

Técnicas metaheurísticas

Álvaro García Sánchez

1. Introducción

Desde los años sesenta han ido apareciendo diferentes métodos de resolución de problemas conocidos por el nombre de técnicas metaheurísticas. Estas técnicas son de especial interés en el caso de variables enteras aunque también se aplican a problemas con variables reales.

Un *heurístico* es un “procedimiento simple, a menudo basado en el sentido común, que se supone que ofrecerá una buena solución (aunque no necesariamente la óptima) a problemas difíciles, de un modo fácil y rápido”. (Zanakis y Evans, 1981)

Los heurísticos se utilizan, por ejemplo, cuando no existe un método exacto de resolución, cuando existe un método exacto que consume mucho tiempo para ofrecer la solución óptima, cuando existen limitaciones de tiempo o como paso intermedio para obtener una solución inicial para la aplicación de otra técnica

Algunos autores (Silver et al., 1980) proponen la siguiente clasificación de métodos de resolución mediante heurísticos:

- Métodos constructivos, que se caracterizan por construir una solución definiendo diferentes partes de ella en sucesivos pasos.
- Métodos de descomposición, dividen el problema en varios más pequeños y la solución se obtiene a partir de la solución de cada uno de estos.
- Métodos de reducción, tratan de identificar alguna característica de la solución que permita simplificar el tratamiento del problema.
- Métodos de manipulación del modelo, obtienen una solución del problema original a partir de otra de otro problema simplificado (con menos restricciones, linealizando el problema, etc.)
- Métodos de búsqueda por entornos, en las que se parte de una solución inicial a la que se realizan modificaciones en sucesivas iteraciones para obtener una solución final. En cada iteración existe un conjunto de soluciones vecinas candidatas a ser nueva solución en el proceso. En este grupo se encuadran las técnicas metaheurísticas.

Las *técnicas metaheurísticas* son procedimientos de búsqueda que tampoco garantizan la obtención del óptimo del problema considerado y que también se basan en la aplicación de reglas relativamente sencillas. A diferencia de los heurísticos, las técnicas metaheurísticas tratan de huir de óptimos locales orientando la búsqueda en cada momento dependiendo de la evolución del proceso de búsqueda.

La aplicación de las técnicas metaheurísticas es especialmente interesante en caso de problemas de optimización combinatoria: problemas en las que las variables de decisión son enteras (o discretas, al menos) en las que, generalmente, el espacio de soluciones está formado por ordenaciones de valores de dichas variables. Sin embargo, las técnicas metaheurísticas se pueden aplicar también a problemas de otro tipo, como con variables continuas, por ejemplo.

La lógica de las técnicas metaheurísticas es similar: el punto de partida es una solución (o conjunto de soluciones) que típicamente no es óptima. A partir de ella se obtienen otras parecidas, de entre las cuales se elige una que satisface algún criterio, a partir de la cual comienza de nuevo el proceso. Este proceso se detiene cuando se cumple alguna condición establecida previamente.

Las técnicas metaheurísticas más extendidas son la siguientes: los algoritmos genéticos, la búsqueda tabú, el recocido simulado, la búsqueda “scatter”, las colonias de hormigas, la técnica conocida por el nombre inglés GRASP. Las redes neuronales, también incluidas entre las técnicas metaheurísticas, son de naturaleza diferente a las tres no se van a considerar.

Todas las técnicas metaheurísticas tienen las siguientes características (Sadiq, S. M. y Habib, Y., 1999):

- Son ciegas, no saben si llegan a la solución óptima. Por lo tanto, se les debe indicar cuándo deben detenerse.
- Son algoritmos aproximativos y, por lo tanto, no garantizan la obtención de la solución óptima.
- Aceptan ocasionalmente malos movimientos (es decir, se trata de procesos de búsqueda en los que cada nueva solución no es necesariamente mejor –en términos de la función objetivo– que la inmediatamente anterior). Algunas veces aceptan, incluso, soluciones no factibles como paso intermedio para acceder a nuevas regiones no exploradas.

- Son relativamente sencillos; todo lo que se necesita es una representación adecuada del espacio de soluciones, una solución inicial (o un conjunto de ellas) y un mecanismo para explorar el campo de soluciones.
- Son generales. Prácticamente se pueden aplicar en la resolución de cualquier problema de optimización de carácter combinatorio. Sin embargo, la definición de la técnica será más o menos eficiente en la medida en que las operaciones tengan relación con el problema considerado.
- La regla de selección depende del instante del proceso y de la historia hasta ese momento. Si en dos iteraciones determinadas, la solución es la misma, la nueva solución de la siguiente iteración no tiene por qué ser necesariamente la misma. En general, no lo será.

Aunque las soluciones que ofrecen las técnicas metaheurísticas no son las óptimas y, en general, ni siquiera es posible conocer la proximidad de las soluciones al óptimo, permiten estudiar problemas de gran complejidad de una manera sencilla y obtener soluciones suficientemente buenas en tiempos razonables.

A pesar de que estas técnicas son relativamente recientes, los campos de aplicación de las técnicas metaheurísticas son numerosos (electrónica, telecomunicaciones, electromagnetismo, etc.), y entre ellos se encuentra el de la Ingeniería de Organización.

A continuación se explican las características más importantes de tres de estas técnicas: la búsqueda tabú, los algoritmos genéticos y el recocido simulado.

2. Búsqueda tabú

2.1. Presentación

Esta técnica fue propuesta por Glover en 1986. Su nombre está relacionado con las acepciones de *tabú* como 'prohibición impuesta por las costumbres sociales como una medida de prevención' y 'prohibición de algo que constituye un riesgo'. Los tabúes sociales suelen pervivir durante un tiempo y luego acaban por extinguirse. Sin embargo, mientras están vigentes, los tabúes no operan como prohibiciones cuando la alternativa prohibida resulta mucho más interesante que el resto.

En el caso de esta técnica, los tabúes se traducen en determinadas prohibiciones en el proceso de búsqueda y el riesgo del que se pretende huir es que la búsqueda quede atrapada en un conjunto de soluciones sin poder salir de él. Para ello, a lo largo de proceso, se almacena información sobre las características de las soluciones visitadas y sobre las operaciones realizadas para obtener dichas soluciones, el proceso tiene memoria de lo ocurrido. Como señalan Glover y Laguna (1997), la memoria, que permite guiar el proceso de búsqueda, es la principal característica de esta técnica; en este sentido, una mala elección puede ofrecer más información que una elección aleatoria y, por tanto, puede ser preferible. Con la memoria es posible orientar la búsqueda hacia soluciones con buenas características a la vez que se exploran nuevas de regiones del espacio de soluciones que son potencialmente interesantes. Existen dos tipos de memoria: la memoria a corto plazo y la memoria a largo plazo. Simplemente con el empleo de la memoria a corto plazo, es posible diseñar un buen proceso de búsqueda, aunque la mejor forma de explotar todas las posibilidades que ofrece esta técnica es el empleo de los dos tipos de memoria.

A veces, como en la analogía del campo social, las prohibiciones dejarán de serlo cuando se cumplan determinadas condiciones y no operarán como tales prohibiciones.

Básicamente, la técnica opera como sigue: dada una solución, se examina un conjunto de soluciones vecinas de aquella y se selecciona la mejor de entre las vecinas, que pasa a ser la nueva solución (aun cuando no sea mejor que la solución

de partida). Sin embargo, no todas las soluciones vecinas son soluciones candidatas a ser nueva solución, algunas están *prohibidas*. Este proceso se repite iterativamente hasta que se cumple alguna condición establecida anteriormente.

Con el procedimiento de la búsqueda tabú, que se detalla en los próximos epígrafes, se pretende evitar la convergencia en torno a óptimos locales. Para ello, se veta el acceso a determinadas soluciones, evitando así (con más o menos éxito, según configuren los parámetros del método) la repetición cíclica de un conjunto de soluciones próximas a un óptimo local y se alienta la exploración de regiones de soluciones potencialmente interesantes. Como indica Glover (1989), con el procedimiento de la búsqueda tabú se trata de evitar la vuelta a *estados de soluciones* anteriores, es decir, a soluciones visitadas anteriormente cuyo mejor movimiento a partir de ella es el mismo que el de la vez anterior en que tal solución fue visitada.

La mayoría de las aplicaciones de la búsqueda tabú se han comenzado a desarrollar a partir de finales de los años ochenta. La búsqueda tabú se ha aplicado, entre otros, a los siguientes problemas: telecomunicaciones, finanzas, ingeniería de organización, medicina, etc.

Se debe apuntar que la búsqueda tabú no es un técnica rígida, sino que se debe confeccionar para cada problema concreto. La forma que adopta el algoritmo depende en gran medida del problema que se pretenda resolver y de la representación que del mismo se adopte. Por lo tanto, este capítulo (de carácter descriptivo) describe los elementos más significativos de la metodología de forma genérica y, en algunos casos, cita ejemplos concretos que sirven para ilustrarla.

2.2. Búsqueda tabú simplificada. Memoria de corto plazo

En este epígrafe se describe la modalidad más sencilla de la búsqueda tabú, que se caracteriza por el empleo, únicamente, de la memoria a corto plazo.

El procedimiento parte de una solución inicial, en la forma de un vector x , a partir de la cual obtiene una nueva solución efectuando cambios elementales, que se denominan *movimientos*, en la solución de partida. A su vez, a partir de esta solución se obtiene, de la misma forma, otra y así sucesivamente.

Dada una solución, el conjunto de soluciones a las que se puede acceder mediante los movimientos considerados se denomina *vecindario*.

De entre todas las soluciones que componen el vecindario, existe un conjunto de ellas que están prohibidas (bien por sus características o bien por el movimiento elemental que condujo a las mismas): son las soluciones tabú y durante un determinado número de iteraciones están vetadas.

La nueva solución, en cada transición, es la mejor solución de entre las vecinas. No es una condición necesaria (a diferencia de otros procedimientos) que la nueva solución sea mejor que la existente.

Esta técnica también tiene en cuenta la posibilidad de que los movimientos prohibidos puedan restringir el área de búsqueda y que ciertas regiones que podrían arrojar buenos resultados queden sin explorar. Para evitar que esto ocurra, existe un *criterio de aspiración*: es posible aceptar una solución clasificada como tabú como nueva solución si cumple con alguna determinada condición.

Por último, el proceso de búsqueda se detiene cuando se cumple algún criterio de finalización.

A continuación se describen los elementos (tipo de solución inicial, tipo de vecindario, etc.) que integran la búsqueda tabú y algunas de las posibles alternativas para cada uno de estos elementos.

Para ilustrar algunos aspectos de la búsqueda tabú, se considerará el problema del viajante con diez ciudades.

Solución inicial

Cada una de las posibles soluciones del problema debe poder ser expresada en forma de un vector o una matriz, x . En el problema del viajante, el elemento $x(i)$ puede representar la posición en la que se visita la ciudad cuyo índice es i o el índice de la ciudad que se visita en lugar i -ésimo (esta última será la que emplearemos).

Para comenzar a aplicar esta técnica es necesario disponer de una solución inicial. Esta solución se puede obtener a partir un heurístico (típicamente de aplicación más sencilla), de forma aleatoria o, sencillamente, a partir de una solución conocida (si el problema se refiere a un sistema ya en funcionamiento, la solución del sistema real es una posible solución de partida).

Función objetivo

Se trata de la función cuyo valor se pretende hacer máximo o mínimo. En cada una de las iteraciones del procedimiento es necesario calcular el valor de la función objetivo de las soluciones candidatas a ser la nueva solución.

Tipos de movimientos

Como se ha indicado anteriormente, la búsqueda tabú explora posibles soluciones a partir de una dada mediante movimientos sencillos. Estos pueden ser, básicamente, de dos tipos:

- Inserción (“insert” en la literatura sajona), en la que un determinado elemento (componente de un vector, columna en una matriz, etc.) de la solución, $x(i)$, pasa a ocupar otra posición, j . El resto de los elementos de la solución quedan desplazados, como se puede apreciar en la figura 1.

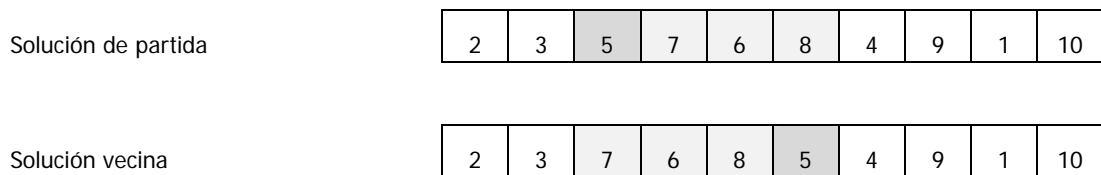


Fig.1. Movimiento de inserción

- Intercambio (“swap”), en el que dos elementos de la solución intercambian su posición (figura 2).

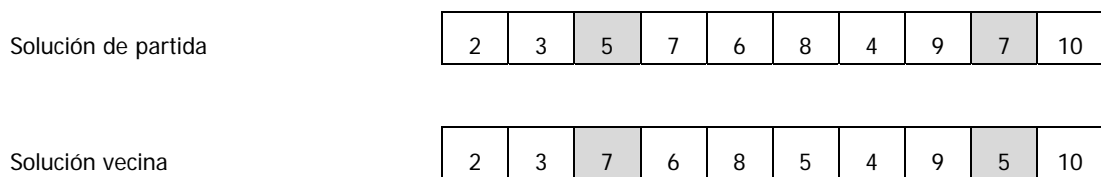


Fig.2. Movimiento de intercambio

En ocasiones, para simplificar, se consideran solamente movimientos de inserción, ya que un intercambio se puede considerar como un caso especial de inserción.

En cada caso, dependiendo de la forma que adopte la solución del problema considerado, se debe definir qué significa un movimiento de intercambio o uno de inserción. Además, es posible restringir el abanico de posibilidades permitiendo

sólo la selección de movimientos que afecten a elementos de la solución que no estén alejados más de una determinada *distancia*. La distancia significará en cada caso una cosa diferente.

Lista tabú

La lista tabú constituye la memoria a corto plazo del proceso de búsqueda. Mediante la lista se alienta la exploración de diferentes regiones de soluciones y se dificulta la convergencia en torno a un óptimo local.

En cada iteración existirá un conjunto de soluciones tabú. Las soluciones son tabú en función de las características de la misma o en función de las soluciones por las que se haya transitado en el pasado reciente. A continuación se explican los dos posibles motivos por los cuales una solución es tabú.

- En primer lugar, una solución puede ser tabú si posee alguna característica que no está permitida. Estas características se llaman atributos. En cada iteración existe un conjunto de atributos que no están permitidas, son los denominados ***atributos tabú***, por ejemplo, el valor de un determinado elemento de la solución.

En la configuración de la búsqueda tabú empleada en cada problema, se debe definir de qué manera se convierten en tabú determinados atributos (características de la solución), de tal manera que al pasar de una solución x_1 a otra x_2 mediante un determinado movimiento, se genere de forma automática el o los atributos tabú asociados a la transición anterior. Asimismo se debe definir la longitud de la lista tabú, es decir, el número de iteraciones durante el cuál dicho atributo va a ser tabú.

Al pasar de una solución a otra mediante un movimiento de inserción, como en la figura, se puede generar el atributo tabú consistente en que la ciudad 5 no puede ser visitada en tercer lugar.

Solución de partida	2	3	5	7	6	8	4	9	1	10
Solución vecina	2	3	7	6	8	5	4	9	1	10

Fig.3. Atributos tabú

- En segundo lugar, una solución puede ser tabú si para llegar a ella se ha realizado un movimiento no permitido. Entonces se habla de *movimiento tabú*. Como en el caso anterior, la definición de la búsqueda tabú debe señalar el tipo de movimientos tabú asociados a cada iteración y el número de iteraciones durante el que dicho movimiento es, efectivamente, tabú.

En el ejemplo anterior, un posible movimiento tabú generado a partir de la transición considerada puede ser el consistente en insertar la ciudad que ocupe el lugar sexto en el tercero.

A continuación se citan algunos ejemplos de atributos tabú:

- que los elementos que ocupan las posiciones i y j en la solución tomen los valores a y b respectivamente;
- que el elemento que ocupe la posición i tome el valor a o que el elemento que ocupe la posición j tome el valor b ;
- que elemento que ocupe la posición i sea mayor que un determinado valor a .

Igualmente, los siguientes son ejemplos de movimientos tabú:

- intercambiar los elementos de las posiciones i y j ;
- mover el elemento que está en la posición i ;
- mover los elementos de las posiciones i y j ;
- mover el elemento i a una posición anterior a la posición a .

En cada aplicación concreta se debe definir si las soluciones se convierten en tabú por haber llegado a ellas mediante movimientos tabú o porque existen atributos tabú. Lo más habitual es trabajar con atributos tabú y no con movimientos tabú, aunque, de hecho, la definición de un atributo tabú está relacionada con los movimientos que hacen que una determinada solución tenga un atributo o no.

Tanto si el algoritmo trabaja con atributos tabú como si lo hace con soluciones tabú, existe una lista denominada *lista tabú*. Esta lista registra la información pertinente para identificar las soluciones tabú y no tabú en cada una de las iteraciones. El número de iteraciones que un movimiento o un atributo es tabú viene dado por la longitud de la lista tabú.

La lista tabú puede ser de dos tipos.

- Lista tabú explícita, en la que se almacena el conjunto de todas las soluciones que están prohibidas porque contienen atributos tabú. Esta alternativa consume una cantidad muy elevada de recursos aunque puede ser interesante cuando se utilizan estrategias que registran y analizan ciertas "soluciones especiales".
- Lista de atributos, en la que sólo se registran los atributos tabú en cada una de las iteraciones. Por ejemplo, una lista tabú de atributos puede ser como la de la figura 4:

Posición en la solución	2	4	1	3	6	8	1	4	2	3
Valor del elemento de la solución	4	5	1	6	2	4	5	7	8	2

Fig.4. Lista tabú de atributos

Se trata de una lista tabú de longitud igual a cinco formada por atributos tabú. Las dos primeras columnas indican, por ejemplo, que una solución que contenga el valor 4 en la segunda posición y el 5 en la cuarta se considerará tabú. Cada par de columnas se interpretan de la misma manera. Cuando se alcanza una nueva solución, aparecen dos nuevas columnas, correspondientes a los atributos de la solución que se abandona que pasan a ser tabú y desaparecen las dos columnas correspondientes a la iteración más antigua.

La lista tabú es uno de los elementos más relevantes de esta técnica metaheurística. Con ella se trata de impedir procesos de búsqueda cíclicos en torno a un óptimo local. Se debe tener presente que una lista tabú muy larga puede impedir que el heurístico explore regiones de soluciones interesantes, mientras que una lista tabú muy pequeña puede conducir a bucles en torno a un óptimo local.

Glover (1989) señala que en las diferentes aplicaciones de la búsqueda tabú se observa que existe un rango de valores amplio para la longitud de la lista tabú que hace que el proceso de búsqueda evite búsquedas cíclicas y, a la vez, ofrezca buenos resultados. Conviene, por ello, al tratar de resolver un problema, realizar un serie de ensayos previos para tratar de buscar cotas a la longitud de la búsqueda tabú. Con estos ensayos se observará que para valores muy pequeños de la longitud, el proceso de búsqueda convergerá muy pronto entorno a óptimos locales; esto se observa cuando aparecen repeticiones periódicas del valor de la función objetivo.

Al contrario, para longitudes muy elevadas se observará un rápido empeoramiento de las soluciones por las que pasa el proceso de búsqueda.

El número de iteraciones durante el cual permanecerá un atributo tabú puede ser diferente. Con ello se pretende alentar la búsqueda en nuevas regiones (cuando el número de iteraciones durante el cual el atributo es tabú es elevado) o para intensificarla en la región en la que está en un determinado instante (cuando el número es pequeño). Esta estrategia constituye una primera aproximación al concepto de estrategia de diversificación y de intensificación, que se comentarán más adelante.

Por ejemplo, si la memoria es explícita, es posible saber si la solución seleccionada en cada iteración ha sido previamente seleccionada, de manera que se puede multiplicar por un determinado factor el tiempo durante el cual el atributo correspondiente será tabú, para hacer menos probable que la búsqueda realice un nuevo ciclo. Para evitar que el número de iteraciones durante el cual un atributo es tabú, L , crezca indefinidamente, es necesario reducir dicho valor con un determinado criterio, por ejemplo, cada vez que transcurra un número determinado de iteraciones (dividiendo L , por ejemplo, por un determinado factor).

Criterio de aspiración

Existen casos en los que una solución tabú perteneciente al vecindario puede dejar de serlo si cumple con un determinado criterio de aspiración. Al permitir la selección de soluciones tabú, la búsqueda gana flexibilidad y permite la exploración de regiones de soluciones que pueden arrojar buenos resultados. A continuación se describen algunos criterios de aspiración:

- Criterio de aspiración por defecto. Se trata del caso más simple, según el cual si todas las soluciones en un determinado instante son tabú se elegirá aquella que ha sido tabú durante un mayor número de operaciones desde que fue catalogada como tal, es la solución “*menos tabú*” porque es aquella a la que le quedan menos iteraciones para dejar de ser tabú.
- Criterio de aspiración determinado por la función objetivo. Una solución dejará de ser tabú si el valor de la función objetivo correspondiente es mejor que el de la mejor solución encontrada hasta el momento (carácter global) o que la mejor solución del espacio de soluciones consideradas en ese instante (carácter local).

- Criterio de aspiración según la dirección de búsqueda. Un movimiento m que introduce un atributo tabú se acepta siempre y cuando si, primero, introduzca una mejora en la función objetivo y, segundo, que en la transición que dio lugar a que dicho movimiento m fuera tabú se realizó un movimiento que mejorara la función objetivo.
- Criterio de aspiración según la influencia del movimiento. Existen movimientos cuya influencia sobre la evolución del algoritmo es mayor que la de otros. Es decir, determinados movimientos conducen a nuevas regiones de soluciones mientras que otros no; los primeros tienen mayor influencia sobre la evolución de la búsqueda que los segundos. En este caso, se debe definir y cuantificar el nivel de influencia de un determinado movimiento. Según este criterio, un movimiento (o un atributo correspondiente a un movimiento) de poca influencia deja de serlo si tras él se ha realizado otro de mayor influencia. Esto es así porque la posibilidad de que tenga lugar un proceso cíclico es mucho menor.

Por ejemplo, en el problema del viajero, si representamos la solución como la ordenación de las ciudades por las que se desplaza, podemos cuantificar la influencia de un movimiento a partir de la distancia entre las ciudades que se ven afectadas por el movimiento.

Lista de candidatas. Exploración del vecindario.

En cada iteración es necesario considerar un conjunto de soluciones de entre las cuales se seleccionará la próxima solución. La exploración de todas las soluciones candidatas puede exigir mucho tiempo. En estos casos conviene explorar el vecindario de una manera más eficiente. A continuación se citan algunas de las posibles formas para generar las candidatas en cada iteración.

- Exploración completa. Una primera forma de estudiar el vecindario consiste en evaluar todas y cada una de las soluciones vecinas que son accesibles mediante los movimientos definidos. En ocasiones esto puede ser muy costoso en términos de tiempo y por eso se emplean otros métodos más eficientes.
- Exploración aleatoria, en la que se acepta como nueva solución la primera solución vecina que se encuentre mejor que la actual. De esta forma es más probable acceder a ciertas regiones con buenas soluciones.

- Exploración por aspiración¹. En este caso se estudia el vecindario solución por solución hasta que se encuentra una solución vecina cuyo valor de la función objetivo supera un determinado umbral. Una vez obtenida esta solución, se estudian otras M soluciones más y se escoge la mejor de entre todas las estudiadas. Generalmente se obliga a este método a estudiar un número mínimo y un número máximo de soluciones.
- Exploración mediante una lista de movimientos de élite. En una determinada iteración se estudia un número elevado de soluciones vecinas. De éstas, se seleccionan las k mejores y se almacenan los movimientos que han conducido a la obtención de dichas soluciones en una lista de movimientos de élite. En la presente iteración se utilizará el primer movimiento de la lista, en la siguiente el segundo y así sucesivamente hasta que finalmente un movimiento cae por debajo de un determinado umbral o transcurre un determinado número de iteraciones. Cuando una de estas condiciones se cumple se construye de nuevo la lista de movimientos de élite y se procede de la misma manera.
- Exploración mediante abanicos de soluciones. Con este método, en una determinada iteración se examina un conjunto de soluciones de las que se escogen las p mejores. A su vez se repite el proceso con estas soluciones obteniendo de nuevo las p mejores. Así un número de veces determinado, al cabo de las cuales se obtiene un conjunto de p caminos a partir de la solución inicial. El camino que se escoge es el mejor de todos ellos. A partir de la solución en la que termina el proceso comienza de nuevo la exploración. Con este método, en ocasiones, los caminos tendrán alguna solución en común, por lo tanto, será necesario generar alguna solución adicional para mantener siempre p caminos.

Criterio de detención

En cada transición se debe estudiar si el proceso de búsqueda se debe detener o no. Existen cuatro criterios que se utilizan típicamente para finalizar el proceso de búsqueda y admitir una solución como buena. Son los siguientes:

1. haber efectuado un determinado número de iteraciones desde el comienzo del proceso de búsqueda;

¹ En la literatura sajona, este método de exploración se conoce como '*aspiration plus*'.

2. haber realizado un determinado número de iteraciones sin mejorar la solución;
3. haber alcanzado un determinado tiempo de computación;
4. haber logrado un valor de la función objetivo suficientemente bueno.

En la figura 5 se muestra un diagrama de flujo en el que se muestra cómo opera la búsqueda tabú con memoria a corto plazo.

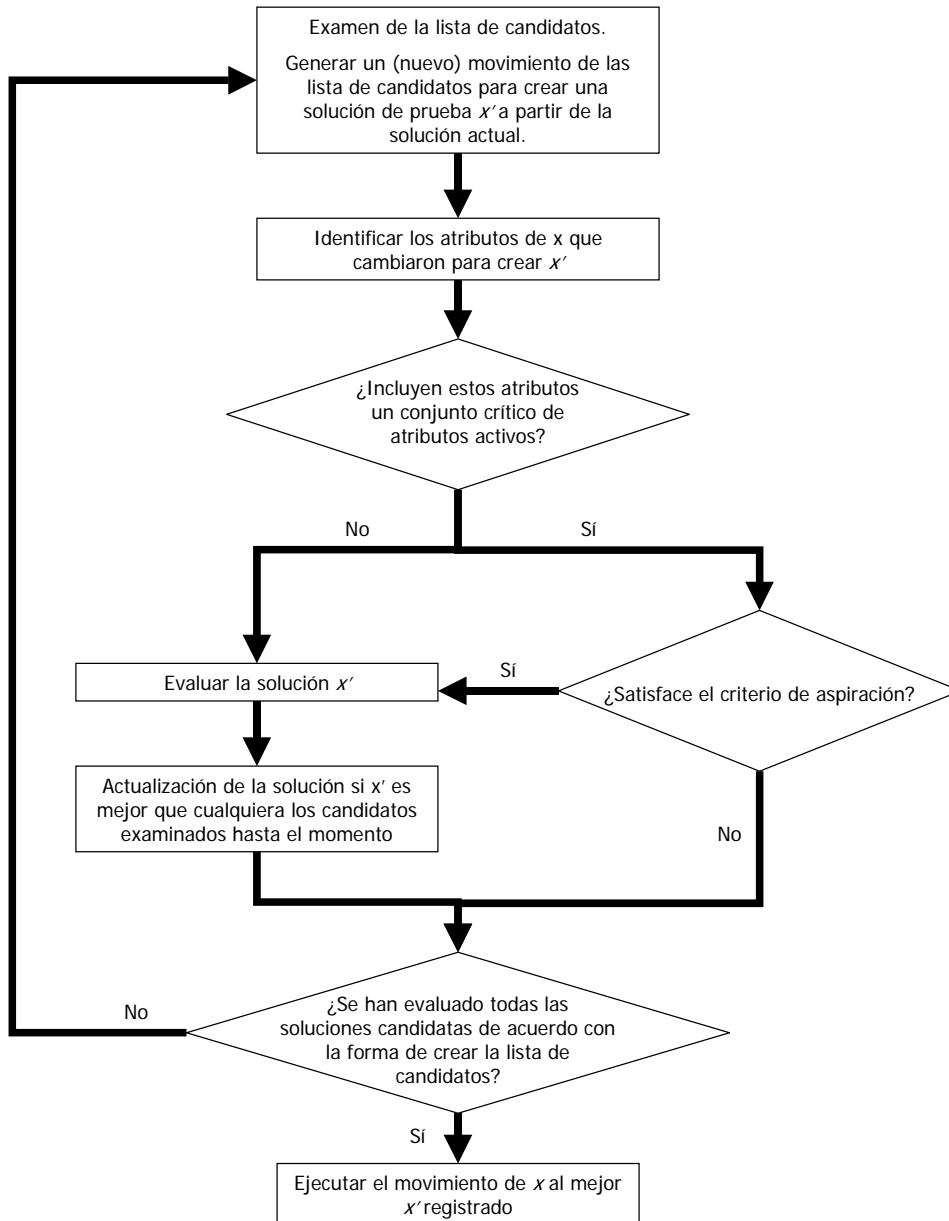


Fig.5. Diagrama de flujo de la memoria a corto plazo

2.3. Búsqueda tabú y memoria de largo plazo

La búsqueda tabú, tal y como se ha formulado en el apartado anterior, incorpora una memoria (en forma de la lista tabú). Esta memoria se refiere sólo al corto plazo y, en ocasiones, puede ser adecuada para la resolución de determinados problemas.

Sin embargo, la búsqueda tabú puede ser más potente si, además de la memoria a corto plazo, se incorporan estructuras de memoria a largo plazo que permitan seguir estrategias más eficientes.

Con el empleo de la memoria a largo plazo se registran soluciones denominadas *de élite* con objeto de intensificar la búsqueda hacia regiones potencialmente atractivas (estrategias de intensificación) o bien para orientar la búsqueda hacia regiones no suficientemente exploradas (estrategias de diversificación).

Memoria de frecuencia

La memoria de frecuencia a largo plazo consta de dos indicadores: el indicador de permanencia y el indicador de transición. Ambos indicadores se expresan en forma de ratios. En el caso del indicador de permanencia, el numerador del ratio es el número de iteraciones durante las cuales la solución seleccionada poseía un determinado atributo. En el indicador de transición, el numerador es el número de veces que un determinado movimiento ha tenido lugar o un determinado atributo sale o entra de la solución.

El denominador, en ambos casos puede ser (1) el número total de iteraciones realizadas en el proceso de búsqueda, (2) el valor del máximo numerador o (3) la suma de todos los numeradores.

En el problema del viajante, un indicador de permanencia puede ser el número de veces que la ciudad C_1 ha ocupado el lugar i -ésimo en la solución. Un posible indicador de transición es el número de veces que intercambian su posición las ciudades C_1 y C_2 .

Estrategias de intensificación

Las estrategias de intensificación modifican las reglas de selección de soluciones vecinas para explorar con mayor profundidad determinadas regiones del espacio de soluciones:

- alentando la aparición de determinadas características de las nuevas soluciones o fomentando determinados movimientos que presumiblemente conduzcan a regiones atractivas o
- volviendo a regiones de soluciones ya visitadas con intención de explorarlas de forma más exhaustiva.

Una forma incorporar las estrategias de intensificación consiste en tener un registro de k soluciones, las k mejores visitadas durante el proceso. Cuando se encuentra una solución mejor que cualquier otra visitada hasta entonces se añade al final de la lista y se toma la primera solución x^1 de la lista como punto de partida del proceso de búsqueda. Además de los atributos activos que estén operando en ese momento, se tienen en cuenta los atributos tabú asociados a la solución x^1 que se generaron cuando esa solución se aceptó como nueva solución en el proceso de búsqueda. De esta manera el proceso de búsqueda vuelve a una solución visitada anteriormente pero, sin embargo, el camino iniciado es otro y los atributos tabú son, en general, diferentes, de modo que cabe esperar que de esta forma se exploren soluciones del entorno de una solución interesante no exploradas con anterioridad.

Otra forma que adoptan las estrategias de intensificación son las denominadas *estrategias de intensificación por descomposición*. Estas estrategias consisten en la construcción de soluciones de partida del proceso de búsqueda mediante la combinación de elementos o atributos interesantes de las soluciones visitadas. Para ello es útil la memoria a largo plazo: los indicadores de permanencia indican si un atributo es atractivo o no, dependiendo de que cuando aparezca, las soluciones que lo contienen sean de calidad o no respectivamente. Si un atributo aparece ligado sistemáticamente con soluciones de buena calidad conviene intensificar la búsqueda en regiones de soluciones que ofrezcan dicho atributo y, al contrario, si las soluciones son de mala calidad conviene abandonar la búsqueda en tales regiones.

Un ejemplo de esta segunda opción para el problema del viajante es la siguiente consiste en identificar las ciudades que están en los extremos del recorrido de muchas soluciones de elevada calidad y dejarlos 'atrapados' en el interior del recorrido para modificar otras partes de la solución.

Estrategias de diversificación

Con las estrategias de diversificación se persigue la exploración de nuevas regiones. Para ello, en primer lugar, se puede modificar el valor de la función objetivo de manera que aumente o disminuya según los atributos que contenga dicha solución hayan aparecido en pocas o en numerosas ocasiones a lo largo de la exploración. En un problema de maximización, se define el valor de un movimiento como el incremento de la función objetivo. Cuanto mayor es el valor de un movimiento, más atractiva es la solución a la que conduce ese movimiento. El valor del movimiento puede modificarse de la siguiente manera:

$$\text{valor del movimiento}' = \text{valor del movimiento} - d \cdot \text{penalización}$$

donde la *penalización* suele depender de la frecuencia del atributo (medida según los ratios descritos antes) y d es un parámetro de diversificación. Las penalizaciones negativas o *incentivos* tienen el objetivo de alentar la aparición determinados atributos y las penalizaciones positivas, inhibir la aparición de dichos atributos. Por su lado, los valores elevados de d indican una mayor tendencia a la diversificación (a explorar nuevas regiones).

Para diversificar el proceso de búsqueda también se puede, periódicamente, reiniciar el proceso con una nueva solución de partida construida a partir de la información almacenada de la memoria a largo plazo, fomentando la aparición de atributos que han aparecido con poca frecuencia e inhibiendo la de aquellos que han aparecido en más ocasiones.

Por otro lado, si la permanencia de un determinado atributo conduce a soluciones tanto de buena como de mala calidad, el atributo puede estar *anclando* la búsqueda a una región sin resultados interesantes y conviene deshacerse de él. Al contrario, si la aparición poco frecuente de un atributo conduce al mismo resultado de antes quizá la búsqueda esté siendo restringida al no incluir regiones con dicho atributo y convenga introducirlo. En ambos casos se trata de diversificar la búsqueda.

Existen diferentes formas de realizar diferentes etapas, unas en las que se intensifica el proceso de búsqueda y otras en las que se diversifica. Sin embargo, no se conoce suficientemente bien cómo combinar intensificación y diversificación de la manera más eficiente.

Estructura del proceso

Las estrategias de diversificación y de intensificación se introducen en el proceso de búsqueda de forma conjunta con los pasos descritos anteriormente para la búsqueda tabú con memoria de corto plazo.

Al final de cada iteración se comprueba si se satisface alguna determinada condición que active una fase de diversificación o de intensificación. Por ejemplo, si transcurrido un determinado número de iteraciones la solución no ha mejorado, se inicia una fase de diversificación: se modifican el valor de los movimientos según se ha descrito anteriormente durante un número de iteraciones o hasta que no se cumpla alguna condición, o se reinicializa el proceso generando una nueva solución, etc.

Igualmente, en determinados instantes y cuando se cumpla una determinada condición, se inicia una etapa de intensificación utilizando, por ejemplo, alguna de las medidas descritas más arriba.

2.4. Oscilación estratégica

Objetivo: permitir la búsqueda en determinadas regiones no exploradas mediante la admisión como soluciones actuales soluciones no factibles. Las soluciones no factibles tienen una penalización. La penalización se modifica según el número de veces que el algoritmo haya transitado por soluciones no factibles. Ejemplo en la pág. 159 de *Essays and surveys in metaheuristics*

2.5. Conclusiones

La búsqueda tabú ofrece un conjunto muy amplio de posibilidades para la resolución de problemas combinatorios. El método se basa en la acumulación de información sobre el proceso de búsqueda.

De forma sucinta comentaremos que, con respecto a la convergencia de la búsqueda tabú, el proceso de búsqueda ordinario, de carácter determinista, no converge a un óptimo local. Sin embargo, se puede demostrar que la búsqueda converge asintóticamente al introducir elementos estocásticos (definiendo funciones de probabilidad que gobiernen el proceso de selección de candidatos, el grado de reversibilidad de determinados movimientos así como la aplicación del criterio de aspiración).

Es posible diseñar procesos de búsqueda sencillos que ofrecen ventajas frente a otros métodos. Se pueden diseñar búsquedas más elaboradas cuyo desarrollo exige más tiempo pero son más eficientes.

En este capítulo introductorio se han comentado los aspectos más relevantes, pero existen muchas otras consideraciones que dan lugar a diferentes alternativas que, en cada caso, hacen que la búsqueda tabú adopte formas particulares.

3. Algoritmos genéticos

3.1. Presentación

La técnica metaheurística de los algoritmos genéticos fue ideada por Holland en 1975 y está inspirada en los procesos de adaptación de los seres vivos. Los resultados de los patrones de evolución de los seres vivos han sido *sobradamente probados* con éxito (a lo largo de la evolución de las especies) y constituyen la base de los algoritmos genéticos. Por un lado, estos patrones permiten que con el transcurso del tiempo se exploren continuamente nuevas posibilidades y, por otro, y en condiciones normales, raramente conducen a la obtención de individuos absolutamente desadaptados e incapaces de sobrevivir.

Partiendo de una población inicial, es decir, un conjunto inicial de soluciones, se realizan manipulaciones por las que se obtienen sucesivas poblaciones. La función de adaptación indica la bondad de las soluciones consideradas en cada momento.

En cada iteración se realizan una serie de operaciones con los individuos de la población, de entre las cuales las más comunes son: la selección, el cruce, la mutación y la inversión. La aplicación de los operadores anteriores permite obtener, típicamente, soluciones con mejores funciones de adaptación.

Los algoritmos genéticos pertenecen al grupo de las técnicas evolucionarias, que son aquellas técnicas que en cada iteración disponen de un conjunto de soluciones a partir de las cuales obtienen un nuevo conjunto de soluciones.

Nomenclatura

Como se ha dicho, los algoritmos genéticos están basados en la observación de la evolución natural de las especies. Existen, por lo tanto, analogías entre la nomenclatura propia de la Biología y la que se emplea en la técnica metaheurística de los algoritmos genéticos.

Los cromosomas de los seres vivos contienen la información de los mismos. Estos cromosomas están, a su vez, formados por genes, y cada uno de los genes es responsable de un rasgo del individuo. Los genes están formados por secuencias de cuatro tipos de aminoácidos.

La información contenida en los cromosomas se llama genotipo. La descodificación (realizada por diferentes enzimas) del genotipo de cada individuo da lugar a un conjunto de características (llamado fenotipo), lo que le confiere al individuo unas determinadas condiciones de adaptación en función del entorno en el que se encuentre.

En los algoritmos genéticos, el equivalente del genotipo es una cadena de caracteres. El equivalente del fenotipo es la solución que resulta de la descodificación de la cadena anterior y, finalmente, la función objetivo hace las veces de entorno y permite evaluar la bondad de las soluciones.

El significado del término ‘individuo’ es distinto en el contexto de los algoritmos genéticos y en el ámbito biológico. En términos biológicos, un individuo es un animal o vegetal de determinada especie. Sin embargo, y haciendo un abuso del lenguaje, al hablar de un individuo nos referimos a una cadena de caracteres que representan una solución o un conjunto de ellas. El conjunto de individuos constituye una población. La figura 6 ilustra la relación entre los algoritmos genéticos y la biología:

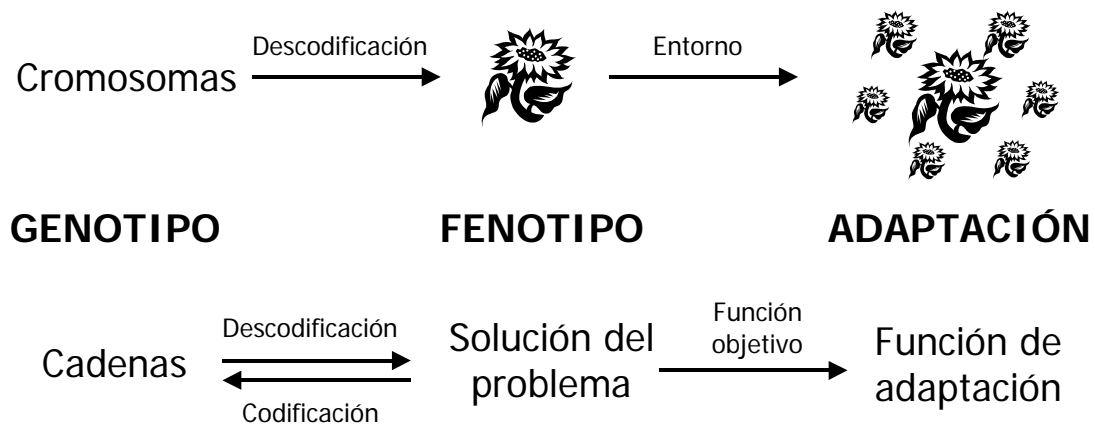


Fig.6. Analogías entre la técnica metaheurística y la Biología

3.2. Método

El método que siguen los algoritmos genéticos no es único y rígido. Existen diferentes métodos propuestos por diversos autores, con características similares pero con diferencias significativas. Todos ellos trabajan con un conjunto de individuos que son sometidos a diferentes operadores genéticos con objeto de obtener nuevos y mejores individuos. Las variantes de los operadores y la forma

en la que se aplican sobre la población determinan las diferencias más notables entre unos algoritmos y otros.

A continuación se describen someramente algunos de los aspectos del método que siguen los algoritmos genéticos. Más adelante se profundiza en los aspectos más relevantes de la técnica.

El espacio de soluciones del problema tratado queda representado por un conjunto finito de cadenas o matrices. Cada una de estas, a las que se les denomina individuos, representa al menos una solución del problema. Debe existir, por lo tanto, un código que establezca una relación entre las diferentes soluciones y las representaciones de las mismas.

En cada instante existe un conjunto de individuos llamado población. A lo largo de proceso de búsqueda (y a diferencia de otras técnicas metaheurísticas) los algoritmos genéticos exploran un conjunto de soluciones y no una sola. De esta forma, se pretende que el proceso de búsqueda no quede atrapado en un óptimo local.

De cada uno de los individuos es posible obtener su función de adaptación a través de la evaluación de la solución a la que representa. Cuanto mayor es dicha función mejor es la solución (su equivalente en el ámbito biológico es la capacidad del individuo para sobrevivir y generar descendencia).

Los individuos mejor adaptados (con mayores valores de la función de adaptación) son seleccionados, con mayor probabilidad cuanto mayor sea la función de adaptación (*selección*). Estos individuos –los padres– se someten a una operación de *cruce*, por el que se generan nuevos individuos –que conforman la descendencia– resultado de la unión de diferentes elementos de la cadena de los progenitores.

Algunos (o todos) los individuos que formaban la población inicial son reemplazados por parte (o todo) el conjunto de los nuevos individuos de la descendencia (*reducción*). En cualquier caso, al comienzo de cada iteración se cuenta siempre con una población de un determinado número de individuos, que permanece constante durante el proceso de búsqueda.

Por otra parte, con una determinada probabilidad las cadenas de los individuos se someten a algún tipo de modificación (*mutación*).

El proceso se detiene cuando se cumple alguna condición relativa al tiempo de proceso o al número de iteraciones o a la bondad de las soluciones obtenidas hasta el momento.

Codificación de la solución

El algoritmo trabaja con la representación de las soluciones y realiza modificaciones sobre las mismas sin conocer cuáles son las soluciones que representan los individuos. Por lo tanto, el proceso desconoce las soluciones con las que trabaja, por esto es importante que la codificación permita una representación adecuada de las soluciones y que las modificaciones que el algoritmo realice sobre dichas representaciones tengan significado respecto del problema que se aborde. De no ser así, lo que se puede obtener es cualquier tipo de individuos sin que mejore la función de adaptación.

Atendiendo al tipo de relación que se establece entre las representaciones y las soluciones podemos distinguir tres tipos de codificación:

- *Uno-a-muchos*: existen individuos que aun siendo diferentes representan la misma solución. En este caso el algoritmo trabaja con las representaciones de las soluciones como si efectivamente fueran diferentes (aunque no lo son –de hecho, el valor de la función de adaptación es el mismo porque la solución que representan es la misma–). Este tipo de codificaciones provoca que la búsqueda sea menos eficiente porque el algoritmo se detiene en la exploración y comparación de individuos que aparentemente son diferentes pero que corresponden a la misma solución.
- *Muchas a una*: muchas soluciones quedan representadas por una sola representación. Esta representación no es adecuada porque adolece de cierta falta de detalle (si un individuo representa a diferentes soluciones puede ocurrir que ese individuo represente a soluciones con características muy diferentes). Además, en cada paso se debe seleccionar cual de las soluciones a las que representa cada individuo se evalúa; típicamente la mejor, por lo que existe un proceso de optimización local del cual las técnicas metaheurísticas tratan de huir en general. Por otro lado, puede ser interesante si permite explorar las soluciones atendiendo a algún rasgo en particular, cuya importancia tenga más peso que cualquier otro. En este sentido, el proceso de búsqueda puede ser más eficiente.

- *Una-a-una*: cada solución queda representada por uno y solo un individuo. Es el mejor tipo de representación, ya que el espacio de búsqueda es el más pequeño posible de entre los que se pueden obtener sin perder nada de información relativa a las soluciones.

Por otro lado, atendiendo a la naturaleza de la representación podemos distinguir dos grandes grupos:

- Codificaciones binarias. Las representaciones son cadenas de ceros y unos. Este tipo de codificación es el que más se ha empleado en la literatura, tratando de obtener ventajas derivadas del empleo de los diferentes operadores basado en el teorema de los esquemas (que se comentará al final).
- Codificaciones no binarias: aunque menos empleadas, pueden ofrecer ciertas ventajas. Exigen un tratamiento diferente al de las codificaciones binarias.

(comentar las ventajas que dicen algunos autores)

Selección de la población inicial

Como en el resto de técnicas metaheurísticas, la solución de partida (la población en este caso) se puede obtener de forma aleatoria o partir de la aplicación de alguna regla sencilla o, simplemente, a partir de soluciones conocidas obtenidas por experiencias anteriores.

La generación de una población de forma aleatoria es ventajosa en cuanto que permite que el proceso de búsqueda comience con un conjunto de soluciones suficientemente diferentes. Por otro lado, al generar una población aleatoria, si el resultado del proceso de búsqueda es satisfactorio, esto será por las bondades del propio proceso y no por el método de generación de la primera población.

Selección

Con este operador se obtiene el conjunto de individuos de la población actual que va a generar la descendencia de una población dada, que formarán parte de individuos de la siguiente población.

En términos biológicos, los individuos más adaptados tienen más probabilidad de sobrevivir y de procrear en mayor medida. La técnica metaheurística, igualmente, selecciona aquellas soluciones cuya función de adaptación sea mayor. Sin embargo, si sólo se eligieran aquellos individuos cuya función de adaptación fuera mayor existiría el peligro de converger en torno a un óptimo local de forma prematura.

La probabilidad de que este operador seleccione a cada individuo es mayor cuanto mayor sea su función de adaptación. Algunas de las formas de establecer esta relación son las siguientes.

- *Selección proporcional* (es el criterio de selección de la propuesta original de Holland). La probabilidad de selección de cada individuo es proporcional a su función de adaptación. Conviene notar que se trata de una probabilidad y no una proporción determinista: los individuos mejor adaptados tienen más probabilidad de ser seleccionados y, por lo tanto, por término medio serán elegidos en mayor proporción, pero no existen garantías de que en una iteración particular los individuos sean seleccionados en proporciones idénticas a sus probabilidades de selección.

$$P_i = \frac{F_i}{\sum_j F_j}$$

donde:

P_i : es la probabilidad de seleccionar el individuo i

F_j : es la función de adaptación del individuo j

- *Selección determinista*. En este caso sí que se trata de proporciones deterministas. Los individuos son seleccionados en una proporción directamente proporcional a su función de adaptación. Con este tipo de selección se acelera la convergencia del algoritmo: si existen soluciones mejor adaptadas que otras pero próximas a un óptimo local, la búsqueda se centrará en la región de dicho óptimo local, a pesar de que puedan existir otros individuos peor adaptados alejados de dicho óptimo y más próximos a otro óptimo local (mejor que el anterior) o incluso más próximo al óptimo global.
- *Normalización*. La selección proporcional plantea algunos problemas. Por ejemplo, no siempre tiene sentido que si a un individuo le corresponde una función de adaptación cuyo valor es el doble que la de otro (está “*doblemente mejor adaptados*”), aquel tenga el doble de probabilidades de ser seleccionado que este. Además, si las diferencias entre los valores de las funciones de adaptación de los individuos son muy elevadas se puede provocar una convergencia prematura del proceso en un entorno de la región donde se ubican los individuos con mayores valores de la función de adaptación. Al contrario, en ocasiones, a pesar de que las diferencias entre diferentes valores

de la función de adaptación no sean elevadas en comparación con los valores absolutos de la misma, puede ocurrir que las soluciones sean bien diferentes en términos del objetivo que se persigue.

Para solventar estos y otros problemas se propone una corrección (normalización) de los valores de la función de adaptación de la siguiente manera:

$$F'_i = \frac{F'_{max} - F'_{min}}{F_{max} - F_{min}} \cdot (F_i - F_{min})$$

donde:

F_{max} y F_{min} representan, respectivamente, el mayor y el menor valor de entre todos los valores de la función de adaptación de los individuos de la población.

F'_{max} y F'_{min} son, respectivamente, el mayor y el menor de valor que toma la función de adaptación normalizada, a partir de la cual se procede como en el caso de la selección proporcional.

La dificultad de esta alternativa radica en que no es sencillo encontrar los valores de F_{max} y F_{min} adecuados.

- *Clasificación.* Según este criterio, se asignan probabilidades de selección directamente proporcionales al lugar que ocupa cada individuo en la clasificación que se establece al ordenar los individuos por valor creciente de la función de adaptación. Es decir:

$$F'_i = \frac{R_i}{\sum_j R_j} = \frac{2 \cdot R_i}{N(N+1)}$$

donde:

R_i es la posición que ocupa el individuo i al ordenar la población por valor creciente de la función de adaptación.

N es el número total de individuos de la población.

Lo que se pretende con este método es seleccionar a los individuos independientemente de la función de adaptación y atendiendo sólo a sus posiciones relativas.

- *Selección por torneo.* A pesar de que los dos métodos anteriores tratan de evitar la relación proporcional entre la función de adaptación y la probabilidad de

selección, la idea de proporción no desaparece por completo al efectuarse una selección proporcional, ya sea con las posiciones que ocupan los individuos o las funciones de adaptación normalizadas.

Este último procedimiento abandona por completo el criterio de selección proporcional y está basado en comparaciones (torneos) entre individuos. En el proceso se van comparando parejas y se va estableciendo una clasificación de los individuos. En primer lugar, se escogen dos individuos de la población al azar. Aquel cuya función de adaptación sea menor pasa a formar parte de la lista de clasificación, y como se trata del primer elemento elegido ocupará el último lugar. Este elemento deja de ser un candidato para participar en nuevas comparaciones porque ya ha sido clasificado. El otro elemento de la comparación (mejor adaptado) sí vuelve a ser candidato para participar en nuevas comparaciones. Repitiendo este proceso se obtiene finalmente una lista con la clasificación definitiva de los individuos de la población.

Esta clasificación, tal y como se obtiene, es la que se emplea en el proceso de selección. Ya no es necesario realizar ningún tipo de asignación de probabilidad proporcional porque, de hecho, el elemento aleatorio se introduce mediante la elección de los individuos que participan en cada comparación.

Cruce

Este operador, conocido también como el de recombinación, es el más importante de esta técnica metaheurística. El objetivo del cruce es combinar elementos de información de diferentes individuos, de modo que las características interesantes que estaban dispersas en diferentes individuos queden reunidas en uno nuevo, confiando en que los individuos obtenidos de esta manera representen soluciones de mejor calidad.

La forma en que se realiza el cruce depende del tipo de representación que se escoja. Para ilustrar algunas posibilidades supondremos que la representación de las soluciones de un determinado problema es de tipo binario y es una cadena de ocho elementos. Los individuos A y B han sido seleccionados para someterlos al operador del cruce:

Individuo A: 1 1 1 0 1 0 0 1

Individuo B: 0 1 0 0 1 1 0 0

Podemos describir los siguientes tipos de cruce (que no son los únicos):

- *Operador de cruce simple.* Según este tipo se cruce, se selecciona un punto de la cadena en cada uno de los padres y se generan nuevos individuos combinando las partes que generan los cortes anteriores, como aparece en la figura 7.

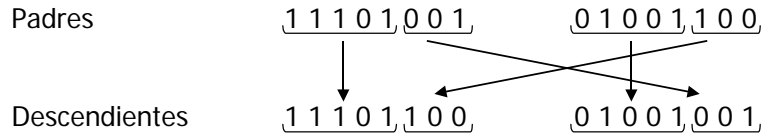


Fig.7. Operador de cruce simple

- *Operador con dos puntos de corte.* Es análogo al anterior, salvo que se seleccionan dos puntos de corte y los padres intercambian los elementos de la cadena que quedan entre dichos puntos para generar los descendientes.

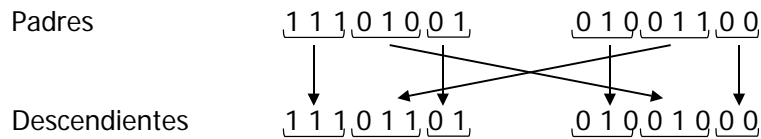


Fig.8. Operador de cruce con dos puntos de corte

- *Operador de cruce conforme a una máscara de cruce.* En este caso, los genes de los descendientes se obtienen de acuerdo con el criterio dado por una *máscara*. La máscara, en el ejemplo, es una cadena de ceros y unos. Para cada posición de la descendencia se tomará el gen del padre 1 si el valor de la máscara para dicha posición es 1 y del padre 2 si el valor del gen es 0.

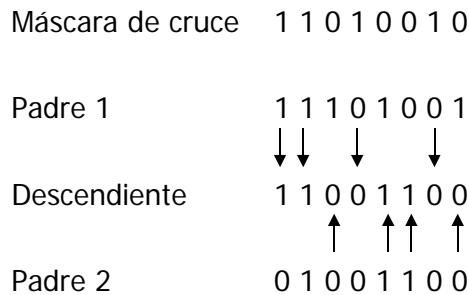


Fig.9. Operador de cruce conforme a una máscara de cruce

Reproducción

En términos biológicos, los individuos mejor adaptados perduran durante varias operaciones. Por eso, el operador de selección escoge aquellos individuos con buenas características y que deben perdurar a lo largo del tiempo y no desperdigarse entre otros individuos nuevos. El operador de reproducción selecciona a los individuos de una determinada población que pertenecen también a la siguiente, sin sufrir ningún otro tipo de modificación.

Mutación

Con los operadores anteriores se obtienen individuos que combinan rasgos que están presentes en los individuos de la población. Sin embargo, pueden existir características que quedan sin explorar si sólo se emplean los operadores anteriores. Con la mutación se introducen modificaciones en los individuos de la descendencia.

Siguiendo con el ejemplo anterior, la mutación consistiría en transformar un cero en uno (o viceversa) con una probabilidad determinada:

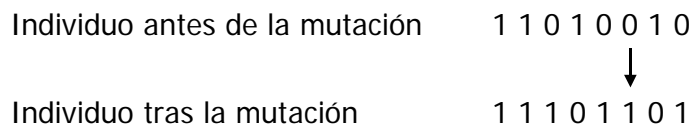


Fig.10. Operador de mutación

Si la probabilidad con la que tiene lugar la mutación es muy pequeña, es difícil que vuelvan a aparecer características que han sido abandonadas por el operador de cruce. En una determinada región puede ocurrir que las si las primeras tres posiciones están ocupadas por elementos '1', el resultado obtenido no sea bueno, por lo tanto los operadores de selección y cruce harán que los esas características se abandonen. Pero si cambia la región en la que tiene lugar la exploración, es decir, si el resto de los elementos cambian notablemente, puede ocurrir que tres '1' en las tres primeras posiciones arrojen un buen resultado; la mutación permite que esas características se vuelvan a recuperar.

Si la probabilidad de mutación es elevada, es probable que alguna de las características adecuadas de la solución se pierda y la técnica se desvíe de regiones potencialmente interesantes sin haber explorado con suficiente profundidad las

regiones visitadas. En el caso extremo de que la probabilidad de mutación fuera uno el proceso se convertiría en una exploración de todo el espacio de soluciones altamente ineficiente al no eliminar la posibilidad de examinar un individuo más de una vez.

Reducción

Inversión

El objetivo de este operador no es modificar las características de ningún individuo ni obtener alguna nueva, sino realizar modificaciones en las representaciones de tal manera que la probabilidad de que los individuos resultantes del cruce en posteriores iteraciones tengan más probabilidad de ser individuos mejor adaptados.

Dado un determinado individuo, como el de la figura 11, es más probable que tras una operación de cruce, el par de cromosomas indicados con círculo permanezcan unidos a que lo hagan los dos cromosomas señalados con un cuadrado. Sin embargo, puede ocurrir que sean los cromosomas indicados con los cuadrados los que ofrezcan mejores funciones de adaptación.



Fig.11. Justificación del operador de inversión

Para evitar que esto ocurre se introduce la mutación. Pero previamente es necesario ampliar la codificación de las soluciones. Los individuos que antes quedaban representados por una cadena de ocho bits ahora quedan representados por ocho parejas de elementos. Cada pareja está formada por el valor del bit y por la posición que dicho bit ocupa en la cadena. El individuo de la figura 11, podría quedar representado de alguna de las siguientes maneras:

(1,1),(1,2),(1,3),(0,4),(1,5),(1,6),(0,7),(1,8)

(1,8),(1,1),(1,3),(0,4),(1,5),(1,2),(1,6),(0,7)

(1,6),(1,8),(1,3),(0,4),(1,5),(1,2),(0,7),(1,1)

Con esta última representación del individuo es mucho más probable que los bits que antes estaban indicados con cuadrados permanezcan en un mismo individuo tras realizar una operación de cruce.

El operador de inversión introduce modificaciones de este tipo para equilibrar las probabilidades de asociadas a que diferentes agrupaciones de bits permanezcan igual tras las operaciones de selección y cruce. Sin embargo, en ningún caso se introduce ninguna modificación en términos del conjunto de soluciones considerados en cada iteración.

Para finalizar, se propone el flujograma de una de las posibles combinaciones de los operadores anteriores, la que describen Pham D.T. y Karaboga D. (2000).

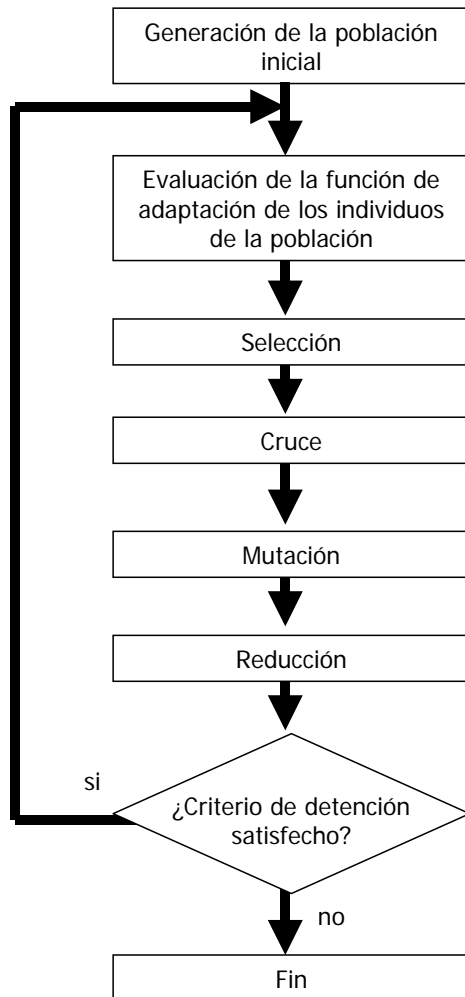


Fig.12. Algoritmo genético de Pham D.T. y Karaboga D. (2000)

3.3. Justificación de la eficacia de los algoritmos genéticos. Teorema de los esquemas

A continuación se ofrece una explicación de la eficacia de los algoritmos que no pretende ser una demostración rigurosa.

Para poder dar una explicación de cómo operan los algoritmos genéticos es necesario introducir el concepto de *esquema*. Cada una de las representaciones de las soluciones con las que trabaja el algoritmo es un vector del espacio de soluciones, que tiene tantas dimensiones como genes.

Un esquema es la región del espacio de soluciones (un hiperplano) que queda determinado cuando se establecen los valores de determinados genes. Si consideramos, por ejemplo, un problema en el que cada solución quede representada por cinco genes, cada uno de los cuales puede tomar cualquier valor entero entre 0 y 9, podemos definir el siguientes esquema:

X2X3X

Este esquema representa el hiperplano del espacio de soluciones definido por todos aquellos vectores cuyo segundo y cuarto genes son 2 y 3 respectivamente.

El *orden* de un esquema viene dado por el número de genes cuyos valores quedan definidos por dicho esquema. Los diferentes vectores pueden ser contemplados como esquemas de orden igual al número de genes.

Por otro lado, la *distancia* de un esquema es la diferencia entre las posiciones que ocupan los genes último y primero definidos por el esquema. En el ejemplo anterior la distancia es $4 - 2 = 2$.

La hipótesis sobre la que se sustentan los algoritmos genéticos es la siguiente: un punto determinado del espacio de soluciones aporta información relevante (es representativo) de los esquemas a los que pertenece. Es decir, si el cromosoma

42231

ofrece una buena función de adaptación, los esquemas que lo contienen (y entre ellos el del ejemplo anterior) probablemente también contengan más cromosomas con buenos valores de la función de adaptación. Como, además, un determinado cromosoma es el resultado de la intersección de un conjunto de esquemas, la búsqueda de una buena solución pasa por la búsqueda de un buen conjunto de esquemas.

Por otro lado, es preferible estudiar esquemas de orden reducido, porque definen regiones más amplias del espacio de soluciones y, a su vez, contienen a los esquemas de mayor orden. Además, la intersección de esquemas de orden reducido define otros de mayor orden. Así, por ejemplo, si los esquemas:

XX3X y 4XXX

ofrecen soluciones mejores que la media del resto de los esquemas considerados, cabe esperar que el esquema

4X3X

represente una región de soluciones potencialmente más interesante que el resto de esquemas.

Sin embargo, aunque existen esquemas de orden reducido mejores que otros, se desconoce cuáles son los mejores y cuales los peores. Una forma de comprobar si, efectivamente, un esquema contiene cromosomas interesantes pasaría por explorar sistemáticamente dichos esquemas. Sin embargo, la exploración sistemática consume el tiempo que las técnicas metaheurísticas tratan de ahorrar.

El mecanismo de los algoritmos genéticos orienta la búsqueda de manera que es capaz de combinar la detección de nuevos esquemas interesantes con la exploración de aquellos conocidos y potencialmente interesantes (un esquema es potencialmente interesante porque algún o algunos cromosomas contenidos en ellos ofrecen resultados superiores a la media).

Por último, en los casos en los que se debe elegir durante *muchas* ocasiones entre dos alternativas, cada una de las cuales ofrece un resultado que responde a una determinada distribución de probabilidad (una mejor y otra peor), la estrategia óptima (en términos de la esperanza de los resultados obtenidos) consiste en incrementar exponencialmente el número de selecciones de la alternativa que en mayor número de veces ha ofrecido mejor resultado.

El caso de los algoritmos genéticos constituye una extensión del problema anterior en el que no se trata de una alternativa sino de muchas. Supongamos el caso en el que sólo existe cruce y que la selección de los individuos se realiza proporcionalmente al valor de la función de adaptación. Si en un determinado instante la población ofrece un conjunto de individuos en los que un determinado esquema H ofrece mejores resultados que la media, es decir:

$$f(H) = (1 + c) \cdot f$$

donde

$f(H)$ es la media de los valores de la función de adaptación de los individuos contenidos en dicho esquema y

f es la media de los valores de la función de adaptación de la población,

la cantidad de individuos que contengan dicho esquema en la siguiente generación aumentará en la misma proporción, esto es:

$$m(H, t+1) = m(H, t) \cdot \frac{f(H)}{f} = (1+c) \cdot m(H, t)$$

donde $m(H, t)$ y $m(H, t+1)$ son, respectivamente, el número de individuos contenidos en el esquema H en el instante t y en el instante $t+1$.

La explicación no es tan sencilla; existen más operadores genéticos que alteran la proporción en la que aumentan los individuos contenidos en un determinado esquema, los individuos están contenidos en varios esquemas (si se consideran esquemas de orden 1, están contenidos en un número igual a las dimensiones del espacio de soluciones considerado), y, por último, no se trata de seleccionar entre dos alternativas en cada iteración, sino entre un conjunto de ellas.

Holland y Goldberg ofrecen demostraciones rigurosas del teorema de los esquemas prestando atención a estas consideraciones y concluyen que con el operador de selección y cruce se espera obtener descendientes mejor adaptados que sus antecesores.

3.4. Conclusión

Los algoritmos genéticos son constituyen una técnica metaheurística ampliamente estudiada con multitud de variantes y con aplicaciones a numerosos campos. Su particularidad consiste en que en cada iteración estudia un conjunto de soluciones a partir de las que realiza operaciones, para combinar características de las mismas con ánimo de obtener poblaciones de mejor calidad.

4. Recocido simulado

4.1. Presentación

La técnica del recocido simulado fue formulada por Kirkpatrick, Gelatt y Vecchi en 1983. Esta técnica está inspirada en el proceso de solidificación de los sólidos. En este proceso, a medida que baja la temperatura el sólido va modificando su configuración. Cada una de las configuraciones tiene asociada una energía determinada. Por otro lado, a medida que la temperatura decrece, el conjunto de configuraciones que puede adoptar el sólido (y por lo tanto los diferentes estados de energía) se va restringiendo.

Con una velocidad de enfriamiento muy rápida se obtienen sólidos no cristalinos, porque partiendo de estados energéticos elevados la temperatura desciende rápidamente y las moléculas no pueden reconfigurarse para adoptar configuraciones cada vez más estables (con menor energía) debido a que, como hemos dicho, las posibles configuraciones se reducen conforme disminuye la temperatura.

El recocido simulado trata de hacer mínima una función, que hace las veces de nivel de energía del sólido. A lo largo del proceso se van explorando diferentes soluciones, que en términos del proceso de solidificación equivalen las diferentes configuraciones que puede adoptar el sólido. Igual que en el proceso de solidificación, en el recocido simulado se define una variable temperatura, T , que influye en las posibles soluciones a las que se puede acceder en cada iteración.

El recocido simulado ha sido aplicado, entre otros, a los siguientes campos: diseño de circuitos, comunicaciones, Investigación operativa, Física y Geofísica, Biofísica, Bioquímica y Biología molecular.

4.2. Descripción del proceso

Partiendo de una solución inicial y de una temperatura inicial, se exploran un número L de soluciones vecinas. Tras examinar cada solución vecina, si ésta es

mejor que la solución de partida, se acepta como nueva solución. Si es peor, se admitirá como nueva solución con una determinada probabilidad:

$$\text{probabilidad de aceptar un solución peor} = e^{-\frac{\Delta f}{T}}$$

donde Δf es el incremento de la función objetivo, $f_{\text{solución candidata}} - f_{\text{solución actual}}$

T es la temperatura

La probabilidad de aceptar una solución cuyo valor de la función objetivo es mejor que la actual es mayor cuanto mayor sea la temperatura y cuanto menor sea el incremento de la función objetivo que se trata de minimizar.

El proceso anterior se repite hasta que se cumple alguna determinada condición.

Por lo tanto, antes de comenzar el proceso es necesario definir una solución inicial, un vecindario y una manera de acceder a las soluciones vecinas, una temperatura inicial, una temperatura final y una ley de enfriamiento y un criterio de detención.

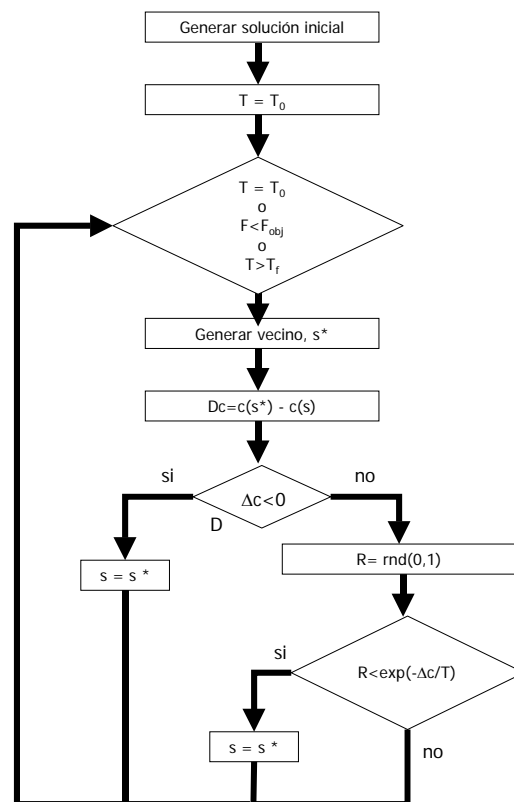


Fig.13. Diagrama de flujo del recocido simulado

Una solución inicial

Como en las técnicas anteriores, la solución de partida puede ser una solución obtenida a partir de una técnica más sencilla, o de forma aleatoria o bien a partir de una solución conocida del problema considerado.

Vecindario

Debe establecerse cómo se generan las soluciones que conforman el vecindario dada una determinada solución y la forma de seleccionar una de ellas, como candidata a ser la nueva solución.

Temperatura inicial

Conviene definir una temperatura inicial de manera que la aplicación de la técnica sea suficientemente robusta, es decir, que la solución alcanzada no dependa de la solución de partida. Esto se traduce en una temperatura inicial suficientemente alta que permita movimientos erráticos al principio del proceso.

Por otro lado, conviene que el proceso sea eficiente y no admita movimientos de empeoramiento *desorbitados*.

Por ello conviene establecer la probabilidad, ϕ , de aceptar una solución S_{cand} que sea un μ por uno peor que la inicial S_0 . De acuerdo con esto, y si f es la función objetivo, es posible determinar la temperatura inicial:

$$\left. \begin{aligned} \Delta f = f(S_{cand}) - f(S_0) = (1 + \mu)f(S_0) - f(S_0) = \mu S_0 \\ \phi = e^{-\Delta f / T_0} \end{aligned} \right\} \quad \phi = e^{-\frac{\mu T_0}{f(S_0)}} \Rightarrow T_0 = \frac{\mu}{-\ln(\phi)} \cdot S_0$$

Velocidad de enfriamiento

La velocidad de enfriamiento puede ser de tipo aritmético, geométrico, etc. Existen estudios que analizan diferentes velocidades.

Temperatura final

En teoría, la temperatura final del proceso de búsqueda debería ser 0. Sin embargo, la probabilidad de aceptar movimientos de empeoramiento se hace muy pequeña

sin necesidad de alcanzar los 0 grados. Lundy y Mees (1986) proponen una temperatura final que viene dada por la siguiente expresión:

$$T_f < \frac{\varepsilon}{\ln(n-1) - \ln(\theta) + \ln(1-\theta)} \approx \frac{\varepsilon}{\ln(n)}$$

donde θ es la probabilidad de que se obtenga una solución cuyo valor de la función objetivo menos el de la óptima global es menor que ε y

n es el número de elementos del espacio de soluciones.

Criterio de detención

Como en las técnicas anteriores, el criterio de detención puede ser haber superado un determinado tiempo de computación, el número máximo de iteraciones o un número de iteraciones sin haber mejorado la solución o haber alcanzado un determinado valor de la función objetivo.

Además, en esta técnica, opera un criterio adicional de detención que es el de la temperatura final, de manera que al alcanzarla, el procedimiento se detiene.

Por último, conviene señalar un criterio de detención adicional para el caso en el que el proceso se detiene cuando durante un número determinado de iteraciones (cada una de ellas a una determinada temperatura) no se ha mejorado la solución. Tras realizar las L exploraciones correspondientes a una determinada temperatura puede ser que no se considere la iteración correspondiente a dicha temperatura en el cómputo de iteraciones en las que la solución no ha mejorado. Esto es así, si el porcentaje de soluciones de empeoramiento aceptadas es superior a un determinado valor. En este caso, se considera que aunque el proceso no ha mejorado la solución, no conviene penalizar (anotándolo en el contador de iteraciones sin mejorar la solución) porque existe la posibilidad de acceder a regiones interesantes. En este caso, el flujograma quedaría como se muestra en la figura:

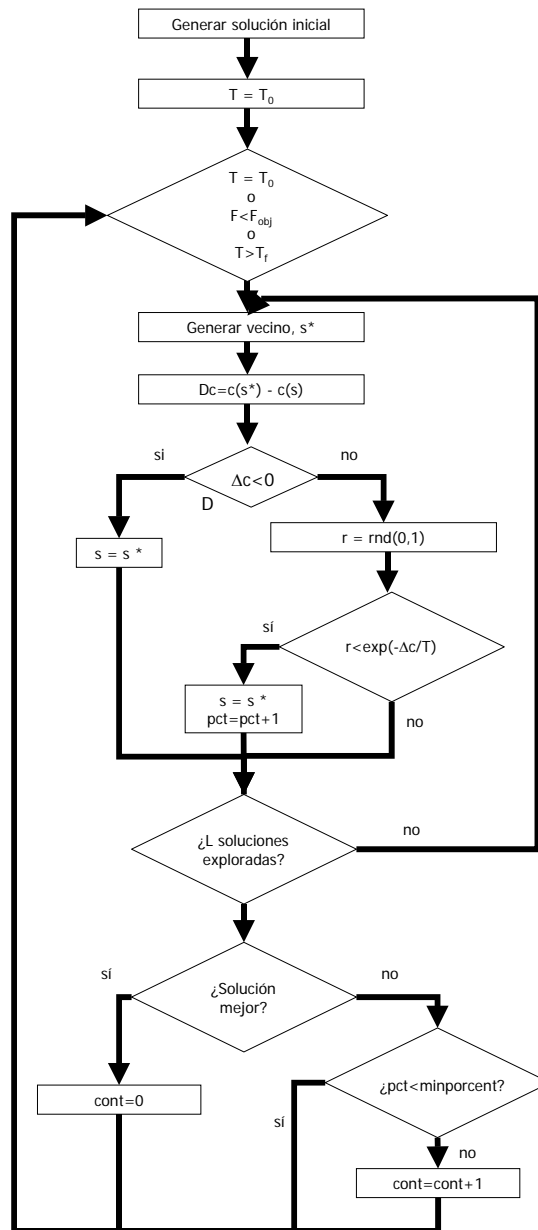


Fig.13. Diagrama de flujo de una variante del recocido simulado

4.3. Variantes del recocido simulado

Existen numerosas variantes del recocido simulado cuyo objetivo es mejorar su eficiencia mediante la reducción del número de iteraciones necesarias para alcanzar una buena solución. A continuación se citan, a título de ejemplo, dos de estas variantes (Aarts, E. y Korst, J., 2001):

- *Criterio de aceptación basado en un umbral determinista.* En esta variante una solución peor que la actual se acepta siempre que el incremento de la función no supere un determinado umbral. Este umbral disminuye a medida que el proceso de búsqueda avanza, el umbral disminuye. En esta variante, una vez seleccionada la solución candidata la el criterio de selección no incorpora ningún elemento aleatorio.
- *Recocido simulado con saltos en la temperatura.* En este caso, la temperatura crece rápidamente o disminuye rápidamente en determinadas fases del proceso con ánimo de permitir la exploración de nuevas regiones en el primer caso o con intención de explorar más a fondo una determinada región del espacio de soluciones.

4.4. Conclusiones

Más

El recocido simulado puede ser formulado en términos de cadenas de Markov, lo cual ha permitido estudiar aspectos relativos a las configuraciones más adecuadas del la técnica y la convergencia del proceso de búsqueda (Aarts et al.,1986, Otten y van Ginneken, 1989, Sadiq y Habib, 1999 y Aats y Korst, 2001). Se ha demostrado que si el recocido simulado opera durante un tiempo infinito la probabilidad de alcanzar una solución no óptima es cero. Sin embargo, para que el proceso sea eficiente se debe establecer un número limitado de exploraciones en cada temperatura. La temperatura inicial debe ser suficientemente alta como para que la probabilidad de aceptar cualquier solución al comienzo sea prácticamente uno.

5. Búsqueda *scatter*

La búsqueda *scatter* fue originalmente propuesta por Glover en 1997. Esta técnica, es una técnica evolucionaria (como los algoritmos genéticos y las colonias de hormigas). **Basado en la combinación de reglas para la programación de la producción de los años sesenta y en la combinación de restricciones en problemas de programación entera y no lineal.**

En cada iteración se dispone de un conjunto de soluciones del problema y a partir de ellas se realizan operaciones para obtener un nuevo conjunto de soluciones que sean de alta calidad y que, al mismo tiempo, sean suficientemente *diferentes*. **(después se verá en qué sentido)** para lograr una búsqueda que abarque.

Para conseguir los objetivos anteriores combina soluciones para obtener otras mejores y mantiene un conjunto de soluciones de referencia que son las que utiliza para obtener las nuevas soluciones.

Premisas (artículo Scatter Search and Path Relinking, p.4)

Etapas:

1. Generación de soluciones iniciales:
a partir de una método particular (aleatorio, con otras reglas, etc.)
utilizando una *imagen fantasma*: una representación fácil del problema con 'límites' que se van relajando hasta que finalmente desaparecen
2. Mejora de las soluciones
3. Actualización de la lista de referencia
4. Generación de agrupamientos

Combinación de soluciones

Típicamente la generación de soluciones se realiza teniendo en cuenta todos los pares de soluciones del conjunto de soluciones de referencia.

Etapas:

1. Generación de un conjunto de soluciones iniciales que sean suficientemente diferentes como para que la diversidad del conjunto permita una búsqueda por todo el espacio de soluciones. A estas soluciones se les aplica algún heurístico para obtener soluciones mejores.
2. Selección de las mejores soluciones de las obtenidas anteriormente. El único criterio en este caso no es el del valor de la función objetivo; una solución puede ser seleccionada si mejora la diversidad del conjunto a pesar de no ofrecer el mejor de valor de la función objetivo de entre las candidatas.
3. Combinación estructurada de las soluciones elegidas anteriormente. Estas soluciones pueden quedar tanto dentro como fuera de la región convexa definida por las soluciones de referencia. La soluciones obtenidas de esta manera si modifican para garantizar que son soluciones 'aceptables'.
4. Aplicación de los heurísticos del paso 1 para mejorar las soluciones obtenidas.
5. Por último se selecciona un conjunto de las mejores soluciones obtenidas en el paso anterior y se incluyen en el conjunto de soluciones de referencia.

Características de la búsqueda dispersa:

- El conjunto de soluciones de referencia contiene información útil sobre la situación (o la forma) de las mejores soluciones.

Para explotar de forma adecuada la información contenida en el conjunto de soluciones de referencia conviene diseñar mecanismos que permitan obtener soluciones fuera de las regiones a las que pertenece las soluciones de referencia y, además, incorporar heurísticos para mejorar las que se han obtenido.

6. GRASP

Greedy randomized adaptative search procedure

Dos fases:

- Fase constructiva. Mediante un proceso constructivo se genera alguna solución inicial de forma 'greedy'. En esta etapa se construye la solución paso a paso. En cada paso existe un conjunto de soluciones (de elementos que se añaden a la

solución en proceso de construcción): RCL (restricted candidate list). Para que en un determinado paso de la etapa de construcción se considere como candidato un elemento, el aumento (o la disminución) de la función objetivo que se logra con dicho elemento debe estar en un determinado rango (que queda fijado por un parámetro, α)

Cada una de estos elementos tiene una probabilidad de ser incorporado a la solución. Existen diferentes formas de asignar probabilidades a los elementos que se incorporan.

- Fase de búsqueda local. A partir de la solución generada se estudia el vecindario con ánimo de encontrar nuevas soluciones.

Greedy Randomized Adaptative Search Procedure

Dos fases:

- Construcción de una determinada solución.
 - En cada fase constructiva existe un conjunto de elementos candidatos para construir la solución (aspecto 'greedy'): lista restringida de candidatos.
 - De estos candidatos se selecciona aleatoriamente alguno de ellos (aspecto aleatorio).
 - Y tras cada fase de la construcción se modifica la lista restringida de candidatos (aspecto adaptativo)
- Optimización local de la primera solución obtenida. A partir de la solución obtenida en la fase constructiva se define un vecindario y se realizan iteraciones mientras exista algún elemento en el vecindario mejor que la solución en cada iteración.

En los extremos: búsqueda 'greedy' pura (en la que la lista de candidatos contiene un solo elemento) y búsqueda aleatoria (en la que la lista de candidatos incluye todos los elementos restantes para construir la solución). La técnica GRASP se debe situar en algún punto entre estos dos extremos. A medida que el algoritmo se aproxima a uno estrictamente 'greedy' se reduce la varianza de las soluciones obtenidas en la fase constructiva, pero la calidad de las soluciones que se obtienen en la fase de optimización local disminuye. Si, al contrario, el procedimiento se hace cercano a la búsqueda aleatoria, las varianzas de las soluciones aumenta mucho y no hay garantías de comenzar la fase de optimización con soluciones

suficientemente atractivas. El tiempo necesario en la primera opción es mucho mayor que en la segunda. Un método intermedio arroja tiempos...

Limitación: cada ciclo es independiente de los anteriores. El procedimiento no aprende de su historia. Alternativas en la fase constructiva:

- Reactive GRASP: el parámetro que determina el compromiso entre la búsqueda aleatoria y la búsqueda 'greedy' se selecciona en cada construcción, dependiendo de las soluciones que se han obtenido anteriormente.
- Perturbación de la función objetivo (en el art. 48 hablan de un problema concreto, falta generalidad)
- Función de sesgo. Una vez definida la lista de elementos candidatos a formar parte de la solución, se asigna una probabilidad a cada uno de ellos en función del orden que ocupan en el conjunto de los elementos de la lista. Se han propuesto diferentes funciones que introducen el sesgo de diferentes maneras.
- Incorporación de la memoria. Para mejorar el proceso global se incorpora una memoria de largo plazo formada por soluciones de élite. Una solución pasa a formar parte de este grupo de soluciones de élite si es 'suficientemente' buena (según algún criterio) y 'suficientemente' diferente del resto de soluciones de élite. La selección de los elementos que se incorporan a la solución en la fase constructiva se relaciona con las veces que dicho elemento aparece en las soluciones de élite, favoreciendo que los elementos que aparecen en las soluciones de élite aparezcan en la fase constructiva.
- Principio de proximidad óptima. Cabe esperar que cerca de buenas soluciones se encuentren otras soluciones, igualmente, buenas. Por eso, Fleurent y Glover (1999) proponen realizar procesos de optimización local no solamente tras finalizar la etapa constructiva sino en diferentes momentos a lo largo de ella (no necesariamente en todos para preservar la eficiencia global del procedimiento).

7. Colonia de hormigas

Maniezzo

Basado en el comportamiento de las hormigas. Las hormigas dejan a su paso un rastro de feromonas que le sirve de medio de comunicación entre ellas. Esta técnica

emplea *hormigas* que crean soluciones al problema considerado. En cada interacción se dispone de un conjunto de hormigas que construyen un conjunto de soluciones. Se trata, por lo tanto, de una técnica evolucionaria, como los algoritmos genéticos y la búsqueda *scatter*.

Cada hormiga construye una solución mediante un método constructivo pasando de unos estados a otros, todas soluciones parciales, cada vez más completos. En términos de un problema de optimización, el rastro de feromonas depositadas por las hormigas se traduce en la probabilidad de pasar de un estado a otro determinado.

Al final, tras conocer la calidad de las soluciones obtenidas por cada una de las hormigas, se modifica el rastro de feromonas asociado a los cambios de estado de acuerdo con algún criterio que premie las transiciones correspondientes a hormigas que han ofrecido buenos resultados.

En la construcción de nuevas soluciones, las hormigas tendrán en cuenta:

- el atractivo que la transición de un estado a otro ofrece de acuerdo con algún criterio que *a priori* permite evaluar la idoneidad de dicha transición;
- la experiencia acumulada a partir de las soluciones construidas en iteraciones anteriores que esta almacenada en el rastro asociado a los movimientos que se realizan para construir una solución.

Comentar el primer artículo de Dorigo con el ejemplo del problema del viajante.

Variante: (bien contado en p.479 libro verde) MMAS: el valor del rastro está acotado inferior y superiormente. Se inicializan los rastros a un valor máximo. A medida que se realizan iteraciones se actualizan los rastros pero nunca quedan por debajo de un determinado valor con el objeto de evitar el estancamiento en torno a un óptimo local. Este artículo tiene, además, otras variantes interesantes.

(en el mismo artículo). Describe el ANTS: como una extensión de las colonias de hormigas: *Approximated Non-deterministic Tree Search*, parecidos a los 'Tree search algorithms' pero que incorporan elementos de las colonias de hormigas: el atractivo de las opciones que se consideran en cada rama y la actualización del rastro.

Referencias y bibliografía

Aarts, E. y Korst, J., (2001). Selected Topics in Simulated Annealing. Capítulo de 'Essays and Surveys in Metaheuristics'. Kluwer Academic Publishers, Massachusetts .

Díaz, A. et al (1996). Optimización Heurística y Redes Neuronales. Paraninfo. Madrid.

Falkenauer, E. (1989). Genetic Algorithms and Grouping Problems. John Wiley and Sons, 1998. Cheichester.

Godberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA.

Glover, F. (1989). Tabu search: Part I, ORSA Journal on Computing 1. 190-206.

Glover, F. (1990). Tabu search: Part II, ORSA Journal on Computing 2 (1990) 4-32.

Holland, John H. (1975). Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor.

Lundy, M. y Mees, A. (1986). "Convergence of an Annealing Algorithm" Mathematical Programming, Vol. 34, p. 111-124

Pham D.T. y Karaboga D. (2000). Intelligent Optimisation Techniques. Springer. London.

Reeves, C. (1995). Modern Heuristic Techniques for Combinational Problems. R., McGraw-Hill. Guildford.

Sadiq, S. M. y Habib, Y. (1999). Iterative Computer Algorithms with Applications in Engineering. Solving Combinatorial Optimization Problems. Wiley. **CIUDAD. AÑO.**

Silver, E. A. et al (1980). A tutorial on Heuristic Methods. European Journal of Operational Research. Vol. 5.

Zanakis, S. H. y Evans, J. R. (1981). Heuristic 'Optimization': Why, When and How to Use It". Interfaces. Vol. 11(5) .