

Heurísticas

Análisis y Diseño de Algoritmos

Dr. Jaime Osorio Ubaldo

Generalmente existen dos estrategias para resolver un problema, los algoritmos exactos (analíticos) y los algoritmos heurísticos (de aproximación).

- 1 **Algoritmos exactos.** Garantizan una solución óptima, siempre que exista; sin embargo, en problemas grandes o muy complejos podría ser inviable ya que el costo computacional crece de forma exponencial con el tamaño del problema.
- 2 **Algoritmos heurísticos.** No garantizan la solución óptima; pero si producen soluciones aproximadas al óptimo, en la mayoría de los casos, en tiempos considerablemente menores.

Algoritmos heurísticos

- 1 Heurística deriva de la palabra griega heuriskein que significa encontrar o descubrir.
- 2 “Un algoritmo heurístico es un procedimiento para resolver un problema de optimización bien definido mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución.”



¿Cuándo usar algoritmos heurísticos?

- 1 En problemas de difícil solución.
- 2 Cuando no se conoce ningún método de solución.
- 3 Aunque existe un algoritmo exacto, su costo computacional es muy alto.
- 4 Es más flexible, permitiendo, por ejemplo, la incorporación de condiciones de difícil modelización.
- 5 Se utiliza como parte de un procedimiento global que garantiza el óptimo del problema, puede ser en la elección de la solución de partida o en un paso intermedio de un procedimiento, como por ejemplo las reglas de selección de la variable a entrar en la base en el método simplex.

¿Son de propósito general?

Algunos algoritmos de resolución de propósito general, como pueden ser los algoritmos exactos de backtracking, existe un procedimiento conciso y preestablecido, independiente en gran medida del problema abordado. En los algoritmos heurísticos esto no es así. Las técnicas e ideas aplicadas a la resolución de un problema son específicas de éste y aunque, en general, pueden ser trasladadas a otros problemas, han de particularizarse en cada caso. Es decir, solo resuelven problemas concretos.

Propiedades de una buena heurística

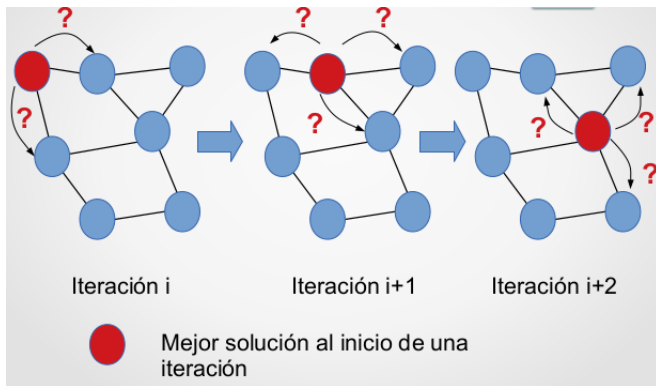
- 1 Permite obtener la solución en un tiempo realista.
- 2 La solución obtenida deber aproximarse a la solución optima.
- 3 Que tenga una baja probabilidad de encontrar soluciones lejos del óptimo.
- 4 Sea intuitivo y tan simple como sea posible.

Tipos de heurísticas

- 1 **Métodos constructivos.** Consisten en añadir paso a paso componentes individuales a la solución hasta que se obtiene una solución factible. El más popular de estos métodos son los algoritmos greedy.
- 2 **Métodos de descomposición.** El problema original se descompone en subproblemas (divide y vencerás).
- 3 **Métodos inductivos .** Se trata de generalizar de versiones pequeñas o más sencillas al caso completo.
- 4 **Métodos de reducción.** La idea es identificar propiedades que se cumplen mayoritariamente por las buenas soluciones e introducirlas como restricciones del problema con la finalidad de reducir el espacio de soluciones.
- 5 **Métodos de búsqueda en la vecindad.** Comienzan con una solución del problema y la mejoran progresivamente.
- 6 **Manipulación del modelo.** La idea es modificar la estructura del modelo con el fin de hacerlo más sencillo de resolver, deduciendo, a partir de su solución, la solución del problema original.

Local search

- 1 Se inicia con una solución, no necesariamente buena.
- 2 Se busca soluciones mejores en sus vecinos.
- 3 No se memorizan soluciones obtenidas.



Seleccionar una solución x_0 cualesquiera.

Repetir

- calcular su función objetivo $f(x_0)$.

- explorar la vecindad y determinar un vecino factible x_v

- si $f(x_v) > f(x_0)$, entonces $x_0 = x_v$

Hasta un máximo número de intentos definido.

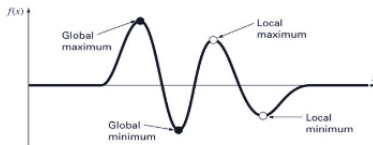
Ventajas y desventajas

1 Ventajas

- Usa poca memoria.
- No demora mucho.
- Generalmente encuentra soluciones aceptables.

2 Desventajas

- Facilmente se estanca en óptimos locales.
- Puede tardarse más de lo usual en problemas donde el óptimo está en una cuasi-meseta.



Problema de la mochica con local search

item	valor	peso
1	10	10
2	30	20
3	65	30
4	40	40
5	60	50

- 1 Supongamos que la capacidad es 90.
- 2 Si elegimos a $x_0 = (1, 2, 4)$ la función factible es $f(x_0) = 80$
- 3 Explorando en el vecino $x_v = (2, 4, 5)$, se observa que no es factible.
- 4 Explorando en el vecino $x_v = (1, 3, 4)$, se obtiene $f(x_v) = 115$
- 5 Ya que $f(x_v) > f(x - 0)$, entonces $x_0 = x_v$.

Explique un problema que puede ser resuelto usando algoritmos heurísticos.