolutionary Computation*, 2005.

[10] N. Hansen and S. Kern. Evaluating the CMA evolution strategy on multimodal test functions. In *Proc. of the Parallel Problem Solving for Nature - PPSN VIII, LNCS 3242*, pages 282–291, 2004.

[11] N. Hansen, S.D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

- [12] N. Hansen and A. Ostermeier. Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation. In *Proceeding of the IEEE International Conference on Evolutionary Computation (ICEC '96)*, pages 312–317, 1996.
- [13] W.E. Hart. *Adaptive Global Optimization With Local Search*. PhD thesis, Univ. California, San Diego, CA., 1994.
- [14] F. Herrera, M. Lozano, and J. L. Verdegay. Tackling Real-coded Genetic Algorithms: Operators and Tools for the Behavioral Analysis. *Artificial Intelligence Reviews*, 12(4):265–319, 1998.
- [15] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina. Real-coded Memetic Algorithms with Crossover Hill-climbing. *Evolutionary Computation*, 12(2):273–302, 2004.
- [16] Peter Merz. *Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies*. PhD thesis, Gesamthochschule Siegen, Germany, 2000.
- [17] Daniel Molina, Manuel Lozano, C. García-Martínez, and Francisco Herrera. Memetic algorithms for continuous optimization based on local search chains. *Evolutionary Computation*. *In press*, 2008.
- [18] P.A. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical report, Technical Report Caltech Concurrent Computation Program Report 826, Caltech, Pasadena, California, 1989.
- [19] P.A. Moscato. *Memetic algorithms: a short introduction*, pages 219–234. McGraw-Hill, London, 1999.
- [20] H. Mülenbein and D. Schlierkamp-Voosen. Predictive Models for the Breeding Genetic Algorithm in Continuous Parameter Optimization. *Evolutionary Computation*, 1:25–49, 1993.
- [21] A.K. Qin and P.N. Suganthan. Self-adaptive Differential Evolution Algorithm for Numerical Optimization. In *2005 IEEE Congress on Evolutionary Computation*, pages 1785–1791, 2005.
- [22] A. Sinha, S. Tiwari, and K. Deb. A Population-Based, Steady-State Procedure for Real-Parameter Optimization. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 514–521, 2005.
- [23] G. Syswerda. Uniform Crossover in Genetic Algorithms. In J. David Schaffer, editor, *Proc. of the Thrid Int. Conf. on Genetic Algorithms*, pages 2–9. Morgan Kaufmann Publishers, San Mateo, 1989.