II MEĐUNARODNI SIMPOZIJUM "INDUSTRIJSKO INŽENJERSTVO"

SIE'98

2nd INTERNATIONAL SYMPOSIUM OF INDUSTRIAL ENGINEERING

Primena genetskih algoritama u nalaženju minimalnog Steinerovog stabla

Ivana Ljubić[1]
mr Jozef Kratica[2]
mr Vladimir Filipović[3]

B2. - 11.

Rezime

U radu je prikazano poređenje dva hibridna genetska algoritma (GA) (bazirana na postojećim heuristikama) za resavanje problema minimalnog Steinerovog stabla (MStT). Egzaktni algoritmi problem rešavaju u eksponencijalnom vremenu. Upoređivane su dve različite formulacije, i dva različita pristupa problemu. Na kraju se ističe značaj ovih algoritama, i potom se navode moguće ideje za dalji rad u ovoj oblasti.

Ključne reči: NP kompletni problemi, mimalno Steinerovo stablo (MStT), genetski algoritmi.

1 UVOD

1.1 NP kompletni problemi

Klasa problema za koje se ne može direktno dokazati da su eksponencijalne složenosti, ali za koje nije pronađen ni algoritam koji ih rešava u polinomijalnom vremenu, predstavlja klasu NP kompletnih problema. Poznati su: problem trgovačkog putnika, bojenje čvorova grafa, problem ranca, nalaženje Hamiltonove konture u grafu... Ako bi se za neki od ovih problema našao polinomijalan algoritam, tada bi se on mogao primeniti i na sve ostale NP kompletne probleme. Više o tome može se naći u [1],[4],[9].

1.2 Steinerov problem na grafu

Steinerov problem na grafu (SPG) je jedan od klasičnih problema kombinatorne optimizacije. On ima široku primenu: kod dizajniranja ingracionih kola, kao i kod dizajniranja velikih mreža (telekomunikacije, naftovodi, trafo-stanice...).

I formulacija problema:

Dato je n tačaka u ravni (skup čvorova W). Povežimo ih granama, a svakoj grani (v_i, v_j) dodelimo težinu koja predstavlja euklidsko rastojanje među čvorovima v_i i v_j. U polinomijalnom vremenu moguće je rešiti MSpT (Minimum spaning tree) problem,([1]) tj. naći stablo najmanje težine koje povezuje sve čvorove. Međutim, ukoliko se postojećem grafu dodaju još neki čvorovi, može se dobiti stablo manje težine. MStT je među tim stablima ono sa najmanjom težinom. Očigledno, da bismo ga jedinstveno odredili, problem svodimo na nalaženje broja Steinerovih (pridodatih) tačaka, i na određivanje njihove pozicije u ravni.
Primeri su dati na slici 1.

slika 1
MStT u jednostavnim situacijama.
Steinerove tačke su boldovane.

II formulacija problema:

Dat je težinski neorjentisani povezani graf $G = (V, E)$. Pozitivna težina (cena) svake grane data je funkcijom $c : E \mapsto \mathbb{R}^+$. Cena grafa G je $c(G) = \sum_{e \in E} c(e)$.
Uočen je fiksni podskup čvorova $W \subset V$. Treba naći povezani podgraf $G' = (V', E')$, takav da $W \subset V'$ i da je $c(G')$ minimalna.
Svaki aciklični podgraf G' takav da $W \subset V'$ je *Steinerovo stablo za* W *u* G, a onaj sa minimalnom cenom je *minimalno Steinerovo stablo za* W *u* G, odn. MStT. Skup $V' \setminus W$ je skup *Steinerovih čvorova*. Dokazano je da je problem NP-kompletan [4].
Optimalna rešenja nađena su egzaktno samo za male dimenzije problema, dok su za veće dimenzije problema u praksi upotrebljive samo heuristike.

1.3 Prost genetski algoritam

Osnovna ideja GA je preuzeta iz prirode - konvergencija u GA je zasnovana na mehanizmu evolucije i prirodne selekcije. U skladu sa tim, posmatramo

[1] Katedra za matematiku, Građevinski fakultet u Beogradu, Bulevar Revolucije 73
[2] Katedra za matematiku, Mašinski fakultet u Beogradu, 27. marta 80
[3] Matematički fakultet u Beogradu, Studentski trg 16

skup od N jedinki koji predstavlja *populaciju* fiksne dužine. Svaka jedinka ima svoju funkciju prilagođenosti (*fitness function*) - koja odgovara vrednosti funkcije koju maksimiziramo u datoj tački.
Jedinke iz populacije zapravo predstavljaju kodirane (po nekom pravilu, najčešće binarno) diskretne vrednosti iz domena. Nad jedinkama se vrši selekcija na osnovu prilagođenosti (bolje prilagođene jedinke će sa većom verovatnoćom preneti svoj genetski materijal u naredne generacije, od onih drugih).
Nad parom selektovanih jedinki vrše se dve osnovne GA operacije: *ukrštanje* i *mutacija*. Na taj način, od dva roditelja dobijamo dva potomka koja zauzimaju njihovo mesto u narednoj generaciji, a koja su (ako su parametri GA dobro izabrani i ako je kodiranje odgovarajuće) bolje prilagođena. Ukratko, jedan opšti GA je sledećeg oblika:

1. Na slučajan način generiši N jedinki - inicijalizuj početnu populaciju.
2. Ponavljaj, do ispunjenja kriterijuma za izlaz:
3. Dekodiraj jedinke i izračunaj njihovu funkciju prilagođenosti, $f(i)$.
4. Selektuj $N/2$ puta parove jedinki i nad njima izvrši ukrštanje i mutaciju.
5. Staru generaciju zameni novom i vrati se na 2.

Kriterijumi za završetak algoritma mogu biti:

- Dostignut je unapred fiksiran broj generacija (zadaje se kao parametar GA).
- Dobijeno je optimalno rešenje (ako se unapred zna).
- Najbolja jedinka se u većem broju generacija ne menja (ovo nije garant da je nađeno rešenje ujedno i optimalno).

Izbor jedinki za ukrštanje se vrši po principu *proste (rulet) selekcije*, odnosno verovatnoća da će i-ta jedinka biti izabrana je broj
$p_i = \frac{f(i)}{\sum_{i=1}^{n} f(i)}$. U prostom GA ukrštanje je jednopoziciono. Za dve binarno kodirane jedinke, na slučajan način se generiše pozicija ukrštanja, prirodan broj $p \in \{1, \ldots, l-1\}$, gde je l unapred fiksirana dužina koda jedinke.

Primer 1:

Pretpostavimo da su u nekoj populaciji ($l = 7$) izabrane jedinke 1011001 i 0101011, i da je generisana pozicija ukrštanja $p = 3$. Dakle,

$$101|1001 \Rightarrow 101|1011$$
$$010|1011 \Rightarrow 010|1001$$

Od interesa je da pomenemo i uniformno ukrštanje korišćeno u algoritmu koji sledi:
Za svaki par izabranih jedinki se na slučajan način generiše maska - binarni niz dužine l. Ako je na i-toj poziciji maske 1, tada prvi potomak uzima gen od prvog roditelja, drugi od drugog, ako je 0 - obrnuto.

Primer 2:

Neka su roditelji isti kao u prethodnom primeru, i neka je generisana maska 1001001. Ukrštanje je tada:

$$1011001 \Rightarrow 1101011$$
$$0101011 \Rightarrow 0111001$$

Mutacija se primenjuje nad ovako dobijenim potomcima na sledeći način: slučajno se generiše pozicija na kojoj se u datoj jedinci komplementira bit, iz 1 u 0, i obrnuto.
Ovim se postiže raznovrsnost genetskog materijala, jer u nekim situacijama gore definisano ukrštanje može da dovede do *preuranjene konvergencije (premature convergence)*, to jest do konvergencije ka nekom lokalnom ekstremumu.

Hibridni GA:

U nekim situacijama je računanje fitness funkcije suviše skupa operacija, pa se umesto prave vrednosti mogu iskoristiti približne (dobijene na primer nekom heuristikom). Isto tako, mogu se vršiti korekcije nađenog rešenja u cilju poboljšanja brzine rada algoritma. Ove ideje su primenjivane u naredna dva GA i predstavljaju samo neke od načina hibridizacije. Za više detalja o tome videti [2].

2 REALIZACIJA GA

2.1 Prva formulacija

Problem je moguće rešiti prostim GA. Primenjuju se klasična selekcija,mutacija i ukrštanje nad kodovima jedinki. Kodiranje je izvršeno na sledeći način: Neka je r broj dodatih Steinerovih tačaka. Kodirajmo x i y koordinate sa po b bitova. Steinerove tačke će se svakako nalaziti unutar konveksnog mnogougla koji opisuje datih n tačaka. Podelićemo dakle na $2^b - 1$ podintervala - na x-osi interval od krajnje leve do krajnje desne tačke iz skupa W, a na y-osi od krajnje gornje do krajnje donje tačke. Na taj način dobili smo 'mrežu' po kojoj vršimo pretraživanje optimalnih r Steinerovih tačaka.
U skladu sa tim, hromozom jedinke iz populacije imače redom x i y binarno kodirane koordinate r Steinerovih tačaka. Dakle dužina hromozoma je $2br$. Jasno je da ovako kodirana jedinka na jedinstven način određuje jedan potpun graf sa

$n+r$ tačaka na kojem se problem *minimalnog razapinjućeg stabla* (MSpT) rešava u polinomijalnom vremenu. Uspešno pretraživanje ([7]) izvedeno je pogodnim izborom parametara:
verovatnoća ukrštanja: $p_c = 0.5$
verovatnoća mutacije: $p_m = 0.01$,
zasnovanog na rezultatima iz [6], dok je eksperimentalno pokazano da je optimalna veličina populacije broj od 30 do 100, (u [7] $N = 50$).

Heuristika

Zbog velikog broja suboptimalnih rešenja i male razlike u dužini tako dobijenih Steinerovih stabala, potrebno je bilo na još neki način ubrzati konvergenciju. Tako je gornji algoritam hibridizovan dodatnom heuristikom, koja fitness funkciju ne računa preko MSpT problema (složenosti $O(n \log n)$), već stablo koje predstavlja MSpT "čisti" od suvišnih Steinerovih tačaka, na osnovu sledećih činjenica:

1. Ako je $|W| = n$, tada je broj mogućih Steinerovih tačaka najviše $n-2$.
2. U MStT stepen Steinerovih čvorova je 3, a ugao izmedju izlaznih grana je 120^o. (videti sliku 1, u geometriji poznata *Toričelijeva tačka* u trouglu)

Tako, Steinerove tačke stepena 1 uklanja zajedno sa incidentnom granom, a za tačke stepena 2 uklanjaju se obe grane i susedni čvorovi povezuju direktno. Čvorovi stepena 3 iterativno se pomeraju u približno optimalnu poziciju. Naime, za dati Steinerov čvor računa se vektor pomeraja kao suma vektora incidentnih grana. Dužina pomeraja se proizvoljno bira, i sve dok ima efekta vrši se pomeranje u datom pravcu i smeru za tu dužinu. Kada to više ne daje rezultata, veličina pomeraja se smanjuje duplo, i postupak se ponavlja sve do odredene tačnosti. Što se čvorova stepena većeg od 3 tiče, oni se relaksacijom drveta (određenim granama se dodeli beskonačna dužina, i na taj način se eliminišu iz minimizacije) svode na stepen jednak 3. Ovim je postignuto ubrzanje u odnosu na gornji algoritam za oko 20 puta.

2.2 Druga formulacija

Pretpostavimo da je $|W| = n$, $|V'| = m$ i $|S| = |V'| - |W| = r$. Za ovaj problem, unapred je poznat skup tačaka preko kojih se vrši minimiziranje, (to je S), ali se ne zna da li u MStT-u učestvuju sve tačke iz S, ili samo neke od njih. Stoga se na neki način one numerišu, kao čvorovi $v_i \in S$, $i = 0, \ldots, r-1$. Ovim se, međutim, na prirodan način nadalje sugeriše kodiranje koje uspešno rešava problem. Dužina koda jedinke je sada r (uporediti to sa dužinom koda prethodnog algoritma!), dakle vreme pretraživanja je znatno kraće, ali sve je to posledica značajno jednostavnije formulacije u ovom slučaju. Kôd jedinke je dakle sada jedan binarni niz $b_0 b_1 \ldots b_{r-1}$, gde vrednost $b_i = 1$ označava da i-ta tačka po redu učestvuje u formiranju Steinerovog stabla.
Primenjeno je uniformno ukrštanje, ali se 'brzina' (verovatnoća) mutacije menja kroz izvršavanje algoritma, po pravilu:

$$p_m = \frac{N}{1-N}(p_{max} - p_{min})N_d + \frac{p_{min} - N p_{max}}{1-N}.$$

N_d je broj različitih Steinerovih stabala predstavljenih jednom generacijom, a p_{max} i p_{min} unapred zadati parametri (granice u kojima treba p_m da se kreće). Teorijski je pokazano da p_m ispod određene granice nema više nikakvog uticaja na GA, ali i da iznad neke granice previše 'razbija' genetski materijal. Ovim je takođe omogućeno da se brzina mutacije linearno povećava, onda kad raznovrsnost genetskog materijala krene da opada.
$N-1$ član inicijalne populacije se bira na proizvoljan način, a jedan član se bira tako da zadovolji gornju ocenu greške, koja garantuje da će se u svakoj sledećoj generaciji dobiti rešenje ne lošije od prethodnog (videti [3]). Naime, pokazano je da za T_{KMB} (videti nastavak teksta) važi:

$$c(T_{KMB}) \leq 2(1 - 1/m')c(MStT),$$

pri čemu je $m' \leq m$ - broj listova u MStT.
I u ovoj situaciji u računu fitness funkcije primenjena je jedna heuristika za aproksimaciju MStT za W u G.

KMB Heuristika (Kou, Markowsky and Berman)

ulaz: graf $G = (V, E)$, cena $c : E \mapsto \mathbb{R}^+$, i podskup W.

1. Konstruisati graf $D(G) = (V, L)$ u odnosu na G, pri čemu su grane $(i, j) \in L$ izračunate kao minimalna rastojanja od i do j preko čvorova iz G. (ovo je na primer polinomijalno rešeno Floyd-Warshalovim algoritmom, [1]). Dakle $D(G)$ je potpun graf.
2. Konstruisati podgraf G_1 od $D(G)$ indukovan skupom W.
3. Naći MSpT T_1 za G_1.
4. Od grana iz T_1 rekonstruisati najkraće puteve iz G. Označimo taj podgraf sa G_2.
5. Naći MSpT T_2 za G_2.
6. Brisanjem svih čvorova $v \in V \setminus W$ stepena jedan, dobijamo aproksimaciju za MStT u oznaci T_{KMB}.

U ovom GA nije primenjivana politika zamene kompletne generacije sledećom (*generational replacement*), već tzv. *steady-state* politika, tj. ukoliko bi genetski materijal nekog od roditelja bio bolji od onog kod potomka, tada bi bolji roditelj prešao u sledeću generaciju umesto lošijeg potomka. Time bi se sprečilo gubljenje dobrih kombinacija (o konvergenciji GA videti [8]).

3 ZAKLJUČAK

Primetimo konačno da se prvi problem može preformulisati u drugi. Naime, tada je skup S potencijalnih Steinerovih tačaka skup svih tačaka koje prave mrežu u ravni - skup po kome vršimo pretraživanje. Prvi algoritam ima značaja u tome što predstavlja jedan od prvih pokušaja rešavanja MStT preko GA. Drugi pak ima teorijskog značaja u dobroj oceni greške, zasnovane na KMB aproksimaciji MStT problema. On se takođe pokazao uspešnim i kod jako velikih dimenzija - testiran je na 60 najvećih primera iz OR-Biblioteke, zajedno sa drugim algoritmima, gde se pokazao ili konkurentnim ili boljim - u smislu: nije bio osetljiv na specijalne slučajeve (kao što npr. branch&bound metod jeste).
Neki pravac kojim bi se dalje moglo nastaviti u smislu primene GA na MStT, verovatno bi morao biti nova hibridizacija novijim heuristikama, ili pak hibridizacija zasnovana na nekim novim saznanjima vezanim za kodiranje i dekodiranje jedinki. Ovo otuda što je prvi algoritam praktično pokazao da prost GA nije od neke velike pomoći u rešavanju MStT problema. Takođe se može raditi na istraživanju nekih novih GA operatora koji bi eventualno doveli do poboljšanja performansi algoritma.

References

[1] D. Cvetković, M. Čangalović, Đ. Dugošija, V. Kovačević-Vujčić, S. Simić, J. Vuleta, "Kombinatorna optimizacija - matematička teorija i algoritmi", *Društvo operacionih istarživača Jugoslavije*, Beograd, 1996.

[2] L. Davis, "Handbook of Genetic Algorithms", *Van Nostrand Reinhold*, New York, 1991.

[3] H. Esbensen, "Finding (Near-)Optimal Steiner Trees in Large Graphs", *6th Internacional Conference on GA's*, 1993, pp.485-492.

[4] M.R. Garey, D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP Completness", *W.H. Freeman and Co.*, 1979.

[5] D. Goldberg, "Genetic Algorithms in Search, Optimization, and Mashine Learning", Adison-Wesley Publishing Company, 1989.

[6] J.J. Grefenstette, "Optimization of control Parameters of GA's", *IEEE Trans. Syst., Man & Cybern.*, Vol. SMC-16, No. 1, 1986, pp. 122-128.

[7] J. Hesser, R. M:anner, O. Stucky, "Optimization of Steiner Trees using Genetic Algorithms", *3th Internacional Conference on GA's*, 1989, pp.231-236.

[8] J. H. Holland, "Adaptation in Natural and Artificial Systems", *The University of Michigan Press*, 1975.

[9] U. Manber, "Introduction to Algorithms - A Creative Approach", *Adison-Wesley Publishing Company*, 1989.

Abstract

This paper shows a comparison between two hybrid GA (based on the existing heuristics) resolving a minimal Steiner tree problem. The exact algorithms resolve this problem in exponential time. Two diferent formulations, and two diferent approaches are compared.

Key words: NP completeness, Minimum Steiner tree problem, Genetic Algorithms.