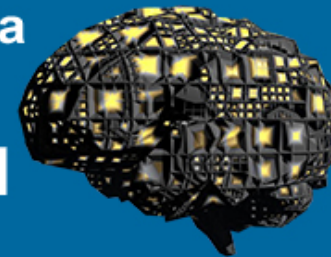


Datorzinātnes un informācijas tehnoloģijas fakultāte

Sistēmu teorijas un projektēšanas katedra

MĀKSLĪGĀ INTELEKTA PAMATI



3. Modulis "Informētas pārmeklēšanas stratēģijas stāvokļu telpā"

3.2. Tēma

Heiristiski informēti pārmeklēšanas algoritmi

Dr.habil.sc.ing., profesors **Jānis Grundspenķis**, Dr.sc.ing., lektore **Alla Anohina**

Sistēmu teorijas un projektēšanas katedra

Datorzinātnes un informācijas tehnoloģijas fakultāte

Rīgas Tehniskā universitāte

E-pasts: {janis.grundspenkis, alla.anohina}@rtu.lv

Kontaktadrese: Meža iela 1/4- {550, 545}, Rīga, Latvija, LV-1048

Tālrunis: (+371) 67089{581, 595}

Tēmas mērķi un uzdevumi

Tēmas mērķis ir sniegt zināšanas par trim heiristiski informētiem stāvokļu telpas pārmeklēšanas algoritmiem un prasmes to realizācijā.

Pēc šīs tēmas apgūšanas Jūs:

- zināsiet kalnā kāpšanas, vislabākā stāvokļa meklēšanas un starveida pārmeklēšanas algoritmu darbību un to raksturojumus;
- būsiet spējīgi realizēt kalnā kāpšanas, vislabākā stāvokļa meklēšanas un starveida pārmeklēšanas algoritmus, lietojot sarakstus Open un Closed.

Heiristiski informēti pārmeklēšanas algoritmi

Heiristiski informētus pārmeklēšanas algoritmus realizē, izmantojot jau apskatītos sarakstus OPEN un CLOSED.

Ar sarakstiem ir saistītas vairākas svarīgas lietas:

- Abos sarakstos stāvokļus ievieto kopā ar heiristiskiem mēriem
- Sarakstā OPEN stāvokļus sakārto atbilstoši heiristiskiem mēriem

Kalnā kāpšanas pārmeklēšanas stratēģija (1)

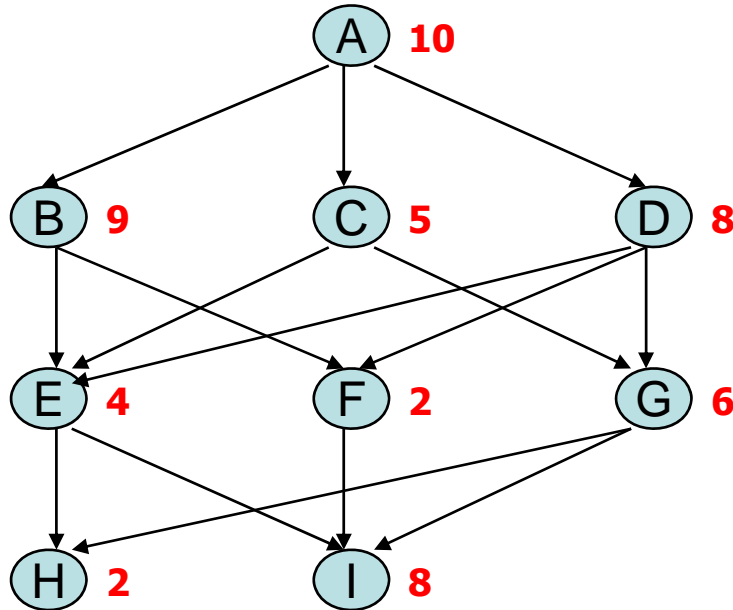
Kalnā kāpšanas algoritms izvērš tekošo virsotni, kas atrodas sarakstā OPEN, un novērtē tās tiešos pēctečus. Pamatojoties uz heiristiskiem vērtējumiem, labāko no šiem pēctečiem izvēlas tālākajai izvēršanai, bet pārējos aizmirst un nekad vairs neapskata, ievietojot tos sarakstā CLOSED. Pārmeklēšana apstājas, kad tā sasniedz stāvokli, kas ir labāks par jebkuru no tā tiešo pēcteču virsotnēm, vai tā sasniedz mērķa stāvokli, vai strupceļa virsotni, kas nav mērķis.

! Sarakstā OPEN vienmēr atrodas tikai viena (labākā) virsotne

Kalnā kāpšanas pārmeklēšanas stratēģija (2)



Piemērs:



6 Stāvokļa heuristisks vērtējums

Dotajā stāvokļu telpā labākais stāvoklis ir stāvoklis ar mazāko heuristisko vērtējumu

A- pārmeklēšanas sākuma stāvoklis

H- pārmeklēšanas mērķis

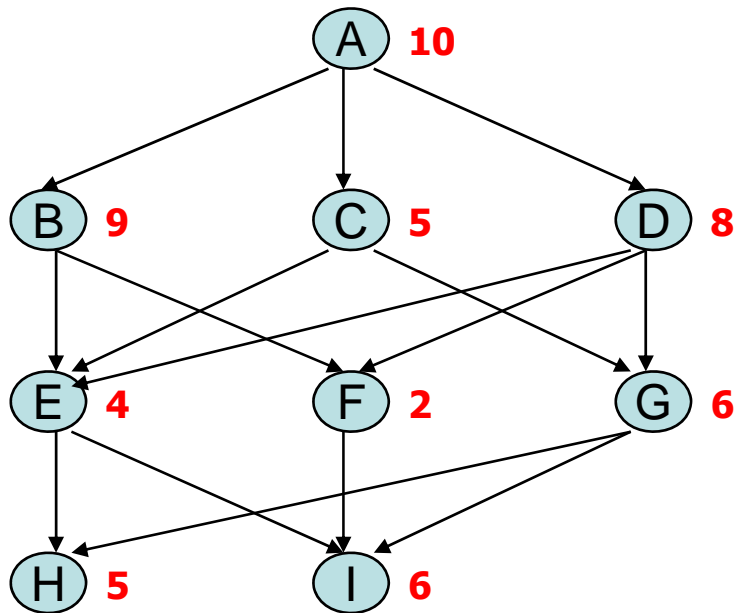
Iterācija	OPEN	CLOSED
0	A(10)	∅
1	C(5)	A(10) B(9) D(8)
2	E(4)	A(10) B(9) D(8) C(5) G(6)
3	H(2)	A(10) B(9) D(8) C(5) G(6) E(4) I(8)

Mērķis ir sasniegts

Kalnā kāpšanas pārmeklēšanas stratēģija (3)



Piemērs:



6 Stāvokļa heiristisks vērtējums

Dotajā stāvokļu telpā labākais stāvoklis ir stāvoklis ar mazāko heiristisko vērtējumu

A- pārmeklēšanas sākuma stāvoklis

H- pārmeklēšanas mērķis

Iterācija	OPEN	CLOSED
0	A(10)	∅
1	C(5)	A(10) B(9) D(8)
2	E(4)	A(10) B(9) D(8) C(5) G(6)

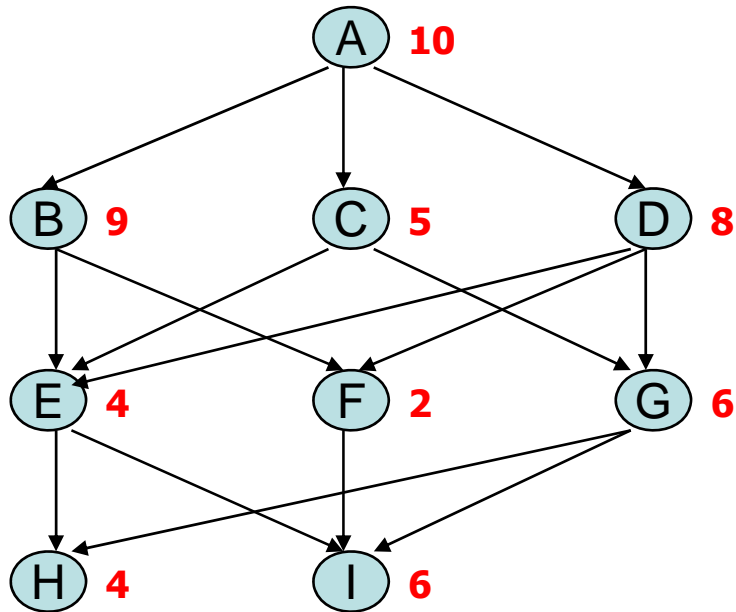
Mērķis nav sasniegts

Algoritms visu laiku virzās uz stāvokli ar mazāku vērtējumu. Taču tas apstājas, nerasniedzot mērķi, jo starp E virsotnes pēctečiem nav pēcteča ar mazāku vērtējumu, nekā stāvoklim E. Iemesls šādai situācijai ir slikti izvēlēta heiristika, jo, ja heiristika ir pareiza, algoritms taisnā ceļā noved pie mērķa.

Kalnā kāpšanas pārmeklēšanas stratēģija (4)



Piemērs:



6 Stāvokļa heiristisks vērtējums

Dotajā stāvokļu telpā labākais stāvoklis ir stāvoklis ar mazāko heiristisko vērtējumu

A- pārmeklēšanas sākuma stāvoklis

H- pārmeklēšanas mērķis

Iterācija	OPEN	CLOSED
0	A(10)	∅
1	C(5)	A(10) B(9) D(8)
2	E(4)	A(10) B(9) D(8) C(5) G(6)

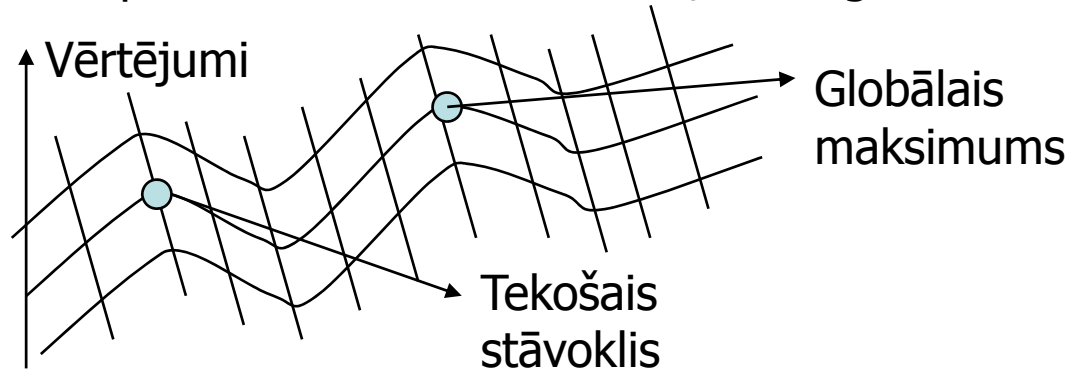
Mērķis nav sasniegts

Algoritms visu laiku virzās uz stāvokli ar mazāku vērtējumu. Taču tas apstājas, nerasniedzot mērķi, jo starp E virsotnes pēctečiem nav pēcteča ar mazāku vērtējumu, nekā stāvoklim E. Atšķirībā no iepriekšējā piemēra virsotnei H ir tāds pats vērtējums kā stāvoklim E, taču tas nav labākais vērtējums. Iemesls šādai situācijai ir slikti izvēlēta heiristika, jo, ja heiristika ir pareiza, algoritms taisnā ceļā noved pie mērķa.

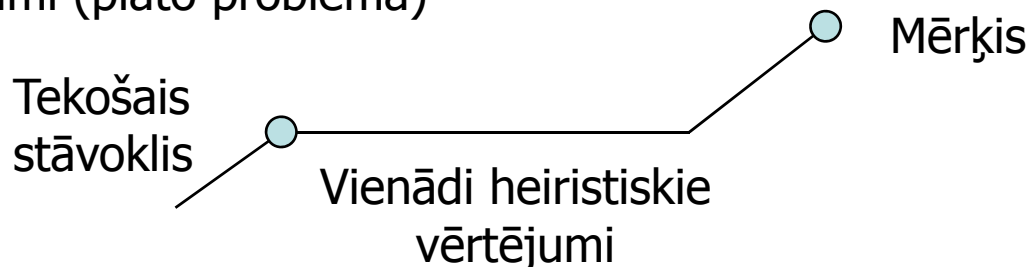
Kalnā kāpšanas pārmeklēšanas stratēģija (5)

Ar šo stratēģiju ir saistītas vairākas problēmas:

- Maldīga heiristika var novest uz dziļu vai pat bezgalīgu ceļu, kas nedod rezultātu. Šis algoritms nesaglabā pārmeklēšanas ceļu un līdz ar to nevar atgūties no kļūdām, jo nerealizē atkāpšanos.
- Algoritms var apstāties lokālā maksimumā, nevis globālā maksimumā



- Algoritms nespēj turpināt pārmeklēšanu, ja visiem kaimiņiem ir vienādi vērtējumi (plato problēma)



Kalnā kāpšanas pārmeklēšanas stratēģija (6)

Tādējādi, kalnā kāpšanas stratēģija negarantē mērķa sasniegšanu. Taču par spīti ierobežojumiem kalnā kāpšanas stratēģiju var efektīvi izmantot, ja heuristika ir pietiekami informatīva.

Turklāt, šī pārmeklēšana taupa datora atmiņas resursus. Tā ir piemērota stāvokļu telpām ar lielu zarošanās koeficientu.

Vislabākā stāvokļa meklēšana (1)

Katrā iterācijā vislabākā stāvokļa meklēšanas algoritms izvēlas vislabāko stāvokli no līdz šim atrastiem, neatkarīgi no tā, kur šis stāvoklis atrodas stāvokļu telpā. Algoritms saglabā visus līdz šim izvērstos stāvokļus. Tas beidz darbību, kad mērķis ir atrasts, vai ir pārmeklēta visa stāvokļu telpa, bet mērķis nav atrasts.

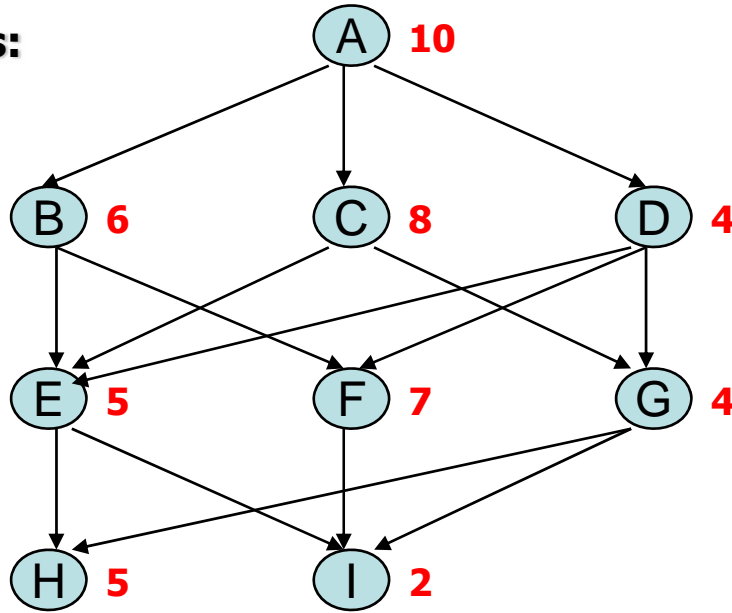
Algoritma raksturojumi:

- Algoritms vienmēr atrod mērķi, ja vien tāds eksistē stāvokļu telpā
- Pārslēdzas no viena zara uz citu
- Saraksts OPEN tiek realizēts kā prioritātes rinda, t.i., šajā sarakstā stāvokļi tiek sakārtoti atbilstoši heuristiskiem mēriem
- Stāvokļu izvēle notiek no OPEN saraksta kreisās puses

Vislabākā stāvokļa meklēšana (2)



Piemērs:



6 Stāvokļa heiristisks vērtējums

Dotajā stāvokļu telpā labākais stāvoklis ir stāvoklis ar lielāko heiristisko vērtējumu

A- pārmeklēšanas sākuma stāvoklis

H- pārmeklēšanas mērķis

Iterācija	OPEN	CLOSED
0	A(10)	∅
1	C(8) B(6) D(4)	A(10)
2	B(6) E(5) D(4) G(4)	A(10) C(8)
3	F(7) E(5) D(4) G(4)	A(10) C(8) B(6)
4	E(5) D(4) G(4) I(2)	A(10) C(8) B(6) F(7)
5	H(5) D(4) G(4) I(2)	A(10) C(8) B(6) F(7) E(5)

Mērķis ir sasniegts

Starveida pārmeklēšana (1)

Starveida pārmeklēšana pārmeklē stāvokļu telpu līmeni pa līmenim. Taču katrā līmenī algoritms pārmeklē tikai ierobežotu skaitu labāko stāvokļu. Šo skaitu nosauc par ***stara platumu***, un to uzdod algoritma darbības sākumā.

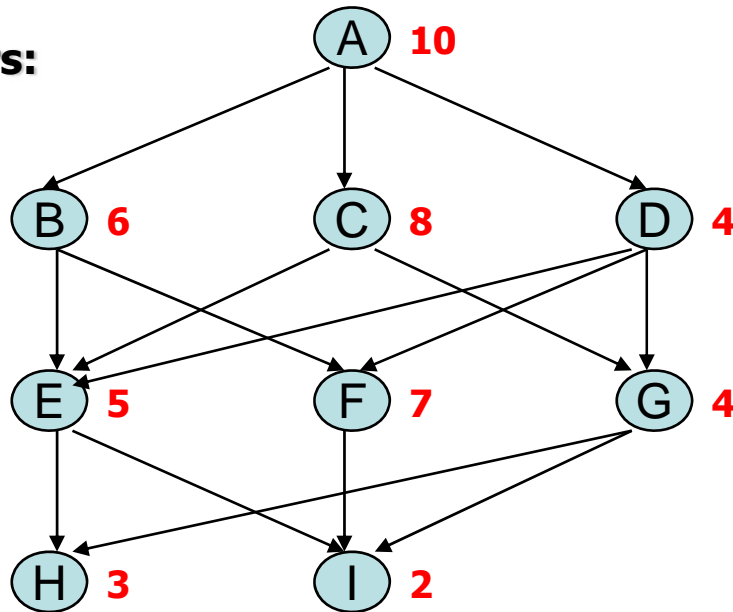
Svarīgas lietas:

- OPEN saraksts vienmēr satur tik daudz virsotņu, cik ir stara platums, izņemot sākuma iterāciju un gadījumu, kad apskatāmo pēcteču skaits ir mazāks par stara platumu
- Apskata visu OPEN sarakstā esošo virsotņu tiešus pēctečus un no tiem izvēlas labākos, bet pārējos ievieto sarakstā CLOSED

Starveida pārmeklēšana (2)



Piemērs:



6 Stāvokļa heiristisks vērtējums

Dotajā stāvokļu telpā labākais stāvoklis ir stāvoklis ar lielāko heiristisko vērtējumu

A- pārmeklēšanas sākuma stāvoklis

H- pārmeklēšanas mērķis

Stara platums ir 2 virsotnes

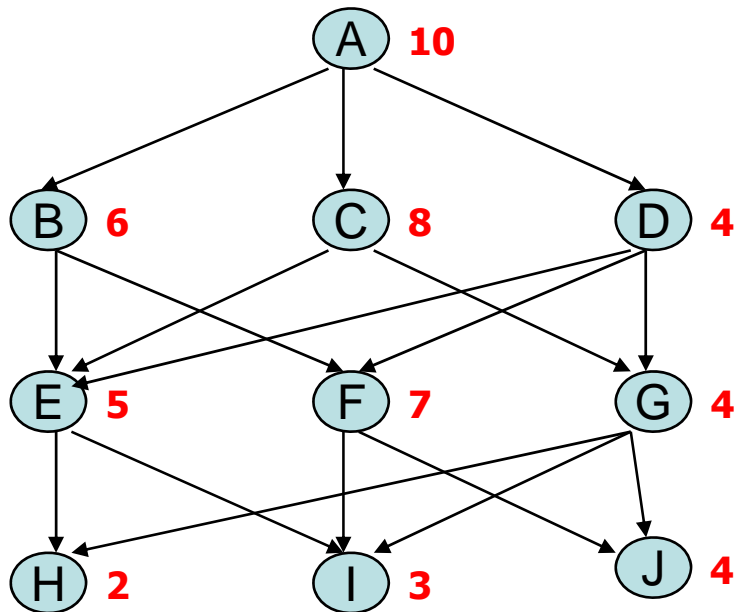
Iterācija	OPEN	CLOSED
0	A(10)	∅
1	C(8) B(6)	A(10) D(4)
2	F(7) E(5)	A(10) D(4) C(8) B(6) G(4)
3	H(3) I(2)	A(10) D(4) C(8) B(6) G(4) F(7) E(5)

Mērķis ir sasniegts

Starveida pārmeklēšana (3)



Piemērs:



6 Stāvokļa heiristisks vērtējums

Dotajā stāvokļu telpā labākais stāvoklis ir stāvoklis ar lielāko heiristisko vērtējumu

A- pārmeklēšanas sākuma stāvoklis

H- pārmeklēšanas mērķis

Stara platums ir 2 virsotnes

Iterācija	OPEN	CLOSED
0	A(10)	∅
1	C(8) B(6)	A(10) D(4)
2	F(7) E(5)	A(10) D(4) C(8) B(6) G(4)
3	J(4) I(3)	A(10) D(4) C(8) B(6) G(4) F(7) E(5) H(2)

Mērķis nav sasniegts

Neskatoties uz to, ka virsotne H tika apskatīta, kā virsotnes E pēctecis, tā tika ievietota sarakstā CLOSED, jo tās vērtējums ir sliktāks par virsotņu J un I vērtējumiem. Iemesls šādai situācijai ir slikti izvēlēta heiristika, jo, ja heiristika ir pareiza, algoritms taisnā ceļā noved pie mērķa.

Starveida pārmeklēšana (4)

Tādējādi, starveida pārmeklēšana negarantē mērķa sasniegšanu. Taču šī pārmeklēšana taupa datora atmiņas resursus un tā ir piemērota stāvokļu telpām ar lielu zarošanās koeficientu.