

OPTIMIZACIÓN ESTOCÁSTICA: BÚSQUEDA LOCAL

ALAN REYES-FIGUEROA
INTELIGENCIA ARTIFICIAL

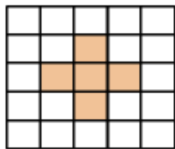
(AULA 12) 23.FEBRERO.2024

Búsqueda Local

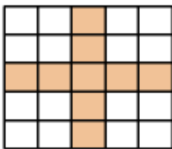
La **búsqueda local** es un método heurístico para resolver problemas de optimización.

La búsqueda local se puede utilizar en problemas que se pueden formular como encontrar una solución que maximice un criterio entre varias soluciones candidatas. Los algoritmos de búsqueda local se mueven de una solución a otra en el espacio de las soluciones candidatas (el espacio de búsqueda) mediante la aplicación de cambios locales, hasta que se encuentra una solución considerada óptima o transcurre un límite de tiempo.

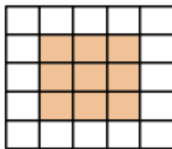
Usan ampliamente el concepto de vecindad de un estado o configuración.



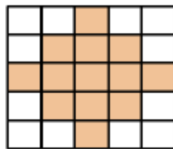
Linear 5 (L5)



Linear 9 (L9)



Compact 9 (C9)



Compact 13 (C13)

Búsqueda Local



Existen muchas variantes de la búsqueda local:

- *hill-climbing*
- *hill-climbing* estocástico,
- *local beam search*,
- búsqueda tabú o *tabu-search*,
- algoritmos de estimación de distribuciones,
- algoritmos y métodos evolutivos:
 - algoritmos genéticos (GA),
 - *particle swarm optimization* (PSO),
 - evolución diferencial (DE).

Hill Climbing

En el análisis numérico, la *escalada* o *hill-climbing* es una técnica de optimización matemática que pertenece a la familia de la búsqueda local. Es un algoritmo iterativo que comienza con una solución arbitraria a un problema, luego intenta encontrar una mejor solución haciendo un cambio incremental en la solución. Si el cambio produce una mejor solución, se realiza otro cambio incremental en la nueva solución y así sucesivamente hasta que no se puedan encontrar más mejoras.

En este tipo de técnicas se suele aplicar una estrategia *greedy*. Dentro de las soluciones candidatas (una muestra de todas las soluciones posibles), se elige aquella o aquellas que maximizan la función objetivo, dentro de este conjunto de soluciones cantidades.

Hill Climbing

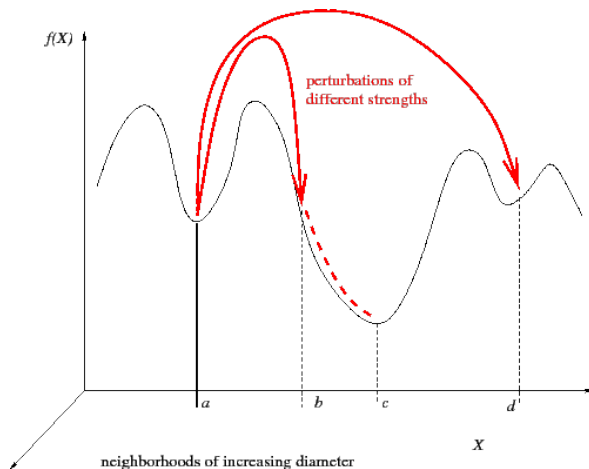
Idea simple:

- Comenzar desde un punto \mathbf{x}_k ,
- Repetir:
 - Moverse al estado o configuración óptima en la vecindad $N(\mathbf{x}_k)$ (mejor vecino),
 - Si no hay vecinos mejores, salir.

Este algoritmo, en la práctica, suele conducir mínimos locales. Se puede mejorar introduciendo **saltos aleatorios** fuera de la vecindad.

- Comenzar desde un punto \mathbf{x}_k ,
- Repetir:
 - Moverse al estado o configuración óptima en la vecindad $N(\mathbf{x}_k)$ (mejor vecino),
 - Si no hay vecinos mejores, salir.
- Dar un salto aleatorio a otra región.

Hill Climbing



Esquema de saltos aleatorios para salir de mínimos locales.

Hill Climbing

```
function HILL-CLIMBING(problem) returns a state
  current ← make-node(problem.initial-state)
  loop do
    neighbor ← a highest-valued successor of current
    if neighbor.value ≤ current.value then
      return current.state
    current ← neighbor
```

“Like climbing Everest in thick fog with amnesia”

Búsqueda Tabú

La *búsqueda tabú* o *tabu search* es una variante de la búsqueda local. Se caracteriza principalmente por llevar una lista de posiciones prohibidas.

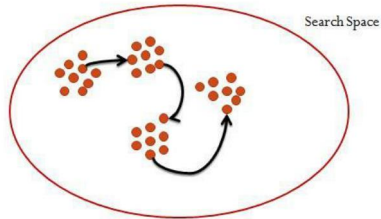
Posición prohibida = configuraciones que se han visitado recientemente.

Estas configuraciones se almacenan en una lista. Usualmente se elige un parámetro entero $M > 0$ que lleva el conteo o tamaño máximo de la lista de posiciones prohibidas, y se establece una posición prohibida como aquella que ha sido visitada en las últimas M iteraciones anteriores.

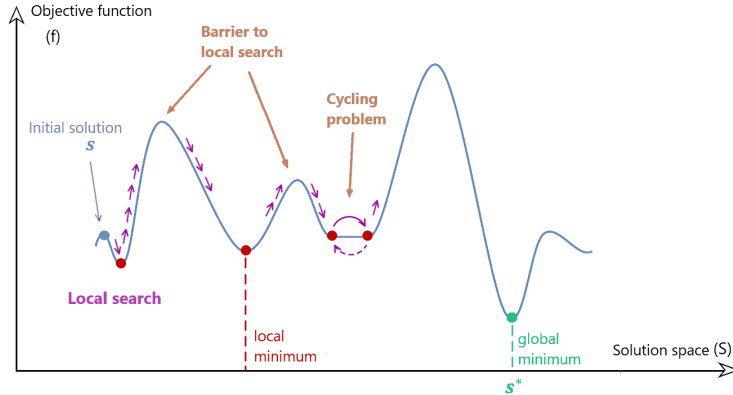
Tiene como propósito añadir un componente de *long-term memory* a los métodos de búsqueda local.

Tabu Search Algorithm

- Diversification
 - Probabilistic
 - Long term memory



Búsqueda Tabú



Una ventaja de la búsqueda tabú es que permite no ciclarse.