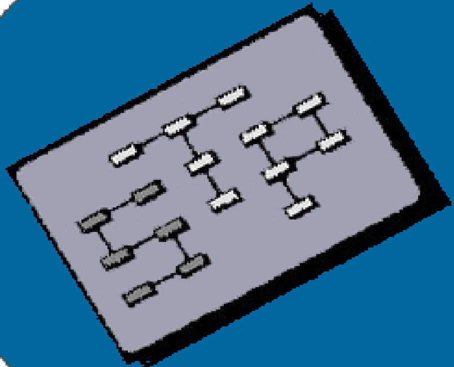
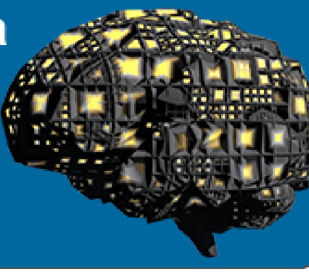


Datorzinātnes un informācijas tehnoloģijas fakultāte

Sistēmu teorijas un projektēšanas katedra

MĀKSLĪGĀ INTELEKTA PAMATI



3. Modulis "Informētas pārmeklēšanas stratēģijas stāvokļu telpā"

3.3. Tēma

Heiristikas efektivitātes vērtēšana

Dr.habil.sc.ing., profesors **Jānis Grundspeņķis**, 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 un prasmes heiristikas efektivitātes vērtēšanā un stāvokļu telpas sarežģītības noteikšanā.

Pēc šīs tēmas apgūšanas Jūs:

- zināsiet trīs dimensijas, pēc kurām vērtē heiristikas efektivitāti;
- būsiet spējīgi noteikt pieļaujamu heiristiku;
- būsiet spējīgi noteikt monotonu heiristiku;
- būsiet spējīgi noteikt informētāku heiristiku;
- pratīsiet novērtēt stāvokļu telpas sarežģītību;
- zināsiet metodes stāvokļu telpas sarežģītības samazināšanai.

Heiristikas efektivitātes vērtēšana

Ja vienā un tajā pašā stāvokļu telpā var definēt vairākas heiristikas, vai kāda no tām nevar būt labāka jeb efektīvāka? Šis jautājums nosaka nepieciešamību novērtēt heiristikas efektivitāti.

Heiristikas efektivitāti vērtē pēc 3 dimensijām:

- Heiristikas pieļaujamība
- Heiristikas monotonitāte
- Heiristikas informētība

Heiristikas pieļaujamība (1)

Daudzos lietojumos ir svarīgi ne tikai sasniegt mērķi, bet sasniegt to pa īsāko ceļu, vai ātrāk, vai ar minimālām izmaksām. Tas attiecas uz lietojumiem, kuros katrs papildus problēmas risināšanas solis maksā pārāk dārgi. Piemērs tam ir robots, kurš darbojas kādā bīstamā vidē, un kuram ir jāplāno ceļš caur šo vidi.

Pārmeklēšanas algoritms ir pieļaujams, ja tas garantē īsākā ceļa līdz mērķim atrašanu, ja vien šāds ceļš eksistē. Pārmeklēšana plašumā ir pieļaujams algoritms. Tas sākumā apskata visus stāvokļu n-tajā stāvokļu telpas līmenī un tikai tad pāriet uz n+1 līmeni. Tādējādi, mērķis tiek sasniegts pa īsāko ceļu.

Taču, kā var noteikt, ka heiristisks algoritms ir pieļaujams?

Heiristikas pieļaujamība (2)

Izmantojot novērtējuma funkciju $f(n)=g(n) + h(n)$, var raksturot pieļaujamo heiristiskas pārmeklēšanas stratēģiju klasi.

Citādi, funkciju $f(n)$ var aprakstīt šādi:

$$\mathbf{f(n) = g(n) + h(n),}$$

kur

$g(n)$ – attālums no sākuma virsotnes līdz virsotnei n, jeb līmenis, kurā n-tā virsotne atrodas stāvokļu telpā, ja tā ir pareiza hierarhija

$h(n)$ – heiristisks novērtējums attālumam no n-tās virsotnes līdz mērķim

$f(n)$ – kopējās izmaksas ceļam no sākuma stāvokļa līdz mērķim caur n-to stāvokli

Heiristikas pieļaujamība (3)

Izmantojot heiristisku novērtējuma funkciju $f(n) = g(n) + h(n)$ kopā ar vislabākā stāvokļa meklēšanas algoritmu, tiek iegūts ***algoritms A***.

funkcija f(n)

+

Vislabākā stāvokļa meklēšana

=

algoritms A

Izrādās, ka algoritms A būs pieļaujams tikai tad, ja jebkurā stāvokļu telpā tas vienmēr beidzas ar optimāla atrisinājuma ceļa atrašanu, ja vien ceļš no sākuma stāvokļa uz mērķi vispār eksistē.

Heiristikas pieļaujamība (4)

Tādējādi, lai noteiktu pieļaujamu heiristiku, ir jādefinē novērtējuma funkcija f^* :

$$\mathbf{f^* (n) = g^* (n) + h^* (n)}$$

kur

$g^*(n)$ – īsākā ceļa izmaksas no sākuma virsotnes uz n-to virsotni

$h^*(n)$ – faktiskās izmaksas īsākajam ceļam no n-tās virsotnes uz mērķi

$f^*(n)$ – faktiskās izmaksas optimālam ceļam no sākuma virsotnes uz mērķi caur n-to virsotni

Heiristikas pieļaujamība (5)

Izmantojot heiristisku novērtējuma funkciju $f^* (n) = g^* (n) + h^* (n)$ kopā ar vislabākā stāvokļa meklēšanas algoritmu, tiek iegūta pieļaujama pārmeklēšanas stratēģija.

funkcija f*

+

Vislabākā stāvokļa meklēšana

=

Pieļaujama pārmeklēšanas stratēģija

Tas ir tāpēc, ka katrs stāvoklis tiek vērtēts saskaņā ar to, vai tas pieder īsākajam ceļam, vai nē. Savukārt, vislabākā stāvokļa meklēšana ņem labāko stāvokli no visiem līdz šim apskatītajiem stāvokļiem un noteikti tā izvēlēsies stāvokli, kurš pieder īsākajam ceļam.

Heiristikas pieļaujamība (6)

Diemžēl, funkcija f* lielākai daļai reālu problēmu neeksistē, tāpēc reāli algoritmā A

$g(n) \approx g^*(n)$, ja algoritms virzās pa optimālu ceļu

$g(n) > g^*(n)$, ja algoritms nevirzās pa optimālu ceļu

Parasti nav iespējams arī izskaitļot $h^*(n)$, bet bieži ir iespējams noteikt vai $h(n)$ ir ierobežots no augšas ar $h^*(n)$, t.i. $h(n) < h^*(n)$.

Algoritmā A, lietojot novērtējuma funkciju f, kurā $h(n) \leq h^*(n)$, tiek iegūts ***algoritms A****.*

Visi algoritmi A* ir pieļaujami.

Heiristikas monotonitāte (1)

Atgriežoties pie algoritma A*, var redzēt, ka tas neprasa, lai g(n) būtu vienāds ar g*(n). Tas nozīmē, ka pieļaujama heiristika sākotnēji var sasniegt stāvokļus, neejot pa optimālu ceļu, bet beigu beigās atradīs optimālu ceļu uz visiem stāvokļiem, kas atrodas ceļā uz mērķi. Šajā sakarā loģisks ir jautājums, vai eksistē "lokāli pieļaujama" heiristika jeb heiristika, kas ļauj secīgi atrast īsāko ceļu uz katru no stāvokļiem pārmeklēšanas gaitā? Tātad, pārmeklēšanas gaitā tiek atvērti daudzi stāvokļi. Vai ir kāda garantija, ka vienu reizi atklātais stāvoklis netiks sasniegts vēlreiz, bet jau pa īsāku ceļu? Uz šo jautājumu atbildi dod monotonitātes īpašība.

Heiristikas monotonitāte (2)

Heiristiska funkcija h ir monotona, ja:

a) Visiem stāvokļiem n_i un n_j, kur n_j ir n_i pēctecis

$$\mathbf{h(n_i) - h\ (n_j) \leq izmaksas\ (n_i,n_j)},$$

kur izmaksas (n_i,n_j) ir faktiskās izmaksas, ejot no n_i uz n_j

b) Mērķa stāvokļa heiristiskais novērtējums ir 0

$$\mathbf{h(mērķis) = 0}$$

Heiristikas monotonitāte (3)

Viens no veidiem kā aprakstīt monotonitātes īpašību ir tas, ka pārmeklēšanas telpa visur ir lokāli saskaņota ar lietoto heiristiku. Atšķirība starp stāvokļa heiristisku novērtējumu un jebkura šī stāvokļa pēcteča heiristisku novērtējumu ir ierobežota ar faktiskām izmaksām, lai nokļūtu no šī stāvokļa uz pēcteci. Tātad, heiristika ir visur pieļaujama, sasniedzot katru stāvokli pa īsāko ceļu no tā priekšteča.

Ja stāvokļu telpas pārmeklēšanai izmanto vislabākā stāvokļa meklēšanu ar monotonu heiristiku, tad atklājot stāvokli vēlreiz nevajag pārbaudīt, vai tas ir sasniegts pa īsāko ceļu.

Jebkura monotona heiristika ir pieļaujama.

Heiristikas monotonitāte (4)

Pierādīsim, ka jebkura monotona heiristika ir pieļaujama. Pieņemsim, ka stāvokļu telpā ir ceļš S_1, S_2, S_3, ..., S_g, kur S_1 ir sākuma stāvoklis un S_g ir mērķis.

Apskatīsim soļu secību:

S_1 -> S_2	$h(S_1) - h(S_2) <$ izmaksas(S_1, S_2) saskaņā ar monotonitātes īpašību
S_2 -> S_3	$h(S_2) - h(S_3) <$ izmaksas(S_2, S_3) saskaņā ar monotonitātes īpašību
S_3 -> S_4	$h(S_3) - h(S_4) <$ izmaksas(S_3, S_4) saskaņā ar monotonitātes īpašību
...	...
S_{g-1} -> S_g	$h(S_{g-1}) - h(S_g) <$ izmaksas(S_{g-1}, S_g) saskaņā ar monotonitātes īpašību

Summējot katru kolonu un ņemot vērā, ka $h(S_g)=0$ iegūsim:

S_1 -> S_g	$h(S_1) - h(S_g) <$ izmaksas(S_1, S_g)

Tas nozīmē, ka monotona heiristika h ir heiristika A* un tā ir pieļaujama.

Heiristikas informētība (1)

Ja ir divas heiristikas, kas sameklē īsāko ceļu uz mērķi, vai kāda no tām nevar būt labāka par citu? Uz šo jautājumu atbild heiristikas informētības īpašība.

Ja abi algoritmi ir algoritmi A*, un tie lieto heiristikas h_1 un h_2, tad ja $h_1(n) \leq h_2(n)$ jebkuram stāvoklim n, tad ***h_2 ir informētākā par h_1***.

Ja heiristika h_2 ir informētākā par heiristiku h_1, tad stāvokļu kopa, kuru pārmeklē, izmantojot h_2, ir apakškopa stāvokļu kopai, ko pārmeklē ar h_1.

Heiristikas informētība (2)

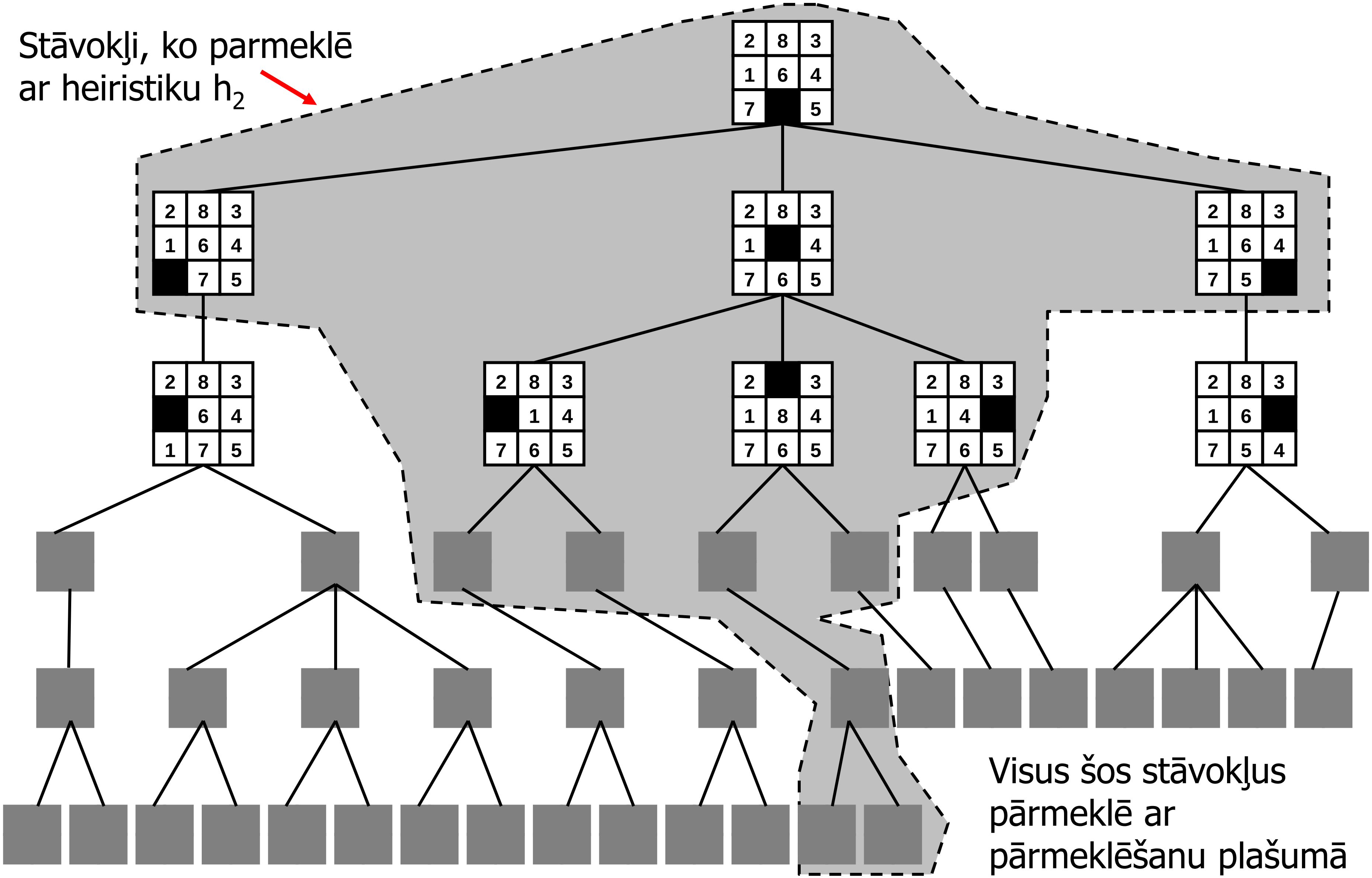
Informētākas heiristikas definīcijas patiesumu var demonstrēt ar spēles ar 8 kauliņiem piemēru. Lai pārmeklētu šīs spēles stāvokļu telpu, izmantosim divus algoritmus: pārmeklēšanu plašumā un algoritmu ar kādu heiristiku.

Varam teikt, ka pārmeklēšana plašumā ir algoritms A* ar heiristiku $h_1(n)=0$ visiem stāvokļiem n. Ir skaidrs, ka $h<h^*$.

Pieņemsim, ka otrs algoritms lieto heiristiku h_2, t.i. kauliņu skaitu, kas atrodas nepareizās vietās. Pēc A* algoritma definīcijas $h_2 \leq h^*$.

No visa aprakstītā seko, ka $h_1 \leq h_2 \leq h^*$. Tādējādi, pārmeklēšana plašumā ir mazāk informēta. Kaut gan abi algoritmi atrod optimālo ceļu līdz mērķim, otrais algoritms (ar heiristiku h_2) pārmeklē mazāk stāvokļu.

Heiristikas informētība (3)



Stāvokļu telpas sarežģītības vērtējums (1)

Stāvokļu telpas sarežģītība ir saistīta ar to, ka augot stāvokļu telpas līmeņu skaitam un telpai zarojoties, pārmeklējamo stāvokļu skaits aug eksponenciāli.

Stāvokļu telpas sarežģītība ir atkarīga no diviem faktoriem:

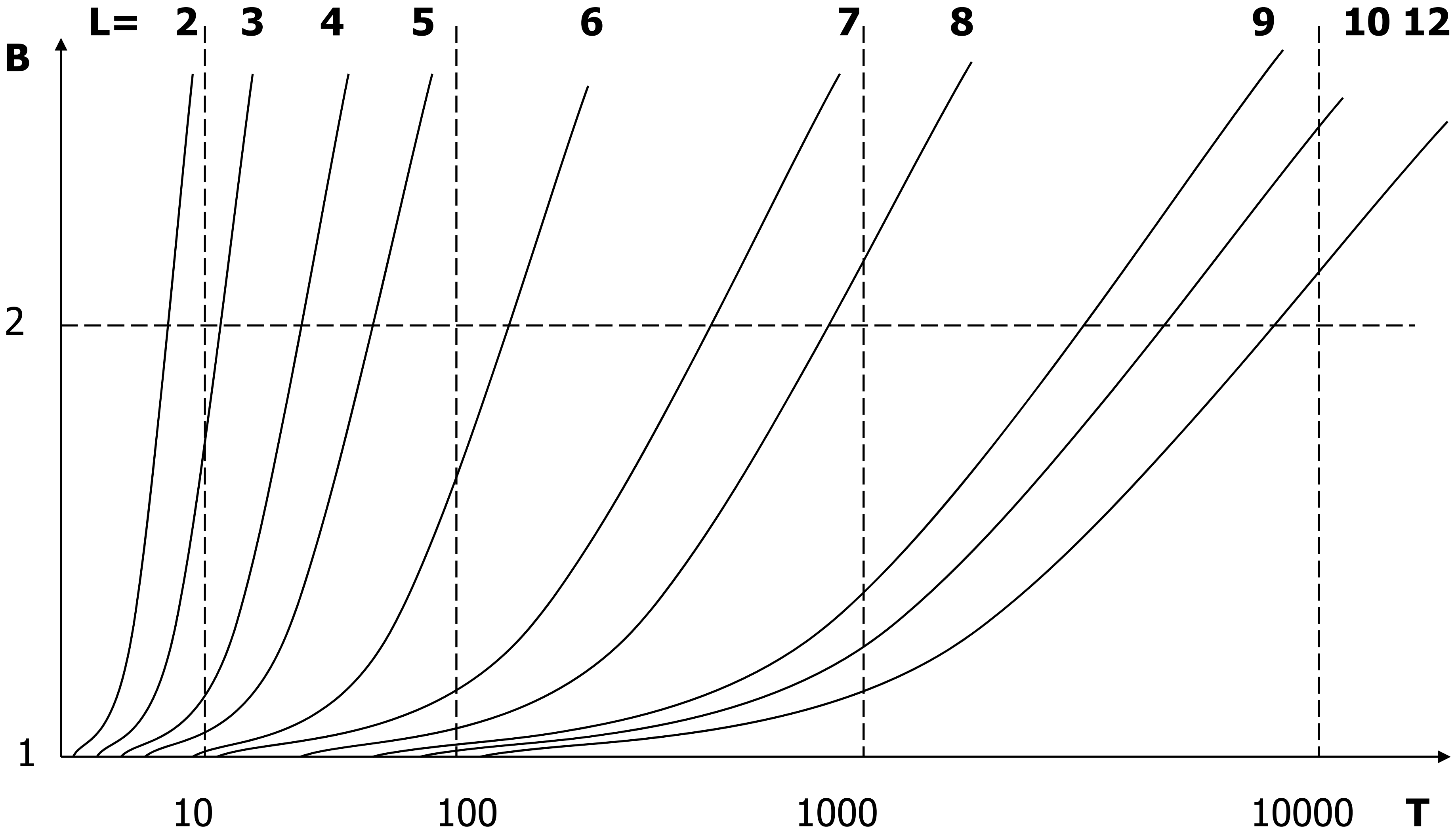
B – vidējā zarošanās koeficienta

L – līmeņu skaita stāvokļu telpā jeb ceļa garuma

Tādējādi, kopējais pārmeklējamo stāvokļu skaits T ir

$$\mathbf{T = B + B^2 + B^3 + \ldots + B^L = \frac{B(\,B^L - 1\,)}{B - 1}}$$

Stāvokļu telpas sarežģītības vērtējums (2)



Stāvokļu telpas sarežģītības vērtējums (3)

Iepriekš dotais attēls atspoguļo stāvokļu telpas vidējā zarošanās koeficienta B, līmeņu skaita L un kopējā pārmeklējamo stāvokļu skaita T savstarpējo atkarību.

Ja vidējāis zarošanās koeficients ir 2 (tas nozīmē, ka lielakajai daļai virsotņu ir tikai 2 pēcteči), tad, lai apskatītu visus ceļus līdz 6. līmenim, būs jāapskata mazliet vairāk par 100 stāvokļiem. Taču, palielinot līmeņu skaitu divreiz (līdz 12 līmeņiem), pārmeklējamo stāvokļu skaits palielinās gandrīz 10 reizes (8190 stāvokļi).

Pārmeklēšanas plašumā resursu patēriņš

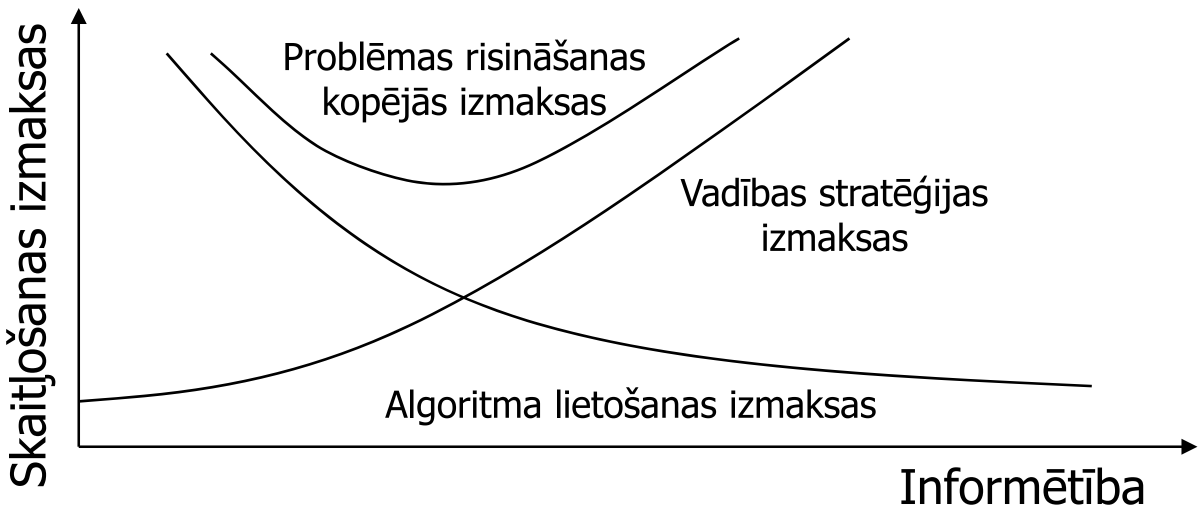
Laika un atmiņas resursu, kas ir vajadzīgi pārmeklēšanai plašumā, novērtējums (vidējais zarošanās koeficients B=10, 1000 virsotņu/sekundē, 100 baiti/virsotne)

Dziļums	Virsotnes	Laika patēriņš	Atmiņas patēriņš
0	1	1 milisekunde	100 baiti
2	111	1 sekunde	11 kilobaiti
4	11 111	11 sekundes	1 megabaits
6	10^6	18 minutes	111 megabaiti
8	10^8	31 stunda	11 gigabaiti
10	10^{10}	128 dienas	1 terabaits
12	10^{12}	35 gadi	111 terabaiti
14	10^{14}	3500 gadi	11 111 terabaiti

Rusels un Norvigs (Russell and Norvig, 1995)

Metodes stāvokļu telpas sarežģītības samazināšanai (1)

- Izmantot informētākas heiristikas



Attēlā var redzēt, ka, palielinoties heiristikas informētībai, algoritmam ir jāpārmeklē mazāk stāvokļu, bet tajā pašā laikā palielinās skaitļošanas izmaksas, jo sistēmai ir jārēķina heiristikā novērtējuma funkcijas vērtības. Problēmas risināšanas kopējās izmaksas ir heiristikas aprēķināšanas un stāvokļu telpas pārmeklēšanas izmaksas. Šīs izmaksas ir jāminimizē.

Metodes stāvokļu telpas sarežģītības samazināšanai (2)

- Transformēt stāvokļu telpu, piemēram, ņemot vērā simetriskuma pazīmi, kā to var darīt spēlē "krustiņi-nullītes", sākumā apskatot tikai 3 iespējamus gājienus: stūrī, centrā un sānu malā.