

## Algoritmos genéticos locales

Carlos García-Martínez

Dept. de Informática y  
Análisis Numérico  
Campus de Rabanales  
Univ. de Córdoba  
14071 Córdoba  
cgarcia@uco.es

Manuel Lozano

Dept. de Ciencias de la Computación e  
Inteligencia Artificial  
ETS Ingeniería Informática  
Univ. de Granada  
18071 Granada  
lozano@decsai.ugr.es

### Resumen

Los *Algoritmos Genéticos Locales* son procedimientos que iterativamente refinan soluciones dadas. Su diferencia con *procedimientos de mejora iterativa clásicos* reside en el uso de *operadores genéticos* para realizar el refinamiento. En este estudio presentamos un nuevo *Algoritmo Genético Local Binario* basado en un *Algoritmo Genético Estacionario*. Hemos comparado el Algoritmo Genético Local Binario con otros procedimientos de mejora iterativa de la literatura. Los resultados muestran que, para un amplio rango de problemas, el Algoritmo Genético Local Binario mejora consistentemente a los otros procedimientos de mejora iterativa.

### 1. Introducción

Los *procedimientos de mejora iterativa* (PMIs) son algoritmos de búsqueda local cuyo objetivo es refinar una solución inicial dada. Para ello, mantienen una *solución actual* e intentan encontrar una solución mejor dentro de la *vecindad* de ésta. En caso de encontrarla, la nueva solución reemplaza a la actual, y se vuelve a actuar del mismo modo. La principal ventaja de los PMIs es que, en muchos casos, pueden localizar un óptimo local con un alto grado de precisión en un tiempo razonable.

En la actualidad, los PMIs despiertan un gran interés principalmente por el hecho de

que son una componente fundamental de *metaheurísticas* que representan el estado del arte de muchos problemas de optimización ([1, 21, 27]). Entre éstas están: la búsqueda local multiarranque ([16]), los procedimientos GRASP, los procedimientos de optimización basados en colonias de hormigas, la búsqueda local iterativa, búsqueda de vecindario variable ([19]), la búsqueda dispersa y los algoritmos meméticos ([20]). En este trabajo, llamamos *metaheurísticas basadas en PMIs* a estos algoritmos de búsqueda.

Una de las principales características de los *algoritmos genéticos* (AGs) ([4, 8]) es que pueden encontrar regiones prometedoras en espacios de búsqueda grandes y complejos. Aunque se reconoce que, bajo su formulación inicial, los AGs no son eficaces a la hora de *refinar* soluciones ([15]), actualmente, el atractivo de estos algoritmos como procedimientos de búsqueda ha motivado el diseño de AGs específicos que actúan como PMIs (es decir, pretenden refinar soluciones dadas, de forma efectiva). De hecho, se han presentado varias propuestas de AGs con este propósito ([3, 13, 15]). En este trabajo, proponemos la denominación de *Algoritmos Genéticos Locales* (AGLs) para esta nueva categoría de PMIs.

Los AGLs presentan ventajas sobre los PMIs clásicos. En complejos espacios de búsqueda, muchas instancias de PMIs pierden la habilidad para seguir un camino hacia el óptimo. Esta dificultad es más evidente cuan-

do el espacio de búsqueda contiene caminos muy estrechos de dirección arbitraria (conocidos como *crestas*). Esto se debe a que estos PMIs intentan dar pasos a lo largo de direcciones ortogonales que no coinciden necesariamente con la dirección de la cresta. Sin embargo, un estudio realizado en [13] apunta que los AGLs tienen capacidad para seguir crestas de dirección arbitraria sin importar su dirección, anchura, o incluso, discontinuidades. Por ello, el estudio de los AGLs se convierte en un campo de interés para el diseño de metaheurísticas basadas en PMIs más efectivas.

En este trabajo, proponemos un *AGL Binario* (AGLB) basado en el modelo de AG *estacionario* que emplea un *método de reemplazo por agrupamiento* (en inglés, método *crowding*) para favorecer la formación de *nichos* en la población. El AGLB cruza iterativamente la *solución actual*, inicialmente dada, con individuos de la población pertenecientes a nichos cercanos a éste. Después, si el nuevo descendiente es mejor que la solución actual, ésta se inserta en la población por medio del método por agrupamiento llamado *selección con torneo restringido* ([7]) y el descendiente se acepta como nueva solución actual. En otro caso, el descendiente es insertado en la población con el mismo procedimiento.

Presentamos un estudio empírico que compara el uso del AGLB frente al uso de cuatro PMIs clásicos, propuestos en la literatura, dentro de dos metaheurísticas basadas en PMIs. En particular, hemos observado que, para un amplio rango de problemas, el AGLB presenta una sólida mejora sobre los PMIs clásicos considerados.

El artículo está organizado de la siguiente forma. En la Sección 2, presentamos los AGLs. En la sección 3, proponemos el AGLB. En la Sección 4, comparamos el AGLB propuesta con cuatro PMIs clásicos. Finalmente, en la Sección 5, proporcionamos las conclusiones de este trabajo.

## 2. Algoritmos genéticos locales

Los AGs son procesos de búsqueda que aplican principios inspirados en *poblaciones genéticas*

*naturales* para evolucionar soluciones a problemas de optimización y búsqueda ([4, 8]). Su idea básica es mantener una *población de cromosomas* que representan soluciones a un problema concreto. El AG evoluciona la población a través de un proceso de competición y variación. Cada cromosoma de la población tiene un *valor de adaptación* (en inglés, *fitness*) que determina cuáles deben considerarse para crear nuevos cromosomas en el proceso de competición, llamado *selección de padres*. Los nuevos se crean mediante *operadores genéticos* tales como el *cruce* y la *mutación*.

A lo largo de su existencia, los AGs se han aplicado esencialmente de dos formas bien diferenciadas:

- Como procesos de búsqueda por sí solos. En este caso, sus componentes se diseñan cuidadosamente para tratar de garantizar fiabilidad y precisión a la vez. Es decir, visitar una amplia representación de las regiones del espacio de búsqueda (*exploración*) y refinar las mejores soluciones encontradas (*explotación*), respectivamente.
- Como un proceso especializado en exploración que se combina con un PMI que refuerza el carácter de explotación sobre las soluciones. Un ejemplo, son los *algoritmos meméticos* ([20]).

Sin embargo, la flexibilidad que ofrece la arquitectura de los AGs nos posibilita el diseño de modelos específicos con el propósito de refinar soluciones. Obtenemos así, una nueva categoría de PMIs, conocida como *algoritmos genéticos locales* (AGL), que surgen como alternativa a los PMIs clásicos para el diseño de metaheurísticas basadas en PMIs ([3, 13, 15]).

## 3. Algoritmo genético local binario

En esta sección, presentamos un AGL Binario (AGLB) que puede emplearse para diseñar metaheurísticas basadas en PMIs. El AGLB es un AG *estacionario* ([25, 28]) que inserta, en cada iteración, un sólo nuevo elemento en la población ( $P$ ) aplicando un *método de reemplazo por agrupamiento*, la *selección por torneo restringido* (STR) ([7]). Hemos escogido

STR porque favorece la formación de *nichos* en  $P$  (agrupaciones de cromosomas de alta calidad localizados en regiones diferentes del espacio de búsqueda). El AGLB mantiene un cromosoma externo, la *solución actual* ( $s^a$ ), que siempre se selecciona como padre para realizar el cruce. En las siguientes secciones describimos, en detalle, las principales componentes de nuestra propuesta de AGL.

### 3.1. Esquema general del AGLB

Supongamos que una metaheurística basada en PMIs aplica el AGLB como PMI para refinar una solución particular. Entonces, el AGLB considera esta solución  $s^a$  y lleva a cabo los siguientes pasos (Ver Fig. 1):

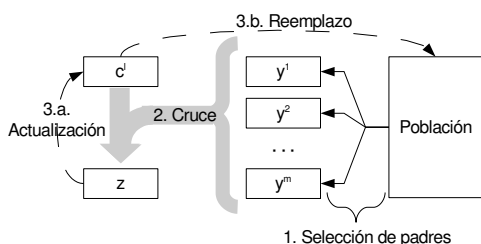


Figura 1: Esquema General del AGLB

1. *Selección de padres.* Se escogen  $m$  cromosomas ( $\{y^1, y^2, \dots, y^m\}$ ) aplicando  $m$  veces el *emparejamiento variado positivo* (Sección 3.2).
2. *Cruce.* Los  $m$  padres escogidos anteriormente se cruzan con  $s^a$  mediante el *operador de cruce uniforme multipadre*, generando un descendiente,  $z$  (Sección 3.3).
3. *Actualización de  $s^a$  y reemplazo.* El mejor entre  $z$  y  $s^a$  se convierte en el nuevo  $s^a$  y el otro se inserta en la población mediante la *selección por torneo restringido* (Sección 3.4).

Estos pasos se repiten hasta alcanzar la condición de parada que se describe en la Sección 3.5.

### 3.2. Emparejamiento variado positivo

Fernandes y otros ([2]) presentaron el emparejamiento variado como mecanismo de emparejamiento para los AGs que fuerza que los padres involucrados en el cruce sean muy semejantes (emparejamiento variado positivo) o muy distintos (emparejamiento variado negativo), imitando así, dos fenómenos similares que ocurren en la propia naturaleza.

La implementación del emparejamiento variado positivo propuesta por estos autores es la siguiente. Se selecciona un primer padre por el método de la ruleta y  $n_{ass}$  cromosomas más mediante el mismo método (en el AGLB el primer padre es  $s^a$  y todos los demás candidatos se escogen de forma aleatoria). Después, se calcula la similitud (distancia Hamming) entre el primer padre y los otros cromosomas, y se elige como segundo padre el de mayor similitud (menor distancia Hamming). El AGLB repite todo este proceso  $m$  veces.

### 3.3. Cruce uniforme multipadre

El AGLB usa una versión *multipadre* del *cruce uniforme parametrizado* ([24]) específicamente diseñada para crear un descendiente cercano a  $s^a$ . Además, combinamos este operador con un mecanismo de *memoria a corto plazo*.

El cruce uniforme multipadre se aplica sobre  $s^a$  y el conjunto de padres seleccionados con el emparejamiento variado positivo ( $y^1, y^2, \dots, y^m$ ). Crea un descendiente,  $z$ , escogiendo genes de  $s^a$ , con probabilidad  $p_f$  (parámetro asociado al operador), y genes de padres aleatorios del conjunto de padres.

Por otro lado, el mecanismo de memoria a corto plazo hace más eficiente el muestreo de descendientes en diferentes regiones del entorno de  $s^a$ . Su idea es obligar a que las diferencias entre el nuevo descendiente y  $s^a$  sean distintas a las diferencias entre  $s^a$  y descendientes generados previamente. Para ello, el mecanismo mantiene una memoria con los genes en los que existen diferencias entre  $s^a$  y algún descendiente generado previamente ( $\{i : z_i^k \neq s_i^a, \forall z^k \text{ descendiente generado de } s^a\}$ ). Entonces, en el nuevo descendiente, se impide

crear diferencias en esos genes. Inicialmente, y cuando un descendiente reemplaza a  $s^a$ , la memoria estará vacía.

Resumiendo, el cruce uniforme multipadre con mecanismo de memoria a corto plazo crea un descendiente  $z$  con las siguientes propiedades:

- $z_i$  es igual a  $s_i^a$  para todos los genes indicados en la memoria a corto plazo.
- Si el gen  $i$  no está marcado en la memoria,  $z_i$  será  $s_i^a$  con probabilidad  $p_f$ . En otro caso,  $z_i$  toma el valor del gen  $i$  de un padre aleatorio  $y^j$ . Se marcará en la memoria el gen  $i$  si  $z_i \neq s_i^a$ .
- Finalmente, si  $z = s^a$ , entonces se cambiará un gen aleatorio  $z_k$  no marcado en la memoria (el cual quedará marcado).

### 3.4. Selección por torneo restringido

El AGLB considera la *Selección por Torneo Restringido* (STR) ([7]) como método de reemplazo por agrupamiento. STR escoge de forma aleatoria  $n_T$  (un parámetro asociado al método) miembros de la población y busca el más parecido a la solución a insertar. Si ésta es mejor lo reemplaza.

La aplicación de STR, junto con el uso de un tamaño de población alto, puede favorecer la creación de nichos en  $P$ . De esta forma, la población del AGLB es capaz de adquirir conocimiento sobre la localización de las regiones prometedoras del espacio de búsqueda y guiar futuras optimizaciones.

### 3.5. Condición de parada

Es importante destacar que, cuando la memoria a corto plazo ha marcado todos los genes (Sección 3.3), entonces, AGLB no podrá mejorar  $s^a$ , porque el operador de cruce sólo produciría soluciones exactamente iguales a  $s^a$ . Por ello, esta condición determina cuando AGLB debe parar y devolver  $s^a$  a la metaheurística.

## 4. Estudio empírico

El propósito de esta sección es comparar el AGLB propuesto, AGLB, con otros PMIs clásicos

propuestos en la literatura. En particular, pretendemos analizar:

- La capacidad de AGLB para refinar soluciones.
- Su adecuación para combinarse con metaheurísticas basadas en PMIs.

Compararemos el AGLB con los siguientes PMIs clásicos propuestos con anterioridad en la literatura para problemas de optimización combinatoria:

- PMI *el primer mejor* (PrimerM) ([1]) cambia genes aleatorios de la solución actual, y selecciona el primero que mejora su calidad.
- PMI *el mejor* ([1]) examina todos los posibles cambios y selecciona el que produce un mayor incremento en la calidad.
- PMI *K-opt* (Kopt) ([17]) examina cambios de  $k$  genes para elegir el mejor encontrado. Para ello, iterativamente cambia el gen que produce un mayor incremento, o el menor detrimento, en la calidad, de entre los genes aún no cambiados, y después selecciona la mejor solución producida para repetir el proceso.
- PMI *RandK-opt* (RandK) ([11, 18]) Igual que K-opt pero, cambiando genes aleatorios que produzcan una mejora, si existen, o el que produzca el menor detrimento en otro caso.

La Tabla 1 enumera los problemas de test usados, su dimensión, máximo número de evaluaciones, valor óptimo (cota<sup>(1)</sup>) o mejor valor conocido en la literatura<sup>(2)</sup>), y referencias. Hemos formulado todos ellos como problemas de maximización.

En la sección 4.1 estudiamos la capacidad de refinamiento de soluciones del AGLB. En la sección 4.2 comparamos la adecuación del AGLB en una metaheurística basada en PMIs frente a la de los otros PMIs.

Tabla 1: Nombre, dimensión, máximo número de evaluaciones, valor óptimo, y referencia de los problemas usados

Índice	Nombre	Dim	Max Evals	$f^*$	Ref
1	BQP(50)	50	$10^5$	2098 <sup>2</sup>	[6]
2	BQP(100)	100	$10^5$	7970 <sup>2</sup>	[6]
3	BQP(250)	250	$10^5$	45607 <sup>2</sup>	[6]
4	BQP(500)	500	$10^6$	116586 <sup>2</sup>	[6]
5	Deceptive(13)	39	$10^5$	390	[5]
6	Deceptive(134)	402	$10^5$	4020	[5]
7	Maxcut(G11)	800	$10^6$	572.7 <sup>1</sup>	[10]
8	Maxcut(G12)	800	$10^6$	621 <sup>1</sup>	[10]
9	Maxcut(G17)	800	$10^6$	No conocido	[10]
10	Maxcut(G18)	800	$10^6$	1063.4 <sup>1</sup>	[10]
11	Maxcut(G43)	1000	$10^6$	7027 <sup>1</sup>	[10]
12	PPeaks(50,100)	100	$10^5$	1	[23]
13	PPeaks(50,150)	150	$10^5$	1	[23]
14	PPeaks(50,200)	200	$10^5$	1	[23]
15	PPeaks(100,100)	100	$10^5$	1	[23]
16	Trap(1)	36	$10^5$	220	[26]
17	Trap(4)	144	$10^5$	880	[26]
18	NkLand(48,4)	48	$10^5$	1 <sup>1</sup>	[12]
19	NkLand(48,12)	48	$10^5$	1 <sup>1</sup>	[12]
20	M-Sat(100,1200,3)	100	$10^5$	1 <sup>1</sup>	[22]
21	M-Sat(100,2400,3)	100	$10^5$	1 <sup>1</sup>	[22]

#### 4.1. Capacidad de refinamiento

En esta sección, estudiamos la capacidad del AGLB para optimizar soluciones. Para ello, implementaremos instancias de la *búsqueda local multiarranque* (BLMA) ([16]) con el AGLB y los otros PMIs. Utilizamos la notación BLMA-⟨PMI⟩ para denominar a los distintos algoritmos resultantes.

La BLMA proporciona soluciones aleatorias de partida a los PMIs y devuelve la mejor solución encontrada. Hemos escogido esta metaheurística para evitar posibles sinergias entre la metaheurística y los PMIs, y así conseguir una comparación equitativa entre el AGLB y los PMIs clásicos.

Todos los algoritmos se han ejecutado 30 veces, consumiendo, cada ejecución, 100.000 evaluaciones o 1.000.000, dependiendo del problema (problemas en los que tras las primeras 100.000 evaluaciones, los algoritmos aún conseguían mejoras significativas) (Ver Tabla 1). El AGLB utiliza una población de 500 individuos,  $p_f = 1 - 7/Dim$  y  $m = 10$  para el operador de cruce,  $n_{ass} = 5$  para el emparejamiento variado positivo y  $n_T = 15$  para la selección por torneo restringido. Es importan-

Tabla 2: Resultados de las instancias de BLMA

	BLMA-Kopt	BLMA-Mejor	BLMA-RandK	BLMA-PrimerM	BLMA-AGLB
1	2098 ~	2095 ~	2095 ~	2098 ~	2098
2	7939 +	7827 +	7887 +	7891 +	7954
3	38008 +	45148 +	38279 +	45551 +	45571
4	114514 ~	114853 +	114806 +	115890 ~	115840
5	390 -	386 -	376 +	381 ~	381
6	2848 +	3891 -	2829 +	3842 -	3834
7	305 +	433 +	185 +	447 +	509
8	302 +	425 +	168 +	438 +	510
9	2743 +	2922 +	2675 +	2931 +	3009
10	593 +	844 +	401 +	866 +	956
11	5485 +	6408 +	5353 +	6456 +	6571
12	0.991 +	1.000 ~	0.978 +	1 ~	1
13	0.973 +	0.995 +	0.949 +	1 ~	1
14	0.955 +	0.974 +	0.878 +	1 ~	0.999
15	0.990 +	0.998 +	0.974 +	1 ~	1
16	220 ~	219 ~	201 +	212 +	220
17	880 -	828 +	782 +	794 +	867
18	0.770 ~	0.759 +	0.765 +	0.762 +	0.773
19	0.764 -	0.740 ~	0.749 -	0.745 -	0.738
20	0.959 ~	0.955 +	0.959 ~	0.958 +	0.960
21	0.936 ~	0.934 +	0.937 ~	0.936 ~	0.937
+	12	15	17	11	
~	6	4	3	8	
-	3	2	1	2	

te destacar que la población del AGLB no se reinicializa en cada iteración de la metaheurística, esto es, la población inicial del AGLB es la final obtenida en la optimización anterior.

La Tabla 2 muestra la media del mejor fitness obtenido por cada algoritmo en cada función. Además, se ha aplicado el *test de Student*, cuando las condiciones de normalidad se cumplen y el de *Wilcoxon-Mann-Whitney* ([9]) cuando no se cumplen, para asegurar si las diferencias en el rendimiento del BLMA-AGLB son significativas comparadas con el de los otros algoritmos:

- Un signo más (+) indica que la media del BLMA-AGLB es mejor que la del algoritmo correspondiente.
- Un signo menos (-) indica que el algoritmo mejora la media del BLMA-AGLB.
- Un signo de aproximación (~) indica que no hay diferencias significativas.

Hemos añadido las últimas tres filas que cuentan el número de mejoras, no diferencias y reducciones, de acuerdo a los test, por algoritmos.

Las últimas tres filas indican que el BLMA-AGLB es significativamente mejor que el resto de algoritmos porque obtiene muchas mejoras y pocas reducciones con respecto a los demás. Por tanto, podemos concluir que el AGLB es capaz de optimizar efectivamente soluciones aleatorias dando mejores soluciones que los PMIs de la literatura.

#### 4.2. Adecuación a metaheurísticas basadas en PMIs

El propósito de esta sección es estudiar el beneficio de utilizar el AGLB, frente a los PMIs clásicos, cuando se aplican dentro de una metaheurística basada en PMIs más avanzada que la BLMA. Para ello, hemos implementado distintas instancias de la *Búsqueda de Vecindario Variable* (BVV) ([19]).

BVV utiliza un conjunto ordenado de vecindarios  $\{N^1, \dots, N^m\}$  (normalmente anidados) para generar las soluciones iniciales para el PMI. BVV obtiene una nueva solución inicial para la PMI utilizando el vecindario actual sobre la mejor solución encontrada ( $N^k(s^{mejor})$ ) anteriormente. Si el PMI obtiene una solución mejor que  $s^{mejor}$ , entonces se toma el primer vecindario como el actual, en otro caso, se elige  $N^{k+1}$ . Normalmente  $N^i(x) \subset N^{i+1}(x)$ , sin embargo, algunos estudios proponen utilizar el espacio definido por  $\{N^i(x) \setminus N^{i-1}(x)\}$  ([14]).

Implementaremos dos familias de BVVs que utilizan operadores diferentes para definir los vecindarios. Ambos aplican perturbaciones sobre  $s^{mejor}$  según una probabilidad  $p$  dada. Los vecindarios de menor índice se corresponderán con los de probabilidad baja (perturbación débil) y los de mayor con los de probabilidad alta (perturbación fuerte):

- BVV1 asigna valores aleatorios a los genes de  $s^{mejor}$  con probabilidad  $p$ . BVV1 sigue la idea de vecindarios anidados.
- BVV2 cambia los genes de  $s^{mejor}$  con probabilidad  $p$ . BVV2 tiene una idea similar a la de vecindarios con intersección vacía.

Ambas familias de BVVs iterarán sobre 9 valores de perturbación ( $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ ), volviendo al primero si aún

Tabla 3: Resultados de las instancias de BVV1

	BVV1- -Kopt	BVV1- -Mejor	BVV1- -RandK	BVV1- -PrimerM	BVV1- -AGLB
1	2098 ~	2098 ~	2094 ~	2098 ~	2098
2	7925 ~	7901 ~	7863 +	7931 -	7900
3	34194 +	45388 +	42282 +	45588 ~	45546
4	114718 +	115694 +	114205 ~	116534 -	115892
5	390 -	389 -	374 +	384 +	386
6	2841 +	3891 -	2824 +	3827 ~	3831
7	308 +	483 +	166 +	511 +	527
8	296 +	474 +	168 +	507 +	521
9	2738 +	2962 +	2677 +	2982 +	3003
10	562 +	895 +	414 +	933 +	943
11	5538 +	6472 +	5386 +	6558 ~	6555
12	0.961 +	0.987 +	0.847 +	0.999 ~	0.998
13	0.923 +	0.939 +	0.875 +	0.995 ~	0.996
14	0.923 +	0.920 +	0.846 +	0.988 +	0.997
15	0.947 +	0.987 +	0.838 +	0.997 +	0.999
16	220 ~	220 ~	198 +	220 ~	220
17	880 -	852 +	778 +	804 +	862
18	0.772 ~	0.772 ~	0.768 ~	0.772 ~	0.773
19	0.767 -	0.753 +	0.754 +	0.758 ~	0.760
20	0.959 ~	0.958 ~	0.959 ~	0.959 ~	0.959
21	0.936 ~	0.936 ~	0.936 ~	0.937 ~	0.936
+	12	12	16	8	
~	6	7	5	11	
-	3	2	0	2	

no se han consumido el máximo número de evaluaciones. El resto de parámetros, problemas de test y condiciones de ejecución serán las mismas a las utilizadas en la sección 4.1.

Las tablas 3 y 4 muestran los resultados de las instancias de los BVVs en el formato indicado en la sección 4.1. Los resultados muestran que las metaheurísticas basadas en AGLB (BVV1-AGLB y BVV2-AGLB) producen mejores resultados que las basadas en otros PMIs, porque obtienen muchas mejoras y pocas reducciones. Por tanto, concluimos que AGLB, frente a otros PMIs, es adecuado para las metaheurísticas basadas en PMIs estudiadas.

Por último, podemos ver que el uso de AGLB en metaheurísticas más avanzadas, como BVV, frente a su uso en BLMA, no aporta diferencias muy significativas. En estudios futuros nos planteamos determinar qué características deben presentar las metaheurísticas para obtener un mayor beneficio de AGLB.

## 5. Conclusiones

En este estudio, hemos presentado el AGLB, un AGL que incorpora mecanismos específicos para la selección de padres, cruce, y reemplazo con la intención de realizar un proceso de me-

Tabla 4: Resultados de las instancias de BVV2

	BVV2- -Kopt	BVV2- -Mejor	BVV2- -RandK	BVV2- -PrimerM	BVV2- -AGLB
1	2098 ~	2098 ~	2097 ~	2098 ~	2098
2	7937 ~	7873 +	7876 +	7916 +	7952
3	34194 +	45387 +	42281 +	45575 +	45593
4	114824 +	115535 +	114207 ~	116476 -	116192
5	390 -	389 -	376 +	383 +	384
6	2841 +	3891 -	2824 +	3882 ~	3883
7	308 +	469 +	166 +	506 +	527
8	296 +	460 +	168 +	502 +	521
9	2738 +	2954 +	2677 +	2979 +	3006
10	562 +	892 +	414 +	935 +	944
11	5538 +	6472 +	5386 +	6567 ~	6572
12	0.977 +	0.996 +	0.934 +	1.000 ~	1
13	0.926 +	0.981 +	0.875 +	0.998 +	1.000
14	0.918 +	0.951 +	0.846 +	0.997 ~	0.997
15	0.974 +	0.994 +	0.921 +	1.000 ~	0.999
16	220 ~	220 ~	205 +	218 +	220
17	880 -	843 +	777 +	796 +	864
18	0.772 ~	0.768 ~	0.767 ~	0.771 ~	0.773
19	0.763 -	0.754 ~	0.752 ~	0.755 ~	0.753
20	0.959 ~	0.956 +	0.959 ~	0.959 ~	0.959
21	0.936 ~	0.936 ~	0.936 ~	0.936 ~	0.936
+	12	14	15	10	
~	6	5	6	10	
-	3	2	0	1	

jora iterativa hacia las regiones prometedoras representadas en su población.

Un estudio experimental, con 21 problemas binarios, ha mostrado que AGLB puede optimizar efectivamente soluciones aleatorias y puede mejorar los resultados de una metaheurística basada en PMIs frente al uso de otros PMIs clásicos presentados en la literatura.

De este estudio surgen varias ideas para investigaciones futuras:

- Analizar el comportamiento del AGLB cuando se usa en otras metaheurísticas basadas en PMIs, como GRASP, optimización por colonias de hormigas o algoritmos meméticos.
- Extender nuestra investigación a otros conjuntos de problemas test (otras codificaciones) y problemas reales.
- Estudiar mecanismos que adapten los parámetros del algoritmo según el estado actual del proceso de búsqueda.

## Agradecimientos

Este estudio fue subvencionado por el proyecto Español, del MEC, TIN2005-08386-C05-01.

## Referencias

- [1] C Blum and A Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308, 2003.
- [2] C Fernandes and A Rosa. A study on non-random mating and varying population size in genetic algorithms using a royal road function. In *Proc. of the 2001 Congress on Evolutionary Computation*, pages 60–66. IEEE Press, Piscataway, New Jersey, 2001.
- [3] C García-Martínez, M Lozano, F Herrera, D Molina, and AM Sánchez. Global and local real-coded genetic algorithms based on parent-centric crossover operators. *European Journal of Operational Research*, 2007. In Press.
- [4] DE Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.
- [5] DE Goldberg, B Korb, and K Deb. Messy genetic algorithms: motivation, analysis, and first results. *Complex Systems*, 3:493–530, 1989.
- [6] VP Gulati, SK Gupta, and AK Mittal. Unconstrained quadratic bivalent programming problem. *European Journal of Operational Research*, 15:121–125, 1984.
- [7] G Harik. Finding multimodal solutions using restricted tournament selection. In LJ Eshelman, editor, *Proc. of the 6th International Conference on Genetic Algorithms*, pages 24–31. Morgan Kaufmann, San Mateo, California, 1995.
- [8] JH Holland. *Adaptation in natural and artificial systems*. MIT Press Cambridge, MA, USA, 1992.
- [9] M Hollander and DA Wolfe. *Nonparametric statistical inference*. John Wiley & Sons, New York, 1973.

- [10] RM Karp. Reducibility among combinatorial problems. In R Miller and J Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
- [11] K Katayama and H Narihisa. A variant k-opt local search heuristic for binary quadratic programming. *Trans. IEICE (A)*, J84-A(3):430–435, 2001.
- [12] SA Kauffman. Adaptation on rugged fitness landscapes. *Lectures in the Sciences of Complexity*, 1:527–618, 1989.
- [13] SA Kazarlis, SE Papadakis, JB Theocharis, and V Petridis. Microgenetic algorithms as generalized hill-climbing operators for GA optimization. *IEEE Transactions on Evolutionary Computation*, 5(3):204–217, 2001.
- [14] L Liberti and M Dražić. Variable Neighborhood Search for the Global Optimization of Constrained NLPs. In *Proceedings of GO*, pages 1–5, 2005.
- [15] M Lozano, F Herrera, N Krasnogor, and D Molina. Real-Coded Memetic Algorithms with Crossover Hill-Climbing. *Evolutionary Computation*, 12(3):273–302, 2004.
- [16] R Marti. Multi-start methods. In F Glover and G Kochenberger, editors, *Handbook of Metaheuristics*, pages 355–368. Kluwer Academic Publishers, Norwell, MA, 2003.
- [17] P Merz. *Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies*. PhD thesis, Department of Electrical Engineering and Computer Science, University of Siegen, Germany, 2000.
- [18] P Merz and K Katayama. Memetic algorithms for the unconstrained binary quadratic programming problem. *Bio Systems*, 79(1–3):99–118, 2004.
- [19] N Mladenovic and P Hansen. Variable neighborhood search. *Computers in Operations Research*, 24:1097–1100, 1997.
- [20] P Moscato. Memetic algorithms: a short introduction. In D Corne, M Dorigo, and F Glover, editors, *New ideas in optimization*, pages 219–234. McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999.
- [21] CC Ribeiro, and P Hansen (eds.). *Essays and Surveys in Metaheuristics*. Kluwer Academic Publishers, Boston, 2001.
- [22] K Smith, HH Hoos, and T Stützle. Iterated robust tabu search for max-sat. In JG Carbonell and J Siekmann, editors, *Proc. of the 16th conference on the Canadian Society for Computational Studies of Intelligence, LNCS 2671*, pages 129–144. Springer, Berlin Heidelberg, 2003.
- [23] WM Spears. *Evolutionary Algorithms: The Role of Mutation and Recombination*. Springer, 2000.
- [24] WM Spears and KA De Jong. On the virtues of parameterized uniform crossover. In *Proc. of the Fourth International Conference on Genetic Algorithms*, pages 230–236. Morgan Kaufman, 1991.
- [25] G Sywerda. Uniform crossover in genetic algorithms. In JD Schaffer, editor, *Proc. of the Third International Conference on Genetic Algorithms*, pages 2–9. Morgan Kaufmann, San Francisco, CA, USA, 1989.
- [26] D Thierens. Population-based iterated local search: restricting neighborhood search by crossover. In *Proc. of the Genetic and Evolutionary Computation Conference, LNCS 3103*, pages 234–245. Springer, Berlin Heidelberg, 2004.
- [27] S Voß, IH Osman, and JP Kelly (eds.). *Meta-Heuristics: The Theory and Applications*. Kluwer Academic Publishers, Boston, 1996.
- [28] D Whitley. The genitor algorithm and selection pressure: why rank-based allocation of reproductive trials is best. In JD Schaffer, editor, *Proc. of the Third International Conference on Genetic Algorithms*, pages 116–121. Morgan Kaufmann, San Francisco, CA, USA, 1989.



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



**CEDI 2007**

II CONGRESO ESPAÑOL  
DE INFORMÁTICA

**ZARAGOZA SPAIN**

11 AL 14 DE SEPTIEMBRE DE 2007

**PROGRAMA**

*Nuevos retos científicos y tecnológicos en Ingeniería Informática*

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