Aproksimacijski postopki za problem trgovskega potnika *

Bojan Mohar
Fakulteta za matematiko in fiziko, Univerza v Ljubljani

1 Uvod

Naj bodo $X_1, X_2, \ldots, X_n$ $(n \geq 1)$ dana mesta, ki jih mora obiskati bogati poslovnež, gospod *T.P.*. Dane so medsebojne razdalje med mesti. Gospod *T.P.*, ki mu bomo rekli kar *trgovski potnik*, mora obiskati vsa mesta in njegov cilj je sestaviti tako zaporedje obiskov, da bo skupna dolžina opravljene poti kar se da majhna.

Naloge podobnega tipa srečamo pri mnogih drugih problemih. V splošnem se lahko zgodi, da se "razdalje" med mesti ne obnašajo kot prave razdalje. Na primer, lahko so negativne, nesimetrične ("razdalja" od X_i do X_j ni enaka kot "razdalja" od X_j do X_i) ali pa celo ne zadoščajo trikotniški neenakosti, ki vedno velja za prave razdalje. Kljub temu problem, kjer ob dani matriki "razdalj" $D = [d(i,j)]_{i,j=1,\ldots,n}$ iščemo tako ciklično zaporedje mest $i_1, i_2, \ldots, i_n, i_1$, $\{i_1, i_2, \ldots, i_n\} = \{1, 2, \ldots, n\}$, da bo vsota zaporednih razdalj $d(i_1, i_2) + d(i_2, i_3) + \cdots + d(i_n, i_1)$ kar se da majhna, imenujemo *problem trgovskega potnika*, včasih tudi *problem potujočega trgovca*. Cikličnemu zaporedju mest $i_1, i_2, \ldots, i_n, i_1$ pravimo *obhod trgovskega potnika*. V obhodu seveda nastopajo vsa mesta, vsako natanko enkrat. Pravim trgovskim potnikom navadno ni težko izbrati ustreznega obhoda, saj število mest, ki jih morajo obiskati, po navadi ni ravno veliko. Vendar pa se že pri relativno majhnih velikostih problema pojavijo hude težave: izkaže se, da splošnih problemov, pri katerih je n večji od 100, že ne znamo dobro reševati[1]. Razlog je nedvomno v tem, da je problem trgovskega potnika NP-težak [2, 6].

Kaj sploh pomeni, da je nek problem NP-težak? Slikovita razlaga je naslednja: Problem je *NP-težak*, če bi kakršenkoli učinkovit (t.j. polinomski) algoritem za ta problem lahko uporabili za učinkovito reševanje stotine drugih problemov, za katere pa doslej še nihče (kljub velikim naporom tisočev znanih matematikov in računalnikarjev) ni uspel najti učinkovitega postopka. Problemi, za katere bi z učinkovito rešitvijo problema trgovskega potnika dobili hitre algoritme, vključujejo na primer naslednje: problem barvanja grafov, problem Hamiltonovega cikla, celoštevilsko linearno programiranje, problem celoštevilskega nahrbtnika itd. Zato je upanje, da bomo našli uporabne postopke za reševanje problema trgovskega potnika, zanemarljivo majhno.

2 Aproksimacijski algoritmi

Omejili se bomo le na poseben primer problema trgovskega potnika. Zahtevali bomo, da za matriko razdalj $D = [d(i,j)]_{i,j=1,\ldots,n}$ velja:

$$d(i,j) = d(j,i), \quad i,j = 1,2,\ldots,n\,. \tag{1}$$

Takemu problemu pravimo *simetrični problem trgovskega potnika*. Že ta posebni primer problema je ne le zelo težak, ampak, kot bomo kmalu videli, je že iskanje približnih rešitev brezupno težek problem.

---

*Prvotna verzija tega članka je bila pripravljena kot gradivo za permanentno izobraževanje učiteljev matematike in računalništva z dne 3. 12. 1992 [5].

[1]Pred kratkim so z uporabo nekaterih novejših metod sestavili programe, ki zmorejo ob podpori najhitrejših računalnikov reševati naloge trgovskega potnika, ki imajo do 1000 mest.

Naj bo I množica podatkov za nalogo trgovskega potnika. Z $\mathrm{Opt}(I)$ označimo skupno vsoto razdalj pri najugodnejšem obhodu trgovskega potnika. Recimo, da imamo postopek, ki za dano nalogo trgovskega potnika poišče neki bolj ali manj dober obhod, ki pa ni nujno optimalen. Recimo, da je pri podatkih I vsota razdalj na obhodu, ki ga vrne naš postopek, enaka $\mathcal{A}(I)$. Če za vsak I velja ocena:

$$\mathcal{A}(I) \leq (1+r)\,\mathrm{Opt}(I)\,, \tag{2}$$

tedaj pravimo, da je naš postopek r*–aproksimacijski*. Po definiciji nam 0–aproksimacijski algoritem vedno vrne optimalno rešitev. Na žalost pa pri splošnem problemu trgovskega potnika nimamo mnogo upanja, da bi znali učinkovito iskati kakršnekoli aproksimativne rešitve. To nam pove naslednji izrek.

Izrek 2.1 *Za vsako dano število $r \geq 0$ je problem iskanja r-aproksimacije simetričnega problema trgovskega potnika NP-težak.*

Dokaz. Znano je, da je problem iskanja hamiltonovega cikla NP-težak. Zato je dovolj videti, da lahko problem hamiltonovega cikla v polinomskem času prevedemo na iskanje r–aproksimacije za problem trgovskega potnika.

Naj bo G graf na n točkah. Zanima nas, ali G vsebuje hamiltonov cikel. Sestavimo matriko $D = [d(i,j)]_{i,j=1,\ldots,n}$ razdalj za nalogo trgovskega potnika po naslednjem receptu: $d(i,j) = 0$, če $i = j$, $d(i,j) = 1$, če sta i in j različna in sta i-ta in j-ta točka v grafu G sosednji, sicer pa naj bo $d(i,j) = (r+1)n$. Hitro se vidi, da ima graf G hamiltonov cikel natanko tedaj, ko za dobljeno nalogo trgovskega potnika obstaja obhod dolžine n, to pa je spet ekvivalentno temu, da obstaja obhod dolžine kvečjemu $(r+1)n$. Zato bi s pomočjo učinkovitega r-aproksimacijskega algoritma za problem trgovskega potnika znali učinkovito iskati hamiltonove cikle v poljubnih grafih.

Dobljena prevedba je očitno polinomske časovne zahtevnosti, saj ob danem grafu G lahko matriko D (podatke za ustrezno nalogo trgovskega potnika) sestavimo brez posebnega truda. $\square$

3 Metrični problem trgovskega potnika

Poseben primer simetričnega problema trgovskega potnika je *metrični problem*, pri katerem ima dana matrika D take lastnosti kot matrika razdalj med n točkami v nekem metričnem prostoru. Razen pogoja simetričnosti (1) zahtevamo še:

$$d(i,j) \leq d(i,k) + d(k,j), \quad i,j,k = 1,2,\ldots,n \quad \text{in} \tag{3}$$

$$d(i,i) = 0\,, \quad i = 1,2,\ldots,n\,. \tag{4}$$

Pogoju (3) pravimo *trikotniška neenakost*. Drugi pogoj (4) je pravzaprav le "lepotni dodatek", saj diagonalni elementi matrike D očitno ne vplivajo na rešitve.

Poseben primer metričnega problema je *Evklidski problem*, pri katerem so mesta, ki naj jih obišče trgovski potnik, točke v nekem Evklidskem prostoru $\boldsymbol{R}^k$, razdalje med njimi pa so enake Evklidskim razdaljam v $\boldsymbol{R}^k$.

Izrek 3.1 *Metrični problem trgovskega potnika je NP-težak.*

Dokaz. Problem hamiltonovega cikla v grafih lahko prevedemo tudi na metrični problem trgovskega potnika. Ob danem grafu G naj bo razdalja med različnima mestoma i in j enaka 1, če sta i-ta in j-ta točka v grafu G sosednji, ter enaka 2 sicer. Hitro se vidi, da tako dobljena matrika razdalj res zadošča pogojem (1), (3) (in (4)). Tedaj obstaja obhod trgovskega potnika, ki je dolžine manj ali enako n, natanko takrat, ko dani graf G vsebuje hamiltonov cikel. $\square$

Kljub temu, da je zaradi izreka 3.1 reševanje metričnega (in tudi Evklidskega) problema trgovskega potnika brezupno težko, pa v nasprotju s splošnim problemom, kjer po izreku 2.1 celo iskanje približnih rešitev ne pride v poštev, obstajajo enostavni in hitri postopki za približno reševanje metričnega problema. Dva taka postopka si bomo ogledali v nadaljevanju.

4 Postopek z uporabo najcenejšega vpetega drevesa

Aproksimacijska algoritma, ki ju bomo spoznali v nadaljevanju, uporabljata najcenejše vpeto drevo, s pomočjo katerega sestavimo neki pomožni graf G'. Potem v G' poiščemo obhod (tako imenovani Eulerjev obhod), s pomočjo katerega nazadnje dobimo obhod trgovskega potnika. Zato si bomo najprej ogledali nekaj osnovnih dejstev o najcenejših vpetih drevesih in Eulerjevih obhodih.

Dan naj bo povezan graf G z n točkami in uteženimi povezavami. Ker je G povezan, ima vsaj eno vpeto drevo. To je podgraf, ki je povezan, vsebuje vseh n točk grafa G, nima pa nobenega cikla. Vpeto drevo T je *najcenejše*, če je vsota uteži povezav iz T najmanjša možna med vsemi vpetimi drevesi grafa G. Iskanje najcenejšega vpetega drevesa v G ni težko. V ta namen lahko uporabimo kar požrešno metodo: drevo gradimo z zaporednim dodajanjem povezav in vsakič dodamo povezavo z najmanjšo težo, če le taka povezava ne napravi cikla (v tem primeru jo lahko zavržemo in vzamemo naslednjo najcenejšo povezavo). Znani sta dve inačici takega postopka, Primov in Kruskalov algoritem. Radovedni bralec bo našel opis teh postopkov in dokaze njihove pravilnosti tudi v slovenščini, na primer v [4]. Najcenejše vpeto drevo torej lahko poiščemo s požrešno metodo. Izkaže se, da se da tak postopek izvesti v času $O(n^2 \log n)$ [4, 6], če predpostavimo, da so uteži (cene) povezav po velikosti omejene. V splošnem pa moramo časovni zahtevnosti dodati še faktor L, ki je enak

$$L = \max\{\lceil \log d(e) \rceil + 1 \mid e \in E(G)\},$$

kjer je $d(e)$ cena povezave e. Število L namreč opisuje maksimalno dolžino (npr. v desetiškem zapisu) števil $d(e)$, ki jih moramo med postopkom med seboj primerjati, urejati itd. Če so ta števila zelo dolga, delo z njimi ni operacija, ki bi se jo dalo opraviti v času $O(1)$, ampak porabimo $O(L)$ časa.

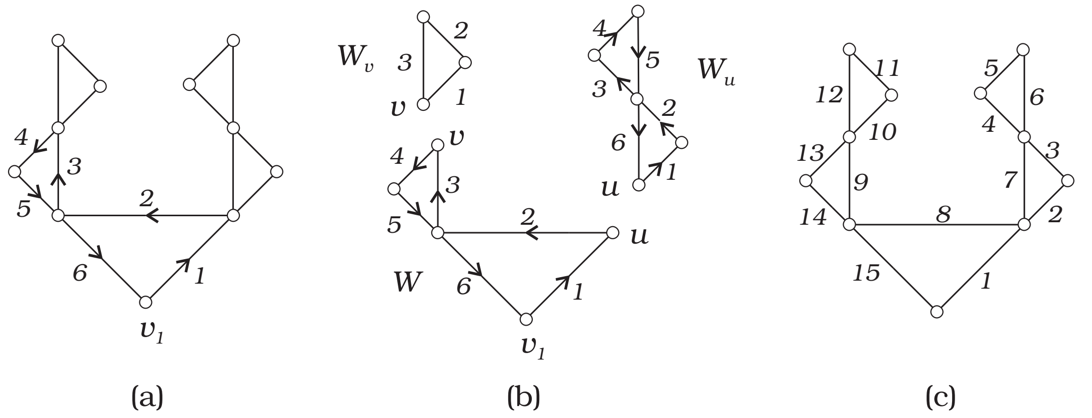
Graf G je *Eulerjev*, če je povezan in vsebuje le točke sode stopnje. Pri tem dovolimo tudi vzporedne povezave in zanke. (Slednje sicer v našem primeru ne igrajo nobene vloge.) Pomembna lastnost Eulerjevih grafov je, da vsebujejo sklenjen obhod, ki uporabi vse povezave grafa, vsako natanko enkrat. Takemu obhodu pravimo *Eulerjev obhod*.

Izrek 4.1 *Graf G brez izoliranih točk vsebuje Eulerjev obhod natanko tedaj, ko je Eulerjev, t.j., G je povezan in vse njegove točke so sode stopnje. Eulerjev obhod v Eulerjevem grafu G lahko poiščemo v linearnem času $O(|E(G)|)$.*

Dokaz. Očitno je, da Eulerjeve obhode lahko vsebujejo le Eulerjevi grafi. Eulerjev obhod namreč v vsako točko pride natanko tolikokrat, kot iz nje odide. Ker graf nima izoliranih točk, je v vsaki točki vsaj ena povezava. Zato Eulerjev obhod obišče vsako točko grafa in graf je torej povezan.

Recimo sedaj, da je G Eulerjev graf. Z indukcijo po številu povezav v G bomo dokazali, da G vsebuje Eulerjev obhod. Če je $|E(G)| = 0$, tedaj ima graf eno samo točko in trivialni obhod dolžine nič je Eulerjev. V splošnem primeru pa izberimo točko $v \in V(G)$. Najprej poiščemo poljuben sklenjen sprehod, začenši v točki v. Iskanje ni težko: odpravimo se v poljubno sosedo točke v. (Ker po predpostavki v ni izolirana točka, ima vsaj enega soseda.) Ker ima točka, v katero tako pridemo, sodo mnogo sosedov, je v njej vsaj ena povezava, ki je še nismo prehodili. Zato lahko nadaljujemo sprehod po neki neprehojeni povezavi. Tak sklep velja tudi v vseh nadaljnjih točkah, ki jih srečamo na našem sprehodu, razen morda tedaj, ko pridemo nazaj v začetno točko v. Ker ima graf le končno mnogo povezav, se to prej ko slej tudi zgodi. Sedaj odstranimo povezave tako dobljenega obhoda iz G in induktivno v vsaki od preostalih povezanih komponent poiščemo Eulerjev obhod. To lahko storimo, saj so v ostanku vse točke sode stopnje — v vsaki točki smo namreč odvzeli sodo mnogo povezav, ki so bile na obhodu. Hitro se vidi, da iz prvotno dobljenega obhoda in Eulerjevih obhodov v preostalem delu lahko sestavimo Eulerjev obhod v G.

Gornjo induktivno (rekurzivno) konstrukcijo lahko zapišemo tudi v obliki algoritma, ki ob pravilni realizaciji porabi le linearen čas:



Slika 1: Iskanje Eulerjevega obhoda

```
procedure Eulerjev_obhod (v)
  { G naj bo graf s točkami sodih stopenj. }
  { Konstruiraj Eulerjev obhod v povezani komponenti, ki vsebuje v. }
begin
  if v nima incidenčnih povezav then vrni prazen obhod [v]
  else
    Začenši v točki v sestavi obhod W,
      ki ne porabi nobene povezave dvakrat.
    Zbriši povezave na W iz grafa G.
    Za vsako točko u na W ponovi:
        S klicem Eulerjev_obhod(u) poišči Eulerjev obhod W_u
            v povezani komponenti grafa, ki vsebuje u.
        Odstrani povezave obhoda W_u iz grafa G.
        Na mestu pojavitve u na W vstavi dobljeni obhod W_u.
    Vrni dobljeni sestavljeni obhod.
  endelse
end.
```

□

Delo gornjega postopka je shematično prikazano na primeru na sliki 1. Oznake na povezavah povedo, katera po vrsti na obhodu je označena povezava.

Dana naj bo simetrična matrika razdalj $D = [d(i,j)]_{i,j=1,\ldots,n}$. Recimo, da je G (vpet) Eulerjev graf na točkah $\{1, 2, \ldots, n\}$. Označimo:

$$d(G) := \sum_{e \in E(G)} d(e)\,,$$

kjer je $d(e) = d(i,j)$, če povezava e povezuje točki i in j.

Izrek 4.2 *Naj bo G Eulerjev graf na točkah $\{1, 2, \ldots, n\}$. Dana naj bo matrika razdalj $D = [d(i,j)]_{i,j=1,\ldots,n}$ za metrični problem trgovskega potnika. Tedaj v času $O(|E(G)|)$ lahko najdemo obhod trgovskega potnika, ki ima vsoto razdalj manjšo ali enako $d(G)$.*

Dokaz. Naj bo W Eulerjev obhod v G. Ker je G povezan, W obišče vse točke $1, 2, \ldots, n$. Zato W lahko zapišemo v obliki

$$W = i_1 W_1 i_2 W_2 \ldots i_n W_n\,,$$

kjer so $i_1, i_2, \ldots, i_n$ ravno vse točke $1, 2, \ldots, n$, vsaka natanko enkrat, vmesne poti W_k pa so lahko tudi trivialne. (V primeru več možnosti za izbiro podzaporedja $i_1, \ldots, i_n$, ki obišče vse točke, vzamemo katerokoli tako podzaporedje.) Zaradi trikotniške neenakosti je vsota cen povezav na W_k gotovo večja ali kvečjemu enaka $d(i_k, i_{k+1})$. Od tod trivialno sledi, da cena dobljenega obhoda $(i_1, i_2, \ldots, i_n)$ ne more presegati skupne cene vseh povezav v G.

Prej smo že videli, da lahko Eulerjev obhod v G poiščemo v linearnem času. Bralcu pa prepuščamo veselje, da sam ugotovi, kako lahko zatem v linearnem času določimo še potrebno podzaporedje $i_1, i_2, \ldots, i_n$. □

Naslednji postopek poišče obhod trgovskega potnika skozi danih n mest:

1. Poišči minimalno vpeto drevo T v polnem grafu na n točkah, kjer so cene povezav enake danim razdaljam za nalogo trgovskega potnika.
2. Podvôji vsako povezavo v T, da dobiš vpet Eulerjev graf.
3. Poišči Eulerjev obhod v tako dobljenem grafu.
4. Iz dobljenega obhoda določi obhod trgovskega potnika.

Izrek 4.3 *Gornji postopek z uporabo minimalnega vpetega drevesa je 1-aproksimacijski algoritem za reševanje metričnega problema trgovskega potnika. Njegova časovna zahtevnost je $O(Ln^2)$, kjer je L največja dolžina števil $d(i,j)$ (zapisanih v desetiškem sistemu z decimalno vejico).*

Dokaz. Videli smo že, da gornji postopek res najde obhod trgovskega potnika. Dokazati moramo le še to, da ta obhod ne more biti več kot dvakrat daljši od optimalnega. Le-tega sicer ne poznamo (in ga je zelo težko najti), a potrebovali ga bomo le za primerjavo s T. Recimo torej, da $\Omega^* = (i_1, i_2, \ldots, i_n)$ določa optimalni obhod trgovskega potnika. Iz trikotniške neenakosti in simetričnosti naših razdalj sledi $d(i_k, i_k) \leq d(i_k, i_{k+1}) + d(i_{k+1}, i_k) = 2d(i_k, i_{k+1})$. Zato je $d(i_k, i_{k+1}) \geq 0$. To velja za vsak indeks k med 1 in n; seveda pa je pri $k = n$ indeks $k+1$ mišljen "po modulu" n, torej enak 1. Naj bo sedaj T' vpeto drevo, sestavljeno iz povezav $i_1i_2, i_2i_3, \ldots, i_{n-1}i_n$. Tedaj je:

$$d(\Omega^*) = d(T') + d(i_n, i_1) \geq d(T') \geq d(T)\,. \tag{5}$$

Eulerjev graf, dobljen iz T, ima vsoto vrednosti povezav enako $2d(T)$. Če W in Ω označujeta dobljeni Eulerjev obhod in ustrezni obhod trgovskega potnika, tedaj od tod, z uporabo trikotniške neenakosti in iz (5) dobimo:

$$d(\Omega) \leq d(W) = 2d(T) \leq 2d(\Omega^*)\,,$$

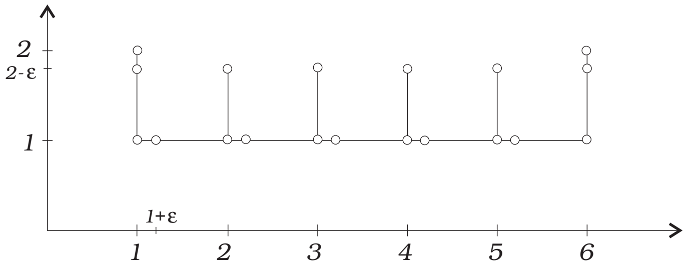
kar smo tudi želeli dokazati.

Poglejmo še časovno zahtevnost našega postopka. Naj bo $n = |V(G)|$. Tedaj iskanje najcenejšega vpetega drevesa porabi $O(Ln^2)$ časa, ostali trije koraki pa se dajo opraviti v linearnem času $O(n)$. Skupna časovna zahtevnost je torej taka, kot pravi izrek. □

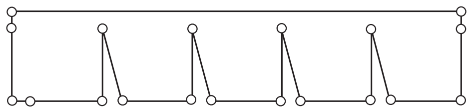
Oglejmo si postopek še na zgledu. Recimo, da imamo $3n+1$ mest, ki ustrezajo točkam v ravnini, kot je prikazano na sliki 2 (za $n = 6$) in katerih medsebojne razdalje so enake običajnim razdaljam v $\boldsymbol{R}^2$. Predpostavimo, da je ε dovolj majhen. Hitro se vidi, da je ustrezno najcenejše vpeto drevo tako, kot je prikazano na sliki 2. Podvojimo mu povezave in vzemimo Eulerjev obhod, ki "se začne" v točki $(1,2)$, gre navzdol, zavije v (2,1), skoči do točke $(2, 2-\varepsilon)$ in nazaj, gre v (3,1), spet gor in dol itd. Ko tako pridemo do $(n,1)$, storimo enako, zatem pa se vrnemo "naravnost" na začetek. Ustrezni obhod trgovskega potnika je prikazan na sliki 3.

Slika 4 kaže, da dobljeni obhod ni optimalen. Če je ε dovolj majhen, pa se hitro vidi, da je dobljeni obhod zelo blizu dvakratni vrednosti obhoda s slike 4.

Opisani zgledi (za poljuben n) dokazujejo, da obdelani postopek v nekaterih primerih najde rešitve, ki so poljubno blizu dvojni vrednosti optimalne rešitve. Zato ta algoritem ni r-aproksimacijski za noben $r < 1$.



Slika 2: Minimalno vpeto drevo



Slika 3: Ustrezni obhod trgovskega potnika

5 Christofidesova izboljšava

Enostavno, a učinkovito izboljšavo postopka iz prejšnjega razdelka je našel Christofides [1]. Pri tem postopku najprej poiščemo najcenejše vpeto drevo T, zatem pa namesto podvojitve vseh povezav poiščemo najcenejše prirejanje med točkami lihe stopnje v T. Natančneje povedano: vzamemo vse točke v T, ki so lihe stopnje v T, in vse možne povezave med njimi. Dobljeni (polni) graf ima sodo mnogo točk, saj ima vsak graf sodo mnogo točk lihe stopnje, torej tudi T. V tem grafu z uteženimi povezavami — uteži se ujemajo z razdaljami iz naloge trgovskega potnika — poiščemo najcenejše maksimalno prirejanje, torej prirejanje, ki pokrije vse lihe točke iz T in je vsota razdalj $d(i,j)$ na njem minimalna. Naj bo M dobljeno najcenejše prirejanje. Če drevesu T dodamo povezave iz M, dobimo Eulerjev graf. Označimo ga z G. V tako dobljenem grafu poiščemo Eulerjev obhod in iz njega dobimo obhod Ω trgovskega potnika. Pokazali bomo, da za dolžino obhoda Ω velja:

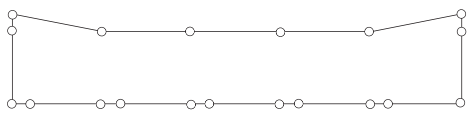
$$d(\Omega) \leq d(G) \leq \frac{3}{2}\,\mathrm{Opt}(I)\,. \tag{6}$$

Z drugimi besedami:

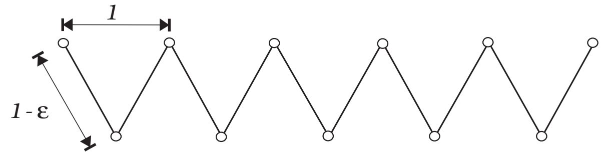
Izrek 5.1 *Zgoraj opisani postopek je $\frac{1}{2}$–aproksimacijski algoritem za metrični problem trgovskega potnika. Njegova časovna zahtevnost je $O(Ln^3)$, kjer je L največja dolžina števil $d(i,j)$ (zapisanih v desetiškem sistemu z decimalno vejico).*

Dokaz. Označimo s T dobljeno minimalno vpeto drevo in z M minimalno uteženo prirejanje na lihih točkah drevesa T. Naj bo W Eulerjev sprehod v $G = T \cup M$ in Ω ustrezni obhod trgovskega potnika. Po izreku 4.2 je vsota razdalj $d(\Omega)$ na obhodu Ω manjša ali kvečjemu enaka $d(G)$, torej:

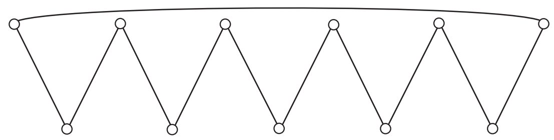
$$d(\Omega) \leq d(G) = d(T) + d(M)\,. \tag{7}$$



Slika 4: Boljši obhod trgovskega potnika



Slika 5: Najcenejše vpeto drevo



Slika 6: Dobljeni obhod trgovskega potnika

V dokazu izreka 4.3 smo videli, da za optimalni obhod Ω^* velja neenakost (5). Naj bodo sedaj $i_1, i_2, \ldots, i_{2m}$ točke lihe stopnje v T, navedene v istem vrstnem redu, kot se pojavijo na optimalnem obhodu Ω^*. Naj bosta M_1 in M_2 naslednji prirejanji:

$$M_1 = \{i_1i_2, i_3i_4, \ldots, i_{2m-1}i_{2m}\} \quad \text{in}$$

$$M_2 = \{i_2i_3, i_4i_5, \ldots, i_{2m}i_1\}\,.$$

Zaradi trikotniške neenakosti velja:

$$d(\Omega^*) \geq d(M_1) + d(M_2) \geq 2\min\{d(M_1), d(M_2)\} \geq 2d(M)\,.$$

Od tod in iz prej dobljenih neenakosti (5) in (7) pa sledi:

$$d(\Omega) \leq d(T) + d(M) \leq d(\Omega^*) + \frac{1}{2}d(\Omega^*) = \frac{3}{2}d(\Omega^*)\,,$$

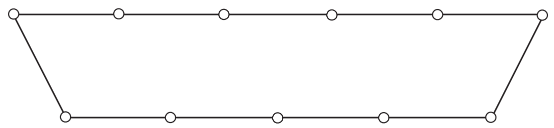
kar smo tudi želeli dokazati.

Poglejmo še časovno zahtevnost našega postopka. Naj bo $n = |V(G)|$. Iskanje najcenejšega vpetega drevesa porabi $O(Ln^2)$ časa, zadnja koraka pa sta tudi sedaj linearna. Korak 2 je časovno najrazsipnejši, saj lahko porabi kubično mnogo časa $O(Ln^3)$. Algoritem za iskanje najcenejšega maksimalnega prirejanja je opisan v [6, poglavje 11] in ga na tem mestu ne bomo podrobneje predstavili. Skupna časovna zahtevnost je torej taka, kot trdi izrek. □

Zgled za Christofidesov postopek bomo spet napravili na primeru, kjer bo dobljena aproksimativna rešitev poljubno blizu trem polovicam vrednosti optimalne rešitve. Zato ta algoritem ni r-aproksimacijski za noben $r < \frac{1}{2}$. Imamo Evklidski problem trgovskega potnika v ravnini. Ustrezne točke so prikazane na sliki 5, kjer je vrisano tudi najcenejše vpeto drevo. Na sliki 6 je prikazan dobljeni obhod trgovskega potnika, na sliki 7 pa še obhod, ki se od dobljenega razlikuje skoraj za faktor 3/2, če je ε dovolj majhen.

6 Boljši aproksimacijski postopki

Za metrični problem trgovskega potnika smo v prejšnjem razdelku opisali $\frac{1}{2}$-aproksimacijski postopek. Seveda bi bilo zanimivo najti hitre algoritme, za katere bi lahko pokazali, da še bolje aproksimirajo optimalne rešitve. Zaenkrat nam najboljšo znano aproksimacijo zagotavlja v prejšnjem razdelku opisani Christofidesov postopek. Vendar pa ni rečeno, da ne obstajajo še boljši algoritmi. Da pa se pokazati, da poljubno dobrih aproksimacij ne moremo dobiti na enostaven način:



Slika 7: Boljši obhod trgovskega potnika

Izrek 6.1 *Naj bo n število mest pri nalogi trgovskega potnika. Tedaj je iskanje* $\frac{1}{3n}$*–aproksimacije za metrični problem trgovskega potnika NP-težko.*

Dokaz izreka 6.1 je podoben dokazoma izrekov 2.1 in 3.1 (prevedba problema hamiltonovega cikla), zato ga opuščamo. Podrobnosti najdemo npr. v [3]. Problem $\frac{1}{3n}$–aproksimacije ostane težak tudi v primeru, ko se omejimo na Evklidski problem v ravnini $\boldsymbol{R}^2$.

Literatura

[1] N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem, v *Proc. Symp. on New Directions and Recent Results in Algorithms and Complexity*, Carnegie-Mellon University, Pittsburgh, 1976.

[2] M. R. Garey, D. S. Johnson, *Computers and Intractability: A guide to the theory of NP–completeness*, San Francisco, Freeman, 1979.

[3] D. S. Johnson, C. H. Papadimitriou, Performance guarantees for heuristics, v *The traveling salesman problem*, Ed. E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan in D. B. Shmoys, John Wiley, 1985, str. 145–180.

[4] J. Kozak, *Podatkovne strukture in algoritmi*, DMFA RS, Ljubljana, 1986.

[5] M. Lokar (urednik), *Problem trgovskega potnika*, Gradivo za 2. seminar, Izbrana poglavja iz računalništva, Oddelek za matematiko, Ljubljana, 1992.

[6] C. H. Papadimitriou, K. Steiglitz, *Combinatorial optimization: Algorithms and complexity*, Prentice-Hall, 1982.