Sesion 3 – 30/09/2021

5.2.2 Estrategias informadas. Funcioens de evaluación

…. pag14

f(n)=g(n) -> búsqueda por anchura

mientras que aumenta la cantidad de informacion, disminuye el espacio de búsqueda

cuales son las condiciones mas optimas?

las tentativas no informales:

-anchura: que pasa por todos los estados, f(n) = g(n), por niveles

A\* : f\*(n) = g\*(n) + h\*(n)

pag17

si tenemos una función h(n)=0, realizamos una búsqueda por anchura

f\*(n)=C\* se cumple para los nodos que formen parte del camino optimo

--pdf parte 2

Algoritmo A\*

listaFrontera es el elemento mas importante para la búsqueda, ya que mientras no este vacia el algoritmo funciona, y si esta vacia no hay solución, se guardan los candidatos para el camino optimo.

listaInterior: las celdas que ya hemos evaluado y no forman parte del camino optimo.

si encontramos la meta devolvemos el camino optimo

sino entonces expandimos el nodo n,vemos las reglas aplicables y repartirlos en 3 posibilidades:

1. si el hijo no esta en la listafrontera, lo agregamos
2. si ya esta en la lista de frontera, ver si el nuevo g es mejor (gm < m.g),y si lo es entonces utilizo g’m y recalculo la f
3. si la nueva g no es mejor, lo dejamos como estaba

-

pg2

a mayor h(n) aumenta la cantidad de informacion, y el coste computacional

**Wiki:**

Para explicar el algoritmo A\*, primero vamos a ver el pseudocodigo:

Alg A\*

listaInterior = vacio

listaFrontera = inicio

mientras listaFrontera no esté vacía

n = obtener nodo de listaFrontera con menor fn = gn+ hn

listaFrontera.del(n

listaInterior.add(n

si n es meta

devolver

reconstruir camino desde la meta al inicio siguiendo los punteros

fsi

para cada hijo m de n que no esté en listaInterior

g’(m) = n.g + c(n, m) //g del nodo a explorar m

si m no está en listaFrontera

almacenar la f, g y h del nodo en (m.f, m.g, m.h)

m.padre = n

listaFrontera.add(m)

sino si g’(m) es mejor que m.g //Verificamos si el nuevo camino es mejor

m.padre = n

recalcular f y g del nodo m

fsi

fpara

fmientras

devolver no hay solución

falg

-----------------

Como podemos ver, la listaFrontera es el elemento más importante para la busqueda, ya que mientras no este vacia el algoritmo funciona, y si esta vacia entonces no hay solucion. En esta lista es donde se guardan los candidatos para el camino optimo, inicialmente solo estará el nodo inicial.

La listaInterior contiene las celdas que ya hemos evaluado y no forman parte del camino optimo, inicialmente estará vacia.

Al principio de lbucle primero obtendremos el nodo con menor coste f de la listaFrontera, y lo borramos de la lista frontera y lo añadimos a la listaInterior.

Si n es el nodo meta entonces devolveremos el camino optimo, al cual podemos acceder mediante punteros sucesivamente desde el nodo final que apuntan a sus nodos padres hasta llegar al nodo inicial.

Sino entonces expandimos el nodo n, calculamos el nuevo coste de camino minimo desde el nodo n al m: g'(m)=g(n)+ c(n,m)

Luego vemos las reglas aplicables y los podemos repartir en 3 posibilidades:

Si el hijo m no está en la listaFrontera, lo agregamos.

Si ya está en la listaFrontera, vemos si el nuevo g es mejor (g'(m) < m.g), y si lo es entonces utilizamos esa g'(m) y recalculamos la f para el nodo m.

Si la nueva g no es mejor (g'(m)) entonces lo dejamos como estaba.

Todo esto se irá repitiendo hasta que no se encuentre el nodo meta y devuelva el camino optimo, o hasta que la listaFrontera este vacia, en cuyo caso querrá decir que no hay solucion.

Autor: Nikita Polyanskiy