

Problema 8-Puzzle

Tema Căutare - Inteligență Artificială

Nicoi Alexandru

Grupa 353

Facultatea de Matematica-Informatica

Universitatea din Bucuresti

Cuprins

1	Euristica 1 admisibilă	2
2	Euristica 2 admisibilă	2
3	Euristica 3 neadmisibilă	2
4	Statistica timp - generare soluție	3
5	Concluzii	3

1 Euristica 1 admisibilă

Prima euristică admisibilă o considerăm numărarea variabilelor care nu îndeplinesc cerința scop a problemei. Cerința scop este reprezentată de faptul că oricare ar fi elementul din matrice, elementele din stânga și deasupra trebuie să fie mai mic decât el.

Considerăm următoarea stare

1 7 4
2 6 8
3 0 5

Aplicând soluția prezentată, parcurgem pe coloane și obținem:

$$0 + 0 + 1 + 0 + 1 + 0 + 0 + 1 + 1 = 4$$

Pentru a ajunge la starea

1 4 7
2 5 8
3 6 0

Costul real este 5

2 Euristica 2 admisibilă

A doua euristică admisibilă pleacă de la ideea de mai sus, dar trebuie luat în considerare și constrângerea legată de distanța manhattan a lui 0 până la starea finală, care este știută ca ultima poziție din matrice (linia 2 - coloana 2 - indexare de la 0)

Considerăm starea anterioară și obținem:

$$0 + 0 + 1 + 0 + 1 + 0 + 0 + 1 + 1 + 1 = 5$$

Pentru a ajunge la starea

1 4 7
2 5 8
3 6 0

Costul real este 5

3 Euristica 3 neadmisibilă

O propunere de euristică neadmisibilă pleacă de la verificarea condiției unei plăcuțe, cum am spus și mai sus, iar pentru a aproxima, vom face următoarea sumă

$$\sum \sqrt{placutaStanga_n^2 + placutaSus_n^2}$$

Pe exemplul de mai sus, calculul iese evident peste costul real, rezultatul fiind 28.98 29. Este evident că această euristică este neadmisibilă

4 Statistica timp - generare soluție

Exemplul 1

1 0 4
7 6 8
3 2 5

ALGORITM	TIMP(MS)	LUNGIME DRUM	COST
BF	557	7	6
DF	54947	16	16
DFI	206	7	6
UCS	2757	7	6
A* (Euristica 1)	11	7	6
A* (Euristica 2)	10	7	6
A* (Euristica 3)	9	11	11
Greedy (Euristica 1)	497	16	17
Greedy (Euristica 2)	521	16	17
Greedy (Euristica 3)	49	12	12

Exemplul 2

1 7 4
2 6 8
3 0 5

ALGORITM	TIMP(MS)	LUNGIME DRUM	COST
BF	200	6	5
DF	12914	18	18
DFI	79	6	5
UCS	351	6	5
A* (Euristica 1)	9	6	5
A* (Euristica 2)	9	6	5
A* (Euristica 3)	2	6	5
Greedy (Euristica 1)	104	13	13
Greedy (Euristica 2)	104	13	13
Greedy (Euristica 3)	2	6	5

5 Concluzii

Algoritmul BF returnează întotdeauna cel mai scurt drum, fiind util când soluția nu este la un nivel foarte adânc în graf.

Algoritmul DF are dezavantajul că durează foarte mult, și nici nu ne returnează un drum scurt.

Algoritmul DFI combină cei doi algoritmi de mai sus, astfel că se aplică DF până la o anumită adâncime, de unde vom scoate cel mai scurt drum.

Algoritmul UCS are dezavantajul că se bazează doar pe costul nodurilor, astfel că timpul este mai mare din cauza tendinței de a urma nodul cu costul cel mai mic.

Algoritmul A^* este cel mai eficient pentru problema 8-puzzle, scoțând un timp foarte bun și un drum de cost scăzut. Acesta se bazează pe euristicele calculate, astfel că o euristică eficient aplicată va eficientiza procesul de căutare.

Algoritmul Greedy, la fel ca A^* , se bazează pe euristici, însă nu este la fel de optim, deoarece acesta nu compară și în funcție de costul calculat până la el.