Bandwidth problem minimizacije

Seminarski rad u okviru kursa
Naučno izračunavanje
Matematički fakultet

Nevena Nikolić, 1021/2018
Luka Kalinić, 1058/2018

Avgust 2019

Sažetak

U ovom radu najpre definišemo, a potom rešavamo *bandwidth* problem minimizacije, koristeći metod promenljivih okolina. Rad se u velikoj meri oslanja na dokument [1].

Sadržaj

1 Uvod

Bandwidth problem minimizacije (eng. *Bandwidth Minimization Problem*) se sastoji u pronalaženju permutacije redova i kolona retke matrice sa ciljem da ne-nula elementi budu sadržani u okviru trake, koja je što je moguće bliža glavnoj dijagonali. Ovaj problem je nastao pedesetih godina prošlog veka i zasniva se na primedbi da kada ne-nula elemente grupišemo u okviru uzane trake oko glavne dijagonale, operacije poput inverzije matrice i računanja determinante postaju vremenski efikasnije.

Primene ovog problema su brojne, a najvažnija je u rešavanju velikih sistema linearnih jednačina. Gausova eliminacija može da se izvede u vremenu $O(nb^2)$ za matricu čija je dimenzija n i *bandwidth* b, što je brže od direktnog $O(n^3)$ algoritma kada je b manje od n. Dokazano je da je *bandwidth* problem minimizacije NP-kompletan, odnosno da nije veoma verovatno da postoji algoritam za njegovo rešavanje koji radi u polinomijalnom vremenu.

Algoritmi za rešavanje *bandwidth* problema minimizacije mogu da se podele na egzaktne i heurističke, a u novije vreme razvijeno je i više metaheuristika. Metaheuristike predstavljaju opšte šablone čijim preciziranjem se dolazi do heurističkih metoda za konkretne probleme. [3].

2 Formulacija problema

Bandwidth problem minimizacije može biti formulisan za matrice ili grafove, pri čemu važi da su obe formulacije problema ekvivalentne. Ekvivalencija je zasnovana na činjenici da se dati graf može predstaviti svojom matricom povezanosti i obratno.

2.1 *Bandwidth* problem minimizacije za matrice

Neka je data binarna (svi elementi su 0 ili 1), retka, simetrična matrica $A = \{a_{ij}\}$. Tada se *bandwidth* matrice A definiše kao

$$B(A) = max\{|i - j| : a_{ij} \neq 0\} \tag{1}$$

Bandwidth problem minimizacije za matrice (eng. *Matrix Bandwidth Minimization Problem*, skraćeno *MBMP*) se sastoji u permutovanju redova i kolona matrice A tako da ne-nula elementi budu sadržani u traci što je moguće bližoj glavnoj dijagonali, odnosno tako da se minimizuje *bandwidth* $B(A)$.

2.2 *Bandwidth* problem minimizacije za grafove

Neka je $G = (V, E)$ konačan neusmereni graf, gde je V skup čvorova i E skup grana. Neka je 1-1 funkcija $f : V \rightarrow \{1, 2, ..., n\}$ funkcija obeležavanja čvorova grafa. Tada se *bandwidth* čvora v definiše kao

$$B_f(v) = max_{i:(i,j)\in E}\{|f(i) - f(j)|\} \tag{2}$$

i *bandwidth* grafa G za f se definiše kao

$$B_f(F) = max\{|f(i) - f(j)| : (i, j) \in E\} \tag{3}$$

Bandwidth problem minimizacije za grafove (eng. *Graph Bandwidth Minimization Problem*, skraćeno *GBMP*) se sastoji u pronalaženju funkcije obeležavanja f koja minimizuje *bandwidth* grafa, odnosno veličinu $B_f(G)$.

Za graf sa n čvorova, broj mogućih obeležavanja je $n!$. Direktan pristup mogao bi da bude isprobavanje svih permutacija i izbor najboljeg rešenja. Međutim, vreme izvršavanja ovog metoda je $O(n!)$, te ovaj pristup postaje nepraktičan već za grafove koji imaju 10 čvorova. Imajući u vidu ekvivalenciju dve navedene verzije problema, u nastavku ćemo podrazumevati *bandwidth* problem minimizacije za grafove.

3 Metode za rešavanje

Kao što smo već pomenuli, algoritmi za rešavanje *bandwidth* problema minimizacije mogu da se podele na egzaktne i heurističke (odnosno metaheurističke).

Egzaktni algoritmi se najčešće baziraju na pristupu poznatom kao *grananje i ograničavanje* (eng. *branch and bound*). Del Korso i Manzini (*Del Corso G.M., G. Manzini*, 1999) su uspeli da reše problem za instance male i srednje veličine. Kaprara i Salazar (*Caprara A. and J.J. Salazar*, 2005) su uspeli da prošire prethodni rezultat uvodeći uže donje granice. Međutim, treba uzeti u obzir i cenu izračunavanja za dobijanje optimalnog rešenja. Stoga ovi metodi mogu rešavati samo probleme čija je veličina dovoljno mala da vreme izvršavanja bude razumno.

Heuristički pristupi zasnovani su na iskustvu i dobijeno rešenje ne mora biti optimalno. Ipak, heurističkim pristupom možemo brzo pronaći rešenje koje je dovoljno dobro za dati problem kombinatorne optimizacije. Metaheuristike, kao što smo pomenuli, predstavljaju tehniku koja je opštija od heuristika i zahtevaju manje računskih pretpostavki, te je poslednjih decenija akcenat stavljen na razvijanje metaheurističkih metoda za rešavanje *bandwidth* problema minimizacije. U literaturi se najčešće pominju tri ovakve metaheuristike: simulirano kaljenje (eng. *Simulated Annealing*, SA), tabu pretraga (eng. *Tabu Search*, TS) i metod promenljivih okolina (eng. *Variable Neighborhood Search*, VNS). U nastavku ćemo se fokusirati na rešavanje *bandwidth* problema minimizacije koristeći metod promenljivih okolina, za koji je eksperimentalno pokazano da ima bolje performanse od ostalih pristupa.

4 VNS pristup rešavanju problema

Metod promenljivih okolina (VNS) je metaheuristika predložena 1997. godine koja se pokazala veoma korisnom za dobijanje približnog rešenja optimizacionih problema. Predstavlja uopštenje lokalne pretrage, gde su definisane okoline po kojima će se sistematski vršiti pretraživanje [2]. VNS se bazira na sledećim principima:

- Lokalni minimum za jedan tip okoline ne mora nužno i da bude lokalni minimum za drugi tip okoline.
- Globalni minimum predstavlja lokalni minimum za sve tipove okolina.

- Za mnoge probleme u praksi, lokalni minimumi više tipova okolina su relativno blizu jedan drugog.

U literaturi postoji nekoliko varijanti osnovnog algoritma, među kojima se najčešće sreću redukovani, osnovni i uopšteni metod promenljivih okolina. U svim slučajevima se, ukoliko nije pronađeno bolje rešenje, prelazi u narednu okolinu, a inače, ukoliko se pronađe bolje rešenje, algoritam se resetuje na prvi razmatrani tip okoline.

Osnovni VNS algoritam uključuje tri postupka: razmrdavanje, lokalnu pretragu i promenu okoline. Razmrdavanje predstavlja pokušaj iskakanja iz tekućeg lokalnog optimuma i pronalaženja novog lokalno optimalnog rešenja koje će biti bliže globalno optimalnom rešenju. Za pronalaženje lokalnog optimuma koristi se lokalna pretraga kako bi se unapredila preciznost pretrage. *Pomeriti se ili ne* (eng. *move or not*) označava menjanje okoline, što nam daje iterativni metod i kriterijum zaustavljanja. Pseudokod osnovnog VNS algoritma može se prikazati na sledeći način:

- Odabrati tipove okolina $U_l(x), 1 \leq l \leq l_{max}$
- Generisati početno rešenje x i postaviti vrednost najboljeg rešenja $f^* = f(x)$
- Dok nije zadovoljen kriterijum zaustavljanja, ponavljati sledeće korake:
  - $l = 1$
  - Dok je $l \leq l_{max}$ ponavljati sledeće korake:
    - Izabrati proizvoljno rešenje x' u okolini $U_l(x)$
    - Primenom nekog tipa lokalne pretrage na početno rešenje x' naći rešenje x'' sa najmanjom vrednošću funkcije cilja
    - Ako rešenje x'' ima manju vrednost funkcije cilja od rešenja x, onda dodeliti $x = x''$ i $l = 1$, a inače $l = l + 1$
  - Po potrebi, ažurirati vrednost f^*
- Ispisati vrednost f^*

4.1 Početno rešenje

Kvalitet početnog rešenja će direktno uticati na performanse algoritma. Dobro početno rešenje može garantovati da će za kratko vreme biti dobijeno rešenje koje je globalno optimalno ili blisko optimalnom.

Dobro početno rešenje može se generisati korišćenjem pretrage u širinu (eng. *breadth first search*, BFS). Osnovna ideja je da bi povezani čvorovi trebalo da imaju bliske oznake. Struktura nivoa (*level structure*) grafa, u oznaci $L(G)$, jeste particionisanje njegovih čvorova u nivoe $L_1, L_2, ..., L_k$ koji zadovoljavaju sledeće uslove:

1. Čvorovi povezani sa čvorom iz nivoa L_1 su ili u L_1 ili u L_2
2. Čvorovi povezani sa čvorom iz nivoa L_k su ili u L_k ili u L_{k-1}
3. Čvorovi povezani sa čvorom iz nivoa L_i (za $1 < i < k$) su ili u L_{i-1} ili u L_i ili u L_{i+1}

Prema tome, mogu se dobiti razumno dobra rešenja. Dakle, procedura za generisanje početnog rešenja se sastoji u primeni BFS algoritma sa nasumičnim izborom početnog čvora; različiti izbori početnog čvora vode različitim početnim rešenjima. Prema strukturi nivoa, sva početna rešenja su bolja od originalnog dodeljivanja. Očigledno, *bandwidth* koji se dobija

ovim metodom ne može biti gori od maksimalnog *bandwidth*-a grafa, jer su susednim čvorovima dodeljeni uzastopni brojevi. BFS metod daje gornju granicu za kvalitetno rešenje.

4.2 Razmrdavanje

Literatura

[1] Abdel Lisser Chen Wang, Chuan Xu. Bandwidth minimization problem. In *MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation*, Nancy, France, November 2014. URL: https://hal.archives-ouvertes.fr/hal-01166658/document.

[2] Stefan Mišković. Metoda promenljivih okolina - beleške sa vežbi. URL: http://ni.matf.bg.ac.rs/vezbe/4/vns.html.

[3] Anđelka Zečević Mladen Nikolić. *Naučno Izračunavanje*. Matematički fakultet, Beograd.