Optimistična heuristika

Neka je $h^*(s)$ prava cijena puta od *s* do ciljnog stanja. Za heuristiku *h* kažemo da je optimistična, ako nikada ne precjenjuje, odnosno ako vrijedi:

$$h(s) \leqslant h^*(s)$$

Kod algoritama pretraživanja, ovo svojstvo je izuzetno važno. Razmotrimo problem pronalaska najkraćeg puta od početnog stanja s_0 do konačnog stanja s_f. Neka algoritam razmatra čvor *n* koji predstavlja jedan konkretan put od početnog stanja s_0 do nekog stanja *s*. Uz oznake koje smo koristili na predavanju, ta je cijena puta (od s_0 do *s*) jednaka $g(s)$. Označimo li s $f(n)$ procjenu ukupne cijene puta čiji je prvi dio određen s *n* (kreće iz s_0 i ide do *s*) i dalje završava u s_f, ta će procjena biti:

$$f(n) = g(n) + h(s)$$

Označimo li sada s $f^*(n)$ pravu minimalnu cijenu puta čiji je prvi dio određen s *n* (kreće iz s_0 i ide do *s*) i dalje ide do konačnog stanja s_f, zbog optimističnosti heuristike (odnosno činjenice da h(s) neće iznositi više od stvarne minimalne cijene puta od *s* do s_f) imamo garanciju da je:

$$f(n) \leqslant f^*(n)$$

Drugim riječima, naša procjena duljine takvog puta sigurno neće biti veća od njegove stvarne duljine.

Zamislimo sada da smo napisali algoritam pretraživanja koji se ponaša slično algoritmu pretraživanja s jednolikom cijenom (*Uniform Cost Search*) samo što čvorove iz liste otvorenih čvorova ne vadi sortirano po cijeni puta od od s_0 do *s* (drugim riječima po $g(n)$), već čvorove vadi sortirano po procijeni ukupne cijene do ciljnog čvora (dakle po $g(n)+h(s)$). Uz pretpostavku da su cijene prijelaza iz čvora u čvor pozitivni brojevi, takav će algoritam nužno biti optimalan (pronaći će najkraći mogući put od početnog stanja s_0 do konačnog stanja s_f). Naime, zamislimo da do ciljnog čvora s_f postoji više puteva; označimo onaj najkraći čvorom n^* a neki dulji čvorom *n*'. Kako je tada:

$$f(n^*) < f(n')$$

algoritam pretraživanja nam garantira da će najprije razmotriti čvor n^*; no time će ustanoviti da taj čvor predstavlja put do ciljnog stanja, zaustavit će pretragu i vratiti taj čvor kao rezultat. Posljedica: ovakav algoritam nikada niti neće otvoriti niti jedan čvor *n* za koji je:

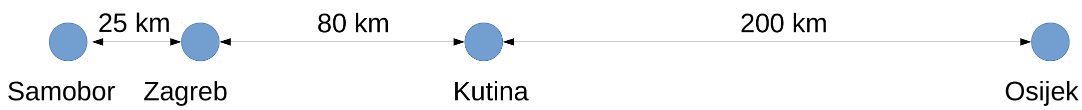
$$f(n) > f(n^*)$$

odnosno čija je cijena puta veća od stvarne minimalne cijene od početnog stanja s_0 do konačnog stanja s_f.

Na ovaj način može se dokazati optimalnost algoritma A*.

Konzistentna heuristika

Razmotrimo primjer duljine puta cestama kroz nekoliko gradova u Hrvatskoj.

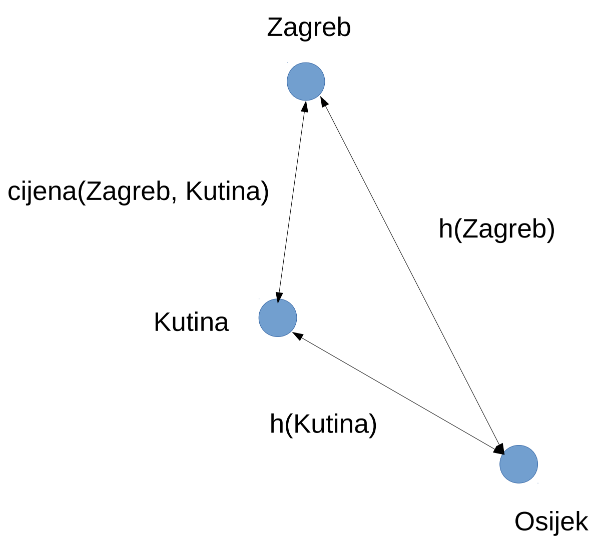


Na slici su prikazane udaljenosti između direktno povezanih gradova. Pretpostavite da razmatramo udaljenosti do grada Osijeka. Stvarna udaljenost od Kutine do Osijeka je 200 km; stoga za niti jednu optimističnu heuristiku ne može vrijediti *h*(Kutina) >200 km. Što možemo zaključiti o *h*(Zagreb)? Razmotrimo jedan specifičan slučaj: radimo s heuristikom koja upravo daje h(Kutina)=200 km. Stvarna udaljenost od Zagreba do Osijeka je 280 km, pa h(Zagreb) ne smije biti veća od toga, da bi bila optimistična. No u tom specifičnom slučaju slijedi ograda:

h(Zagreb) <= h(Kutina) + cijena(Zagreb, Kutina)

odnosno h(Zagreb) ne smije biti veća od 200+80=280.

Prethodni izraz općenito ne mora vrijediti. Primjerice, sasvim je legalno imati heurističku procjenu h(Kutina)=180 km te heurističku procjenu h(Zagreb)=270 km. Takva heuristika i dalje je optimistična (ne precjenjuje). Međutim, takva heuristika ne zadovoljava jedno od osnovnih svojstava geometrije: nejednakost trokuta koja zahtjeva da je izravan put od A do C uvijek kraći ili u najgorem slučaju jednako dugačak kao put kojim od A najprije odemo u B pa potom iz B nastavimo u C. Uz brojke koje smo dali, zaobilazni put je dugačak 80+180=260 km, a direktni 270 km!



Heuristiku koja zadovoljava nejednakost trokuta, odnosno, uz pretpostavku da je $(s_2,c)\in succ(s_1)$, heuristiku koja zadovoljava nejednakost:

$$h(s_1) \leq h(s_2) + c$$

nazivamo **konzistentnom heuristikom**.

Svojstva konzistentne heuristike

Lagano je pokazati **da je konzistentna heuristika nužno optimistična**. Krenite od ciljnog stanja s_f. $h(s_f)$ je po definiciji 0 (jer je s_f ciljno stanje). Pogledamo li sada stanje s koje je direktni prethodnik tog stanja, odnosno stanje za koje vrijedi: $(s_f,c)\in succ(s)$, uvjet konzistentnosti zahtjeva da je

$$h(s)\leqslant h(s_f)+c=0+c=c$$

što znači da heuristička funkcija ne može biti veća od stvarne cijene puta od s do s_f, ali tada slijedi da h(s) doista ne precjenjuje. Sada razmatranje možemo rekurzivno provesti na prethodnicima od s, pa na njihovim prethodnicima, pa na njihovim prethodnicima, … Lagano je vidjeti da je uslijed uvjeta konzistentnosti gornja ograda takve heuristike uvijek stvarna udaljenost do ciljnog stanja, pa takva heuristika nikada ne može precjenjivati čime je po definiciji optimistična.

Konzistentna heuristika također osigurava da je procjena cijene puta od početnog stanja kroz stanje *s* do ciljnog stanja s_f monotono rastuća. Naime, neka je $n_2\in expand(n_1)$ tj. neka iz čvora n_1 odlazimo u čvor n_2, i neka je $s_1=state(n_1)$ te $s_2=state(n_2)$ te neka je c cijena prijelaza iz s_1 u s_2. Krenuvši od uvjeta konzistentnosti heuristike:

$$h(s_1)\leqslant h(s_2)+c$$

lijevoj i desnoj strani možemo dodati $g(n_1)$:

$$g(n_1)+h(s_1)\leqslant g(n_1)+c+h(s_2)$$

Potom možemo uočiti da je $g(n_1)+c$ upravo cijena puta do n_2, pa možemo pisati:

$$g(n_1)+h(s_1)\leqslant g(n_2)+h(s_2)$$

Izraz s lijeve strane po definiciji je procjena cijene puta kroz n_1 a izraz s desne strane procjena cijene puta kroz n_2 pa vrijedi:

$$f(n_1)\leqslant f(n_2)$$

odnosno:

$$f(n_2)\geqslant f(n_1)$$

iz čega vidimo da, kako se od početnog stanja približavamo ciljnom stanju, procjena cijene ukupnog puta je sve veća i veća (ali je odozgo ograničena stvarnom cijenom). Stoga možemo zaključiti da kako napredujemo prema konačnom stanju, procjena cijene puta postaje sve bliža stvarnoj cijeni.

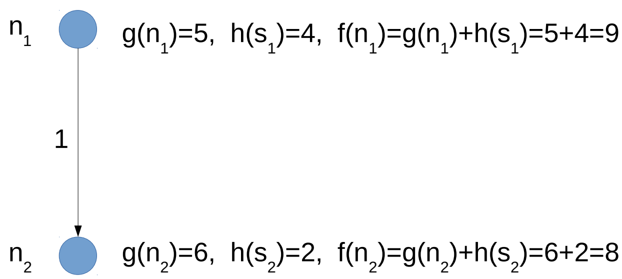
Važnost monotonosti funkcije f(.)

Ako je funkcija f(.) koju koristimo za procjenu ukupne cijene puta do ciljnog stanja monotona, algoritam pretraživanja koji čvorove iz liste otvorenih čvorova vadi sortirano po toj procjeni (počev od najjeftinijeg puta) ima jedno zgodno svojstvo: prvi čvor koji otvori za neko stanje s_i bit će ujedno i čvor koji predstavlja najjeftiniji put od početnog stanja do konačnog stanja a koji prolazi kroz to stanje s_i. Svaki sljedeći čvor koji eventualno generiramo a koji predstavlja put koji također prolazi kroz s_i bit bit će skuplji, pa se performanse algoritma pretraživanja koji radi uz monotonu funkciju f(.) mogu značajno popraviti uvođenjem liste zatvorenih stanja (nije potrebno pamtiti cjelokupne čvorove odnosno stanje plus cijenu).

Osvrnimo se još i na pitanje kako iznos funkcije f(.) utječe na performanse algoritma pretraživanja? Pokazuje se da što je iznos od f(.) bliži stvarnoj cijeni puta, algoritam će ukupno otvarati manje čvorova. Ovo se pak može preslikati na zahtjev da bi heuristička funkcija koju koristimo uvijek trebala imati maksimalni mogući iznos (a da ne precjenjuje). Što je iznos heurističke funkcije $h(s)$ manji u odnosu na stvarnu cijenu puta od s do s_f, to će algoritam otvarati više čvorova – *to smo pokazali na predavanjima*.

Pathmax-korekcija

Razmotrimo još jedan primjer optimistične heuristike ilustriran na slici u nastavku.



Neka je $n_2 \in \text{expand}(n_1)$ tj. neka iz čvora n_1 odlazimo u čvor n_2, i neka je $s_1=\text{state}(n_1)$ te $s_2=\text{state}(n_2)$ te neka je c cijena prijelaza iz s_1 u s_2 (koja na prikazanoj slici iznosi 1). Pretpostavimo da heuristika $h(.)$ je optimistična.

U prikazanom primjeru vidimo da heuristika $h(.)$ nije konzistentna. Posljedica toga je da funkcija $f(.)$ na putu od s_0 do s_f više nije monotona. U prikazanom primjeru procjena ukupne duljine puta koji je predstavljen čvorom n_1 jednaka je 9 ($f(n_1)=9$), a kako je heuristika optimistična, sigurno znamo da put do s_f nije jeftiniji od toga, ali možda je skuplji. Iz čvora n_1 možemo dakle zaključiti da je cijena puta predstavljenog čvorom n_1 pa dalje do s_f barem 9.

Nastavkom puta (odnosno odlaskom iz stanja s_1 dalje u stanje s_2) cijena do tada pređenog puta je porasla: $g(n_2)=g(n_1)+c=5+1=6$, ali je ukupna procjena duljine puta $f(n_2)$ pala jer je heuristička funkcija više izgubila na vrijednosti no što iznosi c, tako da u n_2 imamo $f(n_2)=8$, čime smo izgubili dio informacije koji smo u prethodnom koraku imali – već onda smo znali da nas put košta barem 9, a prethodno smo već komentirali da želimo da funkcija f(.) ima što veći iznos i da bude monotona. Ovom problemu može se doskočiti tako da se kao procjena ukupne duljine puta za zadani čvor uzme što je veće: procjena ukupne duljine puta od roditelja ili naša vlastita procjena. U tom slučaju $f(n_2)$ računamo prema izrazu:

$$f(n_2) = \max(f(n_1),\ g(n_2)+h(s_2))$$

i takav način izračuna procjene ukupne cijene nazivamo *pathmax*-korekcija. Uočite da time ne radimo ništa pogrešno: naprosto odbijamo jednom kada imamo bolju procjenu troška zaboraviti na nju i nastaviti raditi s manje kvalitetnom procjenom.

Pojam dominacije

Neka su A^*_1 i A^*_2 dva optimalna algoritma s optimističnim heurističkim funkcijama h_1 i h_2. Algoritam A^*_1 *dominira* nad algoritmom A^*_2 ako i samo ako:

$$(\forall s \in S) h_1(s) \geqslant h_2(s)$$

U tom slučaju kažemo i da je algoritam A^*_1 *obavješteniji* od algoritma A^*_2. Što je algoritam (odnosno heuristika) obavještenija, to će se otvarati manje čvorova i algoritam će biti usmjereniji u pretraživanju.

Zabilješke uz pojedine slideove

Slide 5: heuristika h_2 je obavještenija od heuristike h_1. Naime, ako ustanovimo da imamo *k* pločica koje su na krivom mjestu, tada svaku moramo pomaknuti barem za jedno mjesto a potencijalno i za više da bi došle na svoje ciljne pozicije. Stoga je $h_1 = k \leq h_2$.

Slide 8: algoritam će otići krivim putem. Naime, gledano iz Pazina, sljedbenici su Motovun i Lupoglav. Kako je h(Motovun)=12 < h(Lupoglav)=13, algoritam najbolji prvi koji gleda samo heurističku procjenu udaljenosti do cilja odabrat će Motovun i nakon toga jedinog sljedbenika: Pazin, čime će pronaći put duljine 20+18=38. Da je krenuo preko Lupoglava, put bi bio dugačak 22+15=37, što je kraće.

Slide 14: obrazložimo svaki od slučajeva
- h1 JE optimistična: stvarna cijena zahtjeva barem onoliko pomaka koliko je pločica na krivim mjestima, ali moguće i više, pa h1 ne precjenjuje
- h2 JE optimistična: kada bi se pločice mogle pomicati jedna preko druge, tada bismo morali napraviti barem onoliko pomaka koliko je zbroj udaljenosti svake od pločica od svoje ciljne pozicije; no kako pločice nije moguće pomicati jedne preko drugih, u stvarnosti će trebati napraviti minimalno toliko pomaka, a moguće i više, pa stvarna cijena veća ili jednaka od h2, čime h2 ne precjenjuje
- h3 JE optimistična: totalno je neinformirana i A* algoritam će pretvoriti u neefikasni Uniform Cost Search, ali svakako ne precjenjuje
- h4 **NIJE** optimistična: i za ciljno stanje tvrdi da je udaljenost do njega samoga 1 čime precjenjuje
- h5 JE optimistična jer upravo daje iste vrijednosti je daje i stvara udaljenost – ona nije niti manja niti veća od stvarne udaljenosti već je savršena
- h6 JE optimistična: naime, ako je iznos od h* veći ili jednak od 2, h6 vraća 2, a ako je h* manja od 2, h6 vraća tu vrijednost; time h6 nikada ne precjenjuje stvarnu udaljenost, ali je vrlo slabo informirana
- h7 **NIJE** optimistična: za ciljno stanje u kojem je h*=0 funkcija h7 bit će 2 čime precjenjuje stvarnu udaljenost.

Slide 17: najjednostavnije će biti izračunati stvarne minimalne udaljenosti do *f* (dakle iznose koje bi davala heuristika h^*) pa onda provjeriti optimalnost i konzistentnost heuristike. Gledajući zadani graf, najlakše je krenuti od ciljnog stanja pa ići unatrag:
- $h^*(c)$ = min distance $(c \rightarrow f) = 1$, $c \rightarrow f$
- $h^*(e)$ = min distance $(e \rightarrow f) = 5$, $e \rightarrow b \rightarrow c \rightarrow f$
- $h^*(d)$ = min distance $(d \rightarrow f) = 7$, $d \rightarrow e \rightarrow b \rightarrow c \rightarrow f$
- $h^*(b)$ = min distance $(b \rightarrow f) = 4$, $b \rightarrow c \rightarrow f$
- $h^*(a)$ = min distance $(a \rightarrow f) = 9$, $a \rightarrow d \rightarrow e \rightarrow b \rightarrow c \rightarrow f$

Za optimističnost sada provjerimo je li heuristika prikazana na slici manja ili jednaka stvarnoj cijeni puta za svaki od čvorova:
- $h(c) = 1 \leq h^*(c) = 1$ zadovoljeno
- $h(e) = 2 \leq h^*(e) = 5$ zadovoljeno
- $h(d) = 7 \leq h^*(d) = 7$ zadovoljeno
- $h(b) = 2 \leq h^*(b) = 4$ zadovoljeno
- $h(a) = 9 \leq h^*(a) = 9$ zadovoljeno

Zaključak: *heuristika je optimistična*.
Pogledajmo je li konzistentna. Da bi bila, moralo bi vrijediti za svako stanje i njegov sljedbenik uvjet konzistentnosti (odnosno nejednakost trokuta):

- $h(a) \leq h(b) + 6 \Rightarrow 9 \leq 2 + 6 \Rightarrow$ uvjet NIJE zadovoljen, *heuristika nije konzistentna*
- dalje onda niti ne gledamo jer je dovoljno da za jedan prijelaz ne vrijedi...

Slide 18: implicitna pretpostavka koja ne piše na slideu je da se koristi optimistična heuristika

Varijanta 1 je sama po sebi jasna. Lista zatvorenih čvorova samo je pomoćni mehanizam kojim smanjujemo višestruka ispitivanja istih stanja kada to nije potrebno.

Varijanta 2: optimalnost slijedi iz činjenice da je uz konzistentnu heuristiku funkcija f(.) monotona, a za taj slučaj smo već objasnili da tada prvo otvaranje čvora s određenim stanjem je ujedno i najjeftiniji put, pa je dovoljno pamtiti samo jesmo li već otvorili neko stanje ili nismo.

Varijanta 3: optimalnost slijedi iz činjenice da pathmax-korekcija funkciju f(.) pretvara u monotonu, a dalje onda vrijedi isto zaključivanje kao u varijanti 2.