

Descomposición Jerárquica no Homogénea de Nichos Basados en Regiones

Daniel Molina¹ and Francisco Herrera²

¹ Departamento de Informática, Universidad de Cádiz, daniel.molina@uca.es

² Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, herrera@decsai.ugr.es

Resumen En los algoritmos evolutivos es muy importante que la población mantenga una adecuada diversidad durante el proceso de búsqueda. Para mantener dicha diversidad es común el uso de técnicas para mantener nichos, como la técnica de *clearing*. En trabajos anteriores se propuso una técnica de nichos basada en regiones que divide el espacio de búsqueda en hipercubos predefinidos de igual tamaño, y se reducía el tamaño de todos ellos durante la búsqueda para aumentar el grado de explotación. En este trabajo proponemos un mecanismo de división del espacio de búsqueda no homogéneo, dividiendo más las regiones más prometedoras. Probamos el nuevo mecanismo de descomposición obteniendo mejores resultados, gracias a una explotación más eficiente.

Keywords: Optimización real, estrategia de nichos, optimización multimodal, búsqueda adaptativa.

1. Introducción

Existen numerosos problemas de optimización complejos que pueden ser abordados de forma exitosa por medio de algoritmos evolutivos (AEs). Pero para que éstos puedan obtener buenos resultados es necesario que realicen una adecuada exploración por todo el espacio de búsqueda, especialmente en la llamada optimización multimodal [8,9], en donde el objetivo no es obtener un único óptimo, si no todos los óptimos posibles. Para que los AEs puedan garantizar un cierto grado de diversidad durante la búsqueda se han desarrollado técnicas, como la estrategias de *niching* [2]. Dentro de estas técnicas se encuentra el *clearing* [7], que presenta una excesiva dependencia del tamaño de los nichos [4,7]. Aunque existen algunos trabajos para intentar calcular los nichos de forma automática [3], requieren conocer tanto el número de óptimos como su distancia, algo no conocido en problemas reales, por lo que su aplicación es muy limitada.

En trabajos anteriores [5,6] hemos propuesto distintos algoritmos meméticos que, entre sus principales características, presentaban una nueva técnica de nichos. Dicha técnica particiona el espacio de búsqueda en hipercubos de igual tamaño denominados regiones. El modelo de regiones presenta la ventaja de que es más sencillo de procesar al no requerir calcular las distancias entre soluciones. Además, para reducir la dependencia respecto al tamaño de las regiones, se

reducía su tamaño durante la búsqueda. Este modelo ofrecía un buen comportamiento, pero, al trabajar con hipercubos del mismo tamaño que son divididos por igual, se incentiva un mismo grado de exploración independientemente de la zona de búsqueda. Sin embargo, es conocido que para obtener buenos resultados es conveniente centrarse más en las zonas más prometedoras [1].

En este trabajo presentamos un nuevo modelo de división, en el que se evalúa a las regiones considerando el valor de *fitness* de sus soluciones, y sólo las consideradas más prometedoras son divididas en regiones de menor tamaño. De esta manera, se permite una mayor explotación en las regiones consideradas más interesantes. Para ello, se utiliza la información de todas las soluciones generadas, aunque no hayan llegado a introducirse en la población. Para probar el nuevo modelo se ha utilizado un algoritmo evolutivo que utiliza *niching* basado en regiones que mantiene únicamente una solución por cada región. Se aplicará sobre dicho algoritmo los distintos criterios de descomposición de regiones (homogénea y no homogénea) y se comparará el comportamiento obtenido con cada una.

Este trabajo se organiza de la siguiente manera. En la Sección 2, presentamos primero el modelo original de nichos con división homogénea. En la Sección 3, se detalla la propuesta del artículo. En la Sección 4, planteamos la parte experimental, y comparamos el modelo propuesto frente al modelo original. Finalmente, en la Sección 5, planteamos las principales conclusiones del trabajo, y planteamos líneas de trabajo futuro.

2. Modelo de Regiones con División Homogénea

En esta sección detallamos el modelo de nichos original propuesto en [5], así como el proceso seguido para descomponer las regiones.

En [5] se redefine la noción de nicho como una división fija del espacio de búsqueda. Para ello, se divide el espacio de búsqueda de cada dimensión en ND partes homogéneas creando así una malla de hipercubos o regiones de igual tamaño. Cada región así creada representa un nicho. La ventaja de definir de esta forma los nichos es que permite identificar fácilmente los nichos existentes únicamente a partir de cada solución evitando el coste computacional de calcular las distancias entre soluciones.

Usando dicha técnica cada vez que el algoritmo genera una nueva solución se comprueba si existe ya otra en la misma región, y en ese caso la reemplaza si posee mejor *fitness*, de forma similar a la técnica de *clearing*. La Figura 1 muestra un ejemplo en un espacio 2D. En la subfigura (a) se muestra el espacio de búsqueda original con una serie de soluciones generadas de forma aleatoria (de fondo está el contorno de la conocida función *Sphere*). En las subfiguras (b) y (c) se muestra el resultado de dividir el espacio de búsqueda con $ND=3$ y $ND=6$. Para cada una de las regiones, las distintas soluciones compiten entre sí para mantenerse en la población. Se puede observar que cuanto menor es ND mayor es la competencia y también la diversidad entre las soluciones finales.

Lo ideal sería que cada región representase una zona con un único óptimo, pero, como no se puede garantizar, el tamaño de las regiones es un parámetro

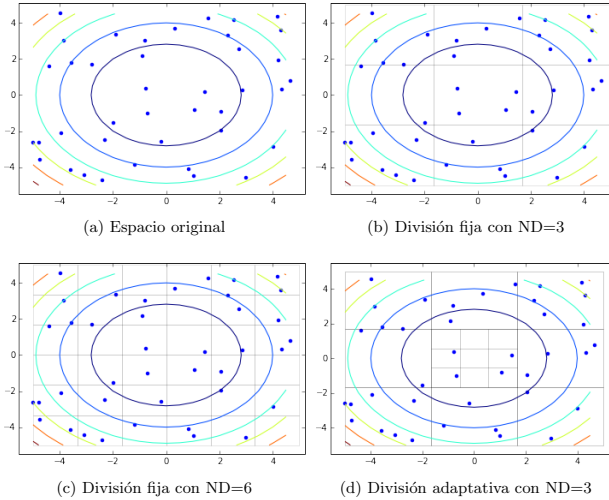


Figura 1: Las distintas estrategias de división del espacio de búsqueda

crítico. Para reducir este inconveniente y poder darle así más robustez al modelo, el tamaño de las regiones se va reduciendo durante la búsqueda.

En el modelo original, el número de división por dimensión, ND , se incrementa u veces durante la búsqueda, realizando una nueva división homogénea del espacio de búsqueda en regiones de menor tamaño. Más en detalle, dado que el algoritmo se ejecuta durante $MaxFE_s$ evaluaciones, cada $MaxFE_s/(u + 1)$ evaluaciones $ND_i = m_u \cdot ND_{i-1}$ donde m_u es el multiplicador.

3. Modelo de Regiones con División No Homogénea

En este trabajo se presenta un nuevo modelo de división de regiones caracterizado en que es no homogéneo, es decir, no todas las regiones del espacio de búsqueda se volverán a dividir en regiones de menor tamaño. El interés de una división no homogénea del espacio de búsqueda es múltiple. Por un lado, las regiones no consideradas suficientemente prometedoras no tienen por qué dividirse de nuevo, al mismo tiempo que fomenta de forma controlada una mayor exploración de las regiones prometedoras, mejorando el comportamiento. Es

más, aunque las regiones mantienen diversidad en la población el modelo de exploración del algoritmo seguirá tendiendo a generar más soluciones en las regiones más prometedoras, compitiendo éstas entre sí. La división no homogénea reduce esta disparidad de comportamiento manteniendo al mismo tiempo una diversidad controlada, fomentando una mayor colaboración entre el algoritmo de exploración y el mecanismo de *niching*.

3.1. Identificación de las Regiones más Prometedoras

El primer paso es identificar las regiones consideradas más prometedoras. Para poder tomar esta decisión se registra para cada región todos los valores de *fitness* de las soluciones generadas en dicha región. Al considerar todas las soluciones generadas por el algoritmo de exploración se utiliza el máximo grado del información disponible (sin aumentar el coste en evaluaciones).

Dado que el número de soluciones generadas en cada región es muy variable, el algoritmo establece un número mínimo de soluciones, min_{reg} . Para toda aquella región en el que se haya generado un número inferior a min_{reg} se considerará que su información no es suficiente, y no se tendrá en cuenta ni para ser elegida como prometedoras ni para identificar las regiones más prometedoras.

El primer paso es eliminar como regiones candidatas a aquellas regiones cuyo mejor valor sea peor que el peor valor de otra región, es decir, dado dos regiones R_m y R_n , se considera que la región R_m mejora claramente a R_n si se cumple la Ecuación (1).

$$max(R_m) < min(R_n) \quad (1)$$

en donde $max(R_i)$ y $min(R_i)$ representan los valores mínimos y máximo de los *fitness* de todas las soluciones generadas en la región R_i .

Una vez eliminadas todas las regiones mejoradas por otras, el resto de regiones se ordenan según la media de los *fitness* de las soluciones generadas en dicha región ($avg(R_i)$), y se consideran las $N_{promising}$ más prometedoras. El algoritmo 1 muestra el pseudocódigo del criterio de elección. Es de destacar que, aunque una región no se haya elegido para ser subdividida, podrán seguir generándose soluciones en dicha región (y compitiendo entre sí) y puede que en un futuro sí sea elegida para ser subdividida.

3.2. Esquema de División de Regiones

Una vez que se han identificado las mejores regiones, éstas se vuelven a dividir usando de nuevo una división por dimensión, con el mismo parámetro ND, pero sobre el tamaño de la región y no sobre el espacio de búsqueda completo. En la Figura 1(d) se observa un ejemplo, en el que el algoritmo ha considerado como prometedor la región central, y la ha subdividido a su vez en regiones de menor tamaño.

Para realizar la división, se define el concepto de espacio divisible, como el espacio que será dividido en regiones de igual tamaño. Cada espacio está identificado por un rango en el espacio de búsqueda. Así pues, para cada solución

Algoritmo 1 Sistema de selección de regiones prometedoras

Para cada región R_i calcular $\min(R_i)$, $\max(R_i)$ y $\text{avg}(R_i)$.
 $\text{Regiones} \leftarrow \{\text{Listado de regiones}\}$
for all $R_m \in \text{Regiones}$ **do**
 for all $R_n \in \text{Regiones}$ **do**
 if $\min(R_m) > \max(R_n)$ **then**
 $\text{Regiones} \leftarrow \text{Regiones} - \{R_m\}$.
 end if
 end for
end for
Ordenar $\{\text{Regiones}\}$ de forma incremental según los valores $\text{avg}(R_i)$.
Devolver los $N_{\text{promising}}$ primeros elementos de la lista ordenada de regiones.

Algoritmo 2 Obtención de la región de la solución sol a partir de la lista de espacios disponibles espacios

$i \leftarrow$ último índice de espacios .
while no encontrado región y $i >= 0$ **do**
 $\min \leftarrow$ rango mínimo de $\text{espacios}[i]$.
 $\max \leftarrow$ rango máximo de $\text{espacios}[i]$.
 $\text{size}_{\text{hyper}} \leftarrow$ tamaño de los hipercubos de $\text{espacios}[i]$.
 if $\text{sol}_d \in [\min_d, \max_d] \quad \forall d \in [1, \dots, D]$ **then**
 $\text{region}_{\text{int}} \leftarrow$ parte entera de $(\text{sol} - \min) / \text{size}_{\text{hyper}}$.
 devuelve el par de valores $i, \text{region}_{\text{int}}$.
 else
 $i \leftarrow i - 1$.
 end if
end while

se almacena el espacio divisible en donde se encuentra, y la región dentro de dicho espacio. El algoritmo mantiene una lista de los espacios divisibles, inicialmente compuesto por un único espacio, que refleja todo el espacio de búsqueda.

Cuando se elige subdividir una región, se crea un nuevo espacio divisible con el área de dicha región, que se almacenará en dicha lista, considerando como valores mínimos y máximos de dicho espacio los valores de la región que se ha subdividido. El tamaño de cada región en el nuevo espacio se calcula dividiendo el nuevo rango de exploración entre el número de divisiones.

3.3. Estructura de la Memoria

Dado que el número de regiones puede ser muy amplio, y más conforme aumenta el número de divisiones, la memoria sólo almacena información sobre las regiones ya exploradas. Más concretamente, almacena la información de cada uno de los espacios divisibles (su rango, y el tamaño de las regiones que lo componen). La Figura 2 muestra un ejemplo que representa la división de la figura 1(d). Como se puede observar, la región $[1, 1]$ ha sido subdividida, por lo que a toda solución entre $[-1.67, 1.66]$ se le asigna una región de menor tamaño.

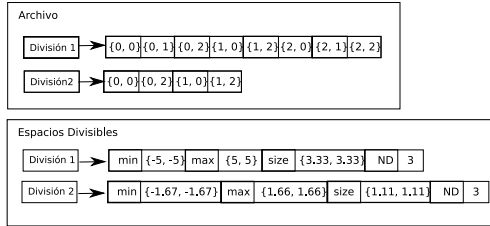


Figura 2: Estructura propuesta de memoria

Así pues, cada solución tiene una región asociada, identificada de forma única con el par (*identificador del espacio, región*). Para calcularlo, se recorre la lista de espacios, comparando el rango de la solución con el rango de cada uno de los elementos. Dado que existe un espacio que incluye todo el espacio de búsqueda, toda solución tendrá un espacio asociado. Una vez encontrado su espacio correspondiente, se le asigna la región dentro de ese espacio. El Algoritmo 2 muestra el pseudocódigo. La lista de espacios divisibles se recorre desde las últimas posiciones hacia adelante para buscar primero los últimos espacios añadidos.

4. Sección Experimental

En esta sección vamos a analizar el comportamiento del nuevo modelo de división de regiones. Para poder compararlo, usaremos ambos modelos sobre un mismo algoritmo, especificado en la Sección 4.1. Una vez definido el algoritmo de referencia, en la Sección 4.2 se detalla el benchmark usado y las condiciones de experimentación, y en la Sección 4.3 se muestran los resultados obtenidos.

4.1. Algoritmo de Referencia

El modelo original se probó con los algoritmos RMA-LSCh-CMA[5] y RMA-wA [6], pero se ha decidido evitar el uso de la búsqueda local para poder obtener conclusiones más claras. Por tanto, se ha decidido utilizar únicamente el componente evolutivo utilizado en ambos algoritmos.

El algoritmo utilizado es un algoritmo evolutivo estacionario, que utiliza como operador de selección la estrategia negativa inversa, y como operador de cruce el operador $BLX - \alpha$. Por cada nueva solución se calcula su región r_n y se consulta el índice de la memoria. Si ya existe dicha región representada por otra solución, ambas se comparan y la nueva solución se introduce únicamente si mejora a la existente. Si r_n no existía entonces s_n compite con la peor solución de la población para reemplazarla. Este proceso se describe en el Algoritmo 3.

Algoritmo 3 Pseudo-código del Algoritmo Evolutivo utilizado

```
1:  $i = 0$ 
2: while  $i < i_{EA}$  do
3:   Selecciona dos padres
4:   repeat
5:     Crea una solución  $s_n$  mediante cruce y mutación
6:     Calcula la región  $r_n$  en donde pertenece  $s_n$ 
7:     until  $r_i$  se deba explorar
8:     Evalúa  $s_n$ ,  $i = i + 1$ 
9:     Recupera de la población conjunto de soluciones  $S_{r_n}$  de la región  $r_n$ 
10:    if  $S_{r_n} \neq \emptyset$  then
11:       $S_{r_n} \leftarrow S_{r_n} \cup s_n$ 
12:      Borra peor individuo de  $S_{r_n}$ 
13:    else
14:      Reemplaza el peor individuo  $s_{worst}$  de la población si  $f(s_{worst}) > f(s_n)$ 
15:    end if
16: end while
```

Los parámetros del algoritmo son los siguientes: Para la descomposición en regiones se usa $ND_0 = 2$, $u = 4$, y $m_u = 1.7$. Se usa un tamaño de población (NP) igual a 30. Todos estos parámetros son los utilizados en el trabajo [6]. Como valor α de cruce $BLX - \alpha$ se usa 0.5 ya que es más común y ampliamente utilizado para mantener diversidad.

4.2. Benchmark Aplicado y Condiciones de Experimentación

Los experimentos han sido realizados usando cuatro funciones bien conocidas [10]. Estas funciones han sido elegidas por sus propiedades: *Sphere* es muy simple y unimodal, por lo que puede mejorar sustancialmente con mayor explotación. *Rastrigin* es fuertemente multimodal, lo hace necesario una búsqueda con diversidad. *Rosenbrock* es de convergencia rápida, pero en la que es difícil encontrar el óptimo. La última función es una versión de *Sphere* con el óptimo desplazado al vector $\{2, \dots, 2\}$ para ver cómo influye un desplazamiento del óptimo.

Para realizar los experimentos se ha decidido utilizar un número de dimensión reducido, $Dim=5$, ya que el modelo de regiones ofrece buenos resultados para dimensiones bajas [6]. Todas la experimentación se ha realizado con un número máximo de evaluaciones Max_{eval} de $2000 \cdot Dim$. Para comparar algoritmos entre sí, cada algoritmo se ha ejecutado 50 veces, y se ha calculado la media.

4.3. Resultados Experimentales

Primero comparamos la convergencia de la población para cada caso. Las Figuras 3-5 muestran para cada función la convergencia de la población (mostrando la evolución del mejor, así como de la media de la población). Se puede observar que en *Sphere* la nueva propuesta es claramente mejor al dividir más

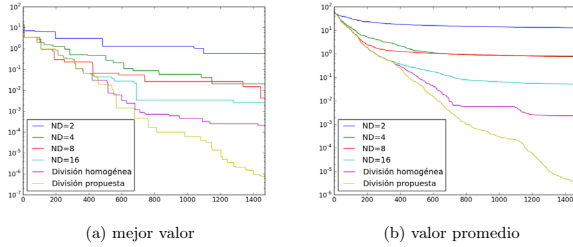


Figura 3: Evolución del *fitness* de la población para la función *Sphere* desplazada

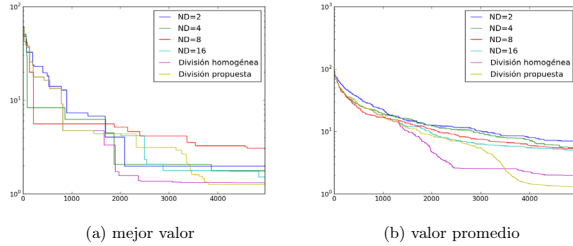


Figura 4: Evolución del *fitness* de la población para la función *Rastrigin*

rápida mente las regiones más prometedoras. En *Rastrigin*, con múltiples regiones prometedoras, el nuevo mecanismo de división mejora más lentamente pero obtiene mejores resultados. En *Rosenbrock* se observan pocas diferencias.

La Tabla 1 muestra una comparativas con los valores de *fitness* utilizando el modelo anterior y el modelo nuevo para distintos valores de dimensión. Se puede observar que en la mayoría de los casos el nuevo modelo ofrece mejores resultados, pero de forma discreta, y se observa que la mejora que supone es mayor cuanto menor es la dimensionalidad. No obstante, es destacable la mejora significativa que supone el nuevo modelo en la esfera desplazada, mostrando un comportamiento más robusto.

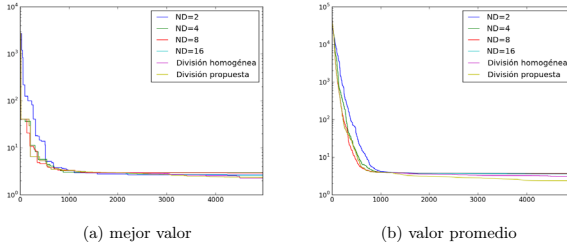


Figura 5: Evolución del *fitness* de la población para la función *Rosenbrock*

Tabla 1: Resultados usando el modelo de división original y el propuesto

Problema	Tipo de división	D=3	D=4	D=5
Rastrigin	homogénea	7.807806e-01	1.107765e+00	1.510784e+00
	no homogénea	7.575266e-01	1.023801e+00	1.444814e+00
Rosenbrock	homogénea	1.004577e-01	4.400635e-01	2.395754e+00
	no homogénea	9.425769e-02	1.066471e+00	2.210630e+00
Sphere	homogénea	8.665659e-06	2.519402e-14	3.751700e-33
	no homogénea	1.307012e-06	9.421586e-14	2.255673e-33
Sphere Desplazada	homogénea	1.567619e-04	2.279989e-04	4.034159e-04
	no homogénea	2.254550e-06	3.201177e-06	2.858141e-06

5. Conclusiones

En trabajos anteriores se propuso una técnica de *niching* que descompone del espacio de búsqueda en hipercubos de igual tamaño, denominados region, sobre los que se suele aplicar un mecanismo de división homogéneo para ir aumentando la explotación durante la búsqueda. En este trabajo se propone un nuevo modelo de descomposición heterogéneo, en el que son las regiones más prometedoras las que se dividen en subregiones de menor tamaño. De esta manera se consigue una exploración más eficiente, dividiendo más las zonas más prometedoras del espacio de búsqueda manteniendo al mismo tiempo una adecuada diversidad.

Tras mostrar en detalle el nuevo modelo de descomposición, y aplicarlo como modelo de nicho a un algoritmo evolutivo, se han experimentado sobre él considerando tanto un número fijo de regiones, como los dos modelos de división: el original, y el propuesto. Comparando, se ha observado que el modelo propuesto mejora los resultados, gracias a una mejora más continua de la población.

Los resultados obtenidos nos animan a plantear como trabajo futuro una mayor experimentación. Se propone estudiar cómo integrarlo de forma exitosa aplicarlo a algoritmos meméticos. Una vez combinado, se propone reemplazar el

modelo de regiones original por el nuevo en algoritmos existentes como [5],[6], para estudiar en detalle si el nuevo modelo permite mejores resultados, y en particular en optimización multimodal utilizando un benchmark adecuado para ello [9]. Otra línea a seguir sería el experimentar modificando el valor ND en las nuevas subdivisiones, incluyendo la posibilidad de distintos valores de ND para cada dimensión, para mejorar la aplicabilidad del modelo para problemas con mayor dimensión.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad bajo el proyecto TIN2013-47210-P, y por la consejería de Economía, Innovación y Ciencia bajo el proyecto P12-TIC-2958.

Referencias

1. Caamaño, P., Bellas, F., Becerra, J.A., Duro, R.J.: Evolutionary algorithm characterization in real parameter optimization problems. *Applied Soft Computing* 13(4), 1902 – 1921 (2013)
2. Das, S., Maity, S., Qu, Q., Suganthan, P.: Real-parameter evolutionary multimodal optimization — a survey of the state-of-the-art. *Swarm and Evolutionary Computation* 1(2), 71 – 88 (2011)
3. Ellabaan, M., Ong, Y.: Valley-adaptive clearing scheme for multimodal optimization evolutionary search. In: *Intelligent Systems Design and Applications, 2009. ISDA '09. Ninth International Conference on*. pp. 1–6 (2009)
4. Goldberg, D., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*. pp. 41–49. L. Erlbaum Associates Inc., Hillsdale, NJ, USA (1987)
5. Lacroix, B., Molina, D., Herrera, F.: Region based memetic algorithm for real-parameter optimisation. *Information Sciences* 262, 15 – 31 (2014)
6. Lacroix, B., Molina, D., Herrera, F.: Algoritmo memético basado en regiones con archivo externo para optimización multimodal. In: *Actas del X Congreso Español de Metaheurísticas, Algoritmos Evolutivos y BioInspirados, MAEB'2015*. pp. 35–42 (4-6 Febrero 2015)
7. Petrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. pp. 798–803 (1996)
8. Preuss, M.: *Multimodal Optimization*, pp. 165–191. Springer London, London (2010)
9. Qu, B., Liang, J., Wang, Z., Chen, Q., Suganthan, P.: Novel benchmark functions for continuous multimodal optimization with comparative results. *Swarm and Evolutionary Computation* 26, 23 – 34 (2016)
10. Sun, W., Yuan, Y.X.: *Optimization Theory and Methods: Nonlinear Programming*. Springer US (Dec 2009)