Primena epsilon-transformacije u problemu pretrage drveta

Jozef Kratica[1], Slobodan Radojević[2], Vladimir Filipović[3], Ana Šćepanović[4]

Rezime

Poznato je da u problemima pretrage drveta postoje slučajevi u kojima mala promena vrednosti čvorova i grana drveta, dovodi do drastične promene u složenosti (vremenu izvršavanja) algoritma. Takva pojava, gde složenost algoritma pretrage prelazi iz eksponencijalne u polinomijalnu (u funkciji od dubine drveta), naziva se fazni prelaz. U ovom radu se prezentira epsilon-transformacija kao metod primene faznog prelaza, gde pretragu eksponencijalne složenosti malom promenom transformišemo u slučaj pretrage polinomijalne složenosti. Međutim, optimalno rešenje pretrage transformisanog drveta, nije i optimalno rešenje pretrage polaznog drveta, ali je blisko optimalnom (suboptimalno rešenje).

Ključne reči: Epsilon-transformacija; Pretraga drveta; Fazni prelaz; Kombinatorna optimizacija;

Abstract

It has been shown that in tree search problems exists cases, in which small change of node and edge values, product large change of problem complexity (execution time). That appearance, in which problem complexity transiting from exponential to polynomial (in the search depth), calling phase transition. In this paper we show epsilon-transformation as method of using phase transition, in which tree search problem of exponential complexity we transforming to search problem of polynomial complexity. However, optimal solution of search on transformed-tree, is not optimal solution of initial problem, but is near to optimal (suboptimal solution).

Keywords: Epsilon-transformation; Tree search; Phase transition; Combinatorial optimization;

[1] mr Jozef Kratica, Katedra za Matematiku Mašinskog fakulteta u Beogradu, ul. 27. marta 80,

[2] dr Slobodan Radojević, Katedra za Matematiku Mašinskog fakulteta u Beogradu, ul. 27. marta 80

[3] Vladimir Filipović, Matematički fakultet u Beogradu, ul. Studentski trg 16

[4] Ana Šćepanović, Matematički fakultet u Beogradu, ul. Studentski trg 16

1. Uvod

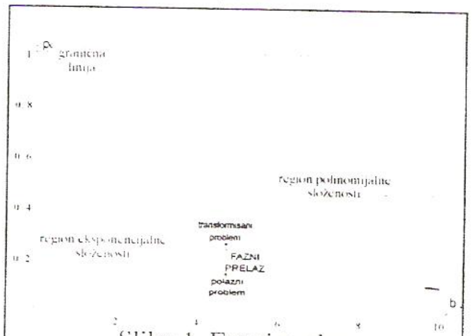
1.1. Fazni prelaz

U prirodi je jedan primer za fazni prelaz, recimo tačka promene agregatnog stanja (tačka topljenja, tačka ključanja, ...). U tim situacijama mala razlika u temperaturi tela karakteriše velike promene u svojstvima i unutrašnjoj energiji tela.

Prvi primer fazog prelaza u oblasti programiranja je dat u problemu pretrage drveta ([8]), a time i u oblasti kombinatorne optimizacije([7][14][18][8]). Zatim je pronadjeno još nekoliko primera primene faznog prelaza u programiranju: modeli asocijativne memorije ([13]), automatsko planiranje ([2]), i u raznim problemima prepoznavanja i klasifikacije ([6][10]).

1.2. Pretraga drveta

Potrebno je naći optimalni završni čvor u drvetu čije su dužine grana nenegativni realni brojevi. Vrednosti u čvorovima su zbirovi dužina grana na putu od korenog čvora do datog čvora (videti sliku 2a). Zbog toga je vrednost svakog čvora naslednika veća ili jednaka od vrednosti njegovog prethodnika. Optimalni završni čvor je čvor na zadatoj fiksnoj dubini, koji ima najmanju vrednost.



Slika 1. Fazni prelaz

Ukoliko su dužine grana nezavisne i jednako raspodeljene slučajne veličine, očekivana složenost nalaženja optimalnog rešenja prolazi faznu tranziciju od eksponencijalne do polinomijalne u funkciji dubine drveta. Vrednost koja utiče na datu promenu je očekivani broj naslednika svakog čvora koji imaju istu vrednost kao i dati čvor (dužina grane je nula). Nazovimo tu veličinu broj istih čvorova. Ukoliko je b očekivani broj naslednika za svaki čvor, a p_0 verovatnoća da je dužina grane jednaka nuli, tada je očekivani broj istih čvorova jednak $b*p_0$. Poznato je ([12][17][16]) da složenost (vreme izvršavanja) algoritma eksponencijalna za $b*p_0<1$, a polinomijalna za $b*p_0>1$, što možemo videti i na slici 1. Pri tranziciji (prelasku) iz eksponencijalnog ($b*p_0<1$) u polinomijalni region ($b*p_0 > 1$), vreme izvršavanja algoritma dramatično opada ([12][17]).

Načini za pretragu drveta su: pretraga po dubini (depth first search - DFS), pretraga po širini (breath first search), različite metode ograničenja grane (branch and bound - BnB), pretraga po veličini (best first search), itd. Različiti načini pretrage drveta, njihove karakteristike, kao i uporedna analiza se mogu detaljnije upoznati u [1] [9] [11] i [15].

Medutim, za svaki konkretan način pretrage važi gorepomenuti način faznog prelaza eksponencijalne u polinomijalnu složenost ([12] [17]).

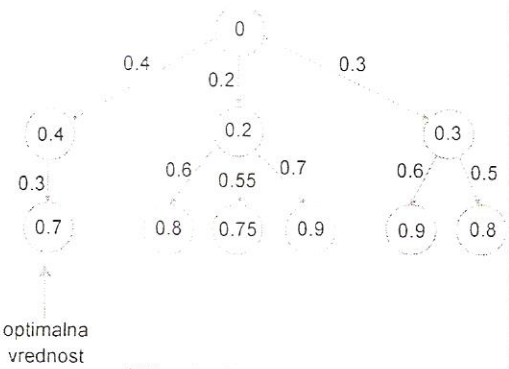
1.3. Drvo pretrage problema kombinatorne optimizacije

Pri rešavanju problema kombinatorne optimizacije se prirodno javlja pojam drveta pretrage. Čvorovi tog drveta predstavljaju stanja, a grane su mogući prelazi sa datog stanja u neko drugo stanje. Recimo u slučaju neke igre, stanja su pozicije, a grane su mogući potezi. Time većinu problema kombinatorne optimizacije možemo svesti na problem pretrage drveta ([1][3][16]). Pri tome u svakom trenutku memorišemo samo nekoliko čvorova drveta pretrage, jer veličina drveta pretrage eksponencijalno zavisi od njegove dubine.

2. Epsilon-transformacija

Ukoliko je drvo takvo da je $b^*p_0>1$, pretraga je polinomijalne složenosti, pa je možemo primeniti bez ikakvih promena, i u tom slučaju dobijamo optimalno rešenje.

Međutim, ako je $b^*p_0\leq 1$, složenost ma koje od gorepomenutih pretraga je eksponencijalna (videti [4][5][9]). U slučaju velike dubine drveta (veće od 10-30), vreme izvršavanja algoritma je enormno veliko. Jedini praktično moguć pristup je primena neke heuristike, gde dobijamo suboptimalno rešenje, koje je relativno blisko optimalnom, uz dostižno vreme izvršavanja algoritma.



Slika 2a Početno drvo

Na primer fazni prelaz iz eksponencijalnog u polinomijalni region složenosti se može ostvariti povećanjem veličine b^*p_0 na vrednost veću od 1. Vrednost b očekivanog broja naslednika je nemoguće povećati, jer je ona fiksna za dati problem. Međutim veličinu p_0 je moguće povećati ukoliko svakoj grani drveta čija je dužina manja od unapred zadatog broja ε dodelimo vrednost nula. Vrednost za ε biramo tako da b^*p_0 bude veće od 1 (videti sliku 1). Dakle ε biramo tako da važi:

$b^*p_0>1$, gde je

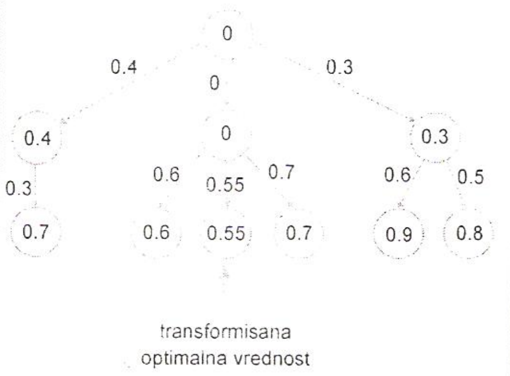
$$p_0=P(\text{transf. dužina grane} = 0) = P(\text{početna dužina grane} < \varepsilon) \qquad (1)$$

Dati postupak se naziva epsilon transformacija.

2.1. Jednostruka primena metode

Ukoliko je unapred poznata vrednost ε, za koju treba primeniti postupak, tada postupak možemo primeniti u jednom prolazu.

Za datu vrednost ε, transformišemo drvo, pri čemu svaka grana čija je dužina bila manja od ε, dobija dužinu 0. Pri tome i vrednosti čvorova dobijaju nove, transformisane vrednosti (videti sliku 2a i sliku 2b).



Slika 2b Transformisano drvo za ε=0.25

Izvršimo pretragu transformisanog drveta nekom od metoda, recimo DFBnB.

Primetimo da završni čvor koji je optimalan u polaznom drvetu, ne mora biti optimalan i u transformisanom drvetu. Na primeru sa slike 2a je optimalan čvor čija je vrednost 0.7, ali je u transformisanom drvetu na slici 2b optimalan čvor čija je početna vrednost 0.75. Iako dati čvor nije optimalan, njegova početna vrednost je bliska optimalnoj.

2.2. Višestruka (iterativna) primena metode

Vrednost ε najčešće nije unapred poznata. U datom slučaju je na početku potrebno odrediti dovoljno veliko ε, koje zadovoljava uslov (1), a zatim postepeno smanjivati ε, ali tako da i dalje zadovoljava uslov (1).

Dakle, sa početnom vrednošću ε, izvršimo tranformaciju drveta, a zatim pretragu (recimo DFBnB). Pri tome beležimo veličinu z - najveću dužinu grane koja je pri transformaciji dobila vrednost nula, odnosno maksimalnu dužinu grane u polaznom drvetu koja je bila manja od ε. Novu vrednost ε biramo na osnovu z, kao ε=f(z), gde je f neka unapred zadata funkcija koja zadovoljava $0 \leq f(z) \leq z$.

Za novu vrednost ε ponovo izvršimo transformaciju drveta i pretragu, i ažuriramo novu vrednost z, za koju je ponovo ε=f(z).

Postupak se završava u slučaju da je ispunjen neki od završnih kriterijuma: određen (konačan) broj iteracija, rešenje u uzastopnim iteracijama se ne poboljšava (ostaje isto), određeno (konačno) vreme izvršavanja, itd.

Funkciju f(z) treba pažljivo izabrati. Ukoliko je vrednost funkcije f(z) manja, time je rešenje bliže optimalnom, ali je složenost (vreme izvršavanja) algoritma veća. Ukoliko je f(z) blizu z, poboljšanje rešenja u novoj iteraciji je manje, ali se složenost algoritma sporije povećava. Takođe su manje šanse da bude $b^*p_0<1$, pa da složenost pretrage bude eksponencijalna (što upravo ovim postupkom želimo da izbegnemo). U literaturi ([12][17]) se kao dobar kompromis navodi izbor funkcije f(z)=z/2.

3. Zaključak

Epsilon-transformacija je jedna od najnovijih metoda za rešavanje problema pretrage drveta čija je složenost eksponencijalna. Uz pomoć teorijskih rezultata o faznom prelazu ([4][5][16][8][7][18]), primenom epsilon-transformacije, pretragu eksponencijalne složenosti transformišemo u pretragu polinomijalne složenosti. Time se algoritam i za veće dimenzije problema (dubina drveta) izvršava u dostižnom vremenu, ali se dobija rešenje koje ne mora biti optimalno, već je blisko optimalnom rešenju (suboptimalno rešenje).

Ukoliko za pretragu drveta primenjujemo neku od metoda ograničenja grane (branch and bound - BnB), moguća su dodatna poboljšanja, primenom ograničenja grane pri pretrazi transformisanog drveta, korišćenjem informacija o polaznom drvetu. Međutim iz ([12]) se može videti da su dobici takvog pristupa vidljivi, ali nisu veliki.

Pošto je za većinu problema kombinatorne optimizacije moguće formirati drvo pretrage, moguća je primena epsilon-tranformacije na veliki broj takvih problema koji su NP-kompletni ili eksponencijalni. Pri tome je bitan izbor vrednosne funkcije koja vrednuje čvorove (ili grane) drveta pretrage na osnovu stanja (ili mogućih prelaza između stanja) datog problema.

4. Literatura

[1] Brassard G., Bratley P., *Algorithmics: Theory and Practice*, Prentice-Hall International, Englewood Cliffs, NJ (1988)

[2] Bylander T., "An average case analysis of planning", *Proceedings AAAI-93*, pp. 480-485, Washington, DC (1993)

[3] Garey M.R., Johnson D.S., *Computers and Intractibility: A Guide to the Theory of NP-Completeness*, Freeman, New York (1979)

[4] Hogg T., Refining the phase transition in combinatorial search, *Artificial Intelligence* 81, pp. 127-154 (1996)

[5] Hogg T., Huberman B.A., Williams C.P., Phase transitions and the search problem, *Artificial Intelligence* 81, pp. 1-15 (1996)

[6] Hogg T., Kephart J.O., Phase transitions in high-dimensional pattern classification, *Comput. Syst. Sci. Eng.* 5 (4), pp. 223-232 (1990)

[7] Huberman B.A., Hogg T., Phase transitions in artificial intelligence systems, *Artificial Intelligence* 33, pp. 155-171 (1987)

[8] Karp R.M., Pearl J., Searching for a optimal path in a tree with random costs, *Artificial Intelligence* 21, pp. 99-117 (1983)

[9] Korf R.E., Chickering D.M., Best-first minimax search, *Artificial Intelligence* 84, pp. 299-337 (1996)

[10] Lau M., Okagaki T., Applications of the phase transition theory in visual recognition and classifications, *J. Visual Commun. Image Representation* 5 (1), pp. 88-94 (1994)

[11] Manber U., *Introduction to Algorithms: A Creative Approach*, Addison-Wesley Publishing Company, Reading, MA (1989)

[12] Pemberton J.C., Zhang W., Epsilon-transformation: exploiting phase transitions to solve combinatorial optimization problems, *Artificial Intelligence* 81, pp. 297-325 (1996)

[13] Shrager J., Hogg T., Huberman B.A., Observation of phase transitions in spreading activation networks, *Science* 236, 1092-1094 (1987)

[14] Stone H.S., Sipala P., The average complexity of depth-first search with bactracking and cutoff, *IBM J Res. Development* 30, pp. 242-258 (1986)

[15] Zhang W., Korf R.E., Performance of linear-space search algorithms, *Artificial Intelligence* 79, pp. 241-292 (1995)

[16] Zhang W., Korf R.E., A study of complexity transitions on the asymmetric traveling salesman problem, *Artificial Intelligence* 81, pp. 223-239 (1996)

[17] Zhang W., Pemberton J.C., "Epsilon-transformation: exploiting phase transitions to solve combinatorial optimization problems", *Tech. Report UCLA-CSD-940003*, Computer Science Department, University of California, Los Angeles, CA (1994)

[18] Zhang W., Korf R.E., "An average-case analysis of branch-and-bound with applications: summary of results", *Proceedings AAAI-92*, pp. 545-550, San Jose, CA (1992)