Metoda reznih ravnina za problem sparivanja stabala

Blažević, Mislav

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Department of Mathematics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Odjel za matematiku**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:126:232382

Sveučilište Josipa Jurja Strossmayera u Osijeku
Odjel za matematiku

Sveučilišni diplomski studij matematike
smjer: Matematika i računarstvo

Metoda reznih ravnina za problem sparivanja stabala

Diplomski rad

Mentor:

izv. prof. dr. sc.
Domagoj Matijević

Kandidat:

Mislav Blažević

Osijek, 2022

Sadržaj

1 | Uvod

Motivirani problemima koji proizlaze iz primijenjenih područja, proučavamo problem sparivanja stabala. U ovom radu taj problem sagledavamo iz perspektive linearnog programiranja. Specifično, bavimo se proučavanjem kombinatornih svojstava reznih ravnina poliedra određenog klikama pridruženog proizvoljnoj instanci problema sparivanja. Prezentirana svojstva motiviraju algoritme koji koristeći dinamičko programiranje u polinomnom vremenu pronalaze određene zanimljive klase reznih ravnina.

Glavni dio ovog rada je rezultat istraživačkog rada (vidi [2]) autora i mentora u suradnji s Stefanom Canzarom[1] i Khaledom Elbassionijem[2]

U poglavlju 2 definiramo osnovne pojmove i prezentiramo relevantne rezultate iz matematičkog konteksta u kojem se algoritmi nalaze.

U poglavlju 3 proučavamo kombinatorna svojstva posebnog slučaja problema i prezentiramo polinoman algoritam koji koristeći ta svojstva optimalno rješava posebni slučaj.

U poglavlju 4 prezentiramo polinoman algoritam koji proširujući algoritam iz poglavlja 3 pronalazi netrivijalnu donju među za problem separacije.

[1]Gene Center, Ludwig-Maximilians-Universität München, 81377 Munich, Germany
[2]Khalifa University of Science and Technology, Abu Dhabi, UAE

2 | Osnovni pojmovi

U ovom poglavlju ćemo definirati neke osnovne pojmove i predstaviti relevantne elementarne rezultate i algoritme u svrhu motiviranja algoritama iz narednih poglavlja.

2.1 Metoda reznih ravnina

Definicija 2.1. *Neka je $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$. Skup $\{x \in \mathbb{R}^n : Ax \geq b\}$ zovemo* ***poliedar****.*

Definicija 2.2. *Neka je $a \in \mathbb{R}^n$, $a \neq 0$ i $b \in \mathbb{R}$. Skup*

$$H_{a,b} = \{x \in \mathbb{R}^n : a^T x = b\}$$

zovemo ***hiperravnina*** *a skupove*

$$\begin{aligned} H^+_{a,b} &= \{x \in \mathbb{R}^n : a^T x \geq b\}, \\ H^-_{a,b} &= \{x \in \mathbb{R}^n : a^T x \leq b\} \end{aligned}$$

zovemo ***poluprostori*** *određeni hiperravninom $H_{a,b}$.*

Lako se može pokazati da su poluprostor i poliedar konveksni skupovi i da je presjek konačno mnogo konveksnih skupova konveksan skup [8]. Iz definicije vidimo da hiperravnina prirodno rastavlja prostor na dva poluprostora čiji je presjek ta hiperravnina. Time je motivirana sljedeća definicija.

Definicija 2.3. *Neka su $x, y \in \mathbb{R}^n$. Za hiperravninu $H_{a,b} = \{x \in \mathbb{R}^n : a^T x = b\}$ ćemo reći da* ***separira*** *x i y ukoliko je $x \in H^+_{a,b} \iff y \in H^-_{a,b}$.*

Neka je $\mathcal{P} \subseteq \mathbb{R}^n$ poliedar i $c \in \mathbb{R}^n$. Problem pronalaska

$$\operatorname*{argmin}_{x \in \mathcal{P}} c^T x$$

zovemo **problem linearnog programiranja**, a problem pronalaska

$$\operatorname*{argmin}_{x \in \mathcal{P} \cap \mathbb{Z}^n} c^T x$$

zovemo **problem cjelobrojnog programiranja**. Problem linearnog programiranja moguće je riješiti u polinomnom vremenu [1], dok za problem cjelobrojnog programiranja nije poznat polinoman algoritam. Stoga problem cjelobrojnog programiranja rješavamo aproksimacijskim algoritmima, heurističkim algoritmima ili egzaktnim algoritmima eksponencijalne složenosti.

Jedna klasa algoritama za rješavanje problema cjelobrojnog programiranja bazirana na algoritmima za rješavanje problema linearnog programiranja je metoda reznih ravnina. U toj metodi generiramo niz poliedara $(\mathcal{P}_k)_k$ takvih da je

$$\mathcal{P} \supseteq \mathcal{P}_1 \supseteq \mathcal{P}_2 \supseteq \cdots \supseteq \mathcal{P} \cap Z^n$$

na način da iterativno pronalazimo hiperravninu koja separira $\operatorname{argmin}_{x\in\mathcal{P}_k} c^T x$ i $\operatorname{argmin}_{x\in\mathcal{P}\cap\mathbb{Z}^n} c^T x$ i presječemo poliedar $\mathcal{P}_k$ s odgovarajućim poluprostorom. Time u svakom koraku sužavamo prostor pretrage što nam može olakšati pronalazak rješenja. Osim toga, tada očigledno imamo

$$\min_{x\in\mathcal{P}} c^T x \leq \min_{x\in\mathcal{P}_1} c^T x \leq \min_{x\in\mathcal{P}_2} c^T x \leq \cdots \leq \min_{x\in\mathcal{P}\cap\mathbb{Z}^n} c^T x.$$

To jest, postupak iterativno poboljšava donju među broja $\min_{x\in\mathcal{P}\cap\mathbb{Z}^n} c^T x$ što je korisno za metodu grananja i ograničavanja. Ukoliko koristimo i ravnine dane Gomoryjevim algoritmom (vidi [1]), nakon konačno mnogo koraka pronalazimo poliedar $\mathcal{P}_k$ takav da je

$$\min_{x\in\mathcal{P}_k} c^T x = \min_{x\in\mathcal{P}\cap\mathbb{Z}^n} c^T x.$$

Međutim, u najgorem slučaju Gomoryjev algoritam ima eksponencijalno mnogo koraka [4] a i u praksi Gomoryjeve ravnine režu jako malen dio poliedra pa ih ne koristimo.

2.2 Poliedar stabilnih skupova

Definicija 2.4. *Neka je $G = (V, E)$ graf i $S \subseteq V$. Za S kažemo da je* ***stabilan skup*** *ukoliko $\{u, v\} \not\subseteq S$ za sve $\{u, v\} \in E$. Za S kažemo da je* ***klika*** *ukoliko $\{u, v\} \in E$ za sve $\{u, v\} \subseteq S$.*

Problem stabilnog skupa je vrlo jednostavan i fundamentalan primjer problema cjelobrojnog programiranja na grafu. Mnoge probleme, uključujući problem sparivanja stabala, prirodno je promatrati kao poseban slučaj problema stabilnog skupa. Stoga razmatramo poliedar tog problema općenito.

Definicija 2.5. *Neka je $G = (V, E)$ graf i $S \subseteq V$. Za $x \in \{0, 1\}^{|V|}$ ćemo reći da je* ***vektor indikator*** *skupa S ukoliko je*

$$x_v = \begin{cases} 1, & v \in S \\ 0, & v \in V \setminus S \end{cases}$$

Lako vidimo da je $S \subseteq V$ stabilan skup ako i samo ako njegov vektor indikator x zadovoljava

$$x_u + x_v \leq 1 \qquad \forall\{u, v\} \in E \tag{I1}$$

$$0 \leq x_v \leq 1 \qquad \forall v \in V \tag{I2}$$

čime je motivirana sljedeća definicija.

Definicija 2.6. *Neka je $G = (V, E)$ graf. Konveksnu ljusku skupa svih indikator vektora stabilnih skupova grafa G zovemo* ***poliedar stabilnih skupova****. Vektor $x \in \mathbb{R}^{|V|}$ koji zadovoljava I1 i I2 zovemo* ***frakcionalan stabilan skup*** *dok skup svih frakcionalnih stabilnih skupova zovemo* ***poliedar frakcionalnih stabilnih skupova*** *grafa G.*

Uočimo da je poliedar stabilnih skupova podskup poliedra frakcionalnih stabilnih skupova. Štoviše, imamo sljedeću karakterizaciju.

Propozicija 2.1. [9] *Neka je $G = (V, E)$ graf, $\mathcal{P}$ poliedar stabilnih skupova i α vektor indikator skupa $S \subseteq V$. Tada je $\alpha^T x \leq 1$ za sve $x \in \mathcal{P}$ ako i samo ako je S klika. Nadalje, ta nejednakost predstavlja stranu poliedra ako i samo ako je klika S maksimalna u smislu inkluzije.*

Time je motivirana sljedeća definicija.

Definicija 2.7. *Neka je $G = (V, E)$ graf. Poliedar*

$$\mathcal{P} = \{x \in \mathbb{R}^{|V|} : \alpha^T x \leq 1 \text{ za svaki indikator klike } \alpha\} \cap \{x \in \mathbb{R}^{|V|} : x \geq 0\}$$

zovemo ***poliedar određen klikama*** *grafa G.*

Neka je $\mathcal{P}_1$ poliedar stabilnih skupova, $\mathcal{P}_2$ poliedar određen klikama i $\mathcal{P}_3$ poliedar frakcionalnih stabilnih skupova. Očigledno imamo $\mathcal{P}_1 \subseteq \mathcal{P}_2 \subseteq \mathcal{P}_3$ s time da je $\mathcal{P}_2 = \mathcal{P}_3$ ako i samo ako graf ne sadrži trokut [9]. Sljedeća propozicija govori kada je $\mathcal{P}_1 = \mathcal{P}_2$.

Definicija 2.8. *Neka je $G = (V, E)$ graf. Najmanji prirodan broj n takav da postoji preslikavanje $f : V \to \{1, \ldots, n\}$ takvo da $f(u) \neq f(v)$ za sve $\{u, v\} \in E$ zovemo* ***kromatski broj grafa*** *i za graf G kažemo da je* ***savršen*** *ukoliko je kromatski broj svakog induciranog podgrafa jednak veličini najveće klike u tom podgrafu.*

Propozicija 2.2. [9] *Neka je $G = (V, E)$ graf. Poliedar stabilnih skupova grafa G jednak je poliedru određenom klikama ako i samo ako je G savršen.*

Imajući na umu gore navedene rezultate, promotrimo sljedeći problem. Neka je $G = (V, E)$ graf, $\mathcal{P}$ poliedar stabilnih skupova grafa G, $w : V \to \mathbb{Z}^+$ težinska funkcija. Problem pronalaska

$$\operatorname*{argmax}_{x \in \mathcal{P}} \sum_{v \in V} w(v) x_v$$

zovemo **problem težinskog stabilnog skupa**. Tom problemu možemo pristupiti metodom reznih ravnina. Specifično, pomoću Algoritma 1 sužavamo prostor pretrage s bilo kojeg poliedra koji sadrži poliedar $\mathcal{Q}$ određen klikama na poliedar $\mathcal{P}$ za koji vrijedi

$$\max_{x \in \mathcal{P}} \sum_{v \in V} w(v) x_v = \max_{x \in \mathcal{Q}} \sum_{v \in V} w(v) x_v$$

nakon čega nekom drugom metodom odabiremo $y \in \mathcal{P} \cap \{0, 1\}^{|V|}$.

Problem pronalaska klike maksimalne težine je $\mathcal{NP}$-hard [7] tako da u općem slučaju se ne možemo nadati efikasnom pronalasku poliedra određenog klikama, ali korištenjem kombinatorne strukture partikularnog problema možemo efikasno pronaći neke klase klika, čime ćemo se baviti u narednim poglavljima.

Algoritam 1 Pronalazak optimuma iz poliedra određenog klikama

1: **procedure** CLIQUE-CUT($\mathcal{P}_1$)
2: $\quad i \leftarrow 1$
3: $\quad$ **while** postoji klika s linije 5 **do**
4: $\quad\quad y^{(i)} \leftarrow \operatorname{argmax}_{x \in \mathcal{P}_i} \sum_{v \in V} w(v) x_v$
5: $\quad\quad S_i \leftarrow$ klika za koju je $\sum_{v \in S_i} y_v^{(i)} > 1$
6: $\quad\quad \mathcal{P}_{i+1} \leftarrow \mathcal{P}_i \cap \{x \in \mathbb{R}^{|V|} : \sum_{v \in S_i} x_v \leq 1\}$
7: $\quad\quad i \leftarrow i + 1$
8: $\quad$ **return** $\mathcal{P}_i$

2.3 Metoda grananja i ograničavanja

Neka je poliedrom $\mathcal{P} \subseteq \mathbb{R}^n$ i vektorom $c \in \mathbb{R}^n$ zadan maksimizacijski problem cjelobrojnog programiranja koji dodatno zadovoljava $0 \leq x \leq 1$ za sve $x \in \mathcal{P}$. To jest, problem pronalaska

$$\operatorname*{argmax}_{x \in \mathcal{P} \cap \{0,1\}^n} c^T x.$$

Ukoliko imamo na raspolaganju efikasan algoritam za računanje netrivijalne gornje ograde težine rješenja tog problema, možemo ga iskoristiti prilikom iscrpnog pretraživanja na način da odbacimo podstablo pretraživanja ukoliko je u njemu gornja ograda rješenja manja od težine najboljeg pronađenog dopustivog rješenja. Za odabir dopustivog rješenja $z \in \mathcal{P} \cap \{0,1\}^n$ možemo primjerice postupiti pohlepno kao u [3] no postoje i druge mogućnosti.

Primjenjujući metodu na problem težinskog stabilnog skupa koristeći poliedar određen klikama za pronalazak gornje ograde i koristeći pohlepnu metodu (ovdje nazvanu GREEDY) iz [3] za pronalazak cjelobrojnog rješenja, dobivamo Algoritam 2. Radi jednostavnosti notacije pretpostavljamo da su vrhovi grafa numerirani od 1 do $|V|$. Ulaz algoritma je poliedar $\mathcal{P}$ unutar kojeg tražimo optimalan težinski stabilni skup grafa $G = (V, E)$ uz težinsku funkciju $w : V \to \mathbb{Z}^+$.

Algoritam 2 Optimalan težinski stabilni skup metodom grananja i ograničavanja

procedure VAL(x)
$\quad$ **return** $\sum_{v \in V} w(v) \mathrm{x}_v$
procedure BNB($\mathcal{P}, i = 1$)
$\quad \mathcal{P} \leftarrow$ CLIQUE-CUT($\mathcal{P}$)
$\quad z \leftarrow$ GREEDY($\mathcal{P}$)
$\quad y \leftarrow \operatorname{argmax}_{x \in \{y,z\}}$ VAL(x) $\qquad \triangleright$ y je globalna varijabla
$\quad$ **if** $i \leq |V|$ **then**
$\quad\quad$ **if** $\max_{x \in \mathcal{P}}$ VAL(x) $>$ VAL(y) **then**
$\quad\quad\quad \mathcal{P}_0 \leftarrow \mathcal{P} \cap \{x \in \mathbb{R}^{|V|} : x_i = 0\}$
$\quad\quad\quad$ BNB($\mathcal{P}_0, i + 1$)
$\quad\quad\quad \mathcal{P}_1 \leftarrow \mathcal{P} \cap \{x \in \mathbb{R}^{|V|} : x_i = 1\}$
$\quad\quad\quad$ BNB($\mathcal{P}_1, i + 1$)

2.4 Sparivanje stabala

Neka je $G = (V, E)$ stablo s istaknutim korijenom, $x, y \in V$. Definiramo relacije

$$\begin{aligned} x < y &\iff \text{vrh } x \text{ je predak vrha } y, \\ x \sim y &\iff x \text{ je predak od } y \text{ ili } y \text{ je predak od } x \text{ ili } x = y. \end{aligned}$$

U ostatku teksta, koristeći definiciju iz [3], sva stabla ćemo smatrati ukorijenjenim i usmjerenim od korijena prema listovima. Posebno, ako je $P \subseteq V$ put, onda za sve $x, y \in P$ vrijedi $x \sim y$. Korijen stabla G ćemo označavati $r(G)$. Ukoliko je $x < y$, jedinstveni put od x do y ćemo označavati $[x, y]$. Za skup $A \subseteq V$ najnižeg zajedničkog pretka elemenata skupa A ćemo označavati $\mathrm{LCA}(A)$. Skup vrhova podstabla s korijenom u $x \in V$ ćemo označavati $\tau(x)$, skup djece vrha x ćemo označavati $c(x)$, a jedinstvenog roditelja vrha $x \neq r(G)$ ćemo označavati $p(x)$.

Definicija 2.9. *Neka su $G_1 = (V_1, E_1)$ i $G_2 = (V_2, E_2)$ stabla. Za $M \subseteq V_1 \times V_2$ kažemo da je* ***sparivanje*** *stabala G_1 i G_2 ukoliko vrijedi*

$$x \leq x' \iff y \leq y' \qquad \forall (x, y), (x', y') \in M. \tag{S1}$$

Nadalje, neka je $w : V_1 \times V_2 \to \mathbb{Z}^+$ težinska funkcija. Broj

$$w(M) = \sum_{(x,y) \in M} w(x, y)$$

ćemo zvati ***težina sparivanja*** *M. Za bilo koje sparivanje M koje zadovoljava*

$$M \in \operatorname*{argmax}_{M' \text{ sparivanje}} w(M')$$

ćemo reći da je ***optimalno****.*

U slučaju kada je barem jedno stablo put, postoji $O(|V_1||V_2|)$ algoritam za pronalazak optimalnog sparivanja [3].

Neka smo u situaciji Definicije 2.9. Dodatno definirajmo graf $G = (V_1 \times V_2, E)$, tako da elementi skupa E odgovaraju parovima bridova koji zadovoljavaju uvjet S1. Odnosno,

$$E = \{((x, y), (x', y')) : x, y \in V_1, y, y' \in V_2 \text{ t.d. } x \leq x' \iff y \leq y'\}.$$

Postoji trivijalna bijekcija između sparivanja grafova G_1, G_2 i klika grafa G. To jest, problem pronalaska optimalnog sparivanja stabala G_1 i G_2 je ekvivalentan problemu pronalaska klike maksimalne težine grafa G, odnosno stabilnog skupa maksimalne težine komplementa grafa G. Stoga uz efikasan algoritam za pronalazak stabilnih skupova u grafu G možemo postupiti kao u poglavlju 2.2. Time je motivirana sljedeća definicija.

Definicija 2.10. *Neka su $G_1 = (V_1, E_1)$ i $G_2 = (V_2, E_2)$ stabla. Za $M \subseteq V_1 \times V_2$ ćemo reći da je* ***antisparivanje*** *stabala G_1 i G_2 ukoliko vrijedi*

$$x \leq x' \iff y \not\leq y' \qquad \forall (x, y), (x', y') \in M. \tag{A1}$$

Postoji $O(|V_1||V_2|)$ algoritam za problem pronalaska antisparivanja maksimalne težine u slučaju kada su G_1 i G_2 putevi [3]. U poglavlju 4 prezentiramo polinoman algoritam za slučaj kada je barem jedan od grafova put. S obzirom da je opći slučaj antisparivanja naizgled težak, razmatramo prirodan posebni slučaj.

Definicija 2.11. *Neka su $G_1 = (V_1, E_1)$ i $G_2 = (V_2, E_2)$ stabla. Za $M \subseteq V_1 \times V_2$ ćemo reći da je* ***si-antisparivanje*** *stabala G_1 i G_2 ukoliko vrijedi*

$$x \sim x' \iff y \not\sim y' \qquad \forall (x, y), (x', y') \in M. \tag{B1}$$

Postoji $O(|V_1||V_2|)$ algoritam za problem pronalaska si-antisparivanja maksimalne težine u slučaju kada je barem jedan od grafova put [3]. U poglavlju 3 prezentiramo polinoman algoritam za opći slučaj. U poglavlju 4 ćemo vidjeti da kombiniranjem tog algoritma s algoritmom za pronalazak antisparivanja za slučaj kada je barem jedan od grafova put dobivamo netrivijalnu donju među težine maksimalnog antisparivanja.

Pojmove **težina antisparivanja**, **optimalno antisparivanje**, **težina si-antisparivanja** i **optimalno si-antisparivanje** definiramo analogno kao i za sparivanja.

3 | Si-antisparivanje

U ovom poglavlju prezentiramo algoritam za pronalazak optimalnog si-antisparivanja. U tu svrhu razmatramo kombinatorna svojstva si-antisparivanja iz kojih proizlazi polinoman algoritam. Naime, pokazat ćemo da si-antisparivanja imaju jednostavnu strukturu i da postoji uređaj koji omogućava konstrukciju polinomnog algoritma koristeći dinamičko programiranje. Također ćemo predstaviti $O(|V_1|^2|V_2|^2)$ algoritam koji koristi navedena svojstva.

3.1 Dekompozicija

Neka je M si-antisparivanje stabala $T_1 = (V_1, E_1)$ i $T_2 = (V_2, E_2)$. Za proizvoljne skupove $A \subseteq V_1$, $B \subseteq V_2$ označavat ćemo

$$\begin{aligned} M(A) &= \{y \in V_2 : (x,y) \in M \text{ za neki } x \in A\}, \\ M^{-1}(B) &= \{x \in V_1 : (x,y) \in M \text{ za neki } y \in B\}. \end{aligned}$$

Za proizvoljne $x \in V_1, y \in V_2$ uvodimo analogne oznake $M(x)$ i $M^{-1}(y)$. Nadalje, označavat ćemo

$$\begin{aligned} V_1^M &= \{x \in V_1 : (x,y) \in M \text{ za neki } y \in V_2\}, \\ V_2^M &= \{y \in V_2 : (x,y) \in M \text{ za neki } x \in V_1\}. \end{aligned}$$

Za proizvoljni podskup skupa V_1 ili V_2 uvodimo analogne oznake. Neka je dodatno P_1 put u T_1. Reći ćemo da je P_1 **si-maksimalan** ukoliko za svaki put Q_1 u T_1 vrijedi $|Q_1^M| \leq |P_1^M|$. Ključna opservacija je da postoji najviše jedan si-maksimalan put u T_1 koji sadrži barem dva elementa skupa V_1^M.

Lema 3.1. [2] *Neka je M si-antisparivanje i P_1 si-maksimalan put u T_1. Tada je $A_1 = V_1^M \setminus P_1^M$ antilanac.*

Dokaz. Pretpostavimo da A_1 nije antilanac. Tada postoje $x, y \in A_1$, $x \neq y$ takvi da je $x \sim y$. S obzirom da je P_1 si-maksimalan, postoje $u, v \in P_1^M$ takvi da je $u \not\sim x$, $u \not\sim y$, $v \not\sim x$, $v \not\sim y$. Međutim, tada imamo $M(u), M(v) \subseteq [r(T_2), \text{LCA}(M(x) \cup M(y))]$, što je kontradikcija. $\square$

Iz definicije si-antisparivanja trivijalno slijedi da je $M(P_1)$ antilanac. Naslućujemo da si-antisparivanja antilance preslikavaju u puteve, što vrijedi do na rubni slučaj.

Lema 3.2. [2] *Neka je M si-antisparivanje i $A_1 \subseteq V_1^M$ antilanac. Tada postoji put $P_2 \subseteq V_2$ takav da je $M(A_1) \subseteq P_2$ ili postoji $x \in A_1$ takav da je $M(A_1 \setminus \{x\}) \subseteq P_2$.*

Dokaz. Pretpostavimo ne postoji takav put P_2. Tada postoje $u, v \in M(A_1)$ takvi da je $u \not\sim v$. Stoga imamo $a \sim b$ za sve $a, b \in M^{-1}(\{u, v\}) \cap A_1$. No A_1 je antilanac, i stoga $M^{-1}(\{u, v\}) \cap A_1$ je jednočlan skup. Neka je x jedinstveni element tog skupa. Iz $x \not\sim y$ za sve $y \in A_1 \setminus \{x\}$ slijedi $u \sim z$ i $v \sim z$ za sve $z \in M(A_1 \setminus \{x\})$. Drugim riječima, $[r(T_2), \text{LCA}(u, v)]$ je traženi put. □

Specifično, zanima nas onaj antilanac, dobiven prikladnim odabirom si-maksimalnog puta u T_1, koji se preslikava u put i time ne ulazi u posebni slučaj prethodne leme.

Lema 3.3. [2] *Neka je M si-antisparivanje. Tada postoje si-maksimalan put $P_1 \subseteq V_1$ i put $P_2 \subseteq V_2$ takvi da je $M(V_1^M \setminus P_1^M) \subseteq P_2$.*

Dokaz. Neka je P_1' proizvoljan si-maksimalan put. Prema Lemi 3.1, $A_1 = V_1^M \setminus P_1'$ je antilanac. Pretpostavimo da ne postoji traženi put $P_2 \subseteq V_2$. Tada postoji $x \in A_1$ takav da elementi skupa $M(x)$ ne leže na istom putu. Tada je put $P_1 = [r(T_1), x]$ također si-maksimalan jer bi u suprotnom postojali $u, v \in P_1' \cap V_1^M$, $u \neq v$, $u \not\sim x$ i $v \not\sim x$ takvi da je $M(u), M(v) \subseteq [r(T_2), \text{LCA}(M(x))]$, što je kontradikcija s $u \sim v$. S obzirom da je P_1' si-maksimalan, maksimalni element (u smislu relacije $<$) skupa $P_1' \cap V_1^M$ zadovoljava $u \not\sim z$ za sve $z \in A_1$. To jest, $A_1 \cup \{u\}$ je antilanac i kao u dokazu Leme 3.2 zaključujemo da postoji put $P_2 \subseteq V_2$ takav da je $M(A_1 \cup \{u\} \setminus \{x\}) \subseteq P_2$. □

Iz prethodnih razmatranja, zamjenom uloga T_1 i T_2, direktno slijedi teorem o dekompoziciji.

Teorem 3.1. [2] *Neka je M si-antisparivanje. Tada postoje si-maksimalan put $P_1 \subseteq V_1$, antilanac $A_1 \subseteq V_1$, si-maksimalan put $P_2 \subseteq V_2$ i antilanac $A_2 \subseteq V_2$ takvi da su $V_1^M = P_1^M \cup A_1$ i $V_2^M = P_2^M \cup A_2$ disjunktne unije, $M(A_1) \subseteq P_2$ i $A_2 = M(P_1)$.*

3.2 Si-antisparivanje stabla i puta

Neka su $T_1 = (V_1, E_1)$ i $T_2 = (V_2, E_2)$ stabla, $x \in V_1$, $v, v' \in V_2$, $v < v'$. Označimo s $\alpha(x, v, v')$ težinu optimalnog si-antisparivanja podstabla ukorijenjenog u x i puta $[v, v']$. Koristimo istu oznaku i u slučaju zamjene uloga T_1 i T_2. Poznato je (vidi npr. [3]) da je taj broj jednak težini antilanca maksimalne težine u podstablu ukorijenjenom u vrhu x s težinskom funkcijom $w : V_1 \to \mathbb{Z}^+$ definiranom izrazom

$$w(x) = \sum_{z \in [v, v']} w(x, z).$$

Elementarna je zadaća pronaći algoritam koji koristeći dinamičko programiranje na stablu računa taj broj u $O(|V_1|)$. Nazovimo taj algoritam TREE-AC. U svrhu pronalaska optimalnog si-antisparivanja, potrebni su nam $\alpha(x, v, v')$ za sve x, v, v'. Algoritam 3 efikasno proširuje TREE-AC računajući tražene vrijednosti u vremenu $O(|V_1|^2|V_2| + |V_1||V_2|^2)$. Uočimo da ga je potrebno izvršiti i sa zamijenjenim ulogama stabala T_1 i T_2.

Algoritam 3 Računanje težine optimalnog si-antisparivanja za svaki par stabla i puta

procedure ALPHA
  for $x \in V_1$ **do**
    $w'(x) \leftarrow 0$ ▷ w' je globalna varijabla
  for $v \in V_2$ **do**
    DFS(v, v)
procedure DFS(v, v')
  for $x \in V_1$ **do**
    $w'(x) \leftarrow w'(x) + w(x, v')$
  $\alpha(\cdot, v, v') \leftarrow$ TREE-AC(T_1, w') ▷ α je izlaz algoritma
  for $v'' \in c(v')$ **do**
    DFS(v, v'')
  for $x \in V_1$ **do**
    $w'(x) \leftarrow w'(x) - w(x, v')$

3.3 Si-antisparivanje stabla i stabla

Neka smo u situaciji Teorema 3.1. Rastavimo članove antilanca A_1 s obzirom na podstablo s korijenom u P_1 u kojem se nalaze. Formalno, za $x \in A_1$ označimo s $l(x)$ najnižeg pretka od x koji se nalazi u P_1. Tada je $\{(x,y) \in A_1 \times A_1 : l(x) = l(y)\}$ relacija ekvivalencije. Neka su $B_1, \ldots, B_n$ klase ekvivalencije te relacije. Učinimo analogno s A_2 i klase ekvivalencije označimo $C_1, \ldots, C_m$. Bez smanjenja općenitosti, pretpostavljat ćemo $l(b_1) < l(b_2) < \ldots < l(b_n)$, za $b_1 \in B_1, \ldots, b_n \in B_n$, i analogno $l(c_1) < l(c_2) < \ldots < l(c_m)$, za $c_1 \in C_1, \ldots, c_m \in C_m$. Nadalje, za $x \in P_1^M$ označimo $L(x) = \{l(c_j) : C_j \cap M(x) \neq \emptyset\}$.

Uz te oznake možemo pokazati da postoji uređaj si-antisparivanja koji omogućava primjenu dinamičkog programiranja.

Teorem 3.2. [2] *Neka smo u situaciji Teorema 3.1 i neka je $x \in P_1^M$. Tada vrijedi*

$$l(b_i) < x \iff u \leq v \qquad \forall u \in M(B_i)\ \forall v \in L(x)$$

Dokaz.

$\boxed{\Longleftarrow}$ Pretpostavimo da postoji B_i takav da je $l(b_i) \geq x$ i $u \leq v$ za neke $u \in M(B_i)$ i $v \in L(x)$. Tada postoji $w \in M(x)$ takav da je $u \sim w$. Neka je $y \in B_i \cap M^{-1}(u)$. Tada je $x \sim y$ zbog $l(b_i) \geq x$, što je kontradikcija s pretpostavkom da je M si-antisparivanje.

$\boxed{\Longrightarrow}$ Pretpostavimo da postoji B_i takav da je $l(b_i) < x$ i $u > v$ za neke $u \in M(B_i)$ i $v \in L(x)$. Tada postoji $w \in M(x)$ takav da je $z \not\sim w$. Neka je $y \in B_i \cap M^{-1}(u)$. Tada je $x \not\sim y$ zbog $l(b_i) < x$, što je kontradikcija s pretpostavkom da je M si-antisparivanje. □

Sada je struktura si-antisparivanja u potpunosti opisana i možemo pokazati kako pomoću dinamičkog progamiranja u polinomnom vremenu pronaći optimalno si-antisparivanje.

Teorem 3.3. [2] *Neka je M optimalno si-antisparivanje stabala $T_1 = (V_1, E_1)$ i $T_2 = (V_2, E_2)$, i neka je $f : (V_1 \times V_2) \cup (V_2 \times V_1) \to \mathbb{Z}^+$ funkcija definirana izrazom*

$$f(u,v) = \max \begin{cases} \max\limits_{\substack{u' \in \tau(u) \setminus \{u\} \\ v' \in \tau(v) \setminus \{v\}}} \left[f(v',u') + \sum\limits_{x \in [u, p(u')]} \sum\limits_{y \in c(x) \setminus [u,u']} \alpha(y,v,v') \right] \\ \max\limits_{v' \in \tau(v)} \alpha(u,v,v') \end{cases} .$$

Tada vrijedi $w(M) = \max \begin{cases} f(r(T_1), r(T_2)) \\ f(r(T_2), r(T_1)) \end{cases}$.

Dokaz.

$\boxed{\geq}$ Neka je $u' \in \tau(u) \setminus \{u\}$ i $v' \in \tau(v) \setminus \{v\}$. Unija $\mathcal{A}$ antilanaca iz podstabala s korijenima iz $c([u, p(u')]) \setminus [u, u']$ je antilanac, pa je $\mathcal{A} \times [v, v']$ si-antisparivanje. Zbog $x \not\sim y$ za sve $x \in \tau(u'), y \in \mathcal{A}$ i $x \sim y$ za sve $x \in [v, v'], y \in \tau(v')$ induktivno zaključujemo da je svaki skup kojeg algoritam nalazi si-antisparivanje.

$\boxed{\leq}$ Neka je M neprazno si-antisparivanje. ako je $P_1^M = \emptyset$ onda je $w(M) \leq \alpha(r(T_1), q_1, q_r)$, gdje su q_1 i q_r redom minimalan i maksimalan (u smislu relacije $<$) element skupa P_2^M. Ako je $P_1^M = \{p_1, ..., p_k\}$, b.s.o. $p_1 < p_2 < ... < p_k$. Lako se vidi da nije smanjenje općenitosti pretpostaviti $l(b_1) < p_1$. Napravimo particiju si-antisparivanja $M = G \cup F$ tako da je $G = M \cap ((V_1 \setminus \tau(p_1)) \times M(V_1 \setminus \tau(p_1)))$ i $F = M \setminus G$. Imamo $w(G) \leq \sum_{x \in [r(T_1), p(p_1)]} \sum_{y \in c(x) \setminus [r(T_1), p_1]} \alpha(y, q_1, q_r)$ gdje su q_1 i q_r redom minimalan i maksimalan (u smislu relacije $<$) element skupa $M(G)$. Nastavljajući induktivno na podstablima s korijenima q_r i p_1 i si-antisparivanjem F dobivamo $w(F) \leq f(q_r, p_1)$. $\square$

Optimalno si-antisparivanje moguće je pronaći u vremenu $O(|V_1|^2 |V_2|^2)$. To je ilustrirano Algoritmom 4. Nakon inicijalnog poziva SELECT-PATH$(r(T_1), r(T_2))$, algoritam alternira između odabira puta i odabira antilanca. Varijablom s efikasno održavamo težinu trenutno razmatranog antilanca koristeći istu tehniku koja je korištena i u Algoritmu 3.

Algoritam 4 Računanje težine optimalnog si-antisparivanja

procedure SELECT-PATH(u, v)
    $\delta \leftarrow 0$
    for $v' \in c(v)$ **do**
        $\mu \leftarrow$ SELECT-ANTICHAIN$(v, u, v', 0)$
        $\lambda \leftarrow$ SELECT-PATH(u, v')
        $\delta \leftarrow \max\{\lambda, \mu, \delta\}$
    return δ
procedure SELECT-ANTICHAIN(v, y, v', s)
    if $c(v') = \emptyset$ **then**
        return $\alpha(u, v, v')$
    $\delta \leftarrow 0$
    for $u' \in c(y)$ **do**
        $s \leftarrow s + \sum_{z \in c(p(u')) \setminus \{u'\}} \alpha(z, v, v')$
        $\mu \leftarrow$ SELECT-PATH(v', u')
        $\lambda \leftarrow$ SELECT-ANTICHAIN(v, u', v', s)
        $\delta \leftarrow \max\{\lambda, s + \mu, \delta\}$
        $s \leftarrow s - \sum_{z \in c(p(u')) \setminus \{u'\}} \alpha(z, v, v')$
    return δ

4 | Antisparivanje

Za problem optimalnog antisparivanja u posebnom slučaju dva puta postoji $O(|V_1||V_2|)$ algoritam [5]. Sljedeći teorem generalizira taj algoritam na slučaj antisparivanja stabla i puta.

Teorem 4.1. [2] *Neka je M optimalno antisparivanje puta $[u,v]$ i stabla ukorijenjenog u x. Tada vrijedi*

$$w(M) = \gamma(x,u,v) := w(v,x) + \begin{cases} \sum\limits_{y \in c(x)} \gamma(y,u,v) & v = u \\ \max \begin{cases} \sum\limits_{y \in c(x)} \gamma(y,u,v) \\ \gamma(x,u,p(v)) \end{cases} & v \neq u \end{cases}$$

Dokaz.

$\boxed{\geq}$ Neka je (k,l) brid odabran na nekom koraku algoritma. Tada svaki element skupa $[u,k] \times \tau(l)$ tvori antisparivanje s (k,l). Induktivno, svaki skup pronađen algoritmom je antisparivanje.

$\boxed{\leq}$ Neka je M antisparivanje. Tada je $M' = M \cup \{(v,x)\}$ antisparivanje za koje vrijedi $w(M) \leq w(M')$. Neka je k jedinstveni minimalni element skupa $\{k' \in [u,v] : (k',x) \in M'\}$. Imamo $(z,w) \notin M'$ za sve $(z,w) \in ([k,v] \setminus \{k\}) \times (\tau(x) \setminus \{x\})$. Stoga postoji particija $M' = P \cup T$, gdje je $P = ([k,v] \times \{x\}) \cap M'$ i $T = ([u,k] \times (\tau(x) \setminus \{x\})) \cap M'$. Očigledno vrijedi $w(P) \leq \sum_{z \in [k,v]} w(z,x)$ i induktivno dobivamo $w(T) \leq \sum_{y \in c(x)} \gamma(y,u,k)$. □

S obzirom da je u fiksan, dinamički program ima $O(|V_1||V_2|)$ stanja i svaki brid stabla algoritam prolazi $O(|V_2|)$ puta. Zaključujemo da je vremenska složenost algoritma $O(|V_1||V_2|)$. Teorem 3.3 stoga sugerira trivijalnu generalizaciju Teorema 4.1 na slučaj antisparivanja stabla i stabla jednostavnom zamjenom α s γ.

Korolar 4.1. [2] *Neka je M optimalno antisparivanje stabala $T_1 = (V_1, E_1)$ i $T_2 = (V_2, E_2)$, i neka je $f : (V_1 \times V_2) \cup (V_2 \times V_1) \to \mathbb{Z}^+$ funkcija definirana izrazom*

$$f(u,v) = \max \begin{cases} \max\limits_{\substack{u' \in \tau(u) \setminus \{u\} \\ v' \in \tau(v) \setminus \{v\}}} \left[f(v',u') + \sum\limits_{x \in [u,p(u')]} \sum\limits_{y \in c(x) \setminus [u,u']} \gamma(y,v,v') \right] \\ \max\limits_{v' \in \tau(v)} \gamma(u,v,v') \end{cases}.$$

Tada vrijedi $w(M) \geq \max \begin{cases} f(r(T_1), r(T_2)) \\ f(r(T_2), r(T_1)) \end{cases}.$

Zaključak

U ovom radu smo razmotrili problem pronalaska reznih ravnina za problem sparivanja stabala. Vidjeli smo da je taj problem ekvivalentan pronalasku klika u prikladno konstruiranom grafu. Kombinatorno smo okarakterizirali prirodno definiranu netrivijalnu klasu tih klika i prezentirali polinoman algoritam koji koristeći dinamičko programiranje pronalazi maksimalnu kliku iz te klase. Efikasan algoritam za pronalazak takvih klika nam u spoju s algoritmom za rješavanje problema linearnog programiranja daje ogradu težine optimalnog rješenja originalnog problema. Ogradu težine optimalnog rješenja moguće je iskoristiti za efikasniji pronalazak samog rješenja. Rezultati ovog rada sugeriraju da je razmatranje analogno definiranih klika za druge probleme potencijalno plodonosan pravac istraživanja.

Literatura

[1] D. Bertsimas, J.N. Tsitsiklis, *Introduction to linear optimization*, Athena Scientific, 1997.

[2] M. Blažević, S. Canzar, K. Elbassioni, D. Matijević, *Anti Tai Mapping for Unordered Labeled Trees*, arXiv [cs.DS], 2021. Dostupno na `http://arxiv.org/abs/2107.08292`

[3] M. Blažević, *Sparivanje usmjerenih acikličkih grafova*, 2020. Dostupno na `https://urn.nsk.hr/urn:nbn:hr:126:101395`

[4] S. Dash, *An Exponential Lower Bound on the Length of Some Classes of Branch-and-Cut Proofs*, Integer Programming and Combinatorial Optimization, IPCO 2002. Lecture Notes in Computer Science, vol 2337. Springer, Berlin, Heidelberg.

[5] V. H. Do, M. Blažević, P. Monteagudo, L. Borozan, K. Elbassioni, S. Laue, F. Rojas Ringeling, D. Matijević, S. Canzar, *Dynamic pseudo-time warping of complex single-cell trajectories*, 23nd Annual International Conference on Research in Computational Molecular Biology, The George Washington University, 2019, 294-297

[6] M. Grötschel, L. Lovász, A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, 1993.

[7] R.M. Karp, *Reducibility among Combinatorial Problems*, Complexity of Computer Computations, 1972. The IBM Research Symposia Series. Springer, Boston, MA.

[8] I. Kuzmanović, K. Sabo, *Linearno programiranje*, Sveučilište Josipa Jurja Strossmayera u Osijeku, 2016.

[9] A. Schrijver, *Combinatorial Optimization — Polyhedra and Efficiency*, Springer, Berlin, 2003.

Sažetak

U ovom radu se bavimo problemom generiranja reznih ravnina za poliedar stabilnih skupova problema sparivanja stabala. Prezentiramo strukturu problema i algoritme koji iz njih proizlaze.

Ključne riječi

sparivanja, stabla, linearno programiranje, rezne ravnine

Cutting plane method for the tree matching problem

Summary

In this paper, we consider the problem of generating cutting planes for the stable set polytope of tree matching problem. We present the structure of the problem and the algorithms which they motivate.

Keywords

matchings, trees, linear programming, cutting planes

Životopis

Rođen sam u Vinkovcima 1996. Godine 2015. završavam srednjoškolsko obrazovanje u Gimnaziji Matije Antuna Reljkovića u Vinkovcima i upisujem sveučilišni preddiplomski studij Matematike na Odjelu za matematiku Sveučilišta Josipa Jurja Strossmayera u Osijeku. Preddiplomski studij završavam 2020. godine i upisujem sveučilišni diplomski studij matematike, smjer: Matematika i računarstvo na istom odjelu.