

Búsqueda Dispersa

Rafael Martí^α y Manuel Laguna^β

Resumen— La Búsqueda Dispersa (Scatter Search) es un método evolutivo que ha resultado muy efectivo en la resolución de problemas de optimización. Aunque presenta similitudes con los Algoritmos Genéticos, difiere de éstos en principios fundamentales, tales como el uso de estrategias sistemáticas en lugar de aleatorias. La Búsqueda Dispersa proporciona un marco flexible que permite el desarrollo de diferentes implementaciones con distintos grados de complejidad. En este trabajo se realiza una revisión del método desde sus orígenes, introduciendo los conceptos básicos para después profundizar en los aspectos más novedosos que dan lugar a algoritmos más eficientes.

Palabras clave—MetaHeurísticos, Búsqueda Local, Optimización Combinatoria.

I. INTRODUCCIÓN

La Búsqueda Dispersa (BD) es un método evolutivo que ha sido aplicado en la resolución de un gran número de problemas de optimización. Los conceptos y principios fundamentales del método, fueron propuestos a comienzo de la década de los setenta, basados en las estrategias para combinar reglas de decisión, especialmente en problemas de secuenciación, así como en la combinación de restricciones (como el conocido método de las restricciones subrogadas). La BD se basa en el principio de que la información sobre la calidad o el atractivo de un conjunto de reglas, restricciones o soluciones puede ser utilizado mediante la combinación de éstas. En concreto, dadas dos soluciones, se puede obtener una nueva mediante su combinación de modo que mejore a las que la originaron.

Al igual que los algoritmos genéticos, el método que nos ocupa se basa en mantener un conjunto de soluciones y realizar combinaciones con éstas; pero a diferencia de éstos, no está fundamentado en la aleatorización sobre un conjunto relativamente grande de soluciones sino en las elecciones sistemáticas y estratégicas sobre un conjunto pequeño. Como ilustración basta decir que los algoritmos genéticos suelen considerar una población de 100 soluciones mientras que en la búsqueda dispersa es habitual trabajar con un conjunto de tan sólo 10 soluciones.

La primera descripción del método fue publicada en 1977 por Fred Glover donde establece los principios de la BD. En este primer artículo se determina que la BD

realiza una exploración sistemática sobre una serie de buenas soluciones llamadas conjunto de referencia. Los siguientes comentarios resumen los principales aspectos de este trabajo:

- El método se centra en combinar dos o más soluciones del conjunto de referencia. La combinación de más de dos soluciones tiene como objetivo el generar centroides.
- Generar soluciones en la línea que unen dos dadas se considera una forma reducida del método.
- Al combinar se deben de seleccionar pesos apropiados y no tomar valores al azar.
- Se deben de realizar combinaciones “convexas” y “no convexas” de las soluciones.
- La distribución de los puntos se considera importante y deben de tomarse dispersos.

En Glover [6] se introduce la combinación ponderada (weighted combination) como el mecanismo principal para generar nuevas soluciones. En esta versión se enfatizan las búsquedas lineales entre dos soluciones y el uso de pesos para muestrear en dicha línea. Asimismo, se introduce el concepto de combinar soluciones de calidad con soluciones diversas. Además, el método incluye una componente de intensificación que consiste en tomar una muestra mayor de la línea que ha producido mejores soluciones.

En este artículo el autor especifica que para trabajar con problemas con variables enteras, binarias o que forman una permutación, hay que diseñar métodos específicos de combinación (notar que no tiene sentido hablar de combinación lineal de dos permutaciones). Para ello se introducen los mecanismos de combinación basados en votos. En estos se definen reglas mediante las que cada solución “vota” para que sus características aparezcan en la solución que se está construyendo. Estos métodos de votos han sido muy utilizados en las rutinas de combinación de los algoritmos de BD y parece que constituyen uno de las claves del éxito de estos métodos. A continuación mostramos un ejemplo sobre un problema de rutas de vehículos introducido en Corberán et al. [4] para ilustrarlos.

Dada una serie de localizaciones en donde hay que recoger a unos estudiantes, el problema consiste en encontrar un conjunto de rutas de modo que cada una sea recorrida por un autobús. A los efectos que aquí nos ocupan podemos omitir el resto de detalles y considerar que tenemos dos soluciones del problema para ver cómo se combinan. Consideremos un ejemplo con 10 localizaciones y dos soluciones (A y B) con dos rutas cada una:

^α Departament D'Estadística i Investigació Operativa, Universitat de València, Burjassot 46100, Valencia, Spain. rafael.marti@uv.es. Parcialmente subvencionado por el Ministerio de Ciencia y Tecnología con código TIC2000-1750-C06-01.

^β Graduate School of Business Administration, University of Colorado, Boulder, 80309-0419, USA. manuel.laguna@colorado.edu

Solución A: $A_1 = \{ 4, 2, 7, 1 \}$, $A_2 = \{ 5, 10, 6, 9, 3, 8 \}$
 Solución B: $B_1 = \{ 2, 6, 8, 10, 9 \}$, $B_2 = \{ 3, 4, 7, 1, 5 \}$

Para poder combinar A y B, tendremos que asociar las rutas de cada una. Si consideramos las dos asociaciones posibles (A_1 con B_1 y A_2 con B_2 ó, A_1 con B_2 y A_2 con B_1) y contamos el número de elementos comunes, podemos tomar la que mayor número de coincidencias presente. Así, asociamos A_2 con B_1 y A_1 con B_2 ya que la primera tiene cuatro elementos en común (6, 8, 9 y 10), y la segunda tiene tres (4, 7 y 1), mientras que la otra asignación presenta $1+2=3$ coincidencias. La nueva solución (N_1 y N_2) se construye en n pasos, en los que una localización se asigna a una ruta en cada paso. La siguiente tabla muestra el proceso.

TABLA I
 COMBINACIÓN DE SOLUCIONES.

Paso	Par	Voto 1	Voto 2	Asignación	Regla de selección
1	(A_1, B_2)	4	3	$N_1 = \{ 4 \}$	azar
2	(A_2, B_1)	5	2	$N_2 = \{ 2 \}$	azar
3	(A_1, B_2)	7	3	$N_1 = \{ 4, 3 \}$	3 antes 7
4	(A_2, B_1)	5	6	$N_2 = \{ 2, 5 \}$	5 antes 6
5	(A_1, B_2)	7	7	$N_1 = \{ 4, 3, 7 \}$	igual
6	(A_2, B_1)	10	6	$N_2 = \{ 2, 5, 6 \}$	azar
7	(A_1, B_2)	1	1	$N_1 = \{ 4, 3, 7, 1 \}$	igual
8	(A_2, B_1)	10	8	$N_2 = \{ 2, 5, 6, 10 \}$	10 antes 8
9		9	8	$N_2 = \{ 2, 5, 6, 10, 8 \}$	8 antes 9
10		9	9	$N_2 = \{ 2, 5, 6, 10, 8, 9 \}$	igual

En el paso 1, comenzamos construyendo la ruta N_1 con el par (A_1, B_2). El primer elemento en la ruta A_1 es el 4, así que esta ruta vota para que el 4 sea el primer elemento de la ruta N_1 (Voto 1). Análogamente el voto asociado con B_2 es para la localización 3 (Voto 2). Como ambas localizaciones ocupan en sus rutas la misma posición (la primera) los dos votos valen lo mismo por lo que desempata al azar y gana el 4. En el paso 2 realizamos lo mismo para la otra pareja de rutas. En el paso 3, A_1 vota por la localización 7 (la 4 y la 2 ya han sido asignadas) y B_2 vota por la 3. Como la 7 está en la tercera posición de la ruta A_1 y la localización 3 está en la primera posición de la ruta B_2 , la regla específica dar preferencia a la localización en una posición anterior, por lo que 3 es seleccionada. Procediendo de igual forma se completa la nueva solución hasta obtener $N_1 = \{ 4, 3, 7, 1 \}$ y $N_2 = \{ 2, 5, 6, 10, 8, 9 \}$.

En 1977 Glover [7] publica una versión más específica del método en donde se recogen y simplifican muchas de las ideas expuestas en trabajos anteriores. Esta publicación tuvo un gran impacto en lo que a la difusión del método se refiere y se ha quedado como la referencia *standard* de la búsqueda dispersa. Numerosos investigadores comenzaron a aplicar la BD a la resolución de problemas de optimización obteniendo resultados de gran calidad. La siguiente sección describe esta versión del método, actualizada según implementaciones y desarrollos posteriores.

Glover, Laguna y Martí [8] estudian las implementaciones más recientes del método en la resolución de problemas de optimización combinatoria. Además, muestran las conexiones entre este método y el denominado Re-encadenamiento de Trayectorias (“Path relinking”). Así, desde un punto de vista espacial, el proceso de generar combinaciones lineales de un conjunto de referencia de soluciones, puede ser visto como el generar caminos entre, y más allá, de estas soluciones. Esto lleva a una concepción más amplia del significado de combinar que es la introducida en el re-encadenamiento de trayectorias (RT).

El RT se basa en el hecho de que entre dos soluciones se puede trazar un camino que las una, de modo que las soluciones en dicho camino contengan atributos de ellas. Las soluciones originales pueden haber sido generadas mediante un método basado en una búsqueda local y estar unidas por un camino, o haber sido generadas por otro método y no estar unidas de ningún modo; en cualquier caso, ahora generaremos un nuevo camino que las una. Las características de dicho camino vendrán especificadas respecto de los atributos que son añadidos o eliminados, o por los movimientos realizados para alcanzar una solución desde la otra. Esto constituye una extensión del concepto de combinación en tanto que se obtienen varias soluciones a partir de dos o más originales.

Consideremos en un problema de permutaciones dos soluciones $x=(1,3,4,2)$ e $y=(2,3,1,4)$. Si aplicamos un método de combinación podemos obtener una determinada solución $z = (1,3,2,4)$, mientras que si tratamos de llegar hasta y partiendo de x , podemos obtener la secuencia $z_1=(2,1,3,4)$, $z_2=(2,3,1,4)$. En este sentido decimos que el RT es una extensión de los métodos de combinación.

Una explicación detallada y actualizada de la búsqueda dispersa puede encontrarse en [7]. Dado que este método se basa en realizar combinaciones y aplicar métodos de búsqueda local, se puede considerar incluido en los llamados algoritmos meméticos (una clase de algoritmos evolutivos en donde la búsqueda local y los métodos de combinación de soluciones se aplican de forma selectiva). Actualmente, existen implementaciones comerciales del método, como *OptQuest*¹ [10], que están compitiendo en la industria con otros métodos de optimización en la resolución de problemas reales. A continuación se describen las partes esenciales de un algoritmo de búsqueda dispersa para pasar después a comentar algunas mejoras y aspectos avanzados del método.

II. EL ALGORITMO DE BÚSQUEDA

El método de BD se basa en combinar las soluciones que aparecen en el llamado conjunto de referencia. En este conjunto se tienen las soluciones buenas que se han ido encontrando. Es importante destacar que el

¹ *OptQuest* es un producto registrado de OptTek Systems.

significado de buena no se restringe a la calidad de la solución, sino que también se considera la diversidad que esta aporta al conjunto. La Búsqueda Dispersa consta básicamente de los siguientes elementos:

1. **Un generador de soluciones diversas.** El método se basa en generar un conjunto P de soluciones diversas (alrededor de 100), del que extraeremos un subconjunto pequeño (alrededor de $b=10$) con el que realizar las combinaciones y que denominamos:
2. **Un conjunto de referencia.** Extraído del conjunto de soluciones diversas según el criterio de contener soluciones de calidad y diferentes entre sí (Calidad y Diversidad). Si el método no logra mejorar a la solución, se considera que el output es la propia solución considerada. Las soluciones en este conjunto están ordenadas de mejor a peor respecto de su calidad.
 - 2.1. **Creación.** Iniciamos el conjunto de referencia con las $b/2$ mejores soluciones de P. Las $b/2$ restantes se extraen de P por el criterio de máxima distancia con las ya incluidas en el conjunto de referencia. Para ello debemos de definir previamente una función de distancia en el problema.
 - 2.2. **Actualización.** Las soluciones fruto de las combinaciones pueden entrar en el conjunto de referencia y reemplazar a alguna de las ya incluidas si las mejoran. Así pues, el conjunto de referencia mantiene un tamaño b constante pero va mejorando a lo largo de la búsqueda. En implementaciones sencillas, la actualización de este conjunto se realiza únicamente por calidad, aunque podemos hacerlo también por diversidad.
3. **Un método de combinación.** BD se basa en combinar todas las soluciones del conjunto de referencia. Para ello, se consideran subconjuntos de 2 o mas elementos del conjunto de referencia y se combinan mediante una rutina diseñada a tal efecto. La solución o soluciones que se obtienen de esta combinación pueden ser inmediatamente introducidas en el conjunto de referencia (actualización dinámica) o almacenadas temporalmente en una lista hasta terminar de realizar todas las combinaciones y después ver qué soluciones entran en éste (actualización estática).
4. **Un método de mejora.** Típicamente se trata de un método de búsqueda local para mejorar las soluciones, tanto del conjunto de referencia como las combinadas antes de estudiar su inclusión en el conjunto de referencia.

El siguiente esquema muestra cómo actúan los elementos descritos en un esquema básico del algoritmo.

Algoritmo de Búsqueda Dispersa

1. Comenzar con $P = \emptyset$. Utilizar el **método de generación** para construir una solución y el **método de mejora** para tratar de mejorarla; sea x la solución obtenida. Si $x \notin P$ entonces añadir x a

P . (i.e., $P = P \cup x$), en otro caso, rechazar x . Repetir esta etapa hasta que P tenga un tamaño prefijado.

2. Construir el **conjunto de referencia** $R = \{x^1, \dots, x^b\}$ con las $b/2$ mejores soluciones de P y las $b/2$ soluciones de P más diversas a las ya incluidas.
3. Evaluar las soluciones en R y ordenarlas de mejor a peor respecto a la función objetivo.
4. Hacer $NuevaSolución = TRUE$

Mientras ($NuevaSolución$)

5. $NuevaSolucion = FALSE$
6. Generar los subconjuntos de R en los que haya al menos una nueva solución.
Mientras (*Queden subconjuntos sin examinar*)
7. Seleccionar un subconjunto y etiquetarlo como examinado.
8. Aplicar el **método de combinación** a las soluciones del subconjunto.
9. Aplicar el **método de mejora** a cada solución obtenida por combinación. Sea x la solución mejorada:

Si($f(x) < f(x^b)$) y x no está en R)

10. Hacer $x^b = x$ y reordenar R

11. Hacer $NuevaSolucion = TRUE$

El algoritmo hace referencia a los subconjuntos de R ya que podemos combinar parejas, tríos o cualquier número de soluciones. Es usual el limitar las combinaciones a parejas, por lo que el punto 6 equivaldría a decir: "Generar todas las parejas de soluciones de R en las que al menos una de las dos sea nueva"; donde por nueva entenderemos que halla entrado al conjunto después de realizar la última combinación de todo R .

Notar que el algoritmo se detiene cuando al tratar de combinar vemos que no hay nuevos elementos en el conjunto de referencia (la variable $NuevaSolución$ está en 0). Este algoritmo puede ser anidado en un esquema global que permita reconstruir el conjunto de referencia cuando éste ya ha sido utilizado. Así, si el límite de tiempo (o evaluaciones) no se ha excedido, una estrategia habitual es regenerar el conjunto de referencia dejando la mitad superior ($b/2$ mejores) y eliminando la mitad inferior. Después, se genera un conjunto P como al comienzo del algoritmo, del que se extraen únicamente las $b/2$ soluciones más diversas con las ya existentes en R . De esta forma obtenemos un nuevo conjunto de referencia en el que mantenemos las soluciones de calidad y renovamos las debidas a diversidad. Ahora se vuelve a combinar como anteriormente sobre este conjunto de referencia (pasos 5 a 11). De este modo se obtiene un esquema cíclico indefinido al que hay que añadirle una variable de control para detenerlo, típicamente esta variable está en función del tiempo o del número de iteraciones (evaluaciones de la función objetivo).

III. DESARROLLO Y MEJORAS

La búsqueda dispersa es un método relativamente reciente que hemos de considerar en desarrollo. Durante

los últimos años se han realizado nuevas contribuciones aplicando esta metodología en la resolución de conocidos problemas de optimización. Algunas de estas aplicaciones han abierto nuevos campos de estudio, ofreciendo alternativas a los diseños conocidos. Laguna y Armentano [11] realizan una revisión de los aspectos clave del método desarrollados durante estos años. A continuación las resumimos de forma esquemática:

- a. Considerar el **uso de memoria** basada en la frecuencia para desarrollar métodos de diversificación eficientes. En lugar de generar las soluciones al azar, se propone construir un método que basado en el número de apariciones de los elementos significativos en la solución, evite la repetición de estructuras similares.
- b. Inicializar el conjunto de referencia a partir de un gran conjunto de soluciones creado con el generador antes mencionado.
- c. **Aplicar la rutina de mejora de forma selectiva.** Las pruebas indican que el aplicar el método de mejora a todas las soluciones generadas y combinadas, no garantiza el obtener mejores resultados finales. Establecer umbrales de calidad para no aplicar la mejora a soluciones que difícilmente van a proporcionar la mejor solución, es un gasto innecesario de tiempo de computación (Ugray et al. [12]). Por otro lado, al aplicar el método de mejora a todas las soluciones se acelera la convergencia de éste, lo cual puede ser deseable si disponemos de poco tiempo de computación, pero debemos de evitarlo si queremos ejecutar el método en un horizonte largo para obtener soluciones de gran calidad.
- d. Es necesario estudiar el porcentaje de tiempo que el método está generando soluciones y el tiempo que está combinando. En esencia esta es la cuestión que se plantea en todos los métodos heurísticos: el **equilibrio entre la intensificación y la diversificación.**
- e. Inicializar el conjunto de referencia con la mitad de soluciones por calidad y la otra mitad por diversidad. Se han realizado experimentos con distintos tamaños y parece ser que esta proporción es la que mejores resultados está dando.
- f. La calidad es más importante que la diversidad al actualizar el conjunto de referencia. Notar que aunque el método comienza con un conjunto de referencia con soluciones de calidad y diversidad, al realizar las combinaciones, sólo entran al conjunto las soluciones por el criterio de calidad (hasta llegar a la etapa de regenerarlo). En Laguna y Martí [9] se han probado actualizaciones de este conjunto por diversidad, obteniendo resultados finales peores.
- g. Comparar las **actualizaciones** del conjunto de referencia **estática y dinámica.** Notar que al

combinar las soluciones podemos aplicar dos estrategias, introducirlas en el conjunto, si procede, nada más generarlas, o anotarlas en una “pila” y cuando terminemos de realizar todas las combinaciones, proceder a la actualización. La primera estrategia es dinámica y más agresiva, en tanto que las soluciones buenas entran rápidamente en el conjunto de referencia, pero dado que este es de tamaño constante, esto implica que hay soluciones que salen sin llegar a haber sido utilizadas para realizar combinaciones. Es necesario comparar ambas estrategias para ver cual proporciona mejores resultados finales.

- h. La mayor parte de las soluciones de calidad proceden de combinaciones de dos soluciones. Asimismo, las buenas soluciones suelen proceder de combinar buenas soluciones. Campos et al. ([1] y [3]) realizan numerosos experimentos realizando un seguimiento a lo largo de la búsqueda de las soluciones de calidad en un problema concreto.
- i. El uso de **múltiples métodos de combinación** ha de ser considerado. Campos, Laguna y Martí [2] realizan un análisis de diferentes métodos de combinación, algunos con elementos aleatorios y otros deterministas, de modo que el algoritmo selecciona el método de combinación probabilísticamente, de acuerdo con los éxitos obtenidos por éste. De la misma forma que los métodos de búsqueda local basados en distintos entornos (Variable Neighborhood Search) están proporcionando muy buenos resultados, hemos de considerar el definir varios “entornos” de combinación para realizar una búsqueda más completa del espacio de soluciones.

IV. CONCLUSIONES

La búsqueda dispersa es un método evolutivo que se encuentra en desarrollo. Sus orígenes se pueden situar en la década de los 70 y, aunque menos conocida que los algoritmos genéticos, se está aplicando en la resolución de numerosos problemas difíciles de optimización.

En la actualidad no existe un esquema único para aplicar la búsqueda dispersa. En este trabajo hemos tratado de introducir aquellos aspectos básicos del método que son ampliamente conocidos, así como revisar las últimas implementaciones realizadas y las cuestiones que son actualmente objeto de estudio.

V. REFERENCIAS

- [1] Campos V., Laguna M. y Martí R., *Scatter Search for the Linear Ordering Problem*, New Ideas in Optimisation, D. Corne, M. Dorigo and F. Glover (Eds.), McGraw-Hill. 331-341. 1999
- [2] Campos V., Laguna M. y Martí R. (2001) *Context-Independent Scatter and Tabu Search for*

- Permutation Problems*, Technical report TR03-2001, Departamento de Estadística e I.O., Universidad de Valencia. 2001
- [3] Campos, V., F. Glover, M. Laguna and R. Martí, *An Experimental Evaluation of a Scatter Search for the Linear Ordering Problem*, to appear in Journal of Global Optimization. 1999
- [4] Corberán A., E. Fernández, M. Laguna and R. Martí, *Heuristic Solutions to the Problem of Routing School Buses with Multiple Objectives*, Technical report TR08-2000, Departamento de Estadística e I.O., Universidad de Valencia. 2000.
- [5] Glover, F., *A Template for Scatter Search and Path Relinking*, in Artificial Evolution, Lecture Notes in Computer Science 1363, J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer and D. Snyers (Eds.), Springer-Verlag, pp. 13-54. 1998.
- [6] Glover, F., Tabu search for non-linear and parametric optimisation (with links to genetic algorithms), Discrete Applied Mathematics 49, 231-255. 1994.
- [7] Glover, F., M. Laguna and R. Martí, *Scatter Search*, to appear in Theory and Applications of Evolutionary Computation: Recent Trends, A. Ghosh and S. Tsutsui (Eds.), Springer-Verlag. 1999.
- [8] Glover, F., M. Laguna and R. Martí, *Fundamentals of Scatter Search and Path Relinking*, Control and Cybernetics, 29 (3), 653-684. 2000.
- [9] Laguna M. and Martí R., *Experimental Testing of Advanced Scatter Search Designs for Global Optimization of Multimodal Functions*. Technical report TR11-2000, Departamento de Estadística e I.O., Universidad de Valencia.
- [10] Laguna M. and Martí R., *The OptQuest Callable Library* to appear in Optimization Software Class Libraries, Voss and Woodruff (Eds.), Kluwer., 2000.
- [11] Laguna M. and Armentano V., *Lessons from Applying and Experimenting with Scatter Search*, Technical Report, University of Colorado, 2001.
- [12] Ugray Z., L. Lasdon, J. Plummer, F. Glover, J. Kelly and R. Martí, *A Multistart Scatter Search Heuristic for Smooth NLP and MINLP Problems*, Technical report, University of Texas at Austin. 2001