

INTRODUCCIÓN

1. ¿Qué es un problema NP?

NP es el conjunto de problemas que pueden ser resueltos en tiempo polinomial por una máquina de Turing no determinista.

Un problema NP es aquel problema para el que se necesita un tiempo polinomial para comprobar la validez de una solución de dicho problema.

Diferencia NP y P

Un problema pertenece a la clase NP si puede ser resuelto en tiempo polinomial pero usando una computadora no determinista, mientras que la un problema de la clase P puede ser resuelto en tiempo polinomial en una computadora determinista.

P vs. NP

P(multiplication) is the class that includes all the problems than can be solve by a reasonable fast program.

NP(Sudoku) is the class that if you have a correct solution you can at least check it in a reasonable amount of time. P E NP.

¿P=NP? → ¿Does being able to quickly recognize NP correct answers mean there's also a quick way to find them?

P stands for polynomial time, where the numbers of steps required isnt that bad compare to the sizes of the problem.

NP is all about polinomyal check it.

2.¿Qué representaciones de soluciones conoces?

Formas de representar soluciones:

- Permutación
- Números Enteros
- Binarios
- Números Reales
- Árboles

Y vectores de estos.

3.¿Qué es la función objetivo de un problema y qué aporta a las Mhs?

Es la función que determina el valor de aptitud de una solución a una instancia de un problema. En una MH aporta una medida de lo “buena” que es una solución. El objetivo de las Mhs es optimizar esta función (maximizar o minimizar según el problema).

4.¿Qué es un método heurístico y qué una MH?

Un método heurístico trata de explotar conocimiento relativo a la instancia del problema a optimizar y, por tanto, sólo es aplicable a este problema. Una metaheurística es un procedimiento genérico para problemas de optimización, que puede aplicarse a cualquiera de ellos. Ciertas Mhs están diseñadas para utilizar alguna clase de heurística en alguna parte de su algoritmo, pero el procedimiento general es universal.

Las heurísticas son usadas cuando no existe una solución óptima bajo las restricciones dadas, o cuando no existe del todo.

Una heurística es una técnica que busca soluciones buenas (casi óptimas o eficaces) a un coste computacional razonable (eficiente), aunque sin garantizar factibilidad u optimalidad de las mismas.

Un proceso que puede resolver un cierto problema, pero que no ofrece ninguna garantía de lograrlo, se le denomina una “heurística” para ese problema.

5.¿Cuándo se debieran utilizar MHs y qué ventajas tienen?

Las MHs se aplican a problemas de optimización. Deberían utilizarse cuando sólo interesan soluciones relativamente buenas, no necesariamente la mejor (óptimo global) y cuando no se tiene un método eficiente para encontrar tales soluciones.

Cuando nos enfrentemos a espacios de búsqueda muy grandes, en los que los algoritmos y técnicas clásicas más eficientes que existen para resolver el problema requieren de tiempos exponenciales.

6.¿En qué términos se pueden clasificar las Mhs?

Se pueden encontrar diferentes formas de clasificar las metaheurísticas, vamos a diferenciar según la característica que tengamos en cuenta:

- Inspiración: naturales o sin inspiración
- Número de soluciones utilizadas: poblaciones o de trayectorias
- Función objetivo: estáticas o dinámicas
- Número de vecindades: unas o varias vecindades
- Uso de memoria: con o sin memoria

7.¿Qué es mejor, diversificar o intensificar?

La diversificación pura () lleva a una MH no guiada por la función objetivo (búsqueda aleatoria) y la intensificación pura se estanca muy rápido en óptimos locales (búsqueda local). Las buenas MHs son las que hacen un uso equilibrado de ambas.

8.¿Qué dice el teorema del No Free Lunch?

No Free Lunch determina que dado un problema de optimización arbitrario, cualquier MH da resultados igualmente malos. Sin embargo, este teorema parte de unas hipótesis demasiado fuertes: los problemas reales son un subconjunto de todos los problemas, lo que desequilibra esta suposición y provoca que existan MHs mejores que otras. No Free Lunch también nos dice que para resolver un problema de forma efectiva es necesario explotar información sobre el mismo (en el caso de las MHs esto consiste en experimentar con varias de ellas).

MÉTODOS DE BÚSQUEDA LOCAL

1.¿Qué es un proceso de búsqueda/optimización local?

La búsqueda local es una MH basada en trayectorias que parte de una solución aleatoria para ir optimizándola paso a paso. Hace uso de una estructura/operador de vecindario que permite obtener soluciones diferentes pero parecidas a la actual. En cada iteración busca una solución mejor a la actual en su vecindario. Es un algoritmo de tipo “hill climbing”.

2.¿Se pueden aplicar métodos de BL sobre cualquier problema?¿Qué es necesario para ello?

Sí. Lo único necesario es definir una estructura de vecindario para una solución dada.

3.¿Qué elementos tienen un procedimiento de Búsqueda Local?

Una representación y una evaluación, como todas las MHs. Una solución inicial, una estructura de vecindario, una estrategia de exploración y una condición de parada.

4.¿Cuándo para un proceso de optimización local?

Para cuando no existe ningún elemento en el vecindario que mejore a la solución actual. Se pueden utilizar numerosas estrategias como el reinicio para evitar que el algoritmo permanezca ocioso.

5.¿Se puede aplicar cualquier estructura de vecindario a cualquier problema, varios o únicamente a un problema?

Suelen ser específicas para cada problema, ya que cada uno puede tener sus restricciones, pero problemas cuyas soluciones se implementen mediante la misma representación pueden utilizar estructuras de vecindario muy similares.

6.¿Qué podrías decir acerca del tamaño de una estructura de vecindario y cómo afecta éste a la BL?

El tamaño de una estructura de vecindario puede afectar enormemente a la

velocidad de convergencia de una BL. Si se utiliza la estrategia de exploración de mejor mejora cada iteración de la búsqueda se retrasará enormemente. Utilizar en este caso la estrategia de primera mejora podría subsanar este problema. Esta última exploración podría ser en orden aleatorio para evitar sesgos en las soluciones obtenidas.

Un ejemplo de esto es optimización de funciones de parámetros reales, donde la cardinalidad del vecindario es demasiado grande como para aplicar mejor mejora de forma eficiente (si se tiene en cuenta precisión infinita sería aplicarla).

7.¿Cómo se puede optimizar la ejecución de un método de Búsqueda Local?

Debido a que en cada exploración se obtienen una solución muy similar a la actual se puede optimizar el cálculo de su aptitud (que supone la mayor parte de la carga computacional) realizándola a partir de la aptitud de la solución actual. En ciertos casos, el cálculo podría incluso realizarse en un tiempo de complejidad $O(1)$.

8.¿Compara las diferentes estrategias de exploración de vecindarios que conozcas?

La estrategia de mejor mejora recorre todo el vecindario y selecciona la solución (si existiera) que supone una mayor mejora sobre la solución actual. Es un método determinista, esto es, para una misma solución inicial, siempre sigue el mismo camino. Se suelen conseguir búsquedas más predecibles y que convergen más rápido.

La estrategia de primera mejora recorre el vecindario y selecciona la primera solución encontrada (si existiera) que mejore a la solución actual. Su determinismo depende del orden de exploración del vecindario. Si éste es aleatorio, se obtendrán trayectorias distintas en cada ejecución, de lo contrario será determinista, igual que mejor mejora. Cada iteración es más rápida (por lo general) que mejor mejora pero son necesarias (de nuevo, por lo general) más iteraciones (convergencia más lenta).

9.Describe con un ejemplo la aplicación de un operador de vecindario en un problema.

En el problema de la mochila una representación típica es un vector de valores booleanos (bits). Un operador de vecindario válido para esta situación sería aquel que interviene N bits de este vector. Cada vecino de la solución actual se diferenciará de la solución actual en N bits. Si la instancia tiene M objetos el tamaño del vecindario será las combinaciones sin repetición de M elementos

tomados de N en N.

10.¿Cuáles son las limitaciones de los métodos de Búsqueda Local y cómo se puede abordar?

La principal limitación de los métodos de BL es que son bastante “miopes”: sólo son capaces de ver a un paso de distancia, atascándose en óptimos locales muy fácilmente.

Existen variaciones sobre la BL que tratan de solventar este problema introduciendo factores de diversificación como perturbaciones (ILS), vecindarios variables (VNS), reglas de aceptación variable (SA), estructuras de memoria (TS)...

ENFRIAMIENTO SIMULADO Y BÚSQUEDA TABÚ

1.¿Cuál es la estrategia en Enfriamiento Simulado y Búsqueda Tabú para superar las limitaciones de la búsqueda local?

Ambas introducen estrategias para que sea posible seleccionar en cada iteración soluciones que no tienen por qué ser las mejores, para no quedarse estancadas en óptimos locales.

2.¿Cuál es el objetivo de usar el parámetro de la temperatura en Enfriamiento Simulado?

El objetivo es variar la exigencia de aptitud de las nuevas soluciones seleccionadas conforme avanza la búsqueda.

3.¿Qué significa enfriar la temperatura en SA?

Significa aumentar la exigencia de aptitud en la selección de nuevas soluciones

4.¿Cómo podemos calcular la temperatura inicial para SA?

Se puede realizar un estudio previo sobre la variación de aptitud media en una sola iteración del algoritmo y calcular la probabilidad inicial deseada con la que seleccionar una solución con una variación de aptitud como la calculada utilizando la fórmula de la regla de aceptación.

Podemos generar un conjunto de soluciones aleatorias, generar un vecino para cada una, y comparar sus fitness. Una vez comparados, aplicar cualquier tipo de operación aritmética.

5.¿Qué es necesario para garantizar que se alcance el óptimo global en SA?

Se garantiza una convergencia al óptimo global si se alcanza una distribución

estacionaria en cada temperatura y ésta disminuye muy lentamente (p. ej. con un esquema logarítmico). Esto es, por lo general, extremadamente lento.

Conforme el esquema de enfriamiento se ralentice más (p. ej. un enfriamiento cada N iteraciones) la probabilidad de alcanzar el óptimo global tiende al 100%. Esto, sin embargo, puede llegar a ser más lento que una búsqueda completa del espacio de soluciones.

6. ¿Qué regla de aceptación es más permisiva en SA, Metropolis o logística?

Metropolis es más permisiva. No solo determina a una probabilidad de aceptación más alta para soluciones peores, sino que además la probabilidad es de 100% para soluciones mejores, mientras que en logística nunca se alcanza una probabilidad de 100%.

7. ¿Cuál es el principal elemento de Búsqueda Tabú?

Las estructuras de memoria a corto plazo y a largo plazo.

8. ¿Qué implementaciones de la STM (memoria a corto plazo) conoces?

Una lista de soluciones tabú (estrategia pobre), de movimientos tabú, de atributos tabú...

9. ¿Qué implementaciones de la LTM (memoria a largo plazo) conoces?

Una contabilización de frecuencias de los elementos seleccionados más a menudo (representación binaria), otra contabilización, pero de los valores más típicos en cada posición (representación entera), una pila de tamaño limitado de las mejores soluciones encontradas...

10. ¿Para qué sirve la tenencia/antigüedad tabú?

Es una forma de desechar tras un periodo de tiempo las reglas tabú y así poder explorar nuevas oportunidades de mejora.

11. ¿Para qué sirve el criterio de aspiración?

Para seleccionar soluciones, incluso si son tabú, que mejoran sobre un cierto umbral la solución actual. Permite no desaprovechar oportunidades de cierta mejora.

12. ¿Para qué sirve la lista de candidatos?

Para hacer más eficiente el recorrido del vecindario en el caso de que sea demasiado grande. La lista de candidatos es un subconjunto (muestra) del vecindario completo, a menudo ordenado según una heurística.

13. ¿Cómo podemos utilizar la memoria a largo plazo para intensificar?

Al converger en un óptimo local, puede generarse una nueva solución inicial (reinicio) utilizando la memoria a largo plazo para generar una solución parecida a las mejores que han sido visitadas (p. ej. seleccionar elementos con una probabilidad directamente proporcional a su frecuencia).

14. ¿Cómo podemos utilizar la memoria a largo plazo para diversificar?

Al converger en un óptimo local, puede generarse una nueva solución inicial (reinicio) utilizando la memoria a largo plazo para generar una solución lo más opuesta posible a las soluciones ya visitadas (p. ej. seleccionar elementos con una probabilidad inversamente proporcional a su frecuencia).

15. ¿Qué condiciones de parada conoces?

Parar (o reiniciar) tras un número de fallos, limitar el número de evaluaciones utilizadas (útil para comparativas entre algoritmos), limitar el tiempo computacional...

ILS, GRASP Y IG

1. ¿Cuál es la motivación para ILS?

ILS modifica el espacio de búsqueda de forma que se disminuye la densidad de malas soluciones haciendo una búsqueda sobre los óptimos locales.

2. ¿Cuáles son los elementos principales de ILS?

Una solución inicial (aleatoria/heurística), un método de BL, un operador de perturbación (esto es, un movimiento aleatorio en un vecindario de mayor orden que el de la BL) y un criterio de aceptación.

3. ¿Qué elementos de ILS proveen diversificación y cuáles intensificación?

La BL aporta intensificación, el operador de perturbación aporta diversificación. El criterio de aceptación puede aportar ambas, dependiendo de su implementación: si se elige la mejor de las soluciones se consigue intensificación, si se elige aleatoriamente se consigue diversificación, así como todos los puntos intermedios.

4. ¿Cómo puede y debe modularse el equilibrio entre intensificación/diversificación en ILS?

Mediante el criterio de aceptación. Si se elige siempre la mejor solución se consigue intensificar, si la selección es aleatoria se consigue diversificar. Si se utilizan criterios probabilísticos dependientes de la diferencia de aptitud (como SA) se puede modular esta relación.

5. ¿Qué criterios de aceptación para ILS conoces?

Aceptar la mejor solución (intensificador), aceptar de forma aleatoria (diversificador) o criterios probabilísticos (Metropolis, logístico) como en SA.

6. ¿Cuál es la clave de los métodos VNS?

Que un óptimo local en una estructura de vecindario no tiene por qué serlo en una estructura distinta, pero un óptimo global sí que lo es para cualquier estructura.

7. ¿Cuáles son las similitudes y diferencias entre ILS y VNS?

Ambas utilizan la estrategia de perturbación pero en VNS se encuentra parametrizada. Además, el criterio de aceptación en VNS es determinista (siempre se escoge la mejor solución).

8. ¿Cuál es la idea de GRASP?

Utilizar un conocimiento del problema (heurística) junto con un factor aleatorio para generar una solución inicial para una BL. De esta forma, se acelera la convergencia.

9. ¿Cómo pueden aleatorizarse los métodos voraces?

En general existen dos métodos: elegir en cada iteración un componente aleatorio elegido entre los mejores según la heurística o elegir el mejor componente según la heurística de un conjunto de componentes aleatorios.

10. ¿Cómo puede y debe modularse el equilibrio entre intensificación/diversificación en GRASP?

Modificando el factor aleatorio en la solución inicial heurística. Un generador voraz puro siempre devolverá la misma solución (intensificación) y un generador aleatorio puro hará que se convierta en una BL ordinaria (diversificación).

Para modular esto se puede modificar el tamaño del conjunto de componentes seleccionado. Si se utiliza la estrategia “muestreo greedy, selección aleatoria”, aumentar el tamaño del conjunto de los N mejores componentes aumentará la diversificación (y viceversa). Si se utiliza la estrategia de “muestreo aleatorio, selección greedy”, aumentar el tamaño del muestreo aleatorio aumentará la intensificación (y viceversa).

11. ¿Cuál es la idea de IG?

Aplicar un constructor voraz sobre soluciones parciales. En lugar de empezar desde cero tras converger, destruir parcialmente la solución encontrada para mantener características de la misma y reconstruirla mediante este constructor

voraz.

12. ¿Cuáles son las alternativas de los elementos de IG?

El constructor voraz puede tener o no una componente aleatoria (que aporta diversificación). También puede explotar, además de la heurística, una estructura de memoria.

El destructor aleatorio puede tener o no una componente heurística. El número de características a destruir puede ser fijo o variable y también puede explotar una estructura de memoria [p. ej. IG con mejora tabú]).

El criterio de aceptación de nuevas soluciones puede ser determinista (siempre la mejor), aleatorio o probabilista.

13. ¿Cómo puede y debe modularse el equilibrio entre intensificación/diversificación en IG?

Variando el aporte de la componente heurística/aleatoria en el constructor y el destructor. También influye el criterio de aceptación utilizado.