

Principios de la Búsqueda Dispersa

SILVIA CASADO

Dpto. de Economía Aplicada, Universidad de Burgos, Plaza Infanta Elena s/n 09001 Burgos,
scasado@ubu.es

RAFAEL MARTÍ

Dpto. de Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Valencia, Dr. Moliner 50, 46100 Burjassot (Valencia), Rafael.Marti@uv.es

Resumen

La Búsqueda Dispersa (denominada en inglés Scatter Search y abreviadamente SS) es un procedimiento metaheurístico basado en formulaciones propuestas en los años sesenta sobre estrategias para combinar soluciones o reglas de decisión y que se ha aplicado con éxito a la resolución de una gran variedad de problemas de optimización. Aunque puede ser clasificado como un método de los que denominamos “evolutivos” o “basados en poblaciones” presenta diferencias importantes con respecto a otros métodos tales como el uso de estrategias sistemáticas en lugar de aleatorias. SS opera sobre un pequeño conjunto de soluciones denominado conjunto de referencia. Básicamente se trata de que mediante la combinación de las soluciones que forman el conjunto de referencia se obtengan nuevas soluciones que mejoren a las que las originaron. En este trabajo se exponen los conceptos y principios básicos de SS así como las múltiples alternativas que ofrece para explotar sus ideas fundamentales. Finalmente se muestra una aplicación del método a la resolución de un conocido problema de localización.

Palabras clave: Metaheurísticos, Métodos Evolutivos, Optimización

1. Introducción

La Búsqueda Dispersa es un método que fue introducido en 1977 (publicado en Glover 1998) como un heurístico para la programación entera, basado en estrategias expuestas en el congreso "Management Science and Engineering Management" celebrado en Austin, Texas en septiembre de 1967. A pesar de eso SS no fue aplicado ni debatido hasta 1990, cuando fue presentado en el "EPFL Seminar on Operations Research and Artificial Intelligence Search Methods" (Lausanne, Switzerland). Un artículo basado en

esta exposición fue publicado en 1994 (Glover 1994), y desde entonces la metodología de Scatter Search se empezó a aplicar con más profusión.

La primera descripción de SS (Glover, 1977) usa una sucesión de principios coordinados para generar soluciones. Concretamente los aspectos más destacados de este trabajo son los siguientes:

- ◆ SS realiza una exploración sistemática sobre una serie de buenas soluciones llamadas conjunto de referencia teniendo en cuenta las características de diversos elementos de la solución .
- ◆ 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.

Básicamente se trata de que mediante la combinación de las soluciones que forman el conjunto de referencia se obtengan nuevas soluciones que mejoren a las que las originaron. Según esto, cuando por ejemplo se crean nuevas soluciones a partir de una combinación lineal de otras dos o más, el conjunto de referencia puede evolucionar según se observa en la Figura 1.

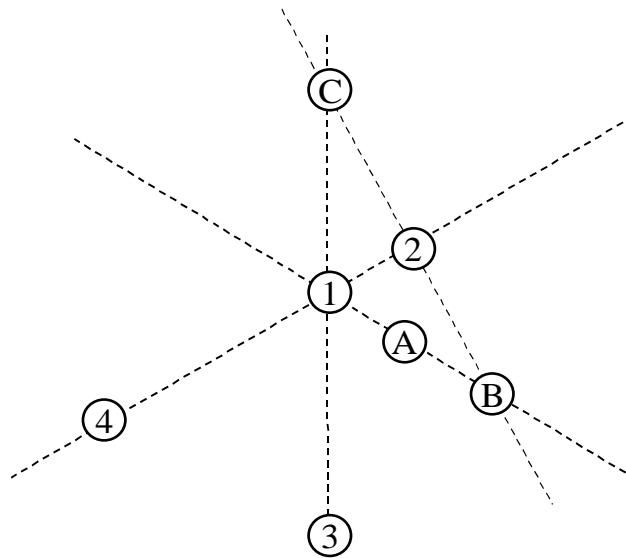


Fig.1.- Conjunto de Referencia

Considerando el conjunto de referencia original como el formado por las soluciones etiquetadas como A, B y C, una combinación no convexa de las soluciones de referencia A y B crea la solución 1. En realidad se crean más soluciones en el segmento definido por A y B, aunque solo se introduce en el conjunto de referencia la solución 1 (el criterio usado para seleccionar las soluciones que forman parte del conjunto de referencia será tratado más adelante). De igual forma, mediante las combinaciones convexas y no convexas de las soluciones del conjunto de referencia y la recién creada se originan los puntos 2, 3 y 4. El conjunto de referencia completo mostrado en la Figura 1 consta de 7 soluciones que posteriormente serán mejoradas con un procedimiento de búsqueda local.

Históricamente los antecedentes de las estrategias para combinar soluciones fueron introducidos en el contexto de los métodos de planificación para obtener mejoras en las soluciones para "*job shop scheduling problems*" (Glover 1963). Se generaron nuevas soluciones mediante la creación de combinaciones numéricamente ponderadas de las soluciones ya existentes. Esta técnica fue motivada por la suposición de que la información está contenida de diferentes formas en diferentes soluciones y por ello esta información puede ser utilizada de un modo más efectivo cuando se combina que cuando la tratamos con estrategias estandar de selección de diferentes soluciones sin tener en cuenta el resto.

Además, en programación entera y no lineal, se desarrollaron procedimientos asociados para combinar restricciones. Hoy en día son muy conocidos los métodos para crear nuevas restricciones de desigualdad llamados *surrogate constraints* (restricciones subrogadas) (Glover 1965 y 1968).

Otros mecanismos de combinación son aquellos basados en "votos". En éstos 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 SS y parece que constituyen una de las claves de su éxito.

Esta forma de operar de SS combinando soluciones hace que se le enmarque dentro de los métodos que llamamos "evolutivos" o "basados en población", es decir, aquellos que combinan soluciones para crear otras nuevas. De hecho se basa en el principio de que la información sobre la calidad o el atractivo de un conjunto de 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.

A pesar de que SS sea un método de los que denominamos "evolutivos" existen algunas diferencias importantes entre los algoritmos genéticos, probablemente el representante más conocido y extendido de los algoritmos evolutivos, y SS. Básicamente son las siguientes:

- Mientras que en SS la selección de las soluciones se hace de forma sistemática y estratégica, en los algoritmos genéticos se realiza de forma totalmente aleatoria.
- SS selecciona las soluciones, para combinarlas posteriormente, de entre un pequeño conjunto de soluciones denominado conjunto de referencia, mientras que los algoritmos genéticos consideran una población de soluciones de mayor tamaño. Así, los algoritmos genéticos suelen considerar una población de 100 soluciones mientras que en SS habitualmente se trabaja con conjuntos de referencia de 10 soluciones.
- Podemos identificar aquí otra diferencia con el resto de métodos evolutivos en los que con frecuencia se emplean métodos de combinación independientes del contexto. Es decir que no utilizan ninguna información o conocimiento sobre el

problema, como el conocido operador de sobrecruzamiento (crossover) en los algoritmos genéticos.

Por otro lado, el método SS no solo consiste en combinar soluciones del conjunto de referencia sino que va más allá y a las soluciones obtenidas tras la combinación se les aplica un procedimiento de mejora que habitualmente es un procedimiento de búsqueda local, aunque en diseños avanzados se puede incorporar al procedimiento de mejora estructuras de memoria. Esta forma de actuar está basada en la suposición de que cuando se combinan soluciones y se aplica un método de mejora sobre las mismas se obtienen mejores resultados que cuando se aplica el método de mejora en las soluciones originales sin haberlas combinado previamente.

En Glover (1998) se recopilan y organizan ideas fundamentales de SS procedentes de trabajos anteriores dando lugar a una versión estandar del método mediante un esquema o plantilla. De ahí la gran importancia de la publicación de este trabajo en lo que se refiere a la difusión del método. Básicamente destacamos las siguientes ideas:

- La información útil sobre la forma o la situación de las soluciones óptimas está normalmente contenida en un conjunto apropiado y diverso de soluciones élite.
- Cuando la combinación de soluciones se usa como una estrategia para explotar tal información, es importante incorporar mecanismos capaces de generar combinaciones que vayan más allá de las regiones abarcadas por las soluciones consideradas. De un modo similar también es importante incorporar procesos heurísticos para transformar las soluciones combinadas en nuevas soluciones. El objetivo de estos mecanismos de combinación es incorporar diversidad y calidad.
- Si se tiene en cuenta múltiples soluciones simultáneamente como base para crear combinaciones, se intensifica la oportunidad de explotar información contenida en la unión de soluciones élite.

2. Método Básico

Scatter Search trabaja sobre un conjunto pequeño de soluciones, denominado conjunto de referencia, combinando sus soluciones para crear otras nuevas. A continuación se

describen los cinco elementos esenciales del método así como el funcionamiento de estos elementos dentro del esquema básico de SS.

1. Método generador de soluciones diversas. Con este método se genera un conjunto de soluciones diversas que en principio no tienen que ser necesariamente factibles. El tamaño de este conjunto, P , suele estar en torno a 100 aunque depende de variantes.

2. Conjunto de referencia (*Refset*). De entre el conjunto de soluciones diversas generado con el método anterior y una vez aplicado el método de mejora, se selecciona el conjunto de referencia, formado por un número pequeño de soluciones, b (alrededor de $b=10$). La mitad de éstas soluciones ($b/2$) serán las de mayor calidad del conjunto de soluciones diversas y la otra mitad se obtiene siguiendo el criterio de la diversidad, es decir se seleccionan aquellas que disten más (según la medida de diversidad considerada en el problema) respecto a las ya incluidas en el conjunto de referencia. Las soluciones seleccionadas se ordenan según su calidad de mayor a menor.

3. Un método generador de subconjuntos. A través de este método se generan subconjuntos de soluciones del conjunto de referencia. Las soluciones de cada uno de estos subconjuntos se combinarán entre sí posteriormente. Un criterio seguido en numerosas ocasiones para obtener los subconjuntos consiste en considerar todos los pares de soluciones del conjunto de referencia, aunque se pueden considerar tríos o subconjuntos formados por cualquier otro número de soluciones.

4. Un método de combinación. Con este método se combinan entre sí las soluciones de cada subconjunto obtenido con el método generador de subconjuntos ya descrito.

5. Método de mejora de soluciones. Este método se usa para tratar de obtener soluciones de mayor calidad que las de partida, aunque en el caso en que aparezcan soluciones no factibles su función consistirá, primero en obtener una solución factible y luego intentar mejorarla. Se aplica tanto al conjunto de soluciones diversas como a aquellas soluciones que se obtienen tras la aplicación del método de combinación a las del conjunto de referencia. Habitualmente como método de mejora se usa un procedimiento de búsqueda local.

El modo de actuación de los elementos descritos anteriormente dentro del esquema básico del algoritmo de SS se muestra a continuación.

Algoritmo de Scatter Search

1- Generación de un conjunto de P soluciones diversas

2-Mejora de dichas soluciones

3-Construcción del *conjunto de referencia* con las b mejores soluciones siguiendo los criterios de calidad y diversidad.

4- Repetir

4.1-Formación de subconjuntos con las soluciones del conjunto de referencia.

4.2-Generación de soluciones nuevas mediante la aplicación del *método de combinación* a las soluciones de los subconjuntos (para obtener soluciones distintas a las de partida)

4.3- Mejora de las nuevas soluciones

4.4-Actualización del conjunto de referencia. (Las nuevas soluciones obtenidas que sean buenas por calidad o por diversidad se incorporan al conjunto de referencia)

Hasta que el conjunto de referencia se estabilice (esto ocurre si durante un ciclo completo no se obtiene ninguna solución que pueda ser incorporada en el mismo)

Tal y como se observa en el esquema del algoritmo de SS, de la combinación de las soluciones del conjunto de referencia se obtienen nuevas soluciones que una vez mejoradas pueden pasar a formar parte del conjunto de referencia, actualizando así dicho conjunto. Dado que el número de soluciones del conjunto de referencia no varía a lo largo de todo procedimiento, la actualización de este conjunto se realizará de forma que las nuevas soluciones sustituirán a las que mejoren en el conjunto de referencia. Es importante destacar que el significado de “*mejores*” no se restringe a la calidad de la solución, sino que también se considera la diversidad que ésta aporta al conjunto. No obstante, aunque la actualización se puede hacer según el criterio de diversidad se ha comprobado que siguiendo el criterio de calidad se obtienen mejores resultados (Laguna y Martí 2003). De esta forma la calidad de las soluciones del conjunto de referencia puede ir mejorando progresivamente.

Finalmente el algoritmo se detiene cuando el conjunto de referencia se estabiliza, es decir, cuando durante un ciclo completo no se obtiene ninguna solución que pueda pasar a formar parte del conjunto de referencia. Llegado a este punto el algoritmo puede reiniciarse volviendo al paso 1 del esquema y repitiendo todo el procedimiento. Una práctica habitual para reinicializar el proceso consiste en obtener un nuevo conjunto de

referencia de la manera siguiente: Partiendo del conjunto de referencia que se había estabilizado (no admitía ninguna nueva solución) se eliminan la mitad de las soluciones ($b/2$), manteniéndose la otra mitad. Concretamente se eliminan las de peor calidad. Para obtener la otra mitad de las soluciones se genera un nuevo conjunto de P soluciones (paso 1 del algoritmo) considerando como objetivo favorecer la diversidad respecto a las soluciones que no se han eliminado. De este conjunto, en sucesivos pasos, se selecciona la solución más diversa respecto a las que ya forman parte del nuevo conjunto de referencia hasta que éste llegue a tener b soluciones. Una vez obtenido el nuevo conjunto de referencia se continua con el paso 4 (4.1, 4.2, 4.3 y 4.4) siguiendo el esquema del algoritmo. En el gráfico de la Figura 2 se muestra el funcionamiento básico del algoritmo de SS.

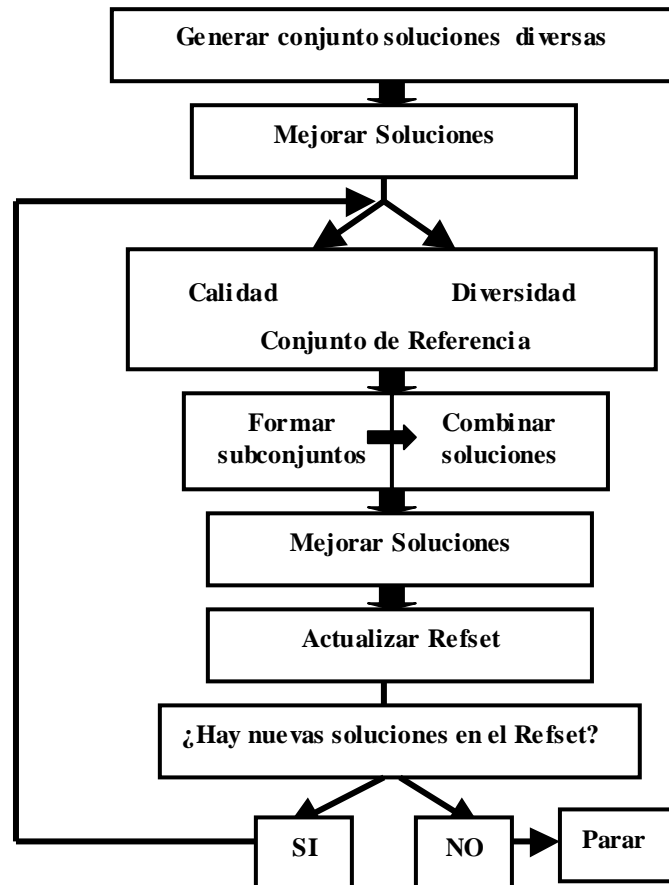


Fig.2.- Funcionamiento del algoritmo de SS

3. Estrategias Avanzadas

Scatter Search puede ser implementado de múltiples formas y ofrece alternativas muy diversas para explotar sus ideas fundamentales. De hecho su mecanismo no está restringido a un diseño único y uniforme, sino que permite diferentes posibilidades que pueden resultar efectivas según el caso.

SS proporciona, por tanto, un marco flexible que permite el desarrollo de diferentes variantes con diversos grados de complejidad entre las que se pueden destacar las siguientes propuestas:

1) Dependiendo del momento en que se realice la actualización del conjunto de referencia podemos distinguir dos variantes:

- a) Estática: La actualización del conjunto de referencia se realiza una vez que se han combinado todos los subconjuntos y se han obtenido todas las nuevas soluciones. De este modo hasta ese momento no se podrá saber si una solución determinada se va a poder incorporar al conjunto de referencia.

Básicamente su funcionamiento se resume en el siguiente esquema:

Algoritmo Scatter Search “estático”

1. *Generar un conjunto inicial P de soluciones diversas*
 2. *Mejorar las soluciones generadas*
 3. *Con estas soluciones construir el conjunto de referencia inicial*
 4. *Repetir*
 - 4.1. *Obtener todos los subconjuntos de elementos del conjunto de referencia*
 - 4.2. *Combinar las soluciones de cada subconjunto para obtener nuevas soluciones*
 - 4.3. *Mejorar las nuevas soluciones obtenidas*
 - 4.4. *Actualizar el conjunto de referencia con estas nuevas soluciones hasta que se estabilice (i.e. no se incluyan nuevas soluciones)*
-

- b) Dinámica: La actualización del conjunto de referencia se realiza cada vez que se genera una nueva solución, de modo que en el momento en que obtiene una solución se sabe si ésta va a formar parte del conjunto de referencia. Esta segunda variante es más agresiva dado que cada vez que se genera una solución apta para entrar en el conjunto de referencia ésta pasa inmediatamente a formar parte de él,

en vez de esperar a que se hayan combinado todos los subconjuntos de soluciones. La ventaja de este tipo de actualización es que en el caso en que el conjunto de referencia contenga soluciones de baja calidad, esas soluciones pueden ser reemplazadas rápidamente y así en las siguientes combinaciones intervendrán soluciones de mayor calidad. Sin embargo, el hecho de que el tamaño de este conjunto se deba mantener constante implica que puedan existir soluciones que salgan del conjunto de referencia sin haber sido combinadas.

Una vez descritas ambas variantes, observamos cómo implementar la variante dinámica resulta ser más complejo que la variante estática. Además, en la variante estática el orden en el cual tiene lugar la formación de subconjuntos y combinación de soluciones no es tan crítico como en la dinámica porque la actualización del conjunto de referencia se realiza una vez combinados todos los subconjuntos. Sin embargo, en la actualización dinámica este orden tiene una gran importancia dado que determina la eliminación de combinaciones potenciales. Por esto, cuando se actualiza según la variante dinámica, es necesario hacer pruebas siguiendo distintos órdenes de formación de subconjuntos y combinación de soluciones.

A continuación se describe el esquema básico de funcionamiento de la variante “dinámica” de SS :

Algoritmo Scatter Search “dinámico”

1. *Generación de un conjunto P de soluciones diversas*
 2. *Mejorar las soluciones generadas*
 3. *Construir el conjunto de referencia inicial Refset*
 4. *Repetir*
 - 4.1. *Siguiendo el orden establecido de formación de subconjuntos, hacer:*
 - 4.1.1 *Obtener el subconjunto de elementos que corresponda*
 - 4.1.2 *Combinar las soluciones de este subconjunto para obtener una nueva solución*
 - 4.1.3 *Mejorar esta nueva solución*
 - 4.1.4 *Actualizar el conjunto de referencia: la solución mejorada se incorpora al Refset si es mejor que alguna de las que ya están en él.*
- hasta que se estabilice el conjunto de referencia (i.e. ninguna nueva solución pueda ser incluida en el Refset)*
-

2) Se pueden implementar diferentes diseños en función del número de elementos que integran los subconjuntos de soluciones que se obtienen del conjunto de referencia. Lo

más común y que permite obtener los mejores resultados es que estén formados por dos elementos (se suelen considerar todos los pares de soluciones posibles). De hecho, en el experimento presentado en Campos y otros (2001) se muestra como al menos el 80% de las soluciones que son admitidas en el conjunto de referencia provienen de combinaciones de subconjuntos formados por dos elementos. No obstante los subconjuntos pueden estar formados por cualquier otro número de elementos o soluciones (por ejemplo subconjuntos de 3 elementos, de cuatro, etc.)

Un procedimiento práctico para generar subconjuntos que permite pasar de subconjuntos de dos elementos a otros formados por un número mayor, teniendo siempre controlado el número de subconjuntos que se van a generar, consiste en crearlos de la siguiente forma propuesta en Glover (1998):

- Subconjuntos de dos elementos: formados por todos los pares de soluciones posibles.
- Subconjuntos de tres elementos: derivados de los subconjuntos de dos elementos añadiendo a cada uno de ellos la mejor solución encontrada que no pertenezca al subconjunto.
- Subconjuntos de cuatro elementos: derivados de los subconjuntos de tres elementos añadiendo a cada uno de ellos la mejor solución encontrada que no pertenezca al subconjunto .
- Subconjuntos formados por los i mejores elementos, desde $i=5$ hasta b .

3) Según la forma de actualizar el conjunto de referencia: Este conjunto inicialmente se forma siguiendo los criterios de calidad y diversidad, de modo que generalmente aproximadamente la mitad de los elementos lo constituyan las soluciones de mayor calidad y el resto se obtengan según el criterio de máxima distancia. Sin embargo a la hora de actualizarlo lo más habitual es hacerlo sólo siguiendo el criterio de calidad. De hecho tal y como se ha comentado anteriormente, se ha probado cómo actualizando el conjunto de referencia solo por diversidad se obtienen peores resultados que considerando sólo el criterio de calidad (Laguna y Martí 2003).

Además, otros aspectos clave del método de SS sobre los cuales se sigue estudiando y que permiten implementarlo según diversas alternativas son los siguientes:

- **Control de la diversidad** cuando se forma el conjunto de referencia: Para garantizar la diversidad cuando se seleccionan las $b/2$ soluciones de mayor calidad de entre las P soluciones generadas por el método generador se puede establecer un umbral de distancia entre estas soluciones de alta calidad, de forma que una solución candidata solo puedan entrar a formar parte del conjunto de referencia si la distancia mínima entre esta solución y las que ya están en el conjunto de referencia sea igual o mayor que ese umbral establecido.

- La incorporación del **uso de memoria** en el algoritmo:

- Para generar soluciones: además de poder generar el conjunto de P soluciones diversas (paso 1 del algoritmo) de forma aleatoria, se puede generar desarrollando algún método que use la memoria basada en la frecuencia con que aparecen los distintos elementos en las soluciones con objeto de evitar la repetición de soluciones similares y favorecer así la diversidad. Es importante destacar que desde sus orígenes, SS, se basa en obtener diversidad de un modo determinista pre-establecido en lugar de recurrir a la aleatoriedad.
- Para su uso en el método de mejora: Cuando se añaden estructuras de memoria al método de mejora, éste deja de ser un simple procedimiento de búsqueda local (procedimiento heurístico) y se transforma en una estrategia metaheurística, es decir, una estrategia maestra que guía y modifica otras heurísticas para producir soluciones más allá de aquellas que normalmente se generan en una búsqueda de óptimos locales (Glover y Laguna 1997). El resultado es un método híbrido que combina dos estrategias metaheurísticas: Scatter Search y el método Tabu Search usado para mejorar las soluciones.

- Búsqueda de un **equilibrio entre diversificación e intensificación**: se trata de estudiar como se va a distribuir el tiempo total de computación, es decir, que porcentaje del tiempo se dedica a generar las soluciones y cuál es el que se dedica a combinarlas.

- Buscar el **tamaño óptimo** del conjunto de referencia: Se debe estudiar si b debe ser un número pequeño tal y como se suele aconsejar o si por el contrario se obtendrían

mejores resultados aumentando este número. Así mismo existe la posibilidad de considerar que b varíe en función del estado de la búsqueda.

- Probar con diversos **métodos de combinación**: En Campos y otros (2005) se analizan distintos métodos de combinación de soluciones, algunos con elementos aleatorios y otros deterministas, de forma que el algoritmo selecciona el método de combinación probabilísticamente, de acuerdo con los éxitos obtenidos por éste.

- Determinar a qué soluciones se debe aplicar el **método de mejora**: La aplicación del método de mejora a todas las soluciones generadas y combinadas no garantiza mejores resultados. En este sentido según Ugray y otros (2001) sería conveniente establecer umbrales de calidad para excluir de la aplicación del método de mejora a aquellas soluciones que difícilmente puedan llegar a ser la mejor solución.

- Formar el **conjunto de referencia inicial** con la mitad de soluciones obtenidas según el criterio de calidad y la otra mitad según el de diversidad. Se han hecho pruebas considerando otras proporciones pero en principio parece ser que ésta es la que permite obtener mejores resultados.

4. Aplicación de SS a un Problema de Localización

Scatter search se ha aplicado a la resolución de una gran variedad de problemas de optimización. En este caso concreto vamos a usar un SS para resolver un problema de localización, concretamente el problema de los p -centros.

El problema de los p -centros es un problema de localización bien conocido que consiste en colocar p servicios como colegios, hospitales o similar y asignar clientes a dichos servicios de forma que se minimice la máxima distancia entre un cliente y su servicio. Se trata de un problema NP-hard tal y como se demostró en Kariv y Hakimi (1979).

Sea $U = \{u_1, u_2, \dots, u_m\}$ un conjunto de usuarios y $V = \{v_1, v_2, \dots, v_n\}$ un conjunto de localizaciones donde colocar servicios. Considérese conocida la distancia d_{ij} entre cada cliente u_i y la localización v_j , el problema consiste en encontrar un subconjunto X de p localizaciones de forma que se minimice

$$\max_{i=1..m} \left\{ \min_{v_j \in X} d_{ij} \right\}$$

El problema se puede formular de forma lineal como sigue

Minimizar z

$$\text{sujeto a: } \sum_{j=1..n} x_{ij} = 1, \quad i = 1..m; \quad (1)$$

$$x_{ij} \leq y_j, \quad i = 1..m; j = 1..n; \quad (2)$$

$$\sum_{j=1..n} y_j = p; \quad (3)$$

$$\sum_{j=1..n} d_{ij} x_{ij} \leq z \quad i = 1..m; \quad (4)$$

$$x_{ij}, y_j \in \{0,1\} \quad i = 1..m; j = 1..n; \quad (5)$$

donde $y_j = 1$ indica que se ha colocado un servicio en v_j (0 en caso contrario); $x_{ij} = 1$ indica que al usuario u_i se le ha asignado el servicio v_j (0 en caso contrario). Este modelo es usado por ejemplo en localización de estaciones de bomberos, policía o ambulancias, unidades de urgencias, etc.

Para resolver este problema de localización se ha diseñado una versión que podríamos denominar ‘estática’ de Scatter Search. La descripción en pseudocódigo de forma general es la siguiente:

Algoritmo Scatter Search “estático”

1. *Generar un conjunto inicial de P soluciones con un método Generador-Diversificador*
 2. *Mejorar estas soluciones con un método de mejora*
 3. *Con estas soluciones construir el conjunto de referencia inicial*
 4. *Repetir*
 - 4.1. *Obtener todos los subconjuntos de pares del conjunto de referencia*
 - 4.2. *Combinar estos subconjuntos para obtener nuevas soluciones*
 - 4.3. *Mejorar estas nuevas soluciones con el método de mejora*
 - 4.4. *Actualizar el conjunto de referencia con estas nuevas soluciones hasta que se estabilice (i.e. no se incluyan nuevas soluciones)*
 5. *Si han transcurrido max_iter iteraciones (pasos 1-4) sin mejora finalizar; sino volver al paso 1 (Reinicializar)*
-

Denotamos por X la solución, total o parcial, en cada momento, es decir, las localizaciones (índices) donde se han colocado servicios, y f el valor de la función objetivo correspondiente a X .

Para formar el conjunto de Referencia, (paso 3) se comienza seleccionando las $b/2$ soluciones de mayor calidad según la función objetivo. Posteriormente para añadir la otra mitad de soluciones se usa la siguiente función o criterio que mide la ‘diversidad’ de una solución candidata X a entrar con respecto a los que ya están en Refset

$$Difmin(X, Refset) = \min \{ dif(X, X') / X' \in Refset \};$$

donde $dif(X, X') = |X - X'|$, es decir el número de elementos (localizaciones) de la solución X que no están en X' .

La actualización de Refset (paso 4.4.) se realiza considerando la calidad de las soluciones. Es decir, se incorporan aquellas nuevas soluciones que mejoren la función objetivo de alguna de las soluciones existentes en Refset. A continuación se describen el método Generador-Diversificador, el método de mejora y el de combinación.

4.1 Método Generador-Diversificador

Nuestro método diversificador esta basado en constructivos tipo GRASP. GRASP (greedy randomized adaptive search procedure), es un método heurístico que construye soluciones usando una función voraz y aleatoriedad controlada. La mayoría de las implementaciones GRASP incluyen un procedimiento de búsqueda local para mejorar las soluciones generadas. GRASP fue originalmente propuesto en el contexto de problemas de cubrimientos de conjuntos (Feo y Resende 1989). Un tutorial clásico se puede encontrar en Feo y Resende (1995) y más recientemente en Pitsoulis y Resende (2002).

En nuestro caso la evaluación proporcionada por la función voraz Δ_j en cada paso es el valor de la función objetivo que se obtendría si se añade un servicio a j . El método diversificador consta de los siguientes pasos:

Procedimiento Avido-Aleatorio

Hacer $X = \emptyset$

Mientras $|X| < p$ hacer

- Determinar $\forall j \in V-X$ el valor Δ_j de *f* si se añadiera j a X
 - Determinar $\Delta_{max} = \max \{ \Delta_j / j \in V-L \}$ y $\Delta_{min} = \min \{ \Delta_j / j \in V-L \}$
 - Construir $L = \{ j \in V-L / \Delta_j \leq \alpha \cdot \Delta_{min} + (1-\alpha) \cdot \Delta_{max} \}$
 - Elegir $j^* \in L$ aleatoriamente
 - Hacer $X = X \cup \{j^*\}$
-

El parámetro α ($0 \leq \alpha \leq 1$) controla el nivel de aleatoriedad. A mayor valor de α menor nivel de aleatoriedad. Con este uso de aleatoriedad controlada se consigue una muestra de soluciones en la que normalmente la mejor de ellas supera a la encontrada con una elección totalmente determinística, (con $\alpha = 1$). Una selección adecuada de α permite un equilibrio entre diversificación y calidad de las soluciones.

La primera vez que se emplea el método generador-diversificador (paso 1) no hay ‘historia’ acerca de cuantas veces un elemento ha formado parte de las soluciones del conjunto de referencia. Sin embargo, esta información puede ser utilizable cuando el método se usa para reinicializar el proceso. La información se registra en el siguiente vector

$freq(j) =$ Número de veces que cada localidad j de V ha pertenecido a las soluciones del conjunto de referencia

La información registrada en $freq(j)$ se usa para modificar los valores Δ_j en el método diversificador de la siguiente manera

$$\Delta'_j = \Delta_j - \beta \Delta_{max} \frac{freq(j)}{freq_{max}}$$

donde $freq_{max} = \max \{ freq(j) : \forall j \}$. Con los valores modificados Δ'_j se calculan Δ'_{min} y Δ'_{max} y se ejecuta el método diversificador con estos valores para construir la lista de candidatos L . Obsérvese que con $\beta = 0$, el método diversificador modificado coincide con el original. Altos valores de β fuerzan a la selección de elementos que menos han aparecido. El uso de información dada por la frecuencia en el método diversificador está inspirada en Campos y otros (2005). Hay que destacar que la incorporación de

memoria a la construcción, hace que ésta ya no sea de tipo GRASP en sentido estricto, pues el muestreo del espacio de soluciones ya no es aleatorio e independiente.

4.2 Método de Mejora

El método de mejora usado tiene su origen en Mladenovic y otros (2001) donde se hacen adaptaciones al problema de los p -centros de 3 heurísticos clásicos para el problema de las p -medianas tomados de Mulvey and Beck (1984). Uno de estos tres heurísticos, el procedimiento “Alternate”, es el que hemos seleccionado como método de mejora y se describe a continuación:

Procedimiento Alternate

Repetir

- *Para cada servicio j de X , determinar el subconjunto de los puntos de U que tienen a j como servicio más cercano*
- *Para cada uno de estos subconjuntos de usuarios resolver el problema del 1-centro*
- *Hacer X' el conjunto de soluciones de estos p problemas, y f' su valor*
- *Si $f' < f$ hacer $X = X'$ y $f = f'$*

hasta que no haya cambios en X

4.3 Método de Combinación

Se obtienen nuevas soluciones combinando pares del conjunto de referencia (paso 4.2). El número de soluciones generadas de cada par depende de la relativa calidad de las soluciones que son combinadas. Considérese x^t y x^q las soluciones del conjunto de referencia que son combinadas, donde $t < q$. Se asume que el conjunto de referencia está ordenado de forma que x^1 es la mejor solución y x^b la peor; entonces el número de soluciones generadas de cada combinación es:

tres si $t \leq Tam_Ref1$ y $q \leq Tam_Ref1$

dos si $t \leq Tam_Ref1$ y $q > Tam_Ref1$

uno si $t > Tam_Ref1$ y $q > Tam_Ref1$.

Cada par de soluciones del conjunto de referencia se usa para generar nuevas soluciones. Para ello se usa la estrategia denominada Path Relinking. La idea básica es construir un camino que una las dos soluciones. Algunas de las soluciones intermedias

en dicho camino se utilizan como puntos iniciales a los que se les aplica la fase de mejora tal y como muestra la Figura 3.

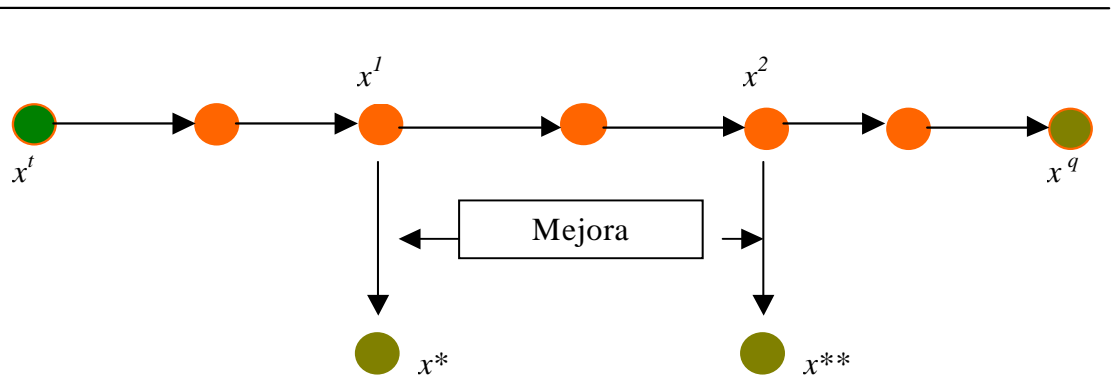


Figura 3.- Generación de nuevas soluciones usando Path Relinking

El camino que une dos soluciones dadas, x^t y x^q , se construye como sigue. Inicialmente se hace $x = x^t$. En los siguientes pasos se añade a x un elemento de x^q que no esté en x y se elimina un elemento que no esté en x^q . De esta forma la solución intermedia x en cada paso tiene un elemento más en común con x^q . En cada paso se elige el mejor entre estos posibles cambios.

Path Relinking es una estrategia tradicionalmente asociada a Tabu Search. La idea que subyace es que en el camino entre dos buenas soluciones, se espera que haya soluciones de parecida calidad (incluso en algún caso mejor). Para una mayor ilustración ver Glover, Laguna y Martí (2000).

4.4 Resultados Computacionales

Inicialmente, para mostrar el funcionamiento de SS se han hecho una serie de pruebas usando los ejemplos de la librería OR-Lib correspondientes a valores de $p \leq 10$. Los valores de los parámetros que ha usado SS en este caso son: $P = 12$, $b = 6$, $\alpha = \beta = 0,8$ y $max_iter = 5$. En estos ejemplos $U=V$, es decir las localizaciones donde colocar las facilidades coinciden con los usuarios. En la Tabla 1 se muestran los resultados.

n	p	Scatter Search	Tiempo Mejor Solución
100	5	121	0,98
100	10	98	1,16

<i>n</i>	<i>p</i>	<i>Scatter Search</i>	<i>Tiempo Mejor Solución</i>
100	10	93	1,25
200	5	82	4,39
200	10	63	4,75
300	5	57	6,88
300	10	49	10,69
400	5	45	10,7
400	10	39	16,64
500	5	40	16,91
500	10	37	27,2
600	5	38	23,42
600	10	32	34,78
700	5	30	31,77
700	10	28	45,52
800	5	29	60,94
800	10	27	76,11
900	5	28	54,86
900	10	24	73,91

Tabla 1.- Resultados para ejemplos de OR-Lib, con $p \leq 10$

Además, se han hecho pruebas con problemas reales. Los datos de los problemas reales se refieren a diversas provincias en el norte de España, concretamente Ávila, León, Salamanca, Segovia y Burgos. Con estas experiencias se quiere analizar para cada provincia dónde situar una serie de unidades de diabetes entre las diferentes localidades que pueden acoger las mismas (por tener algún tipo de instalación que pueda considerarse adecuada). Los valores de p considerados son siempre menores o iguales a 10. Esto se debe a que las autoridades sanitarias correspondientes establecieron que, por diversas restricciones presupuestarias, el número máximo de unidades de diabetes que se podían abrir era diez.

En cada caso se ha considerado la matriz de tiempos (en minutos) entre todas las poblaciones origen, y las poblaciones que, potencialmente, pueden ser destinos. Para hallar estos tiempos de recorrido se ha usado la información sobre carreteras suministrada por el CNIG (Centro Nacional de Información Geográfica), considerando diferentes velocidades según el tipo de tramo (Nacionales, Autonómicas, Provinciales etc...). Con esta información sobre la red de carreteras se ha calculado la matriz de tiempos usando el conocido algoritmo de Dijkstra.

Estos problemas reales se han resuelto, además de con el SS descrito anteriormente, con un algoritmo basado en la estrategia metaheurística Búsqueda en Entornos Variables (VNS) tomado de Mladenovic y otros (2001). El criterio de parada que se ha considerado para ambas estrategias es un tiempo de computación máximo de 400 segundos. En estos ejemplos $U \neq V$ y los valores de m y n para cada provincia se muestran en la Tabla 2 junto con los resultados que se obtienen.

	<i>M</i>	<i>n</i>	<i>P</i>	VNS	<i>Tiempo mejor sol</i>	SS	<i>Tiempo mejor sol.</i>
<i>Ávila</i>	248	156	5	36	4.12	36	0,08
	248	156	10	25	106.4	23	3.72
<i>Burgos</i>	452	152	5	61	0.22	61	0.05
	452	152	10	41	7.20	41	0.05
<i>León</i>	211	184	5	47	13,4	47	2.84
	211	184	10	33	4.64	33	4.44
<i>Salamanca</i>	362	150	5	45	0,8	45	0,36
	362	150	10	31	227.68	31	5.68
<i>Segovia</i>	209	119	5	31	35.4	31	1
	209	119	10	22	87.88	22	2

Tabla 2.- Resultados para ejemplos de diversas provincias del norte de España ($p \leq 10$)

Si observamos los resultados obtenidos por ambos algoritmos vemos como aunque VNS obtiene en prácticamente todos los casos las mismas soluciones que SS, éste último alcanza sus soluciones en un tiempo considerablemente inferior al que emplea VNS.

5. Conclusiones

La Búsqueda Dispersa es una estrategia metaheurística que tiene su origen en los años setenta y se ha aplicado con éxito a la resolución de numerosos problemas de optimización. Aunque se trate de un método evolutivo presenta diferencias respecto a los algoritmos genéticos, probablemente el representante más conocido y extendido de los algoritmos evolutivos. En este trabajo se introducen los aspectos básicos de Búsqueda Dispersa, así como las múltiples alternativas que ofrece para explotar sus ideas fundamentales. Además se muestra una aplicación a la resolución de un conocido problema de localización.

Bibliografía

- Campos V., Glover F., Laguna M. y Martí R. (2001). “ *An experimental Evaluation of Scatter Search for the Linear Ordering Problem*” *Journal of Global Optimization* 21, 397-414.
- Campos V., Laguna M. y Martí R. (2005).”*Context-Independent Scatter and Tabu Search for Permutation Problems*”, *INFORMS Journal on Computing*, 17(1), 111-122.
- Glover F. (1963). “*Parametric Combinations of Local Job Shop Rules*”. Chapter IV. ONR Research Memorandum n°. 117, GSIA, Carnegie Mellon University, Pittsburgh, PA.
- Glover F. (1965). “*A Multiphase Dual Algorithm for the Zero-One Integer Programming Problem*”. *Operations Research*, vol. 13, n° 6, pp. 879-919.
- Glover F. (1968). “*Surrogate Constraints*”. *Operations Research*, vol. 16, pp. 741-749
- Glover F. (1977). “*Heuristics for Integer Programming Using Surrogate Constraints*”. *Decision Sciences*, vol. 8, pp. 156-166.
- Glover F. (1994).” *Genetic Algorithms and Scatter Search: Unsuspected Potentials*”. *Statistics and Computing*, 4, 131-140.
- Glover F. (1998). “*A Template for Scatter Search and Path Relinking*. J.k. Hao, E. Lutton, E. Ronald, M. Schoenauer, D. Syner (eds.). *Artificial Evolution*. Springer LNCS 1363, pp. 13-64.
- Glover F. y Laguna M. (1997). “*Tabu Search*”. Kluwer Academic Publishers, Boston.
- Glover, F., Laguna, M. y Martí, R. (2000) “*Fundamentals of Scatter Search and Path Relinking*”. *Control and Cybernetics*, vol. 39, no. 3, pp. 653-684.
- Kariv O. y Hakami S.L. (1979) “*An algorithmic approach to network location problems. Part I: The p-center Problem*”. *SIAM J. Appl. Math.*, 37, pp. 513-538.
- Laguna M. y Martí R. (2003). “*Scatter Search: Methodology and implementations in C*”. Kluwer Academia Publishers, Bostón.
- Mladenovic N., Labbe M. y Hansen P. (2001). “*Solving the p-center Problem with Tabu Search and Variable Neighborhood Search*”. *Les Cahiers du GERAD*, G-2000-35.
- Mulvey J.M. y Beck M.P. (1984) “*Solving Capacitated Clustering Problems*”. *European Journal Operations Research*, 18 (3), pp. 339- 348.
- Pitsoulis L.S. y Resende M.G.C. (2002). “*Greedy Randomized Adaptive Search Procedures*”. In *Handbook of Applied Optimization*, P. M. Pardalos and M. G. C. Resende (Eds.), Oxford University Press, pp. 168-182.
- Ugray Z., Lasdon, L., Plummer J., Glover F, Kelly J. y Marti R.(2001). “*A Multistar Scatter Search Heuristic for Smooth NLP and MINLP Problems*”, to appear in *INFORMS Journal on Computing*.