

Algoritmo Memético con Intensidad de Búsqueda Local Adaptativa*

Daniel Molina	Francisco Herrera	Manuel Lozano
Dept. de Lenguajes y Sistemas Informáticos	Dept. de Inteligencia Artificial y Ciencias de la Computación	Dept. de Inteligencia Artificial y Ciencias de la Computación
Campus de Cádiz	ETS Ingeniería Informática	ETS Ingeniería Informática
Univ. de Cádiz	Univ. de Granada	Univ. de Granada
11001 Cádiz	18071 Granada	18071 Granada
daniel.molina@uca.es	herrera@decsai.ugr.es	lozano@decsai.ugr.es

Resumen

Existen métodos de búsqueda local (BL) que presentan muy buen comportamiento en optimización continua, gracias en parte a un esquema adaptativo que permite adaptarse mejor a la estructura del problema.

Dentro de estos métodos se encuentra el algoritmo evolutivo para optimización continua denominado *Covariance Matrix Adaptation* (CMA-ES), que, mediante adaptación, es capaz de obtener una gran capacidad de explotación.

Este trabajo presenta un algoritmo memético para optimización continua que aplica el CMA-ES con una intensidad adaptativa para poder explotar con mayor intensidad aquellas soluciones consideradas más prometedoras. Para adaptar dicha intensidad se propone un esquema en el que la BL puede aplicarse más de una vez sobre un mismo individuo usando una intensidad fija, y el uso de una memoria que almacena tras cada aplicación de la BL el estado final de sus parámetros, para que en la siguiente aplicación sobre el mismo individuo pueda continuar desde el estado anterior en donde se quedó.

Finalmente, nuestra propuesta es comparada con las distintas propuestas presentadas en la Sesión Especial de Optimización Continua

del *CEC'2005*, destacando como mejor opción que todas ellas.

1. Introducción

Muchos problemas de ingeniería pueden expresarse como problemas de optimización continua. Debido a su importancia y a su naturaleza intratable numéricamente, se han convertido en un importante tema de investigación. Los algoritmos evolutivos (AEs) [1], han originado un interés creciente para su aplicación en optimización continua debido a sus buenas propiedades al abordar espacios de búsqueda complejos [13].

Estos AEs, capaces de obtener una buena exploración global, pueden ser mejorados mediante la incorporación de un método de búsqueda local (BL) que mejore la precisión de las soluciones encontradas [12], generando algoritmos híbridos, como los algoritmos meméticos (AMs) [16]. Los AMs pueden ser descritos como AEs que aplican un proceso de BL dentro de su ciclo evolutivo [14].

Recientemente, se ha propuesto el algoritmo *Covariance Matrix Adaptation* (CMA-ES) [11], que presenta muy buen comportamiento en funciones unimodales pero no se comporta bien en multimodales [8]. Sería interesante poder aplicarlo como método de BL dentro de un algoritmo memético, ya que en ese caso sus problemas de exploración con múltiples ópti-

*Este trabajo ha sido financiado por el MEC a través del proyecto TIN2005-08386-C05-01.

mos no son tan relevantes. Pero presenta el gran problema de que, por su comportamiento adaptativo, requiere una alta intensidad, por lo que no es fácilmente integrable dentro de un enfoque híbrido.

En este trabajo proponemos un AM, denominado Algoritmo Memético con Aplicación Adaptativa de BL (*Memetic Algorithm with Adaptive Local Search Intensity, MA²LSI*), que permite aplicar CMA-ES con un valor de intensidad que depende de las características del individuo a aplicar, mayor para los más prometedores.

Para poder aplicar una intensidad adaptada al individuo nuestro modelo presenta dos características: Por un lado, se puede aplicar la BL repetidamente a un mismo individuo, con una intensidad fija, mientras se siga considerando prometedor. Por el otro, se utiliza una memoria que almacena tras cada aplicación de la BL los valores finales de sus parámetros, permitiendo que aplicar la BL repetidamente a una misma solución sea equivalente a aplicarla una única vez con mayor intensidad. De esta forma, las soluciones que durante más tiempo se consideren prometedoras recibirán mayor número de veces la BL, lo que, es nuestro modelo, es equivalente a aplicar la BL con mayor intensidad a las soluciones más prometedoras.

Este trabajo está estructurado de la siguiente forma: En el apartado 2 se presenta el modelo de AM propuesto. En el apartado 3 se presenta un estudio experimental comparando la propuesta con una serie de algoritmos de optimización real. Por último, en el apartado 4 se exponen las principales conclusiones obtenidas.

2. Modelo MA²LSI Propuesto

En este apartado vamos a describir los distintos elementos de la propuesta planteada: El Algoritmo de Exploración, la BL, y el modelo de hibridación.

2.1. Algoritmo de Exploración

Como algoritmo exploratorio, nuestra propuesta utiliza un algoritmo genético estacionario (AGE) [1], con una población de 60 individuos. Aplica como operador de cruce el operador *BLX* - 0,5 [3]. Como operador de selección aplica una variante del emparejamiento variado inverso o NAM [4]: El primer padre es elegido de forma completamente aleatoria, y para elegir el segundo se muestrean N_{NAM} individuos (aplicamos $N_{NAM} = 3$), y se selecciona aquel más distante al primero (usando como medida de distancia la distancia euclídea). Como estrategia de reemplazo aplica el RW [6], es decir, cada nueva solución reemplaza a la peor de la actual población si lo mejora.

Como mutación aplica el operador BGA [15]. En este operador de mutación se transforma el individuo c en el individuo c' según la ecuación siguiente.

$$c'_i = c_i \pm rang_i \cdot \sum_{k=0}^{15} \alpha_k 2^{-k} \quad (1)$$

En donde $rang_i$ define el rango de mutación, se ha aplicado el valor usual $0,1 \cdot (b_i - a_i)$. El signo (+ ó -) es elegido cada vez con una probabilidad de 0,5 y α_k toma para cada variable un valor 0 ó 1 determinado aleatoriamente, mediante la expresión $p(\alpha_i = 1) = \frac{1}{16}$.

Este operador de mutación se aplica a un número relativamente alto de cromosomas (al 12,5 %) para fomentar diversidad en el proceso de búsqueda.

2.1.1. CMA-ES

En esta propuesta aplicamos la Adaptación de Matriz de Covarianza, *Covariance Matrix Adaptation* (CMA-ES) como nuestro método de BL [11, 8]. A continuación describimos los principios generales del algoritmo. El (μ, λ) CMA-ES es un algoritmo que emplea una función de distribución de probabilidad para generar μ soluciones, y, posteriormente, emplea las λ mejores para refinar la propia función de distribución de probabilidad para producir mejores soluciones, en un proceso iterativo. Finalmente, el algoritmo devuelve la mejor solu-

ción encontrada. Puede consultar [11, 8] para obtener una descripción más detallada.

El (μ, λ) CMA-ES ofrece buenos resultados en optimización real, posee una alta velocidad de convergencia con la que es capaz de alcanzar soluciones con alto nivel de precisión en muchas funciones unimodales [10], pero en funciones multimodales, no presenta tan buenos resultados como otras técnicas [9].

Este método requiere únicamente dos parámetros de entrada (para el resto de parámetros de la búsqueda los autores proponen valores por defecto): el centro inicial de la distribución, y el valor σ (varianza inicial de la distribución). Como centro inicial aplicaremos la solución a mejorar en cada caso, y como σ la mitad de la distancia al individuo más cercano.

2.2. Memoria de la BL

Una característica fundamental del modelo propuesto es el uso de una memoria que almacena el estado de la BL tras aplicarse a un individuo. La memoria permite centrar la búsqueda hacia los mejores individuos, haciendo posible que sea equivalente el aplicar varias veces (n_{num}) la BL a un individuo con intensidad n_{int} que aplicar la BL una única vez con intensidad $n_{num} \cdot n_{int}$. Sin dicha memoria, esta equivalencia no sería posible en métodos de BL que (como el CMA-ES) adaptan uno o varios parámetros durante el proceso de búsqueda.

Los parámetros que definen el actual estado de la búsqueda (explícita o implícitamente) son almacenados junto con los individuos. Así pues, si el mismo individuo es seleccionado de nuevo para ser mejorado mediante la BL, el método de BL puede continuar bajo las mismas condiciones en las que se interrumpió la anterior vez.

2.3. Modo de Aplicación de la BL

En nuestro modelo empleamos para el ratio de esfuerzo de BL ($ratio_{BL}$) un valor fijo y definido *a priori*, definido como el ratio entre el número de evaluaciones durante las diferentes aplicaciones de la BL y el número total de evaluaciones. Mantener fijo este ratio nos permite cambiar la intensidad de la BL sin tener que

calcular el número de individuos que podrán ser mejorados sin aumentar demasiado el esfuerzo invertido en la BL.

El esquema general del método de hibridación es el siguiente:

1. Generar aleatoriamente una población inicial.
2. Aplicar el AGE para evolucionar la población durante n_{frec} evaluaciones. El parámetro n_{frec} es el número de evaluaciones del AGE entre dos aplicaciones consecutivas de la BL. El valor n_{frec} es automáticamente calculado para mantener constante el $ratio_{BL}$, mediante la expresión
$$n_{frec} = n_{int} \frac{1 - ratio_{BL}}{ratio_{BL}}$$
 donde n_{int} es la intensidad de la BL y $ratio_{BL} = \text{Evaluaciones en la BL} / \text{Número Total de Evaluaciones}$.
3. Seleccionar un conjunto de individuos susceptibles de ser mejorados. Incluiremos en dicho conjunto a todos los individuos sobre los que no se haya aplicado la BL; o que al aplicarse obtuviese una mejora significativa (mayor que un valor umbral $mejoraMin$) y una varianza σ final mayor que otro valor umbral σMin (para evitar aplicarla repetidamente sobre óptimos locales).
4. Se selecciona un individuo de dicho conjunto, para lo cual aplicamos una medida que indique lo prometedor que consideramos cada individuo. En este trabajo seleccionamos como la solución más prometidora aquella con mejor valor $fitness$.
5. Si el individuo tiene asociado un estado previo de la BL (valores de los parámetros) en la memoria correspondiente, recuperarlos. En otro caso obtener los parámetros de la BL asignándoles sus valores por defecto (cuando corresponda).
6. Aplicar la BL sobre el individuo seleccionado empleando los parámetros de la BL obtenidos en el paso anterior, utilizando una intensidad de BL (n_{int}) fija.

7. Almacenar en la memoria de la BL el estado de la BL (valores de los parámetros obtenidos al final del proceso de mejora).
8. Ir al paso 2.

Por tanto, a un individuo con buen *fitness* (prometedor) se le aplica periódicamente la BL mientras: El AGE no obtenga una nueva solución mejor; la última mejora obtenida no sea significativa; y la varianza autoadaptada para las ejecuciones siguientes no sea demasiado pequeña (la BL haya convergido).

3. Estudio Experimental

En este apartado vamos a presentar un conjunto de experimentos que nos permita estudiar el comportamiento y rendimiento de nuestra propuesta. Para mostrar el rendimiento de nuestra propuesta hemos considerado el uso de las funciones recomendadas en la *Sesión Especial de Optimización Real del 2005 IEEE Congress of Evolutionary Computation (CEC'2005* en lo sucesivo) [17]. El uso de estas funciones nos permitirá comparar los resultados obtenidos con nuestro *MA²LSI* con todos los algoritmos propuestos en dicho congreso. Cada algoritmo es ejecutado sobre las funciones con distintos valores de dimensión: 10, 30 y 50.

Cada algoritmo es ejecutado 25 veces con un número de evaluaciones de $10000 * dimensión$, y se calcula el error medio de las 25 ejecuciones. Dicho valor medio es el valor utilizado en las comparaciones. En [17] se puede obtener una descripción detallada de las distintas funciones.

En las comparativas hemos hecho uso de los métodos de comparación no paramétricos detallados en [5] ya que, como se muestra en dicho artículo, para las funciones de tests consideradas, no pueden emplearse las funciones paramétricas (t-test, ...) al no cumplir las condiciones requeridas para tal fin. Puede consultar dicho trabajo para obtener mayor información.

3.1. Parámetros de la propuesta

Los parámetros concretos empleados en nuestra comparativa son:

- Para el AGE utilizamos una población de 60 individuos, $N_{NAM} = 3$. Aplicamos el BGA al 12,5% de los individuos.
- A la hora de aplicar la BL utilizamos una combinación diferente de ratio e intensidad inicial (*ratio; n_{int}*) para cada dimensión, elegidos mediante un proceso de ajuste previo: (0,8; 100) para dimensión 10; (0,5; 500) para dimensión 30; y (0,5; 1000) para dimensión 50.
- En el esquema de hibridación se aplica $mejoraMin = 1e - 8$ y $\sigma Min = 1e - 8$ (se igualan al valor umbral del error).

3.2. Estudio de la influencia de la memoria de la BL

Antes de comparar nuestra propuesta con el resto, estudiamos la conveniencia del uso de la memoria aplicada a la BL, para observar si el enfoque adaptativo supone una mejora significativa.

Dimensión	R^+	R^-	Ref.	A/R
10	216	109	89	A
30	246	79	89	R
50	252	73	89	R

Tabla 1: Test de Wilcoxon Sin Usar Memoria de BL Versus Usándola, $p=0,05$

La Tabla 1 contiene el resultado de aplicar el test de Wilcoxon de la variante propuesta aplicando y sin aplicar la memoria. El formato de la tabla es el siguiente: Cuando el valor en R- es menor que en R+ (se muestra en negrita el menor) nuestra propuesta es mejor que la considerada, y peor en el caso contrario. Además, cuando el menor valor de entre R+ y R- es menor que el valor de referencia definido por el test de Wilcoxon, la diferencia es estadísticamente significativa. Utilizaremos como valor de referencia aquel con valor 0,05 de control de probabilidad de error. Una descripción más detallada de dicho formato de tabla se encuentra en [5].

Puede observarse que el uso de la memoria es mejor en todos los casos, y que dicha

mejora es mayor conforme aumenta la dimensionalidad (y, por tanto, la complejidad). Con dimensión 30 y 50 la diferencia ya es estadísticamente significativa.

3.3. Comparativa con el CEC'05

En este apartado vamos a comparar los resultados obtenidos con nuestra propuesta (MA^2LSI) con todos los algoritmos propuestos en la Sesión Especial de Optimización Continua, en el *IEEE Congress on Evolutionary Computation* del 2005 (CEC'2005).

Dentro de las funciones del grupo de *benchmark* hay 25 funciones, de las cuales las 5 primeras son unimodales. Dado que nos interesa principalmente observar el comportamiento de nuestra propuesta en las multimodales, aplicamos los tests sobre las 20 funciones restantes (f6-f25).

Para poder analizar la escalabilidad de nuestra propuesta, se analizará de forma separada para cada uno de los valores de dimensión considerados (10, 30 y 50).

Dimensión	Valor Iman-Davenport	F_F	A/R
10	5,74	1,84	R
30	8,49	1,84	R

Tabla 2: Test de Iman-Davenport de MA^2LSI y propuestas del CEC'05

Para poder aplicar el test no paramétrico de Holm, es necesario que el test de Iman-Davenport identifique que existe alguna diferencia estadísticamente significativa, por lo que lo aplicamos primero. La Tabla 2 muestra los resultados obtenidos. Como en cada caso el valor de Iman-Davenport es mayor que el valor de referencia F_F (ver [5]), se identifica una diferencia significativa para cada dimensión, por lo que procedemos a un análisis más minucioso. (En dimensión 50 sólo tenemos los datos de G-CMA-ES y L-CMA-ES, por lo que para dicho dimensión sólo podemos aplicar el test de Wilcoxon).

3.4. Dimensión 10

La Tabla 3 muestra el resultado de comparar las distintas propuestas del CEC'05 y MA^2LSI con aquel que posee en menor ranking medio, el G-CMA-ES. Se ordena para cada función los algoritmos por *fitness*, y se calcula para cada algoritmo su posición media (ranking medio). Dicha tabla lista para cada función de forma ordenada los distintos algoritmos (de menor a mayor ranking medio). La tabla está dividida por una línea horizontal en dos partes, en la superior se encuentran los algoritmos estadísticamente peores que el tomado de referencia, y en la inferior aquellos que aún poseyendo peor ranking medio, la diferencia no llega a ser catalogada como significativa. Una descripción más detallada de dicho formato de tabla se encuentra en [5].

Se observa que MA^2LSI posee el segundo mejor ranking medio, y es estadísticamente equivalente a aquel con el mejor, G-CMA-ES.

Algoritmo	Z	P	$\frac{\alpha}{i}$
CoEvo	5,745	9,20e-9	0,0045
BLX-MA	3,749	0,0017	0,005
EDA	3,661	0,0003	0,0055
K-PCX	3,398	0,0006	0,0062
SPC-PNX	3,267	0,0010	0,0071
L-CMA-ES	3,047	0,0023	0,0083
DE	2,653	0,0079	0,01
BLX-GL50	2,061	0,0393	0,0125
DMS-L-PSO	1,513	0,1303	0,0167
L-SADE	1,250	0,2113	0,025
MA^2LSI	1,228	0,2195	0,05

Tabla 3: Test de Holm CEC'2005 y MA^2LSI Versus G-CMA-ES, dimensión 10

Para comparar los resultados de nuestra propuesta con el resto, hemos aplicado el test de Wilcoxon. La Tabla 4 muestra los resultados, en el mismo formato que la Tabla 1. Se puede observar que MA^2LSI es mejor que 8 algoritmos (y es estadísticamente mejor en 3 de ellos), y sólo es ligeramente peor que otros tres, aunque no se detectó que fuese una diferencia estadísticamente relevante.

Algoritmo	R^+	R^-	Ref.	A/B
BLX-GL50	110	100	52	A
BLX-MA	161	49	52	R
CoEvo	184	26	52	R
DE	98	112	52	A
DMS-L-PSO	115,5	94,5	52	A
EDA	109	101	52	A
G-CMA-ES	56,5	153,5	52	A
K-PCX	180	29,5	52	R
L-CMA-ES	130,5	79,5	52	A
L-SaDE	82,5	127,5	52	A
SPC-PNX	112	98	52	A

Tabla 4: Test de Wilcoxon CEC'2005 Versus MA^2LSI , $p = 0,05$, dimensión 10

Algoritmo	Z	P	$\frac{\alpha}{7}$
CoEvo	6,379	1,774e-10	0,00625
DE	3,464	0,0005	0,0071
SPC-PNX	3,002	0,0027	0,0083
K-PCX	2,5403	0,0110	0,01
BLX-MA	2,482	0,0130	0,0125
L-CMA-ES	2,222	0,0262	0,0167
BLX-GL50	1,8763	0,0606	0,025
G-CMA-ES	1,1547	0,2482	0,05

Tabla 5: Test de Holm CEC'2005 Versus MA^2LSI , dimensión 30

3.5. Dimensión 30

En este apartado vamos a comparar las distintas propuestas del CEC'2005 para dimensión 30 con MA^2LSI . Nótese que el conjunto de algoritmos del CEC'2005 comparados es ligeramente menor (son 8 de los 11 empleados con dimensión 10), ya que algunas propuestas presentadas al congreso no presentaron los resultados completos para dimensión 30.

La Tabla 5 muestra el resultado de comparar las distintas propuestas del CEC'05 con respecto a MA^2LSI , que en este caso posee el menor ranking medio. Se observa que MA^2LSI destaca como mejor algoritmo que todos ellos, y que, según el test de Holm, es estadísticamente mejor que tres de ellos. Para comparar con mayor detalle nuestra propuesta con cada algoritmo, aplicamos de nuevo el test de Wilcoxon. La Tabla 6 muestra el resultado, con el mismo formato que en el apartado

anterior. Se puede observar que es estadísticamente mejor que prácticamente todos (únicamente respecto a G-CMA-ES la mejora no se llega a identificar como estadísticamente significativa).

Algoritmo	R+	R-	Ref.	A/R
BLX-GL50	166	44	52	R
BLX-MA	198	11,5	52	R
CoEvo	210	0	52	R
DE	199,5	10,5	52	R
G-CMA-ES	139	71	52	A
K-PCX	174	36	52	R
L-CMA-ES	165	45	52	R
SPC-PNX	169,5	40,5	52	R

Tabla 6: Test de Wilcoxon CEC'2005 Versus MA^2LSI , $p = 0,05$, dimensión 30

3.6. Dimensión 50

En este apartado vamos a comparar los resultados del MA^2LSI para dimensión 50. Sin embargo, al igual que pasaba con dimensión 30, el número de algoritmos que ofrece sus resultados para dicha dimensión es muy reducido (únicamente dos: L-CMA-ES y G-CMA-ES). Sin embargo, en nuestro caso son los dos algoritmos que más nos interesan, al estar compuestos por el mismo método que nosotros utilizamos, CMA-ES (son propuestos por su propio autor). El comparar nuestra propuesta con estas variantes nos permite comprobar las ventajas de mejorar el CMA-ES mediante el uso de un AM, frente a los otros enfoques aplicados por estos dos: multiarranque (L-CMA-ES) y un multiarranque con adaptación de parámetros (G-CMA-ES).

Al ser sólo dos algoritmos, aplicamos directamente el comparador de Wilcoxon. La Tabla 7 muestra los resultados, que confirman la bondad de MA^2LSI (es mejor que ambos, y respecto a L-CMA-ES la mejora se considera estadísticamente significativa).

3.7. Comparativa MA^2LSI con otras variantes de CMA-ES

En este punto es interesante resumir el comportamiento del MA^2LSI frente a estos dos

Algoritmo	R+	R-	Ref.	A/R
L-GMA-ES	176	33,5	52	R
G-CMA-ES	154	56	52	A

Tabla 7: Test de Wilcoxon CEC'2005 Versus MA^2LSI , $p = 0,05$, dimensión 50

algoritmos, para cada dimensión. Esto nos permite confirmar que el buen comportamiento se debe a nuestro modelo global, y no principalmente al método de BL, tal y como es sugerido en [2].

Algoritmo	R+	R-	Ref.	A/R
D10	130,5	79,5	52	A
D30	165	45	52	R
D50	176	33,5	52	R

Tabla 8: Test de Wilcoxon L-CMA-ES Versus MA^2LSI , $p = 0,05$

Algoritmo	R+	R-	Ref.	A/R
D10	56,5	153,5	52	A
D30	139	71	52	A
D50	154	56	52	A

Tabla 9: Test de Wilcoxon G-CMA-ES Versus MA^2LSI , $p = 0,05$

Las Tablas 8 y 9 muestran los resultados de comparar nuestra propuesta con el L-CMA-ES y el G-CMA-ES, respectivamente. De dichas tablas podemos concluir que:

- Nuestra propuesta es estadísticamente mejor que L-CMA-ES, es decir, que la variante memética es mejor que una variante multiarranque, lo cual confirma que el uso de un AM ofrece una mejora significativa. Es más, la Tabla 8 muestra que la mejora es mayor conforme aumenta el valor de dimensión. A partir de dimensión 30 la diferencia es significativa.
- Respecto al G-CMA-ES, que se comporta como el mejor algoritmo del congreso (ver [7]), la Tabla 9 a partir de dimensión 30 nuestra propuesta presenta mejor comportamiento (y la mejora es mayor conforme aumenta la dimensionalidad).

3.8. Resultados de los experimentos

Resumiendo, nuestra propuesta mejora a todas las propuestas presentadas en el CEC'2005. Con dimensión 10 nuestra propuesta es el segundo mejor algoritmo, y estadísticamente equivalente al mejor. Sin embargo, con dimensión 30 nuestra propuesta es siempre el mejor algoritmo, y estadísticamente mejor que casi todas (únicamente respecto al G-CMA-ES la mejora no es suficiente para clasificarse como estadísticamente mejor). Y conforme la dificultad aumenta la mejora respecto al otro mejor algoritmo, G-CMA-ES, se incrementa aún más, demostrando que nuestra propuesta mejora a todas las propuestas del CEC'2005. Además, su mejor comportamiento en los problemas más difíciles (mayor dimensionalidad) hace que nuestra propuesta sea de gran interés.

4. Conclusión

En este trabajo hemos propuesto un nuevo modelo de hibridación que permite el uso dentro de un memético de métodos de BL avanzados que consiguen buenos resultados, pero que requieren un valor de intensidad alto.

En esta propuesta se emplea una nueva técnica de hibridación que nos permite centrar más la BL (mayor intensidad) sobre las soluciones más prometedoras. Nuestro modelo alcanza esta adaptación de la intensidad de la BL mediante la posibilidad de aplicar de forma reiterada la BL sobre una misma solución, y un sistema de memoria que permite que cada nueva aplicación suponga una continuación de la aplicación anterior. Dada la dificultad de establecer *a priori* una intensidad correcta a cada solución, esta habilidad para adaptar la intensidad de la BL de una forma robusta posee gran valor.

Hemos propuesto un AM que hace uso de este modelo de hibridación usando el CMA-ES como método de BL, y lo hemos comparado con las distintas propuestas del CEC'05, obteniendo que la propuesta presentada era mejor que todas ellas, especialmente al aumentar la complejidad de los problemas.

El éxito de nuestra propuesta es princi-

palmente debido al mecanismo de intensidad adaptativa, que nos permite que se concentre el esfuerzo de la BL hacia las soluciones más prometedoras, aprovechando la mejora alcanzada mediante la aplicación de BL dentro del algoritmo exploratorio.

Además, el nuevo modelo de hibridación presentado permite que un nuevo tipo de BLs (las que presentan un comportamiento adaptativo y requieran una alta intensidad para poder emplearse adecuadamente) puedan ser integradas dentro de AMs, evitando los problemas de los enfoques más tradicionales, con lo que se amplía el uso de AMs a una nueva gama de métodos de BL avanzados.

El modelo de hibridación propuesto posee cierta flexibilidad, tanto en el diseño de catalogar a las soluciones como prometedoras, como en su ordenación, que será motivo de estudios futuros.

Referencias

- [1] T. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, 1997.
- [2] B.G.W. Craenen and A.E. Eiben. Hybrid Evolutionary Algorithms for Constraint Satisfaction Problems: Memetic Overkill? In *The IEEE Congress on Evolutionary Computation*, volume 4, pages 1922–1928, 2005.
- [3] L. J. Eshelman and J. D. Schaffer. Real-coded Genetic Algorithms in Genetic Algorithms by Preventing Incest. *Foundation of Genetic Algorithms 2*, pages 187–202, 1993.
- [4] 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.
- [5] S. García, D. Molina, M. Lozano, and F. Herrera. An experimental study about the use of non-parametric tests for analysing the behaviour of evolutionary algorithms in optimization problems. In *Spanish*. In *MAEB 2007 (Quinto congreso español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados)*, 2007.
- [6] D. E. Goldberg and K. Deb. A Comparative Analysis of Selection Schemes used in Genetic Algorithms. *Foundations of Genetic Algorithms*, pages 69–93, 1991.
- [7] N. Hansen. Compilation of Results on the CEC Benchmark Function Set. In *2005 IEEE Congress on Evolutionary Computation*, 2005.
- [8] N. Hansen and S. Kern. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In Xin Yao at al., editor, *Parallel Problem Solving for Nature - PPSN VIII, LNCS 3242*, pages 282–291. Springer, 2004.
- [9] N. Hansen and S. Kern. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In *Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII, Proceedings*, pages 282–291. Springer, Berlin, 2004.
- [10] N. Hansen, S.D. Mller, and P. Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 1(11):1–18, 2003.
- [11] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [12] W.E. Hart. *Adaptive Global Optimization With Local Search*. PhD thesis, Univ. California, San Diego, CA., 1994.
- [13] 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.
- [14] N. Krasnogor and J. Smith. A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issue. *IEEE Transactions on Evolutionary Computation*, 9(5):474–488, 2005.
- [15] H. Mlenbein and D. Schierkamp-Voosen. Predictive Models for the Breeding Genetic Algorithm in Continuous Parameter Optimization. *Evolutionary Computation*, 1:25–49, 1993.
- [16] P. A. Moscato. *Memetic Algorithms: a Short Introduction*. D. Corne, M. Dorigo and F. Glover, editors. McGraw-Hill, London, 1999.
- [17] P.N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical report, Nanyang Technological University, May 2005. Also available as <http://www.ntu.edu.sg/home/EPNSugan/>.

Una heurística Beam Search para el problema de Equilibrado de Líneas de Montaje, <i>Joaquín Bautista, Jordi Pereira</i>	187
Algoritmo Memético con Intensidad de BL Adaptativa, <i>Daniel Molina, Francisco Herrera, Manuel Lozano</i>	195
Un Algoritmo Genético Celular Híbrido para el Problema de Ensamblado de Fragmentos de ADN, <i>Bernabé Dorronsoro, Gabriel Luque, Enrique Alba</i>	203
Evolución de modelos jerárquicos de reglas en problemas anidados y no anidados, <i>Francesc Teixidó-Navarro, Ester Bernadó-Mansilla</i>	211
Tests no paramétricos de comparaciones múltiples con algoritmo de control en el análisis de algoritmos evolutivos: Un caso de estudio con los resultados de la sesión especial en optimización continua CEC'2005, <i>Salvador García, Daniel Molina, Manuel Lozano, Francisco Herrera</i>	219
Metaheurísticas multiobjetivo para optimizar el proceso de difusión en MANETs metropolitanas, <i>Enrique Alba, Bernabé Dorronsoro, Francisco Luna, Antonio J. Nebro, Coromoto León, Gara Miranda, Carlos Segura</i>	229
Evolución Diferencial y Algoritmos Genéticos para la planificación de frecuencias en redes móviles, <i>Eugénia M. Bernardino, Anabela M. Bernardino, Juan Manuel Sánchez Pérez, Miguel A. Vega Rodríguez, Juan Antonio Gómez Pulido</i>	237
Algoritmos genéticos locales, <i>Carlos García-Martínez, Manuel Lozano</i>	245
Selecting an Appropriate Statistical Test for Comparing Multiple Experiments in Evolutionary Machine Learning, <i>José Otero, Luciano Sánchez, Jesús Alcalá</i>	253
Datos GPS como conjuntos borrosos. Aplicación a la verificación de taxímetros, <i>José Ramón Villar, Adolfo Otero, José Otero, Luciano Sánchez</i>	261
Using a fuzzy mutual information measure in feature selection for evolutionary learning, <i>Javier Grande, Maria del Rosario Suárez, Jose Ramón Villar</i>	269

Actas de las I Jornadas sobre Algoritmos
Evolutivos y Metaheurísticas
JAEM'07

Editadas por
Enrique Alba
Francisco Chicano
Francisco Herrera
Francisco Luna
Gabriel Luque
Antonio J. Nebro

Zaragoza, 12 y 13 de Septiembre de 2007