Dr Milunka Damnjanović

Elektronski fakultet
Beogradska 14
18000 Niš

HEURISTIČKI ALGORITMI I PROJEKTOVANJE VEZA INTEGRISANIH KOLA

HEURISTICS AND ICs ROUTING

SADRŽAJ. S obzirom da je dokazano da problemi projektovanja veza pripadaju kategoriji nedeterminističkih polinomnih problema, nameće se upotreba heurističkih algoritama za njihovo rešavanje. U njihovoj osnovi su empirijska znanja, a problem rešavaju nalazeći približno optimalno rešenje. U ovom radu su, pored osnovne postavke heurističkih algoritama uopšte, opisani neki algoritmi razvijeni za globalno trasiranje veza na integrisanim kolima.

ABSTRACT. Typical subproblems derived from the original layout problems by some abstraction and simplification are shown to be *NP*-complete, which encourages us to use heuristics for layout design. They tend to use the empirical knowledge to aid in discovery, and they find the approximation of the objective function. In this paper, heuristics in common is discussed and some heuristics specially developed for layout design are described.

1. UVOD

Najšire prihvaćena mera osobina jednog algoritma je njegovo vreme izvršavanja na računaru, tj. vreme potrebno da se dobije konačni rezultat. Postoje dva problema pri definisanju vremena izvršavanja nekog algoritma: vreme značajno varira od jednog do drugog računara i od jednog do drugog ulaza. Prvi problem se može prevazići izražavanjem vremenskih zahteva jednog algoritma pomoću broja elementarnih koraka kao što su aritmetičke operacije, poredjenja, instrukcije grananja, i tako dalje. Da bi se izbegao drugi problem, mogu se razmotriti svi ulazi zadane veličine n i definisati kompleksnost algoritma za tu veličinu ulaza kao najgori slučaj ponašanja nad ovim ulazima. Slično se može definisati kompleksnost srednjeg slučaja za algoritam za tu veličinu ulaza. Dakle, vremenska kompleksnost T(n) jednog algoritma se može izraziti kao funkcija veličine ulaza n.

Definišimo *polinomni algoritam* kao onaj kod koga se vremeizvršavanja, tj. broj elementarnih operacija nad bitovima koje on obavlja, za ulazni niz dužine n može izraziti u obliku polinoma P(n). *P* je klasa svih problema koji se mogu rešiti takvim algoritmom.

Definišimo *deterministički algoritam* kao onaj kod koga za svako stanje postoji najviše jedno sledeće stanje, tj. algoritam može da radi samo jednu stvar u jednom trenutku. Onda su *nedeterministički algoritmi* oni kod kojih za bilo koje dato stanje mogu postojati više od jednog sledećeg stanja. Drugim rečima, nedeterministički algoritmi mogu da rade više stvari istovremeno.

NP je klasa svih problema koji se mogu rešiti nedeterminističkim algoritmima koji imaju polinomno vreme izvršavanja ($P \subseteq NP$).

Problem P je *NP-težak* (*NP*-hard) ako se algoritam za njegovo rešavanje može koristiti za dobijanje determinističkog polinomnog algoritma za svaki problem u *NP*. Dakle, P je *NP*-težak ako je težak najmanje kao bilo koji problem u *NP*.

NP-težak problem u *NP*, naziva se *NP-potpuni* (*NP*-complete).

Da bi se dokazalo da je problem *NP*-težak, treba dokazati da ako je dat deterministički polinomni algoritam za problem, može se taj algoritam koristitit da bi se dobili deterministički polinomni algoritmi za svaki problem u *NP*. Da bismo dokazali da je jedan problem *NP*-potpuni, moramo dalje još dokazati da je problem u *NP*.

NP-potpuni problemi su najteži u sledećem smislu:

1) Nijedan *NP*-potpuni problem se ne može rešiti nekim poznatim polinomnim algoritmom (nasuprot ogromnom trudu mnogih briljantnih istraživača u toku nekoliko decenija).

2) Možemo transformisati bilo koji deo *NP*-problema u deo *NP*-potpunog problema u polinomnom vremenu. Dakle, ako postoji polinomni algoritam za bilo koji *NP*-potpuni problem, postoje polinomni algoritmi za sve *NP*-probleme.

S obzirom da je nemoguće rešiti *NP*-potpune probleme u polinomnom vremenu, osim za neke specijalne slučajeve, istraživači su predložili polinomne heurističke algoritme koji pronalaze približno optimalno rešenje za sve probleme.

2. HEURISTIČKI ALGORITMI

Poreklo naziva ovih postupaka koji u korenu ima grčku reč "eureka" ukazuje na korišćenje empirijskog znanja kao sredstva za generisanje postupaka. Nekoliko kriterijuma se koriste za ocenu heurističkog algoritma. To su sledeći [1].

- *Kvalitet rešenja* se meri na dva načina: približavanjem vrednosti ciljne funkcije optimalnoj vrednosti i sposobnošću algoritma da generiše fizičko rešenje kad god ono postoji. Nekoliko tehnika su pogodne za merenje koliko blizu optimalnoj vrednosti prilazi heuristički postupak. To su: analiza najgoreg slučaja, analiza verovatnoće, statistička analiza, karakterizacija dobrih i loših problema i mnoge empirijske analiza. Najuverljivija forma analiza su empirijske analize.

- *Vreme izvršavanja i memorija* su osobine koje se, takodje, mogu tretirati tehnikom najgoreg slučaja, verovatnoćom ili empirijskom analizom.

- *Teškoća implementiranja* uglavnom podrazumeva komplikovano kodiranje i široke zahteve podataka. Medjutim, ove karakteristike je teško meriti.

- *Fleksibilnost* je vrlo važno svojstvo heurističkih algoritama jer su oni tipično projektovani za rešavanje realnih problema. Posebno je značajno da su moguće izmene u modelu, ograničenjima i funkciji cilja.

- *Robustnost (snaga)* je svojstvo koje obuhvata više poželjnih karaktetristika algoritma. Tu spada analiza osetljivosti i mogućnost generisanja granica dobijenog rešenja. Jedno proširenje generisanja granica je generisanje dobre karakterizacije, što znači da bi se istovremeno sa generisanjem rešenja, generisale neke informacije kao dualno rešenje. Dobra karakterizacija omogućava korisniku da ovu informaciju koristi za dokazivanje optimalnosti dobijenog rešenja ili za dokazivanje da se rešenje nalazi unutar procentualnog odstupanja od optimalnog rešenja. U mnogim slučajevima algoritam ne generiše fizički ostvarivo rešenje. U tim slučajevima bi bilo poželjno da algoritamgeneriše informaciju koja će korisniku omogućiti da odredi zašto fizičko rešenje nije generisano.

- *Jednostavnost i mogućnost analize* je svojstvo koje predstavlja prednost algoritma u odnosu na ekstremno kompleksne algoritme koje je teško analizirati u pogledu fleksibilnosti, kvaliteta rešenja, itd.

- *Interaktivno računanje* je poželjno u mnogim poslovima gde se interakcijom čovek - mašina može menjati tok postupka.

Uopšte, za ocenu heurističkih algoritama se mogu koristiti i drugi, novorazvijeni kriterijumi koji mogu biti od značaja i samo za odredjenu kategoriju heurističkih algoritama.

Heuristički pristup rešavanju problema je svojstven različitim kategorijama zadataka:

- konstruktivni algoritmi koji generišu rešenje postepenim dodavanjem elemenata (napr. čvorova ili grana) rešenju sve dok se ne postigne prihvatljivo rešenje koje predstavlja fizičko rešenje,

- algoritmi poboljšanja koji startuju sa fizičkim rešenjem koje se popravlja nizom izmena ili dodavanja,

- dekompozicija problema kod kojih izlaz jednog može biti ulaz narednog,

- algoritmi deljenja koji ceo problem dele na skup manjih potproblema, zatim rešavaju svaki potproblem i na kraju integrišu u konačno rešenje,

- sužavanje prostora za fizičko rešenje na skup rešenja za koje postoji efikasan algoritam, i tako dalje.

S obzirom da je dokazano [2] da kategoriji *NP*-potpunih (*NP*-complete) problema pripadaju problemi prevodjenja grafa u layout - definisanje minimalne oblasti u koju se može prevesti graf, minimizacija najduže grane grafa, definisanje broja preseka pri deobi grafa, heuristički algoritmi predstavljaju postupke nalaženja njihovih rešenja. Takodje je dokazano da kategoriji *NP*-potpunih problema pripadaju i problemi povezivanja - planarno povezivanje skupa tačaka na rešetki [3], povezivanje modula u dva sloja [2], dvoslojno i višeslojno kanalno trasiranje [4].

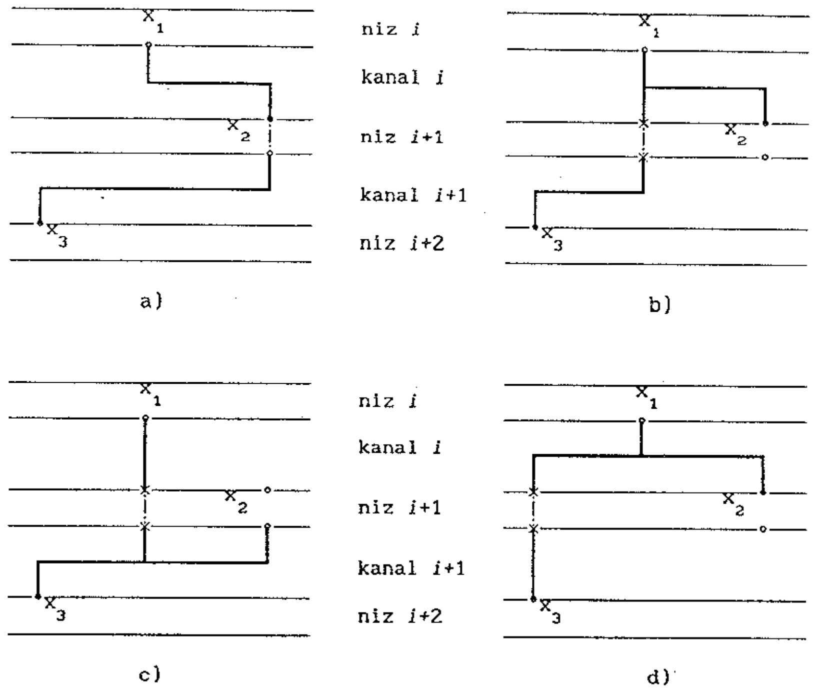
U ovom radu će biti posvećena pažnja algoritmima za globalno trasiranje veza, tj. povezivanje zadanog broja tačaka u ravni, najčešće stablom minimalne dužine realizovanim u rektilinearnoj geometriji (geometriji kod koje su dozvoljena samo dva pravca prostiranja linija, paralelno osama ortonormiranog sistema). S obzirom da se takvo stablo naziva rektilinearnim Steiner-ovim stablom, ceo problem je nazvan Steiner-ovim problemom.

3. HEURISTIČKI ALGORITMI U GLOBALNOM TRASIRANJU

Potreba za globalnim trasiranjem veza ili predtrasiranjem javila se uvodjenjem kanalnog algoritma za trasiranje veza jer je on zahtevao oblasti za trasiranje veza bez diskontinuiteta bilo koje vrste. Zato je bilo potrebno izdvojiti i definisati takve oblasti (koje su nazvane kanalima), a ceo problem trasiranja veza razdvojiti na potprobleme od kojih se svaki rešava unutar jednog kanala. To znači da za svaku vezu treba generisati globalni oblik i njene segmente dodeliti kanalima pre startovanja trasiranja veza unutar kanala (što je nazvano finalnim trasiranjem). Prethodno razvijeni algoritmi opšteg tipa za trasiranje veza (Lee-ov i Hightower-ov) nisu postavljali zahteve u pogledu diskontinuiteta na površini za trasiranje što je omogućavalo trasiranje na celoj površini ploče (čipa). Medjutim, povećanjem stepena integracije elektronskih kola do nivoa LSI i VLSI, podela problema trasiranja čipa na potprobleme je postala potreba, nezavisno od korišćenog algoritma. Tako je globalno trasiranje veza dobilo veliki značaj i razvijen je veliki broj algoritama za njegovu realizaciju, a značajan broj algoritama su heurističkog tipa.

Kod "heurističkog algoritma optimalne dekompozicije" [5], Aoshima i Kuh za svaku vezu *n* generišu graf $G_n = (V_n, E_n)$. Čvorovi V_n grafa G_n su pinovi veze *n* i svi dostupni prolazi. Izmedju čvorova *i* i *j* postoji grana iz E_n ako su *i* i *j* susedi u istom horizontalnom kanalu. Svakoj grani u E_n dodeljuje se težina koja se odredjuje razmatranjem već generisanih veza. Težina grane je jednaka najvećoj gustini dela kanala na kome leži grana pri čemu se gustinom dela kanala (intervala) naziva broj dodeljenih segmenata veza intervalu. Zadatak je naći min-max Steiner-ovo stablo (Steiner-ovo stablo sa minimiziranom maksimalnom težinom grane). Medjutim, s obzirom da se veze razmatraju sekvencijalno, jedna po jedna nezavisno od ostalih, ovakvim načinom dekompozicije moguće je zauzeti sve prolaze kroz nizove ćelija čak i ako to nije neophodno. Naime, neki pinovi imaju ekvipotencijalne pinove u drugom kanalu, te oni mogu zameniti prolaze. Zato su Aoshima i Kuh predložili i drugi heuristički algoritam koja pokušava da minimizira upotrebu prolaza. Ovaj postupak, nazvan "praktični pristup", zahteva generisanje grafa G_d koji za razliku od G_n uključuje samo pinove veze *n*, a ne i prolaze kroz nizove. Zadatak je generisati min-max povezano stablo.

Oba algoritma Aoshima-e i Kuh-a imaju nedostatak da predstavljaju krajnosti. Prvi, tzv. optimalni heuristički postupak, ne ograničava korišćenje prolaza kroz nizove ćelija te se prekomernim korišćenjem prolaza pri dekompoziciji prvih nekoliko veza može onemogućiti dekompozicija ostalih veza ako je broj prolaza ograničen (što je obično slučaj). Drugi postupak, tzv. praktični pristup nastoji da ne koristi prolaze čime se umanjuje mogućnost nalaženja dobre dekompozicije. Na primer, algoritam praktičnog pristupa bi za vezu (x_1, x_2, x_3) sa slike 1 generisao oblik prikazan na slici 1 a) jer pin x_2 ima ekvipotencijalni pin sa druge strane niza. Očigledno je da su slučajevi prikazani na sl. 1 b) - d) bolja rešenja jer veza na sl. 1 a) ima paralelne horizontalne segmente veza, te je duža od ostalih varijanti.



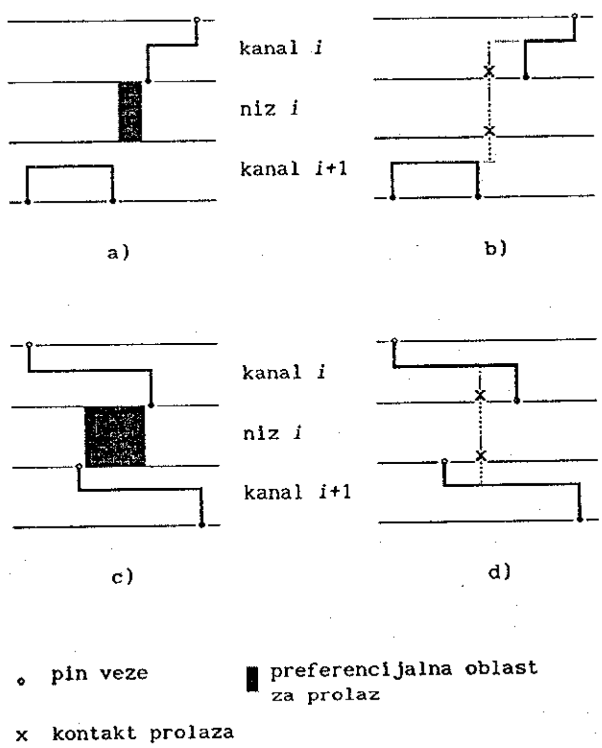
× kontakt prolaza kroz niz
∘ pin veze

Slika 1

Heuristički postupak predložen u [6] rešava ove nedostatke nalazeći kompromis. Naime, optimalna dekompozicija Aoshima-e i Kuh-a je modifikovana time što se u skup čvorova V_n grafa G_n uključuje prolaz kroz niz ćelija ako se nalazi na rastojanju najmanje s. Očigledno se za $s = 1$ dobija isti graf G_n, a za dovoljno veliko s dobija se graf G_d. Ova tehnika je nazvana "spacing".

Drugi pristup istih autora je postavljanje okvira ili prozora u kome je dozvoljeno generisati vezu pri čemu se polazi od minimalnog pravougaonika opisanog oko pinova, a može se povećati ukoliko dekompozicija nije uspešna. U tom smislu bi heuristički algoritmi Aoshima-e i Kuh-a usvajali ivice čipa kao okvir. Autori L.Lin, S.Sahni i E.Shragowitz su ovu tehniku nazvali "windowing".

Za globalno trasiranje na čipu tipa GEM, nezavisno od ovih postupaka, na Elektronskom fakultretu u Nišu razvijen je heuristički postupak [7], [8], [9] koji predstavlja postupak sličan prethodno opisanim postupcima spacing i windowing. Naime, uveden je heuristički pristup *"preferencijalne oblasti"*. Ovim pristupom se u skup čvorova V_n grafa G_n uključuju pinovi veze i prolazi koji se nalaze u preferencija-



Slika 2

lnoj oblasti (osenčena oblast na slici 2 a) i c). Preferencijalna oblast je oblast u kojoj treba da se nalazi prolaz kroz niz da bi globalna veza imala minimalnu dužinu, slika 2 b) i d). Medjutim, da bi se, bez uticaja na kvalitet veze, smanjio broj korišćenih prolaza, u preferencijalnoj oblasti se ispituje postojanje ekvipotencijalnog pina na drugoj strani niza i ako postoji takav pin, usvaja se i ne koristi se prolaz kroz niz. Ako ne postoji takav pin, usvaja se prolaz kroz niz u preferencijalnoj oblasti. Ako ne postoji slobodan prolaz unutar oblasti, oblast se širi sve dok se ne nadje ili ekvipotencijalni pin ili slobodni prolaz kroz niz.

4. ZAKLJUČAK

S obzirom na nedeterminističku polinomsku prirodu problema trasiranja veza, nameće se korišćenje heurističkih algoritama za njegovo rešavanje. U ovom radu su, posle opisa heurističkih algoritama uopšte, opisani neki srodni algoritmi za globalno trasiranje veza. Jedan od njih razvijen je na Elektronskom fakultetu u Nišu i uspešno primenjen za projektovanje gejtovskih polja tipa GEM. Njegova programska realizacija je ugradjena u programski paket ISPGM [10].

5. LITERATURA

[1] Ball, M. and M. Magazine, "The Design and Analysis of Heuristics", Networks, Vol.11, No.2, 1981, pp. 215-219.

[2] Ohtsuki, T., editor, "Layout Design and Verification", North-Holland, Elseiver Science Publishers B.V., 1986.

[3] Richards, D., "Complexity of Single-layer Routing", IEEE Trans. on Computers, Vol.C-33, No.3, 1984, pp. 286-288.

[4] Chen, Y.K.and M.L. Liu, "Three-layer Channel Routing", IEEE Trans. on CAD, Vol.CAD-3, No.2, 1984, pp. 156-163.

[5] Aoshima, K. and E. Kuh, "Multi-Channel Optimization in Gate-Array LSI Layout", Proceedings of the ICCAD 83, 1983, pp. 1005-1008.

[6] Lin, L., S.Sahni and E.Shragowitz, "Enhanced Heuristic for Multi-channel Optimization in Gate Array Layout", CAD, Vol. 21, No. 2, March 1989, pp. 66-70.

[7] Damnjanović, M.S. i V.B. Litovski, "Jedno rešenje povezivanja u gejtovskoj matrici GEM21", Tehnika - Elektrotehnika, Vol. 37, No. 3, 1988, pp.337-340.

[8] Damnjanović, M.S. and Litovski, V.B., "A Survey of Routing Algorithms in Custom IC Design", Journal of Semicustom ICs, Vol.7, No. 2, 1989, pp. 10-18.

[9] Damnjanović, M.S., Radenković, T.S., "Routing for Gate Arrays in an Integrated Design-Software Environment", Facta Universitatis, Series: Electronics and Energetics, Vol. 4, No. 1, 1991, pp.67-80.

[10] Litovski, V.B., "New Results in Integrated Software Development for Gate-Array Design", Proc. of the Third Mideuropean Conference on Custom/Application Specific Integrated Circuits, Sopron, Hungary, April 1991, pp. 69-83.