Imprimir

Sesión 3-Tema 2: Estrategia de búsqueda A*

Tema 2: Estrategias de búsqueda (parte 2) – Resumen

A continuación, desarrollaremos el algoritmo de búsqueda aditiva A*, y para ello, hay que aclarar ciertos conceptos.

El algoritmo A* se considera que hace una búsqueda óptima porque siempre encuentra la mejor solución posible con el camino de mínimo (o máximo) coste posible, y para ello tenemos la siguiente función de mínimo coste:

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

g*(n): Coste mínimo al nodo actual (nodos pasados)

h*(n): Coste mínimo desde el nodo actual hasta la meta (nodos futuros)

f*(n): g*(n) + h*(n) (función evaluación óptima)

C*: Coste final óptima (desde nodo inicial hasta la meta - solución)

Hay ciertos conceptos que debemos tener en cuenta para esta búsqueda óptima:

g(n) >= g*(n): cuando es estrella implica que es el mejor camino posible a ese nodo, por lo que el resto de camino sí o sí deben ser igual o más grande, sino es que había un camino mejor y hay que actualizar g*(n).

¿f*(n) = C*?: Cuando es el camino óptimo, si 'n' es un nodo del camino óptimo, cualquier f*(n) del camino óptimo es igual a C* (f(n) = g(n) + h(n) -> camino hasta ahora + especulación hasta meta)

h(n) <= h*(n): Es una aproximación hasta alcanzar h*, cuanto más cerca de la meta, explora menos nodos, esta es una heurística admisible, garantiza coste mínimo. Sí h(n) > h*(n), ese camino no tiene solución.

Si un algoritmo A usa una heurística admisible, es un algoritmo A* (búsqueda óptima).

Explicación algoritmo A*:

Escoges un nodo (el que tenga una mejor f(n)), si es meta, termina el algoritmo. Exploras sus hijos (nodos adyacentes) si estos no pertenecen a la lista interior (nodos ya explorados). Si el nodo hijo ya está en la lista frontera (nodos prometedores), actualizas h(n), g(n) (comparando su g(n) anterior, el cual tiene un padre diferente porque ya se ha explorado), f(n) y pasas al siguiente nodo. SI no estaba en lista frontera, hace lo mismo y lo añade a la lista frontera y la interior.

Diferentes heurísticas para h(n):

Heurística óptima: Calcular la distancia manhattan entre 2 puntos, geométricamente es la mejor (óptima) para distancia entre bloques.

Heurística admisible: Explora más estados, tarda más que la óptima, pero encuentra la solución óptima, Usa la hipotenusa entre los 2 puntos para la función admisible.

Inconvenientes mantener la admisibilidad:

- Consume mucho tiempo en discriminar caminos insignificantes.
- Aumenta la lista frontera e interior ya que explora todos los nodos, por lo que tiene dificultades de espacio en memoria.
- No es práctico para problemas grandes.

Soluciones a los problemas anteriores:

- Relajación de la optimalidad – Técnicas de admisibilidad-e (épsilon):

F(sol) <= (1+e)C*

- Ponderación dinámica: Anticipamos la profundidad de la solución (cuando podamos), N, aplicamos a función: f(n) = g(n)+h(n)+ε[1- d(n)/N] h(n). La variable d(n) comienza en 0 (muy lejos de N) hasta d(n) = 1 (casi o igual a N), es decir a la solución.
- Estimación de costes: Seleccionamos nodo prometedor de lista frontera, sumamos 1+ e a ese f(n) para obtener los nodos de alrededor no más lejos de esa suma ((1+e) + f(n)) y obtienes una lista focal. Aplicas segunda heurística y te quedas el mejor. Lf = {n: f(n) ≤ (1+ε) min(f(m))}

- Relajación de la optimalidad – Ajuste de pesos: fw(n) = (1-w)g(n)+w h(n). La heurística h(n) <= h*(n) se comporta como función aditiva. En función de 'w', si es 0, la función es muy lenta, si cómo máximo es ½, la función poda mucho pero está bien, si sobrepasa ½ poda demasiado, no encuentra solución óptima o puede que incluso ninguna solución.

Algoritmo de búsqueda heurística A*

Índice WIki individual