**4 - Búsqueda informada y exploración**

**4.1 – Estrategias de búsqueda informada (heurísticas)**

* La estrategia de búsqueda informada usa conocimiento específico sobre el problema
* La aproximación general es **búsqueda primero el mejor**
  + Es un caso particular del algoritmo general de búsqueda de árboles o grafos
  + Se selecciona un nodo para la expansión basada en una **función de evaluación**
    - Mide la distancia al objetivo, f.
  + Se puede implementar con una cola de prioridad, que mantendrá la frontera en orden ascendente de valores f.
* Hay una familia entera de algoritmos de búsqueda primero mejor con funciones de evaluación diferentes
  + Un componente clave de estos algoritmos es una **función heurística** h(n) = costo estimado del camino más barato desde nodo n a un nodo objetivo

Modos para de usar la información heurística para dirigir la búsqueda

**Búsqueda voraz primero el mejor**

* Trata de expandir el nodo más cercano al objetivo
* Se evalúan los nodos usando solamente la función heurística: f(n) = h(n)
* Es algoritmo avaro porque en cada paso trata de ponerse tan cerca del objetivo como pueda
* Si no somos cuidados en descubrir estados repetidos, la solución puede que nunca se encontrará
* Se parece a la búsqueda primero en profundidad en el modo que prefiere seguir un camino hacia el objetivo, pero volverá atrás cuando llegue a un callejón sin salida
* No es óptima y es incompleta (puede irse hasta abajo y no probar las otras opciones)

**Búsqueda A\*: minimizar el costo estimado total de la solución**

* Evalúa los nodos combinando g(n), el costo para alcanzar el nodo, y h(n), el costo de ir al nodo objetivo
  + f(n) = g(n) + h(n)
  + g(n) nos da el costo del camino desde el nodo inicio al nodo “n” y h(n) el costo estimado del camino más barato desde “n” al objetivo:
    - f(n) = costo más barato estimado de la solución a través de n
    - Es razonable intentar primero el nodo con el valor más bajo de g(n) + h(n)
    - Resulta más razonable que con tal de que la función heurística h(n) satisfaga ciertas condiciones, la búsqueda A\* es tanto completa como óptima
* A\* es óptima si h(n) es una **heurística admisible,** con tal de que h(n) nunca sobrestime el costo de alcanzar el objetivo
  + Son optimistas por naturaleza, piensan que el costo de resolver el problema es menor que es en realidad
* **Consistencia:** una heurística es consistente si para todo nodo “n” y cada sucesor “n’ “ de n, generado por cualquier acción a, el costo estimado de alcanzar el objetivo desde n no es mayor que el costo de alcanzar n’ más el costo estimado de alcanzar el objetivo desde n’
  + H(n) <= c(n, a, n’) + h(n’)
    - Esto es una forma de **desigualdad triangular**
      * Especifica que cada lado de un triángulo no puede ser más largo que la suma de los otros dos lados
      * En nuestro caso el triangulo está formado por n, n’ y el objetivo mas cercano a n
* El hecho de que los f-costos no disminuyan a lo largo de cualquier camino significa que se pueden dibujar **curvas de nivel** en el espacio de estados
  + Dentro de la curva de nivel, todos los nodos tienen f(n) menor que ese valor
* El algoritmo A\* es **óptimamente eficiente** para cualquier función heurística, ningún otro algoritmo garantiza expandir menos nodos que A\*

**Búsqueda heurística con memoria acotada**

* **Búsqueda recursiva del primero mejor (BRPM)**
  + Algoritmo recursivo que intenta imitar la operación de búsqueda primero el mejor, pero utiliza solo un espacio lineal
  + Estructura similar a la búsqueda primero en profundidad pero es seguir indefinidamente hacia abajo en el camino actual
  + Mantiene la pista del f-valor del mejor camino alternativo disponible desde cualquier antepasado del nodo actual
  + Si el nodo actual excede este límite, la recursividad vuelve atrás al camino alternativo
  + BRPM sustituye los f-valores de cada nodo a lo largo del camino actual
  + Mantiene la pista del f-valor del mejor camino alternativo disponible desde cualquier antepasado del nodo actual
  + Si el nodo actual excede el límite, la recursividad vuelve al camino alternativo , BRPM sustituye los f-valores de cada nodo a lo largo del camino con el mejor f-valor de su hijo
  + Es un algoritmo óptimo y la función heurística h(n) es admisible
  + A\*PI y BRPM sufran de utilizar muy poca memoria
* Parece sensible usar toda la memoria disponible
  + Dos algoritmo que hacen esto son A\*M y A\*MS, memoria acotada y memoria simplificada respectivamente
  + A\*MS
    - Avanza como A\*, expandiendo la mejor hoja hasta que la memoria esté llena
    - Para retirar, el algoritmo retira el peor nodo hoja (f-valor mas alto)
    - Se devuelve hacia atrás, a su padre
    - El antepasado de un subárbol olvidado sabe la calidad del mejor camino en el subárbol, A\*MS vuelve a generar el subárbol solo cuando todos los otros caminos parecen peores que el camino olvidado

**Aprender a buscar mejor**

* Podría un agente aprender a buscar mejor?
* Si, el método se poya sobre un concepto llamado **espacio de estados metanivel**
  + Cada espacio captura el estado interno de un programa que busca en un **espacio de estados a nivel de objeto**
  + Cada acción en el espacio de estados es un paso de cómputo que cambia el estado interno

**Palabras clave**

|  |  |
| --- | --- |
| **Búsqueda informada** | Usa conocimiento específico sobre el problema más allá de la definición del problema en sí mismo |
| Búsqueda primero el mejor | Se selecciona un nodo para la expansión basada en una función de evaluación |
| **Función de evaluación** | Mide la distancia al objetivo |
| **Función heurística** | costo estimado del camino más barato desde nodo n a un nodo objetivo |
| **Búsqueda voraz primero el mejor** | Trata de expandir el nodo más cercano al objetivo |
| **Búsqueda A\*** | Evalúa los nodos combinando g(n), el costo para alcanzar el nodo, y h(n), el costo de ir al nodo objetivo |
| **Heurística admisible** | Son optimistas por naturaleza, piensan que el costo de resolver el problema es menor que es en realidad |
| **óptimamente eficiente** | Ningún otro algoritmo óptimo garantiza expandir menos nodos |
| **Búsqueda recursiva del primero mejor** |  |