PRÀCTICA 1: CERCA INFORMADA

Nom: Pol Morilla Mendez

En aquest informe veureu les diferents qüestions preguntades sobre la pràctica respostes una per una de forma seqüencial.

• Formalitzeu el problema definint els estats i els operadors:

En aquest problema concret, els estats als que es pot arribar durant l'execució del mateix són un seguit de valors numèrics que representen les diferents alçades a les que ens trobem de mentre explorem el mapa. Aquestes alçades són les següents:

{0, 1, 2, 3, 4, 5, 6, 7}

Com bé s'especifica a la pràctica, aquestes alçades comporten un cost depenent de l'alçada a la que ens trobem i a l'alçada a la que volem anar(moure's comporta un cost).

Per altre banda els diferents operadors aplicables a aquest problema són els que ens permeten desplaçar-nos per la matriu d'alçades. Els operadors disponibles en el nostre problema són els següents:

{pujar, baixar, esquerra, dreta}

Cada un d'ells comporta un desplaçament d'una sola casella cap a la direcció especificada.

• Doneu 3 heurístiques ben diferenciades (no tenen per què ser les 3 millors, però han de ser ben diferents) per intentar trobar el camí més ràpid des de l'estat inicial al final. Teniu en compte que la informació que poden fer servir les heurístiques són les característiques de les caselles actuals i finals (coordenades i alçada de cadascuna).

A continuació mostrarem les diferents heurístiques que he dissenyat per el problema que estem tractant en aquest document.

- *Distància de Manhattan:*

  La primera heurística en la que vaig pensar per aplicar al problema és la distància de Manhattan. Tenint en compte que el que estem calculant a cada pas són formes el millor possibles per arribar a una casella d'una matriu creiem que és convenient utilitzar una mètrica com aquesta.

  La heurística en aquest cas queda definida formalment com segueix:

$$h(x) = \sum_{i=1}^{d} \left| a_i - b_i \right|$$

- *Diferència entre alçades:*

  Un altre forma de heurística en la que he pensat és en simplement mirar la diferència d'alçades entre la posicions a les que ens podem moure a cada moment i comparar-la amb l'alçada del destí.

  $$h(x) \; = \; h_d - h_i$$

  En el cas de la fórmula anterior, 'hd' fa referència a la alçada del destí, mentre que 'hi' fa referència a l'alçada dels nodes que podem escollir d'entre els nodes als que ens podem desplçar.

- *Distància euclidiana:*

  Com bé coneixem, la distància més curta és sempre la distància en línia recta. Si bé en el cas del problema que volem analitzar no ens és permès moure'ns en direcció diagonal, sabem que la distància més curta que podem trobar entre un punt i un altre és aquesta distància euclidiana:

  $$h(x) \; = \; \sqrt{\sum_{k=1}^{n} (p_{ik} - p_{jk})^2}$$

• Per cada heurística, indiqueu si són o no admissibles respecte al temps. No cal que les 3 heurístiques que dissenyeu siguin admissibles, però almenys una d'elles ho hauria de ser.

A continuació es fa un petit anàlisis corresponent a les diferents heurístiques en el cas d'una matriu amb els diferents estats que s'han discutit anteriorment.

Distància de Manhattan:

En el cas de la distància de Manhattan podem veure que no és admissible respecte al temps, ja que posant per exemple un cas simple en el que volem fer un camí que requereix un sol desplaçament, sempre serà distància igual a 1. En canvi amb el cost real, si les alçaces tenen una diferència negativa, el valor del cost serà igual a ½(cost menor al que donaria la distància Euclidiana). Donant aquest cas d'exemple, podem dir que la nostra heuristica **no és admissible** tot i que en molts casos serà menor al cost real i per tant ens pot ajudar a trobar una solució òptima al problema.

Diferència entre alçades:

Una heuristica que hem dissenyat per assegurar una heuristica asmissible en el temps és la de la diferència d'alçades(sense tenir en compte un valor absolut). Com bé sabem, de vegades treure restriccions als costos reals d'un algoritme pot resultar útil per obtenir finalment una heuristica admissible.

Per comprobar que la nostra heuristica és admissible en el temps només cal posar un altre cop el cas més simple en el que volem anar d'un punt A a un punt B que requereix tan sols un sol desplaçament. Podem veure que en el cas d'una diferència positiva o 0, per calcular el cost real fem 1+(diferència entre alçades).

En aquest cas veiem que la nostra heurística és menor, ja que en cap moment sumem 1 al nostre valor. Per altre banda cal comprovar que en el cas d'una diferència negativa, en el cas del cost real el que es fa és assignar el valor ½. En el cas de la heuristica, donaria un valor negatiu, com que tenim en compte que el valor més favorable és un valor negatiu, aquesta heuristica demostra ser **admisible respecte al temps.**

Distància euclidiana:

Finalment, l'altre heuristica que hem escollit és la distància euclidiana. En aquest cas ens trobem com en el cas de la distància de Manhattan. Per verificar que no és una heuristica admissible en el temps(ja que no sempre ens dona un valor menor o igual al cost). Com en el cas de la distància de Manhattan, el valor mínim que podem obtenir per desplaçar-nos d'una casella a un altre és 1, mentre que si per exemple en el cas més simple tinguéssim una diferència d'alçades negativa, obtindriem un cost real de ½. Per tant podem veure que no en tots els casos tenim un cost heuristic menor, però també és veritat que en molts casos és aixi. Per tant podem veure que **no és una heuristica admissible**.

En les dues qüestions següents se'ns demana crear dos codis que implementin l'algorisme Best First i l'algorisme A*. He pensat que en comptes d'adjuntar el codi en aquest document no té gaire sentit, ja que conjuntament amb el document es proporciona el codi, per tant passem directament a la següent pregunta.

- **Proveu ambdós algorismes amb les 3 heurístiques per a diferents problemes (el de l'enunciat i, almenys, un altre mapa que dissenyeu vosaltres) indicant:**

    - **La solució (camí) que s'ha trobat amb el temps que li correspon.**
    - **El nombre de nodes que ha "tractat" l'algorisme de cerca per trobar el camí (és a dir el nombre d'iteracions de cerca que ha fet).**
    - **Si la solució trobada és l'òptima respecte al temps o no.**

Per fer aquesta part, tenint en compte que disposem d'una heuristica admissible amb el temps tal i com s'ha argumentat anteriorment, el que farem per saber si la solució donada és o no òptima respecte al temps el que farem és intentar veure si els resultats que ens dónen són els mateixos o no(determinant llavors que una solució donada no és la òptima i que l'altre si que la és).

Primer de tot el que farem serà evaluar les 3 heuristiques diferents amb el mapa proporcionat a la pràctica. Cal aclarar que per cada tipus d'heuristica i per cada algorisme farem dues proves diferents, una amb el mapa proporcionat a l'enunciat de la pràctica i l'altre amb un que hem creat:

Els mapes utilitzats són els següents(les caigudes són representades amb el número '-1':

Matriz1(proporcionat a la pràctica):

1,0,-1,1,3,2,3,4,3,1
2,1,-1,2,4,2,2,4,2,2
5,3,-1,2,3,2,-1,3,3,3
3,3,1,3,4,3,-1,1,2,2
2,2,2,3,6,4,-1,1,2,1
-1,-1,-1,-1,3,3,-1,0,2,-1
-1,-1,-1,-1,2,4,-1,2,2,-1
2,3,4,3,1,3,-1,3,2,-1
3,5,6,5,2,3,-1,5,3,-1
5,6,7,6,4,4,-1,6,4,5

Matriz2 (modificant el primer mapa):

1,4,-1,1,3,2,3,4,3,1
2,1,-1,2,4,2,1,5,2,3
5,3,-1,2,3,1,-1,3,3,3
3,3,1,3,4,3,-1,1,-1,2
2,2,2,3,6,4,3,1,5,1
-1,1,-1,2,3,2,-1,0,2,-1
-1,-1,-1,-1,2,4,-1,2,2,-1
2,3,4,3,1,3,-1,3,2,-1
3,5,6,5,-1,3,-1,5,3,2
5,6,7,6,5,4,-1,6,4,5

Primer de tot provarem les diferents heurístiques amb el primer mapa:

<u>Proves amb heuristica de la distància de Manhattan:</u>

BestFirst (anar del punt [0][0] al [9][9]):

Número d'iteracions: 50

Camí solució: [(0,0), (1,0), (2,0), (3,0), (4,0), (4,1), (4,2), (4,3), (4,4), (4,5), (3,5), (2,5), (1,5), (1,6), (1,7), (2,7), (3,7), (4,7), (5,7), (6,7), (7,7), (8,7), (9,7), (9,8)]

Temps d'execució: 9 ms

Per determinar si él camí que ens dóna és òptim comparem el camí resultat amb el cas del resultat de l'algorisme A* amb la heuristica de la diferència entre alçades(que és la única que és admissible). El resultat no és el mateix, per tant determinem que el resultat no és òptim amb el temps.

A(anar del punt[0][0] al [9][9]):*

Número d'iteracions: 52

Camí solució: [(0,0), (0,1), (1,1), (2,1), (3,1), (3,2), (3,3), (3,4), (3,5), (2,5), (1,5), (1,6), (1,7), (1,8), (1,9), (2,9), (3,9), (4,9), (4,8), (5,8), (6,8), (7,8), (8,8), (9,8)]

Temps d'execució: 13 ms

Per determinar si él camí que ens dóna és òptim comparem el camí resultat amb el cas del resultat de l'algorisme A* amb la heuristica de la diferència entre alçades(que és la única que és admissible). El resultat no és el mateix, per tant determinem que el resultat no és òptim amb el temps.

Proves d'heuristica amb la diferència entre alçades:

BestFirst (anar del punt [0][0] al [9][9]):

Número d'iteracions: 69

Camí solució: [(0,0), (0,1), (1,1), (2,1), (3,1), (3,2), (3,3), (2,3), (2,4), (2,5), (1,5), (1,6), (1,7), (2,7), (3,7), (4,7), (5,7), (6,7), (6,8), (7,8), (8,8), (9,8)]

Temps d'execució: 10 ms

Per determinar si él camí que ens dóna és òptim comparem el camí resultat amb el cas del resultat de l'algorisme A* amb la heuristica de la diferència entre alçades(que és la única que és admissible). El resultat és el mateix, per tant determinem que el resultat és òptim amb el temps.

A(anar del punt[0][0] al [9][9]):*

Número d'iteracions: 69

Camí solució: [(0,0), (0,1), (1,1), (2,1), (3,1), (3,2), (3,3), (2,3), (2,4), (2,5), (1,5), (1,6), (1,7), (2,7), (3,7), (4,7), (5,7), (6,7), (6,8), (7,8), (8,8), (9,8)]

Temps d'execució: 17 ms

Com que estem tractant amb l'algorisme A* i sabem que la heuristica que estem utilitzant és admissible, sabem que la soulció que se'ns proporciona és òptima respecte al temps.

Prova heuristica amb la distància euclidiana:

BestFirst (anar del punt [0][0] al [9][9]):

Número d'iteracions: 48

Camí solució: [(0,0), (1,0), (2,0), (3,0), (4,0), (4,1), (4,2), (4,3), (4,4), (4,5), (3,5), (2,5), (1,5), (1,6), (1,7), (2,7), (3,7), (4,7), (5,7), (6,7), (7,7), (8,7), (8,8), (9,8)]

Temps d'execució: 11 ms

Podem veure que el camí que obtenim no és el mateix que el que ens dona l'algorisme A* amb l'heuristica admissible, determinem que no és la solució òptima.

A(anar del punt[0][0] al [9][9]):*

Número d'iteracions: 51

Camí solució: [(0,0), (0,1), (1,1), (2,1), (3,1), (3,2), (3,3), (3,4), (3,5), (2,5), (1,5), (1,6), (1,7), (2,7), (3,7), (4,7), (5,7), (6,7), (7,7), (7,8), (8,8), (9,8)]

Temps d'execució: 17 ms

Com que veiem que el camí de la solució és diferent al del camí òptim de l'algorisme A* amb l'heuristica admissible, determinem que el camí no és òptim respecte al temps.

A continuació repetirem les proves anteriors, però en comptes de fer servir el primer mapa(proporcionat per el propi enunciat de la pràctica), ho executarem amb el segon mapa que hem creat(Matriz2).

<u>Prova heuristica amb la distancia de Manhattan:</u>

BestFirst (anar del punt [0][0] al [9][9]):

Numero d'iteracions: 35

Camí solució: [(0,0), (1,0), (2,0), (3,0), (4,0), (4,1), (4,2), (4,3), (5,3), (5,4), (5,5), (4,5), (4,6), (4,7), (5,7), (6,7), (7,7), (8,7), (9,7), (9,8)]

Temps d'execució: 6 ms

Per comprovar si la solució és la òptima farem com en les proves del mapa anterior, comprovant el valor que ens dóna l'algorisme A* amb la heuristica B. En aquest cas podem comprovar que els resultats són diferents, per tant determinem que el camí no és l'òptim.

A(anar del punt [0][0] al [9][9]):*

Número d'iteracions: 32

Camí solució: [(0,0), (1,0), (1,1), (2,1), (3,1), (3,2), (4,2), (4,3), (5,3), (5,4), (5,5), (4,5), (4,6), (4,7), (5,7), (6,7), (6,8), (7,8), (8,8), (9,8)]

Temps d'execució: 11 ms

Per comprovar que l'algorisme no és òptim hem fet el mateix que fins ara, comparant el resultat amb el que dóna òptim.

Prova heuristica amb la diferència d’alçades:

BestFirst (anar del punt [0][0] al [9][9]):

Número d’iteracions: 70

Camí solució: [(0,0), (1,0), (1,1), (2,1), (3,1), (3,2), (4,2), (4,3), (5,3), (5,4), (5,5), (4,5), (4,6), (4,7), (5,7), (6,7), (6,8), (7,8), (8,8), (8,9)]

Temps d’execució: 13 ms

En aquest cas, a diferència del mapa anterior, quan executem aquesta heuristica admissible amb el BestFirst no dóna el mateix valor que amb el A*. Per tant el valor que ens dóna en aquest cas no és òptim respecte el temps.

A(anar del punt [0][0] al [9][9]):*

Número d’iteracions: 70

Camí solució:[(0,0), (1,0), (1,1), (2,1), (3,1), (3,2), (3,3), (2,3), (2,4), (2,5), (3,5), (4,5), (4,6), (4,7), (5,7), (6,7), (6,8), (7,8), (8,8), (9,8)]

Temps execució: 16 ms

En aquest cas com que estem utilitzant una heuristica que és admissible en el temps determinem que el resultat que ens dóna l’algorisme A* és òptim respecte al temps.

Prova heuristica de la distància euclidiana:

BestFirst (anar del punt [0][0] al [9][9]):

Número d’iteracions: 30

Camí solució: [(0,0), (1,0), (2,0), (3,0), (4,0), (4,1), (4,2), (4,3), (5,3), (5,4), (5,5), (4,5), (4,6), (4,7), (5,7), (6,7), (7,7), (8,7), (8,8), (9,8)]

Temps d’execució: 8 ms

Altre cop podem veure que la solució és diferent al valor proporcionat per l’algorisme A* amb heuristica admissible, per tant no és òptima respecte el temps.

A(anar del punt [0][0] al [9][9]):*

Número d'iteracions: 30

Camí solució: [(0,0), (1,0), (1,1), (2,1), (3,1), (3,2), (4,2), (4,3), (5,3), (5,4), (5,5), (4,5), (4,6), (4,7), (5,7), (6,7), (7,7), (7,8), (8,8), (9,8)]

Temps d'execució: 14 ms

En aquest cas tampoc és òptim respecte el temps ja que el camí proporcionat no coincideix amb el que ens proporciona l'algorisme A* amb la heuristica admissible.

• Per a cada heurística que heu dissenyat, hauríeu trobat una solució si haguéssiu aplicat l'algorisme hill climbing? No cal implementar l'algorisme hill climbing només justificar-ho.

Per saber si una heuristica permet que l'algorisme Hill Climbing trobi o no un resultat esperat el que s'ha de fer és verificar si la heuristica decreix de forma monòtona segons s'executa l'algorisme.

Cas de la heuristica de la distància de Manhattan:

Creiem que el cas de la distància de Manhattan no hauria de perquè trobar la solució, ja que si que és veritat que l'algorisme pren decisions de forma que la distància es va reduïnt, però segons s'executa l'algorisme, podria donar-se el cas de que trobem alguna casella invàlida que ens faci canviar la ruta i que l'algorisme ens faci anar per llocs que en comptes de fer reduïr la distància ens la faci augmentar.

Cas de l'heuristica de la diferència entre alçades:

En aquest cas ens trobem que tampoc tenim perquè trobar la solució, ja que en aquest cas mirem la diferència entre alçades del mapa segons anem prenent decisions sobre quin camí agafar. En aquest cas, donat que la matriu és gran, és fàcil pensar que en algun moment no poguem trobar un valor que de casella que tingui aquest decreixement monòton.

Cas de l'heuristica de la distància Euclidiana:

En aquest cas crec que tampoc hauriem de perquè trobar la solució, ja que ens trobem en el mateix cas que la distància de Manhattan, podria ser que ens haguéssim de desviar degut als desnivells pels que no podem passar per la matriu i això eviti que els valors heuristics decreixin de forma monòtona tal i com es desitja.