

Iskanje s tabuji

Jerica Valjavec

Fakulteta za matematiko in fiziko, Ljubljana

26. maj 2011

- zapomni si podatke o zgodovini postopka
- pri vsaki ponovitvi najde novo (boljšo, včasih tudi slabšo) rešitev
- nove rešitve išče v okolici trenutne rešitve

$s = s_0$; /začetna rešitev

prepovedani seznam;

srednjeročni spomin;

dolgoročni spomin;

while (zaustavitveni kriterij)

 poišči najboljšo dosegljivo sosednjo rešitev s' ;

$s = s'$;

 posodobi prepovedani seznam;

 posodobi srednjeročni spomin;

 posodobi dolgoročni spomin;

if (pogoj za usmerjeno preiskovanje) usmerjeno preiskovanje;

if (pogoj za razpršeno preiskovanje) razpršeno preiskovanje;

- lahko se pojavijo cikli
- loči med obiskanimi in neobiskanimi rešitvami
- obiskane rešitve hrani na prepovedanem seznamu
- prepovedani seznam predstavlja kratkoročni spomin

- hrani že obiskane rešitve (lahko tudi poteze ali attribute potez)
- pri vsaki ponovitvi se posodobi
- zahteva veliko časovno in prostorsko zahtevnost
- izboljšava: hrani konstantno število prepovedanih rešitev
- njegova dolžina vpliva na potek postopka
- dolžino se določi glede na strukturo pokrajine problema

- poteza lahko doró obeta (generira boljšo rešitev), vendar je na prepovedanem seznamu
- če prepovedana poteza zadošča aspiracijskem kriteriju, jo lahko sprejmemo (uporabimo)
- dosegljiva okolica rešitve so tiste rešitve, ki niso na prepovedanem seznamu ali zadoščajo aspiracijskem kriteriju

Oblike prepovedanega seznama:

- statična oblika: statično dolžino seznama določimo glede na velikost problema oz. glede na velikost okolice rešitve
- dinamična oblika: dolžina prepovedanega seznama se spreminja brez uporabe informacij o zgodovini postopka
- prilagodljiva oblika: dolžina prepovedanega seznama se spreminja glede na zgodovino postopka (npr. glede na obnašanje v zadnjih nekaj ponovitvah)

- skrbi za usmerjeno preiskovanje (postopek usmeri v obetavno smer)
- hrani najboljše najdene rešitve
- postopek usmeri v iskanje med rešitvami, ki imajo take lastnosti kot najboljše najdene rešitve
- lahko kar ponovo zažene postopek z najboljšo najdeno rešitvijo in fiksira komponente, ki so najbolj obetajoče

- srednjeročni spomin se predstavi kot nedavni spomin
- definirane morajo biti komponente, povezane z rešitvami
- nedavni spomin za vsako komponento beleži število uspešnih ponovitev postopka v katerih je ta komponenta prisotna v obiskanih rešitvah

Usmerjeno preiskovanje

- začetek usmerjenega preiskovanja sproži določeno število ponovitev brez napredka
- usmerjeno preiskovanje ni vedno uporabno
- učinkovitost je odvisna od strukture problema

- skrbi za razpršeno preiskovanje (postopek usmeri v še ne raziskana področja)
- dolgoročni spomin se predstavi kot frekvenčni spomin
- definirane morajo biti komponente, povezane z rešitvijo
- frekvenčni spomin si za vsako komponento beleži število kolikokrat je komponenta prisotna v vseh obiskanih rešitvah
- razpršeno preiskovanje se uporabi po danem številu ponovitev brez napredka

Tri metode razpršenega preiskovanja:

- ponovni zagon razpršenega preiskovanja: pogleda najmanjkrat obiskano komponento, vzame najboljšo rešitev s to komponento, nato pa postopek ponovno zažene na tej rešitvi
- nenehno razpršeno preiskovanje: v oceni trenutne rešitve uporabi frekvenco komponent rešitve, pri tem so pogoste obiskane rešitve slabše ocenjene
- strateško nihanje: dovoljuje uporabo vmesnih rešitev (rešitev, ki sploh niso možne); postopek vodi prek teh rešitev nato pa pride nazaj po pravo rešitev

- ni vedno uporabno (odvisno od pokrajinske strukture problema)
- npr. če so vse dobre rešitve skoncentrirane na enem mestu (in so tesno skupaj), potem razpršeno preiskovanje ni uporabno