

Artificial Intelligence

ENRIQUE PELÁEZ Ph.D.

Escuela Superior Politécnica del Litoral – ESPOL

Facultad de Ingeniería en Electricidad y Computación – FIEC

Semestre 2022 – II

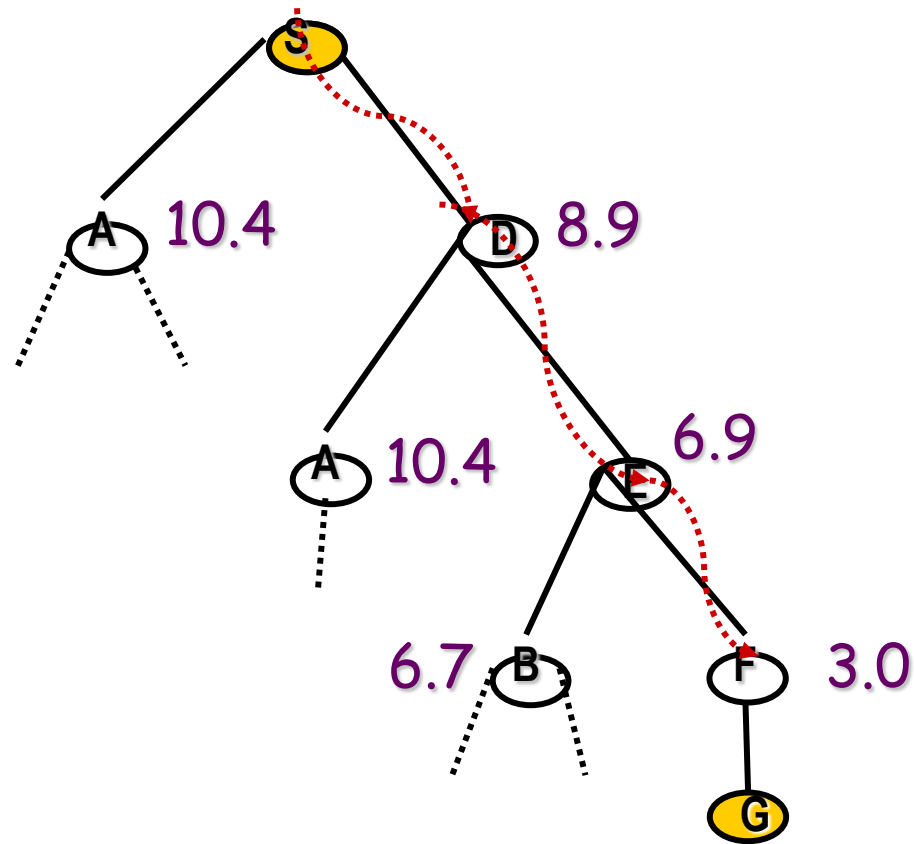
Agenda

- Búsqueda heurística
 - Hill Climbing
 - Simulated Annealing
 - Best – First

Búsqueda heurística

- Ejemplo: considere el problema de conducir entre ciudades: podríamos considerar cada ciudad como un nodo en un árbol, y las rutas entre ciudades, como un operador o regla, con alguna información heurística para ayudar al viajero a moverse de una ciudad a otra. otro.

Búsqueda heurística



- Es como la búsqueda primero a lo profundo, pero:
- En lugar de seleccionar el siguiente nodo de izquierda o derecha,
- Selecciona el nodo con el mejor valor heurístico.

Hill Climbing

- ***Hill Climbing***
 - Es una versión mejorada del algoritmo de generación y prueba.
 - Desde el procedimiento de prueba, hay una retroalimentación para ayudar al generador de la solución a decidir, en qué dirección ir a continuación y moverse en el espacio de la solución.
 - En este proceso, la búsqueda finaliza si no hay un próximo estado o nodo razonable para mover.

Simple Hill Climbing

1. Evaluar el estado inicial.
2. Repetir hasta que se encuentre una solución o no haya nuevos operadores para aplicar:
 - Seleccionar un nodo y aplicar la función heurística
 - Evaluar el nuevo estado:
3. objetivo → terminar
4. mejor que el estado actual → nuevo estado actual

Hill Climbing: Disadvantages

Maximo local

- Es el mejor estado o nodo en comparación con sus vecinos, pero no es el mejor entre otros, que podría estar adelante en el espacio de la solución.

Hill Climbing: Desventajas

Meseta

- Un área plana del espacio de búsqueda en el que todos los estados vecinos tienen el mismo valor.

Hill Climbing: Desventajas

Acantilado

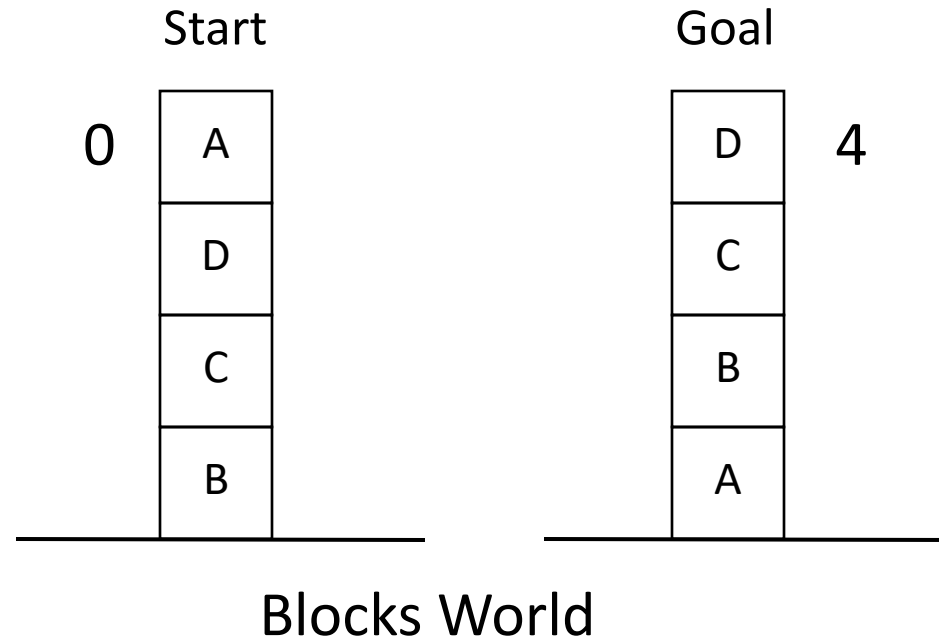
- Es un tipo especial de máximo local, que es imposible de cruzar con movimientos simples.

Hill Climbing: Desventajas

Salidas

- Retroceder a algún nodo anterior e intentar ir en una dirección diferente.
- Dar un gran salto para intentar ingresar a una nueva sección del espacio de solución.
- Moverse en varias direcciones a la vez.

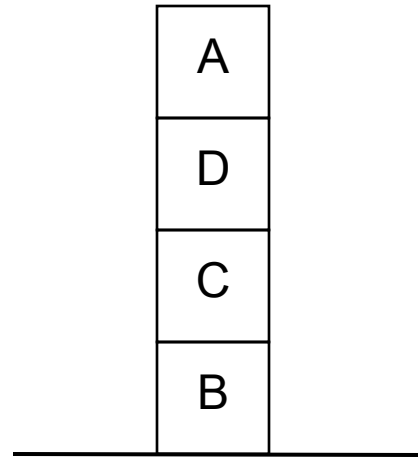
Hill Climbing: Meseta



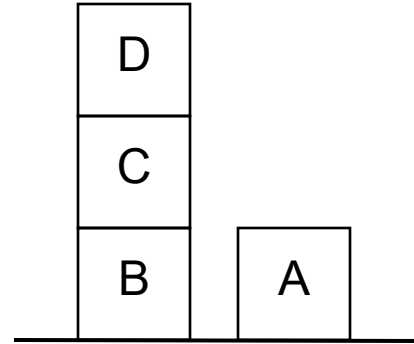
Local heuristic – plateau problem:

+1 for each block that is located on the position it is supposed to be located on.
-1 for each block that is located in a wrong position.

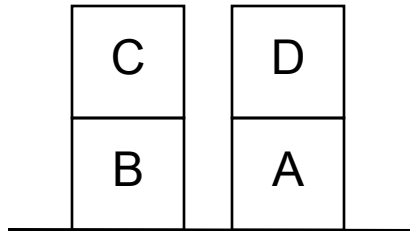
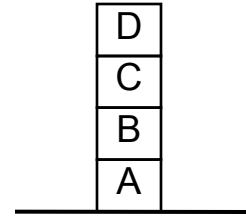
Hill Climbing: Meseta



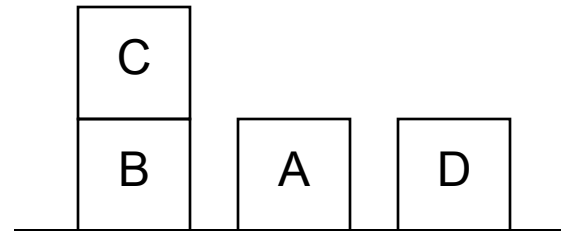
0



2



$1-1+1-1=0$



$1-1+1-1=0$

Simulated Annealing

- Es una versión mejorada del algoritmo Hill Climbing.
- Al principio, este algoritmo explora una buena parte del espacio de la solución, de tal manera que la solución final podría no verse afectada por el estado inicial.
- Como consecuencia, la posibilidad de quedar atrapado en un máximo local, mesa o acantilado, es pequeña.

Simulated Annealing

1. Evaluar el estado inicial.
2. Repetir hasta que se encuentre una solución o no haya nuevos operadores para aplicar:
 - Establecer T según un estado inicial
 - Selecciona y aplica un nuevo operador
 - Evaluar el nuevo estado:
 - goal \rightarrow terminar
 - $\Delta E = \text{Valor (estado actual)} - \text{Valor (nuevo estado)}$
 - $\Delta E < 0 \rightarrow$ nuevo estado actual
 - else \rightarrow nuevo estado actual con probabilidad $e^{-\Delta E/kT}$

Best-First Search

- **Best - First**

1. **Lista open:** contiene los nodos generados.

A los nodos se les ha aplicado la función heurística; pero aún no han sido probados; es decir, sus sucesores aún no se han generado. Esta cola puede considerarse como una cola de prioridad, en la que los elementos con mayor prioridad son los que tienen valores más altos, asignados por la función heurística.

1. **Lista close:** contiene los nodos que se han examinado y no son el objetivo esperado.
2. **Función heurística:** permite que el algoritmo busque primero las rutas más prometedoras.

Best-First Search

Función heurística:

$$f(n) = g(n) + h(n)$$

- $g(n)$ = costo de la ruta más barata desde el estado inicial al nodo n .
- $h(n)$ = costo de la ruta más barata desde el nodo n a un estado objetivo.

Best-First Search

- Greedy search:

$h(n)$ = costo estimado de la ruta más barata desde el nodo n hasta un estado objetivo.

Ni óptimo ni completo

Best-First Search

- Uniform-cost search:

$g(n)$ = costo de la ruta más barata desde el estado inicial hasta el nodo n .

Óptimo y completo, pero muy ineficiente

Best-First Search

- Algoritmo A

$$f(n) = g(n) + h(n)$$

Best-First Search

- **Best - First**

- La función heurística ***f*** se puede definir como la suma de otras dos funciones, llamadas ***g*** y ***h***.
- La función ***g*** es el costo para alcanzar el estado actual desde el estado inicial.
- La función ***h*** es el costo estimado para alcanzar el estado objetivo desde el estado actual.

$$f(n) = g(n) + h(n)$$

Best-First Search

[illegible]