Steinerova stabla

Vojković, Fran

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Fran Vojković

STEINEROVA STABLA

Diplomski rad

Voditelj rada:
prof. dr. sc. Robert Manger

Zagreb, 2021.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Parentibus et omnibus magistrus meis

Sadržaj

Uvod

Klasični problem Steinerovog stabla glasi ovako: Zadan je skup točaka u metričkom prostoru, navedene točke zovu se terminalne točke. Treba pronaći najkraću mrežu koja međusobno povezuje sve terminalne točke. Rješenje problema ima oblik stabla i zove se Steinerovo minimalno stablo koje osim terminalnih točaka može sadržavati i dodatne točke koje se zovu Steinerove točke. Navedeni problem jedan je od poznatijih problema kombinatorne optimizacije sličan problemu nalaženja minimalnog razapinjućeg stabla uz novinu, odnosno dodatak pomoćnih točaka takozvanih Steinerovih točaka. Problem minimalnog Steinerovog stabla ima mnoge primjere u području računalnih mreža. Postoji opsežna literatura o Steinerovom problemu kao i mnogo instanci problema nastalih uslijed raznih primjena Steinerovog problema. Navodimo tri, danas glavne varijante Steinerovog problema: *Euklidski Steinerov problem* (ESP), *pravocrtni Steinerov problem* (RSP -*Rectilinear Steiner problem*) i *Steinerov problem u mrežama*. U prvom poglavlju iskazani su osnovni pojmovi i rezultati teorije grafova potrebni za daljnje razmatranje Steinerovih stabla. Drugo poglavlje opisuje Euklidski Steinerov problem te su prikazani algoritmi pomoću kojih se rješava navedni problem. Treće poglavlje bavi se Steinerovim problemom u mrežama, iskazani su algoritmi pomoću kojih konstruiramo egzaktna i približna rješenja.

Poglavlje 1

Uvod u teoriju grafova

U ovom poglavlju navodimo neke osnovne definicije i rezultate koji su potrebni za daljnje razmatranje Steinerovog problema.

Definicija 1.0.1. *Za uređen par $\mathcal{G} = (V, E)$ kažemo da je* graf*, pri čemu je $V = \{v_1, v_2, \ldots\}$ skup vrhova grafa, a E skup dvočlanih podskupova od V koji zovemo skupom bridova grafa $\mathcal{G}$.* Mreža*, odnosno* težinski graf *je par $(\mathcal{G}, \ell)$, gdje je $G = (V, E)$ neusmjeren graf te $\ell \colon E \to \mathbb{R}_0^+$* težinska funkcija*. Broj $\sum_{e \in E} \ell(e)$ zovemo* težinom grafa*.*

U ovom radu pretpostavit ćemo da su skupovi bridova i vrhova konačni osim ako ne naglasimo drugačije. Za graf se ponekad koristi i drugi naziv, mreža. Slika 1.1 prikazuje graf $\mathcal{G} = (V, E)$ s pripadnim skupom vrhova $V = \{v_0, v_1, v_2, \ldots, v_8\}$ te skupom bridova $E = \{e_1, e_2, \ldots, e_8\}$.

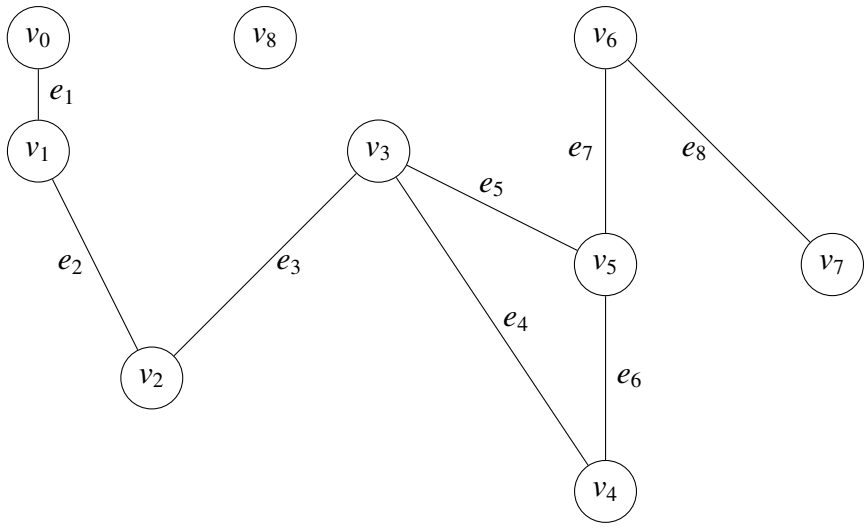
Za vrh $v \in V$ i brid $e \in E$ kažemo da su *incidentni* ako postoji vrh $w \in V$ takav da je $e = \{v, w\}$. U primjeru sa slike 1.1 vrh v_2 i brid e_3 su incidentni jer postoji vrh v_3 za koji vrijedi $e_3 = \{v_2, v_3\}$. Također, za vrhove $v, w \in V$ kažemo da su *susjedni* ako postoji brid $\{v, w\} \in E$, u prethodnom primjeru vrhovi v_1 i v_2 su susjedni. *Petlja* je brid koji spaja vrh sa samim sobom.

Definicija 1.0.2. Stupanj vrha *v u oznaci $d(v)$ je broj bridova koji su incidentni sa vrhom v. Petlja stupnju vrha s kojim je incidentna doprinosi sa* 2*.*

U grafu sa slike 1.1 imamo sljedeće stupnjeve vrhova, $d(v_1) = 2$, $d(v_2) = 2$, $d(v_5) = 3$. Vrh čiji je stupanj jednak 0 zovemo *izolirani vrh*, kao što je v_8.

Teorem 1.0.3. *Za proizvoljan graf $\mathcal{G} = (V, E)$ vrijedi da je broj bridova grafa $|E|$ jednak sumi stupnjeva svih vrhova podijeljeno sa* 2*, tj.*

$$|E| = \frac{\sum_{i=1}^{n} d(v_i)}{2} \tag{1.1}$$



Slika 1.1: Primjer grafa.

Dokaz. Tvrdnja teorema je očita, proizlazi iz toga što svaki brid dvaput doprinosi sumi vrhova. □

Definicija 1.0.4. Šetnja *u grafu* $\mathcal{G} = (V, E)$ *je niz vrhova* $(v_1, v_2, \ldots, v_n)$ *pri čemu su vrhovi* v_i *i* v_{i+1} *susjedni za* $i = 1, 2, \ldots, n-1$. Staza *je šetnja u kojoj su svi bridovi različiti, no mogući su isti vrhovi.* Put *je staza u kojoj su svi vrhovi različiti, osim eventualno prvog – takav put nazivamo* ciklusom.

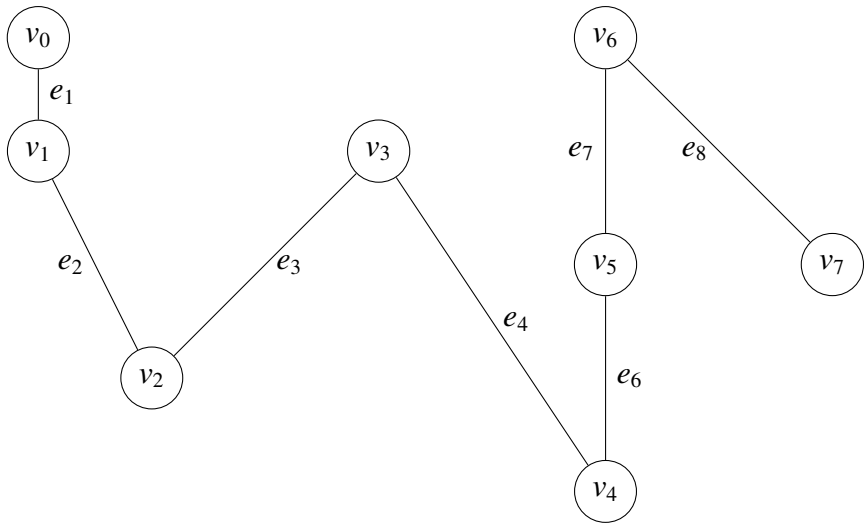
U prethodnom primjeru (v_3, v_4, v_5, v_6) je primjer puta koji nije ciklus, dok je (v_3, v_4, v_5, v_3) primjer jednog ciklusa. U daljem razmatranju promatrat ćemo *jednostavne grafove*, odnosno neusmjerene grafove bez višestrukih bridova. Za graf kažemo da je *povezan* ako postoji put između svaka dva vrha grafa, primjer sa slike 1.1 nije povezan graf jer ne postoji put do vrha v_8.

Definicija 1.0.5. *Neka je* $\mathcal{G} = (V, E)$ *graf. Za graf* $\mathcal{G}$ *kažemo da je stablo ako ne sadrži cikluse i izolirane vrhove te ako postoji put između svaka dva vrha iz* V.

Teorem 1.0.6. *Neka je* $\mathcal{G} = (V, E)$ *konačan graf. Tada su sljedeće tvrdnje ekvivalentne:*

1. $\mathcal{G}$ *je stablo,*
2. $\mathcal{G}$ *ne sadrži cikluse i vrijedi* $|E| = |V| - 1$,
3. $\mathcal{G}$ *je povezan graf i vrijedi* $|E| = |V| - 1$.

Primjer stabla dan je na slici 1.2.



Slika 1.2: Primjer stabla.

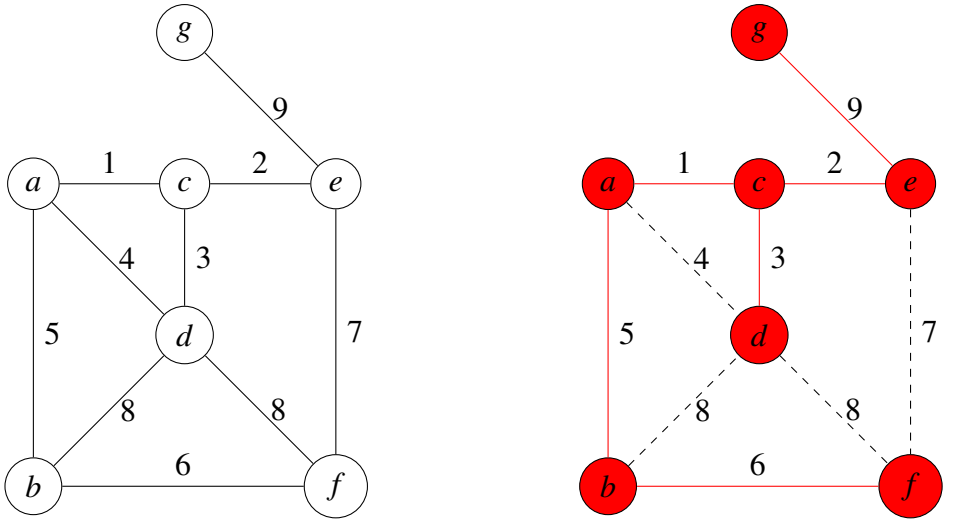
Definicija 1.0.7. *Neka je $\mathcal{G} = (V, E)$ graf.* Podgraf *grafa $\mathcal{G} = (V, E)$ je graf $\mathcal{G}' = (V', E')$ za kojeg vrijedi $V' \subseteq V$ i $E' \subseteq E$ te za svaki brid $e = \{v, w\} \in E'$ vrijedi da su oba njegova vrha u V', tj. $v, w \in V'$.*

Slika 1.2 upravo prikazuje podgraf grafa sa slike 1.1.

1.1 Minimalno razapinjuće stablo

Definicija 1.1.1. Težinski graf *je par $(\mathcal{G}, \ell)$, gdje je $G = (V, E)$ neusmjeren graf te $\ell\colon E \to \mathbb{R}_0^+$* težinska funkcija. *Broj $\sum_{e \in E} \ell(e)$ zovemo* težinom grafa.

Prirodno se postavlja pitanje može li se odrediti minimalno (u smislu težine) povezano stablo. Odgovor je potvrdan te navodimo dva algoritma kojima možemo konstruirati takvo minimalno razapinjuće stablo.



(a) Zadano stablo.

(b) Minimalno razapinjuće stablo.

Slika 1.3: Primjer Kruskalova algoritma 1

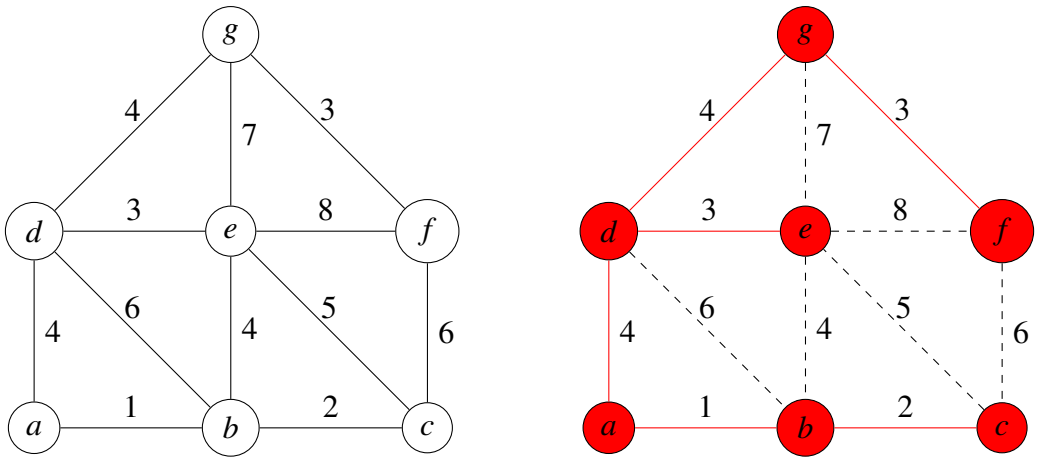
Algoritam 1 Kruskalov algoritam

Ulaz: Neka $G = (V, E)$ povezan graf te $\ell: E \to \mathbb{R}_0^+$ težinska funkcija.
Izlaz: Minimalno razapinjuće stablo (V, S).

1. Postavimo $S = \emptyset$.
2. Sve dok graf (V, S) nije povezan ponavljaj:
   a) Odaberi brid $e \in E \setminus S$ minimalne težine takav da u $(V, S \cup \{e\})$ ne postoji ciklus. Postavi $S = S \cup \{e\}$.

Ilustrirat ćemo Kruskalov algoritam na primjeru sa grafom $\mathcal{G} = (V, E)$ od 7 vrhova, prikazanom na slici 1.3. Algoritam počinje tako da postavimo skup $S = \emptyset$, skup S označava skup bridova minimalnog razapinjućeg stabla kojeg ćemo konstruirati algoritmom. Najprije odaberemo brid minimalne težine iz E te ga dodamo u skup S, pa imamo $S = \{\{a, c\}\}$. U drugom koraku također odaberemo brid najmanje težine iz E, no pazimo da taj brid ne stvara ciklus ako ga dodamo u skup S. Nakon drugog koraka dodali smo brid $\{e, c\}$ u skup S. Na isti način nastavimo dok graf (V, S) nije povezan, navedeni graf predstavlja traženo minimalno razapinjuće stablo prikazano na slici 1.3b. Složenost Kruskalova algoritma je $\mathbb{O}(|V|\log(|V|))$.

Navodimo i drugi poznati algoritam za konstrukciju minimalnog razapinjućeg stabla.



(a) Zadano stablo. (b) Minimalno razapinjuće stablo.

Slika 1.4: Primjer Primova algoritma 2

Algoritam 2 Primov algoritam

Ulaz: Neka $G = (V, E)$ povezan graf te $\ell: E \to \mathbb{R}_0^+$ težinska funkcija. Označimo $|V| = n$.
Izlaz: Minimalno razapinjuće stablo (V, F).

1. Za proizvoljni $v_0 \in V$ definiramo $T = \{v_0\}, S = V \setminus \{v_0\}, F = \emptyset$.
2. Sve dok je $|F| < n - 1$ ponavljaj:
   a) Odaberi brid $e = \{v, w\} \in E$ minimalne težine takav da je $v \in T, w \in S$. Postavi $T = T \cup \{w\}, F = F \cup \{e\}, S = S \setminus \{w\}$.

Primov algoritam ćemo ilustrirati na primjeru sa grafom $\mathcal{G} = (V, E)$ od 7 vrhova, prikazanom na slici 1.4. Početno izaberemo proizvoljan vrh, neka je to a. Definiramo sa $T = \{a\}$ i $T = V \setminus \{a\}$ pomoćne skupove vrhova te sa $F = \emptyset$ skup bridova. U prvom koraku odaberemo brid minimalne težine iz E takav da se jedan vrh tog brida nalazi u skupu T, a drugi u skupu S. U primjeru to je očito brid $\{a, b\}$, pa ga dodamo u skup F, a vrh b izbacimo iz skupa S te ga dodamo u skup T. Kako je $|F| = 1 < 7 - 1$ nastavljamo sa drugim korakom na isti način. Odaberemo brid $\{a, c\}$ zato što je najmanji takav da se jedan vrh nalazi u skupu T, a drugi u skupu S. Također, kao i u prethodnom koraku dodamo brid u skup F, a vrh c izbacimo iz skupa S te ga dodamo u skup T. Nakon 6 koraka algoritma dolazimo do konačnog rješenja prikazanog na slici 1.4b, odnosno konstruirali smo minimalno razapinjuće stablo za zadano stablo.

Poglavlje 2

Steinerov problem u Euklidskom prostoru

Povijesno gledano, Steinerov problem je zapravo generalizacija *Fermatovog problema*, koji glasi: Neka su u prostoru dane tri točke, potrebno je odrediti točku u prostoru tako da je suma njenih udaljenosti do zadane tri točke minimalna. Kako bi ilustrirali problem pretpostavimo da su dane tri točke u ravnini: $(0,0), (2,0)$ i $(1, \sqrt{3})$, kao na slici 2.1.

Kako bi definirali Steinerov problem u Euklidskom prostoru i usporedili ga sa problemom minimalnog razapinjućeg stabla potrebna nam je definicija metričkog prostora.

Definicija 2.0.1. *Neka je $N \neq \emptyset$ neprazan skup i $f: N \times N \to \mathbb{R}$ nenegativno, simetrično preslikavanje koje zadovoljava nejednakost trokuta. Odnosno f je preslikavanje takvo da vrijedi:*

$$(\forall x, y \in N) f(x,y) \geq 0 \tag{2.1}$$

$$f(x,y) = 0 \iff x = y \tag{2.2}$$

$$(\forall x, y \in N) f(x,y) = f(y,x) \tag{2.3}$$

$$(\forall x, y, z \in N) f(x,y) \leq f(x,z) + f(z,y). \tag{2.4}$$

Kažemo da je f funkcija udaljenosti, odnosno metrika *na skupu N. Tada uređeni par $(N, f) =: \mathcal{S}$ zovemo* metrički prostor.

Napomena 2.0.2. *Primjetimo, u prethodnoj definiciji aksiomi 2.1 i 2.3 su suvišni jer slijede iz aksioma 2.2 i 2.4. Aksiom 2.1 dobijemo na sljedeći način: $(\forall y, x) 0 = f(x,x) \leq f(x,y) + f(y,x) = 2f(x,y)$. Slično i za aksiom 2.3, kako je $f(x,x) = 0$ po 2.2, imamo iz 2.4 da vrijedi $f(x,y) \leq f(y,x)$. Ako zamijenimo x sa y i z sa y iz 2.4 slijedi $f(y,x) \leq f(x,y)$, odnosno vrijedi 2.3.*

Slijedi intuitivan opis Steinerovog problema u metričkom prostoru koji ćemo kasnije formalizirati. Neka je dan skup točaka V u metričkom prostoru. Problem se svodi na nalaženje mreže, najkraće ukupne duljine, koja povezuje sve točke iz V. Optimalno rješenje poprima strukturu stabla te od tuda dolazi naziv rješenja Steinerovo minimalno stablo (SMT–*Steiner minimum tree*). Kao što je prethodno opisano SMT može sadržavati točke koje nisu sadržane u skupu V, te točke nazivamo *Steinerove točke*, dok točke iz skupa V nazivamo *terminalne točke*, odnosno regularne točke. Slijedi formalna definicija Steinerovog problema.

Steinerov problem u metričkom prostoru: Neka je dan skup terminalnih točaka $V = \{v_1, \ldots, v_n\}$ u metričkom prostoru $\mathcal{S} = (W, l)$, pri čemu je $W \supseteq V$. Potrebno je naći Steinerovo stablo T za dani skup terminalnih točaka V, odnosno stablo T takvo da je ukupna duljina bridova u T minimalna, tj. potrebno je minimizirati $l(T) = \sum_{e \in T} l(e)$. Duljinu brida $l(e)$ računamo kao udaljenost odgovarajućih vrhova u skladu sa metrikom. Pri čemu stablo T sadrži sve terminalne točke te može sadržavati dodatne Steinerove točke koje se ne nalaze u skupu V, odnosno nalaze se u nadskupu W.

Manjom modifikacijom problema nalaženja Steinerovog stabla dolazimo do srodnog problema, nalaženja minimalnog razapinjućeg stabla. Ako prethodnom problemu isključimo mogućnost dodavanja Steinerovih točaka u stablo T, odnosno ako se ograničimo samo na skup terminalnih točaka dolazimo do problema minimalnog razapinjućeg stabla. Slijedi formalni opis problema.

Problem minimalnog razapinjućeg stabla u Metričkom prostoru: Neka je $V = \{v_1, \ldots, v_n\}$ skup terminalnih točaka u metričkom prostoru $\mathcal{S} = (W, l)$, pri čemu je $W \supseteq V$. Minimalno razapinjuće stablo T za dani skup točaka V je stablo čija je ukupna duljina svih bridova minimalna, $l(T) = \sum_{e \in T} l(e)$. Duljinu brida $l(e)$ računamo kao udaljenost odgovarajućih vrhova u skladu sa metrikom, pri čemu stablo T sadrži samo terminalne točke iz zadanog skupa V.

Kako bi ilustrirali razliku između minimalnog razapinjućeg stabla i Steinerovog minimalnog stabla, vratimo se na primjer sa slike 2.1. Skup terminalnih točaka se sastoji od tri zadane točke. Slika 2.1a prikazuje minimalno razapinjuće stablo čija je ukupna duljina uz standardnu euklidsku normu jednaka 4. Dok slika 2.1b prikazuje Steinerovo minimalno stablo koje sadrži jednu Steinerovu točku te je ukupna duljina tog stabla $2\sqrt{3}$. Problem Steinerovog stabla je NP-težak dok je problem minimalnog razapinjućeg stabla rješiv u polinomijalnom vremenu zahvaljujući *Primovom* i *Kruskalovom* algoritmu.

Steinerov problem u Euklidskom prostoru zapravo je ekvivalentan Steinerovom problemu u metričkom prostoru uz napomenu da se kao težinska funkcija ℓ koristi Euklidska

(a) Minimalno razapinjuće stablo. (b) Steinerovo minimalno stablo.

Slika 2.1: Usporedba minimalnog razapinjućeg stabla i Steinerovog minimalnog stabla za zadane tri točke u ravnini.

norma, tj. za dan brid $e = \{v_1, v_2\}$ pri čemu su $v_1 = (x_1, y_1), v_2 = (x_2, y_2) \in \mathbb{R}^2$ vrijedi:

$$\ell(e) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{2.5}$$

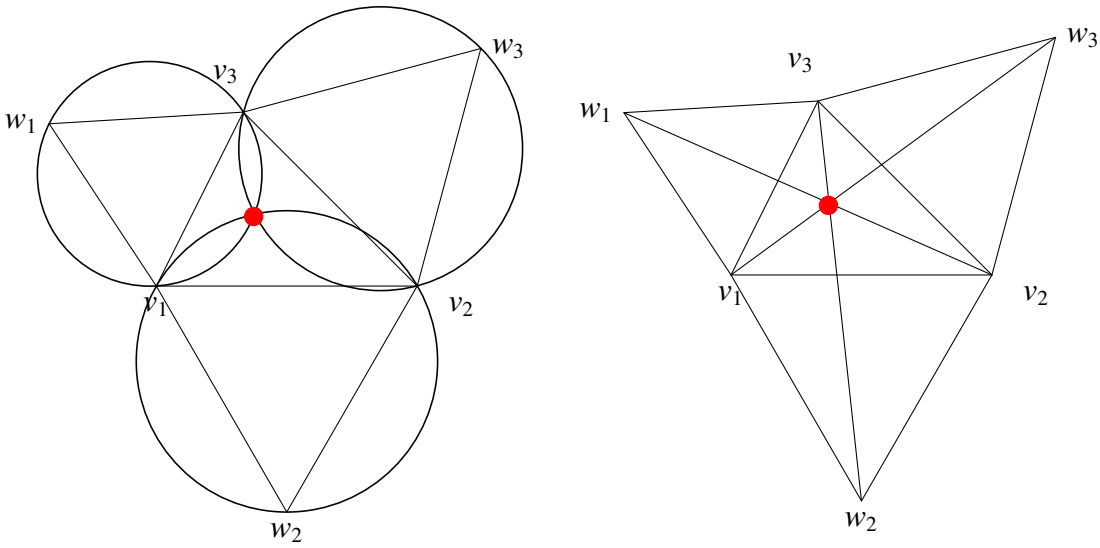
2.1 Frematov problem

Prethodno smo spomenuli *Frematov problem* koji predstavlja pojednostavljenje Steinerovog problema za dane 3 točke u Euklidskoj ravnini. Geometrijsko rješenje navedenog problema dali su *Torricelli* i *Simpson*. Prijedlog njihovih rješenja svodi se na sljedeće. Označimo 3 zadane točke u ravnini sa v_1, v_2, v_3 te konstruirajmo trokut $\Delta v_1 v_2 v_3$.

Torricellijevo rješenje je da se konstruiraju jednakostranični trokuti na svakoj od stranica trokuta $\Delta v_1 v_2 v_3$ te se konstruiranim jednakostraničnim trokutima opišu kružnice. Tada je točka u kojoj se sjeku sve tri kružnice tražena točka.

S druge strane Simpson je također predložio da se konstruiraju jednakostranični trokuti na svakoj od stranica trokuta $\Delta v_1 v_2 v_3$. Simpsonovo rješenje svodi se na povezivanje dužinom, svakog od vrhova koji se ne nalazi u trokutu $\Delta v_1 v_2 v_3$ sa nasuprotim vrhom u trokutu $\Delta v_1 v_2 v_3$. Navedena dužina se naziva *Simpsonova dužina*. Točka u kojoj se sijeku Simpsonove dužine je tražena točka.

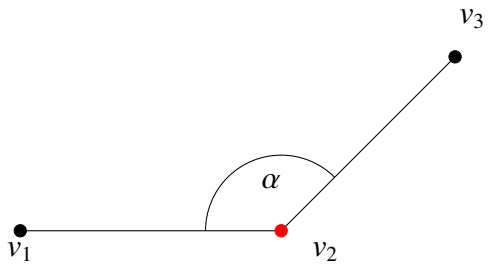
Primjer Torrichelijeve metode dan je na slici 2.2a, a primjer Simpsonove metode dan je na slici 2.2b. U tom primjeru dane su tri točke v_1, v_2, v_3 u ravnini te prvo konstruiramo trokut $\Delta v_1 v_2 v_3$. Također konstruiramo i jednakostranične trokute $\Delta v_1 v_2 w_2$, $\Delta v_1 v_3 w_3$ i $\Delta v_1 w_1 v_3$. Sada ovisno o algoritmu konstruiramo kružnice nad jednakostraničnim trokutima, odnosno Simpsonove dužine. Na kraju dobijemo isti rezultat, kao što je i prikazano na slici 2.2, odnosno istu točku u ravnini čija je udaljenost najmanja u odnosu na zadane 3 točke. Primjetimo da točka koju smo našli primjenom prethodnih metoda čini kut od 120°sa svaka dva vrha trokuta $\Delta v_1 v_2 v_3$.



(a) Torricellijeva metoda.

(b) Simpsonova metoda.

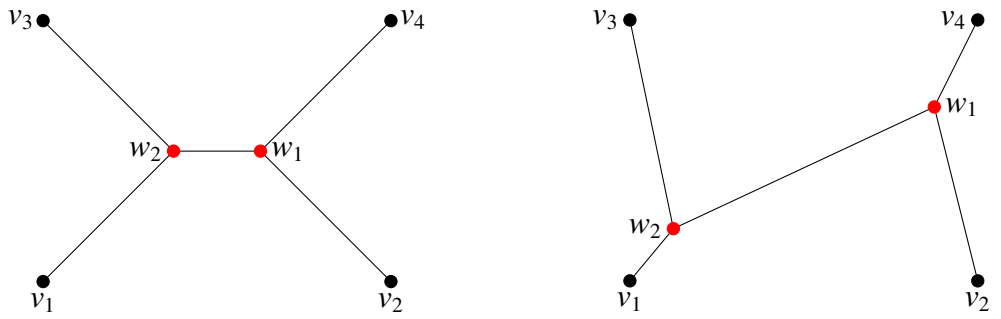
Slika 2.2: Usporedba Torricellijeve i Simpsonove metode za Fermatov problem.



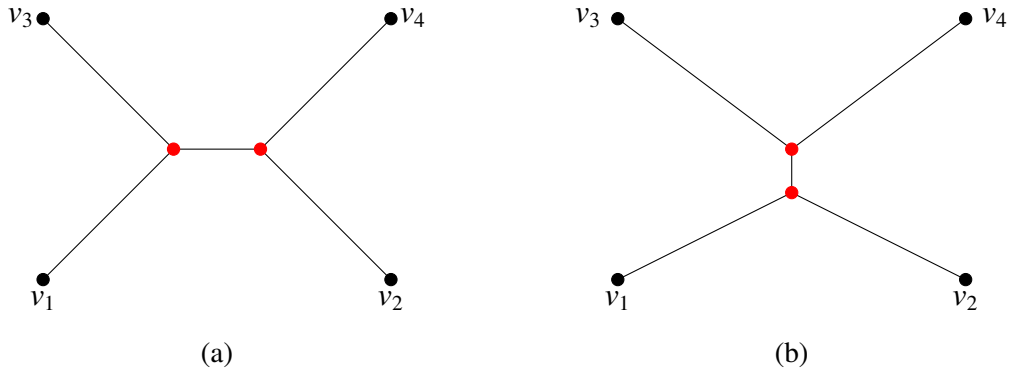
Slika 2.3: Fermatov problem sa kutem α između točaka većim od 120°.

Prethodno opisane metode ne daju rješenje kada u trokutu $\Delta v_1 v_2 v_3$ postoji kut veći od 120°. U tom slučaju kod Simpsonove metode sve 3 dužine se ne sijeku u istoj točki, a kod Toricellijeve metode dobijemo točku koja se nalazi izvan trokuta $\Delta v_1 v_2 v_3$. Navedeni problem u metodama možemo "popraviti", u slučaju kada postoji kut u trokutu $\Delta v_1 v_2 v_3$ koji je veći od 120° tražena točka je zapravo jedna od zadanih točki. Na slici 2.3 prikazan je upravo takav primjer, dane su 3 točke v_1, v_2, v_3 u prostoru te je kut α između njih veći od 120°. U tom slučaju očito je tražena točka upravo točka v_2.

Dakle, možemo poopćit prethodno razmatranje na sljedeći način. Za dane 3 točke v_1, v_2, v_3 u ravnini konstruiramo trokut $\Delta v_1 v_2 v_3$. U slučaju da u trokutu $\Delta v_1 v_2 v_3$ postoji kut veći od 120° tada vrh kod kojeg je kut veći od 120° je tražena točka koja rješava Fermatov problem. Inače, ako su svi kutevi trokuta $\Delta v_1 v_2 v_3$ manji od 120° rješenje Fermatovog



Slika 2.4: Ekvivalentne topologije.



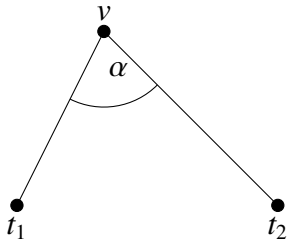
Slika 2.5: Različite topologije.

problema možemo dobiti primjenom Simpsonove ili Torricellijeve metode.

2.2 Svojstva Steinerovih stabla u Euklidskom prostoru

Steinerov problem u Euklidskom prostoru možemo iskazati na sljedeći način. Neka je dano n terminalnih točaka u ravnini, potrebno je odrediti stablo, tj. Steinerovo stablo, koje sadrži svih n zadanih točaka te čija je ukupna duljina minimalna, obzirom na Euklidsku normu. Uz dodatak da konstruirano stablo može sadržavati i dodatne, Steinerove, točke kako bi ukupna duljina stabla bila minimalna. Navodimo specifična svojstva za Euklidski Steinerov problem.

Promotrimo razne topologije proizvoljnog stabla T, odnosno promotrimo ekvivalentne i različite topologije stabla. Topologija nam ukazuje samo na povezanost između vrhova, a ne položaj vrhova u prostoru te duljinu bridova unutar stabla. Slika 2.4 prikazuje dva stabla ekvivalentne topologije, dok slika 2.5 prikazuje različite topologije. Na slikama,



Slika 2.6: Stablo sa kutem $\alpha < 120°$

crni vrhovi označavaju terminalne točke dok crveni vrhovi označavaju Steinerove točke. Na slici 2.4 očito su oba vrha, v_1 i v_2 povezana sa istom Steinerovom točkom kao i vrhovi v_3 i w_4, no ukupna duljina stabla je očito različita. Na slici 2.5 prikazana su dva stabla različitih topologija. Na slici 2.5a prikazano je stablo u kojemu su terminalne točke v_1 i v_3 povezane istom Steinerovom točkom kao i vrhovi v_2 i v_4 dok su terminalne točke v_3 i v_4 sa slike 2.5b povezane s jednom Steinerovom točkom, a terminalne točke v_1 i v_2 su povezane drugom.

Prethodno je opisan Fermatov problem u slučaju da su zadane 3 točke u prostoru. Motivirani tim primjerom možemo primijetiti da u Steinerovom stablu ne može postojati vrh u kojem dva brida čine kut manji od 120°. U tom slučaju, ako je kut između 2 brida u istom vrhu manji od 120°, tada postoji Steinerovo stablo koje sadrži tu točku te je njegova ukupna duljina manja od prethodno zadanog stabla. Navedeni uvjet iskazan je sljedećom lemom.

Lema 2.2.1. *Neka je $T = (V, E)$ minimalno Steinerovo stablo u ravnini. Tada niti jedna dva brida stabla T istog vrha ne čine kut manji od* 120°.

Dokaz. Pretpostavimo da postoji vrh $v \in V$ u Steinerovom minimalnom stablu T te bridovi $e_1 = \{v, t_1\} \in E$ i $e_2 = \{v, t_2\} \in E$ koji čine kut manji od 120°. Polazna situacija je prikazana na slici 2.6.

Prethodno je izložen Fermatov problem za nalaženje točke u prostoru kako bi udaljenost od preostale 3 točke bila minimalna. Kao što je opisano u prethodnom potpoglavlju i ovdje možemo naći takvu točku primjenom Simpsonove ili Torricellijeve metode. Na taj način konstruirano stablo ima manju ukupnu udaljenost pa zadano stablo nije minimalno. □

Prethodna lema nam daje uvjet na kut između dva brida istog vrha Steinerovog stabla, iz čega proizlazi da u Steinerovom stablu ne postoji vrh čiji je stupanj veći od 3.

Iz prethodnog proizlazi da su sve Steinerove točke u stablu stupnja točno 3. U protivnom, pretpostavimo da postoji Steinerova točka stupnja 1. Tada odbacivanjem navedene

točke imamo Steinerovo stablo manje ukupne duljine. Također u slučaju da postoji Steinerova točka stupnja 2, odbacivanjem navedene točke i povezivanjem susjedna dva vrha imamo stablo manje ili jednake duljine od polaznog. Dakle, lema 2.2.1 pokazuje da su sve Steinerove točke u Steinerovom stablu stupnja točno 3 te da pripadni bridovi čine kut od 120°. Pomoću teorema 1.0.3 i 1.0.6 iskažimo uvjet na broj Steinerovih točaka u Steinerovom stablu.

Teorem 2.2.2. *Steinerovo stablo $T = (V, E)$ ima najviše $|V|-2$ Steinerovh točaka.*

Dokaz. Neka je $T = (V, E)$ Steinerovo stablo sa n terminalnih točaka $v_1, \ldots, v_n$ te k Steinerovih točaka $w_1, \ldots, w_k$. Prema teoremu 1.0.6 slijedi da je $|E| = n + k - 1$. Također svaka Steinerova točka je stupnja 3, tj. $d(w_j) = 3$, $\forall j \in \{1, \ldots, k\}$ i svaka terminalna točka ima stupanj barem 1. Iz teorema 1.0.3 slijedi

$$n + k - 1 \geq \frac{3k + n}{2} \tag{2.6}$$

$$n - 2 \geq k \tag{2.7}$$

□

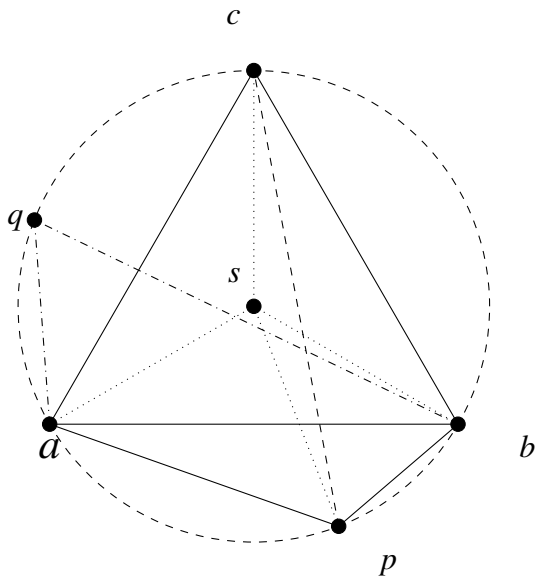
2.3 Algoritmi za Steinerov problem u Euklidskom prostoru

U ovom poglavlju prvo je iskazan algoritam koji riješava prethodno promatrani Fermatov problem. Taj algoritam će kasnije biti temelj za *Melzakov algoritam* koji će rješavati Steinerov Euklidski problem. Naknadno ćemo pokazat i numerički algoritam za rješavanje Steinerovog problema, zvat ćemo ga *Smithova numerička metoda*. Posljednji algoritam koji iskazujemo u ovom poglavlju je aproksimacijski algoritam koji nam daje približno rješenje problema.

2.3.1 Algoritam za Fermatov problem

Prvo navodimo lemu bitnu za dokaz korektnosti algoritma za Fermatov problem.

Lema 2.3.1. *Neka je Δabc jednakostraničan trokut sa opisanom kružnicom C. Tada sve točke p na manjem segmentu od C između vrhova a i b čine kut od* 120°*, tj vrijedi $\measuredangle apb =$* 120°*. Također sve točke q na većem segmentu između a i b čine kut od* 60°*, tj vrijedi $\measuredangle aqb =$* 60°*.*



Slika 2.7: Ilustracija leme 2.3.1.

Dokaz. Neka je p proizvoljna točka na manjem segmentu kružnice C. Kut $\sphericalangle apc$ je ekvivalentan kutu $\sphericalangle aqb$ zbog simetrije, pa je dovoljno pokazati $\sphericalangle apb = 120°$ i $\sphericalangle apc = 120°$. Kako je s centar kružnice C i trokuti Δasp i Δbsp su jednakokračni zato što su bridovi $\overline{as}$, $\overline{bs}$, $\overline{ps}$ svi jednaki radijusu kružnice C. Pa uz činjenicu da je Δabc jednakostraničan, tj. vrijedi $\sphericalangle asb = 120°$ te imamo

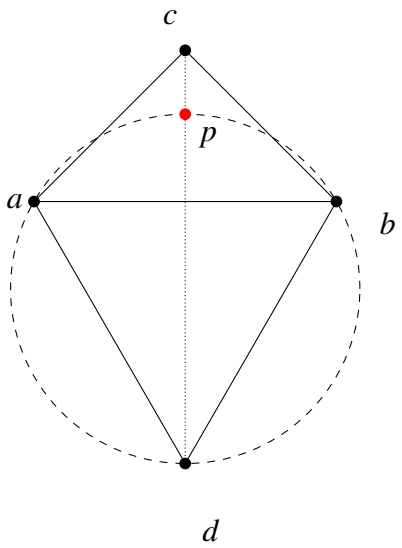
$$\sphericalangle apb = \frac{1}{2}(180° - \sphericalangle asp) + \frac{1}{2}(180° - \sphericalangle bsp) = 180° - \frac{1}{2}\sphericalangle asb = 120°. \tag{2.8}$$

Slično i za trokute Δcsp, Δasp, trokut Δcsp je jednakokračan pa je

$$\sphericalangle apc = \frac{1}{2}(180° - \sphericalangle asp) + \frac{1}{2}(180° - \sphericalangle csp) = 180° - \frac{1}{2}(\Delta asb + \Delta bsc) = 60° \tag{2.9}$$

□

Navodimo algoritam za Fermatov problem, bez smanjenja općenitosti možemo pretpostaviti da za dane tri točke u prostoru niti jedan kut u tom trokutu ne iznosi više od 120°.



Slika 2.8: Ilustracija algoritma 3.

Algoritam 3

Ulaz: Neka su $a, b, c \in \mathbb{R}^2$ točke u ravnini takve da su svi kutevi trokuta Δabc manji od 120°.

Izlaz: Torricellijeva točka p za zadane točke a, b, c.

1. Konstruiraj jednakostraničan trokut Δabd takav da je vrh d na suprotnoj strani od vrha c, obzirom na dužinu $\overline{ab}$.
2. Konstruiraj kružnicu C opisanu trokutu Δabd.
3. Torricellijeva točka se nalazi na presjeku dužine $\overline{cd}$ i kružnice C.

Teorem 2.3.2. *Neka je p Torricellijeva točka nađena pomoću algoritma 3 te neka su točke a, b, d definirane kao u algoritmu 3. Tada vrijedi*

$$|\overline{ap}|+|\overline{bp}|= |\overline{dp}|. \tag{2.10}$$

Iz leme 2.3.1 slijedi da algoritam 3 očito pronalazi Torricellijevu točku p za dane točke a, b, c u ravnini. Prethodno smo već pokazali da ako točka p čini kut od 120° sa vrhovima trokuta a, b, c to je Torricellijeva točka.

Ilustrirajmo algoritam 3 na konkretnom primjeru. Neka su $a = (0, 0), b = (4, 0), c = (2, 2)$ tri točke zadane u ravnini. Najprije nađemo točku d tako da postavimo uvjet na

jednaku udaljenost točke d od vrhova a i b pa imamo sljedeći sustav

$$\begin{cases} x^2 + y^2 = 16 \\ (x-4)^2 + y^2 = 16 \end{cases} \tag{2.11}$$

Kao rješenje sustava dobijemo točku $d = (2, -2\sqrt{3})$. Za drugi korak algoritma potrebno je konstruirati kružnicu opisanu trokutu Δabd. Kako bi konstruirali kružnicu potrebno je odrediti središte kružnice kao i radijus. Navedeno ćemo dobiti korištenjem uvjeta da vrhovi trokuta Δabd nalaze na kružnici, tj. vrijedi

$$\begin{cases} p^2 + q^2 = r^2 \\ (4-p)^2 + q^2 = r^2 \\ (2-p)^2 + (-2\sqrt{3} - q)^2 = r^2 \end{cases} \tag{2.12}$$

Iz navedenog, rješavanjem sustava 2.12 dobijemo sljedeću jednadžbu opisane kružnice: $(x-2)^2 + (y + 2/\sqrt{3})^2 = 16/3$. U trećem koraku algoritma prvo je potrebno odrediti jednadžbu pravca na kojem leži dužina $\overline{cd}$. U ovom primjeru jednadžba takvog pravca je vrlo jednostavna, kako obje točke imaju istu x koordinatu slijedi da je jednadžba pravca $x = 2$. Kako bi odredili presjek pravca na kojem leže točke c i d sa kružnicom, uvrstimo pravac u jednadžbu kružnice i dobijemo da je točka presjeka $(2, 2\sqrt{3}/3)$. Na kraju provjerimo dobiveno rješenje

$$\begin{aligned} |\overline{ap}| + |\overline{bp}| &= \frac{4\sqrt{3}}{3} + \frac{4\sqrt{3}}{3}, \\ |\overline{dp}| &= \frac{8\sqrt{3}}{3}. \end{aligned} \tag{2.13}$$

Iz 2.13 vidimo da vrijedi tvrdnja teorema 2.3.2 pa zaključujemo da je rješenje ispravno. Cijeli primjer prikazan je na slici 2.8.

2.3.2 Melzakov algoritam

Melzakov algoritam za dani skup terminalnih točaka i topologiju daje Steinerovo stablo. Algoritam se temelji na prethodno iskazanom algoritmu koji rješava Fermatov problem sa 3 točke u ravnini. U ovom slučaju pretpostavljamo da nam je poznata topologija traženog Steinerovog stabla, odnosno povezanost između vrhova stabla, tj. poznat nam je broj terminalnih i Steinerovih točaka u traženom stablu. Svako Steinerovo stablo sa minimalno jednom Steinerovom točkom p će imati barem dvije terminalne točke. Označit ćemo ih sa a i b. Za takve terminalne točke a i b kažemo da su susjedne. Kako znamo da su sve

Steinerove točke stupnja barem 3, postoji još jedna točka c takva da postoji brid $\{p, c\}$. Iskažimo sada *Melzakov algoritam* koji se sastoji od dvije faze i nekoliko potkoraka.

Algoritam 4 Melzakov algoritam

Ulaz: Neka je zadan skup od n terminalnih točaka i topologija.
Izlaz: Steinerovo minimalno stablo obzirom na zadanu topologiju.

1. Dok postoji bar jedna Steinerova točka radi sljedeće:

   a) Odaberi dvije terminalne točke a i b takve da postoji Steinerova točka p za koju vrijedi da postoji put (a, p, b). Neka je točka c treća susjedna točka točke p, odnosno takva da postoji put (p, c). Izračunaj točke d_1 i d_2 koje čine jednakostraničan trokut sa točkama a i b. Izbaci točke a i b iz topologije.

   b) Ako je točka c terminalna, odaberi $d_i, i \in \{1, 2\}$ tako da trokut Δabd_i leži izvan trokuta Δabc.

   c) Inače, generiraj dva nova problema, jedan za d_1, drugi za d_2 te u svakom zamjeni Steinerovu točku p sa koordinatma točke $d_i, i \in \{1, 2\}$. Rekurzivno riješi oba problema.

2. Dok stablo T ne sadrži sve početne terminalne točke radi sljedeće:

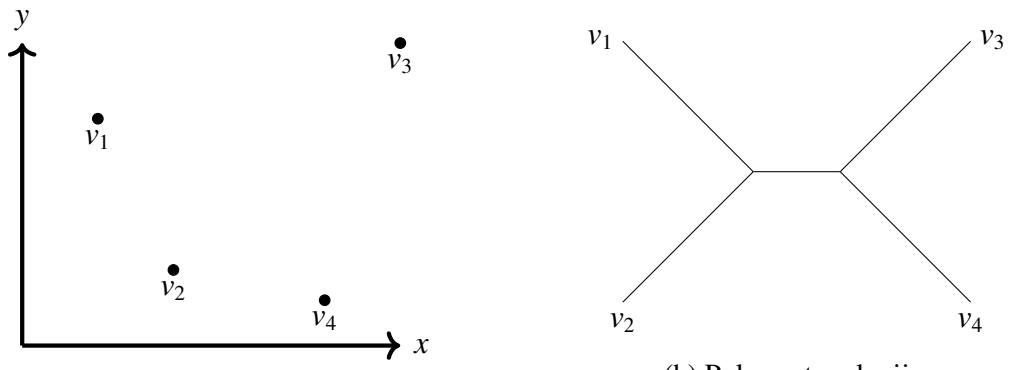
   a) Ako ne postoji više Steinerovih točaka, spoji terminalne točke bridovima u skladu sa trenutnom topologijom.

   b) Inače, neka su a i b terminalne točke koje su izbačene u prvom korak, neka je d točka koja je dodana, a c susjedna točka od d u trenutnom Steinerovom stablu.

      i. Ako kružnica opisana trokutu Δabd sječe dužinu $\overline{dc}$ (izuzev u točki d) spoji točke a, p i b bridom te odbaci brid $\overline{dp}$.

      ii. Inače stani, za danu topologiju ne postoji Steinerovo stablo.

Pokažimo algoritam na sljedećem primjeru. Neka su zadane terminalne točke $v_1 = (3, 5)$, $v_2 = (4, 3), v_3 = (7, 6), v_4 = (6, 2.6)$, te topologija sa dvije Steinerove točke s_1, s_2 takve da je s_1 povezana sa v_1 i v_2, a s_2 sa v_3 i v_4. Početno stanje prikazano je na slici 2.9.



(a) Terminalne točke.

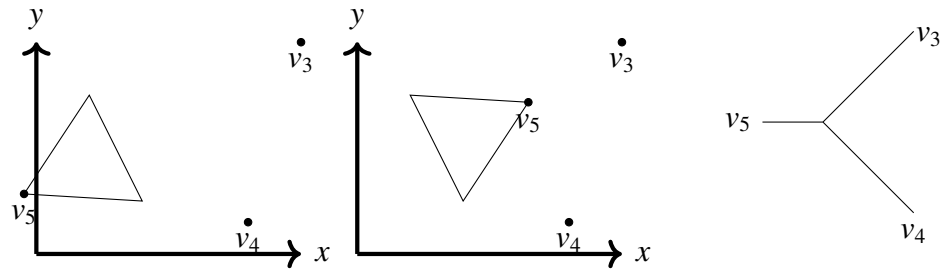
(b) Polazna topologija.

Slika 2.9: Ulazni podaci za Melzakov algoritam.

Prvo izaberimo točke a i b na sljedeći način. Postavimo $a = v_1$ i $b = v_2$ budući da su susjedne Steinerovoj točki p. Nađemo pripadne d_1 i d_2 tako da za d_i vrijedi da su trokuti Δabd_i jednakostranični, mora vrijediti

$$\begin{aligned} |\overline{ad}| &= |\overline{ab}| \\ |\overline{bd}| &= |\overline{ab}|. \end{aligned} \tag{2.14}$$

Iz gornjeg dobijemo dva rješenja za $d_i, i \in \{1, 2\}$, $d_1 = 1/2(7 - 2\sqrt{3}, 8 - \sqrt{3})$ te $d_2 = 1/2(7+2\sqrt{3}, 8+\sqrt{3})$. Zamjenom točke p sa d_1, odnosno d_2 dobijemo dva nova potproblema koja su prikazana na slikama 2.10a i 2.10b. Također, topologija za potprobleme prikazana je na slici 2.10c.

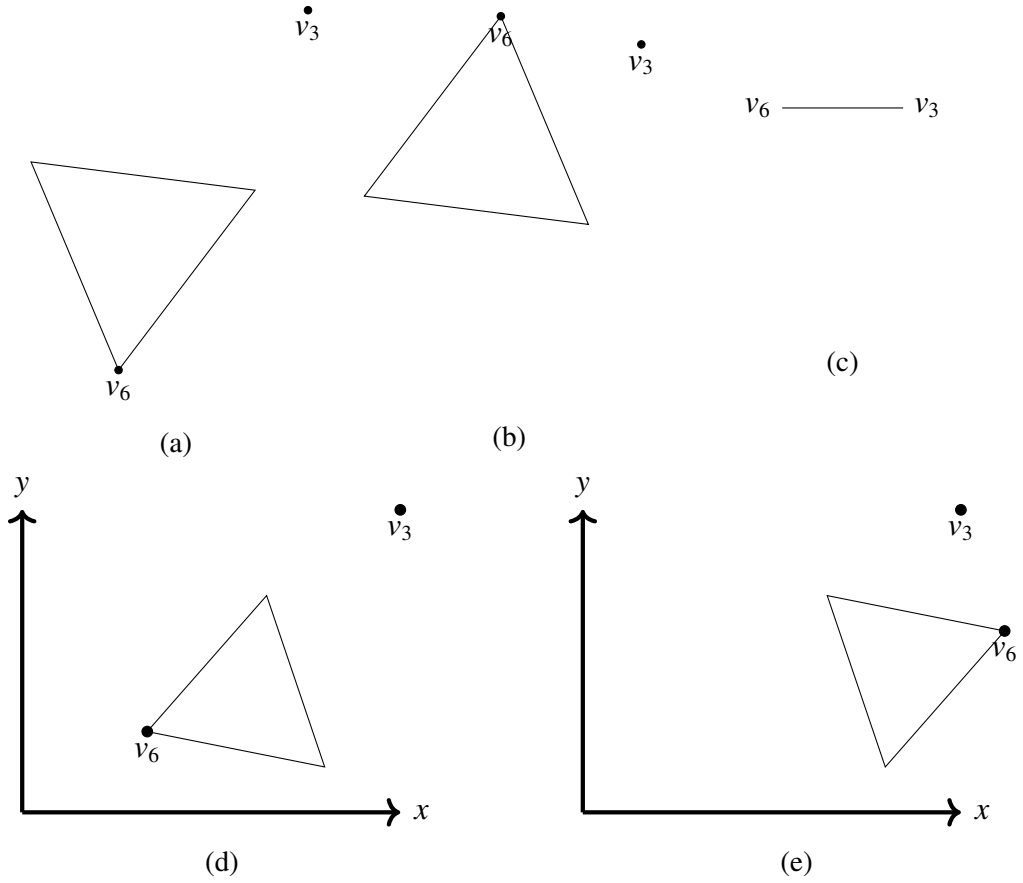


(a) (b) (c)

Slika 2.10: Prvi korak Melzakova algoritma.

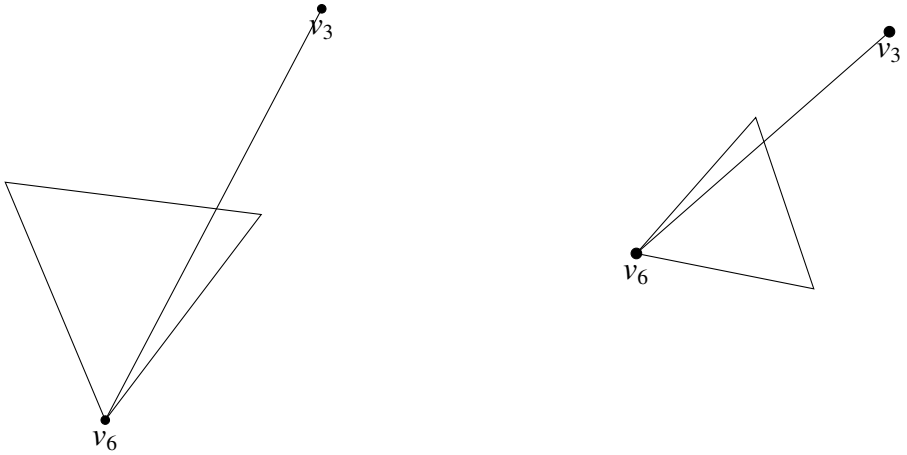
Sada je opet potrebno odabrati točke a i b za potprobleme sa slika 2.10a i 2.10b. Stavimo $a = v_5$ i $b = v_4$ tako da imaju susjednu Steinerovu točku p. Treća točka, točka c koja je susjedna Steinerovoj točci p je terminalna točka v_3 pa za svaki od potproblema

računamo pripadne točke d_i koristeći uvjet 2.14. Potrebno je odabrati za svaki od potproblema $d_i, i \in \{1, 2\}$ tako da trokut Δabd_i leži izvan trokuta Δabc. Za prvi potproblem dobijemo dvije vrijednosti $d_{1,2}$ prikazane na slikama 2.11a i 2.11b te izaberemo slučaj sa slike 2.11a kako prikazani trokut leži izvan trokuta Δabc. Na isti način dobijemo dvije vrijednosti $d_{1,2}$ za slučaj sa slike 2.10b koje su prikazane na slikama 2.11e i 2.11d te odaberemo slučaj sa slike 2.11d. U novoj topologiji prikazanoj na 2.11c odbačene su točke $a = v_5$ i $b = v_4$ i zamijenjene sa v_6.

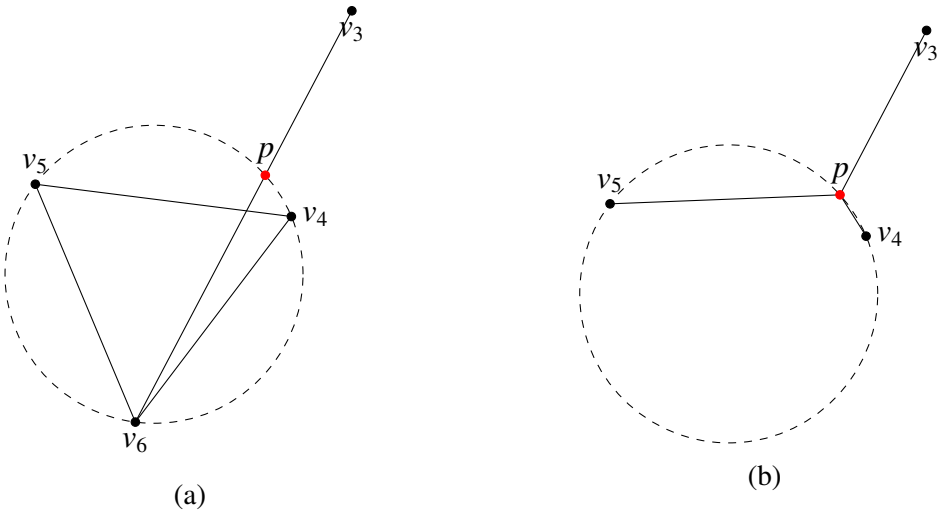


Slika 2.11: Drugi korak Melzakova algoritma.

Krenimo sa drugim dijelom algoritma. Najprije spojimo preostala dva vrha v_3 i v_6 bridom za oba problema sa slika 2.11a i 2.11d kao što je prikazano na 2.12.

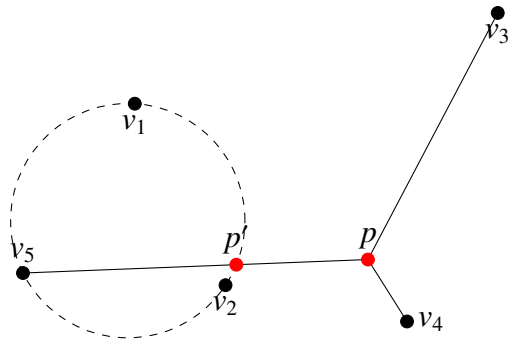


Slika 2.12: Treći korak Melzakova algoritma.

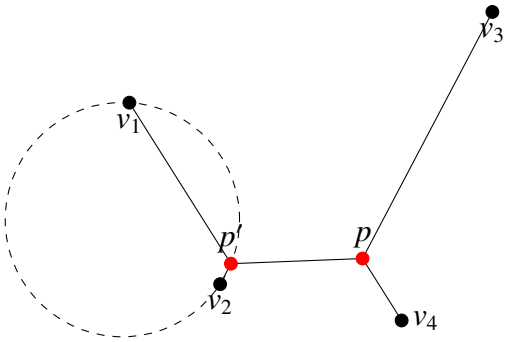


Slika 2.13: Četvrti korak Melzakova algoritma.

Promotrimo slučaj sa slike 2.12b. Kako kružnica opisana trokutu Δabd ne sječe dužinu $\overline{dc}$ zadovoljen je kriterij zaustavljanja i u tom slučaju za danu topologiju ne postoji Steinerovo stablo. U slučaju sa slike 2.12a, zadnje terminalne točke koje su odbačene su bile v_4 i v_5 te su zamijenjene terminalnom točkom v_6. Tada je točka c bila zapravo v_3. Kružnica opisana trokutu $\Delta v_4 v_5 v_6$ sječe dužinu $\overline{v_3 v_6}$ pa označimo to sjecište sa p kao na slici 2.13a. Sada spojimo točku p sa v_4 i v_5 te odbacimo dužinu $\overline{pv_6}$ kao na 2.13b.



(a)



(b)

Slika 2.14: Zadnji korak Melzakova algoritma.

Terminalne točke koje su odbačene u prethodnom koraku su bile v_1 i v_2 te su zamijenjene sa terminalnom točkom v_5. Također točka c je bila Steinerova točka p. Kružnica opisana trokutu $\Delta v_1 v_2 v_5$ sječe dužinu $\overline{pv_5}$, označimo točku sjecišta sa p' kao na 2.14a. Sada spojimo p' sa točkama v_1 i v_2 te odbacimo brid $\overline{p'v_5}$ i dobili smo Steinerovo minimalno stablo prikazano na 2.14b.

Složenost *Malzakova* algoritma za rješavanje Steinerovog problema uz danu topologiju je određena brojem Steinerovih točaka. Svaka Steinerova točka može generirati do dva potproblema koja je potrebno izračunati. Na taj način, kako znamo da je broj Steinerovih točaka u stablu ograničen sa $n-2$, složenost algoritma je $O(2^n)$.

2.3.3 Numerički algoritam

Ukupnu duljinu Steinerovog stabla za danu topologiju možemo promatrati i kao funkciju ovisnu o Steinerovim točkama. Takva funkcija je konveksna. Za funkciju realnu $f\colon \mathbb{R} \to \mathbb{R}$

kažemo da je *konveksna* na intervalu $I \subseteq \mathbb{R}$ ako

$$(\forall x_1, x_2) f(\frac{x_1 + x_2}{2}) \leq \frac{f(x_1) + f(x_2)}{2}. \tag{2.15}$$

Za konkretno razmatranje Steinerovog problema bit će nam potrebna karakterizacija konveksne funkcije. Dva puta derivabilna funkcija f je konveksna na intervalu I ako i samo ako je $f''(x) \geq 0, \forall x \in I$. Ako je funkcija konveksna te ima jednu stacionarnu točku za koju vrijedi $f'(x) = 0$, to će biti točka lokalnog minimuma. Iz prethodnog proizlazi da možemo naći Steinerove točke pomoću parcijalnih derivacija funkcije ukupne duljine Steinerovog stabla. Time dobivamo sustav nelinearnih jednadžbi pa su nam potrebne numeričke metode za rješavanje tog problema. U tu svrhu koristimo *Smithovu numeričku metodu*, iterativan algoritam koji u svakom koraku točnije određuje trenutni položaj Steinerovih točaka.

Algoritam 5 Smithov algoritam

Ulaz: Neka je zadan skup od n terminalnih točaka i topologija $T = (V, E)$ sa k Steinerovih točaka. Neka $s_k = (x_k, y_k)$ označava k-tu Steinerovu točku te neka su koordinate susjednih točaka od s_k dane sa $w_j = (u_j, t_j)$.

Izlaz: Steinerovo minimalno stablo obzirom na zadanu topologiju.

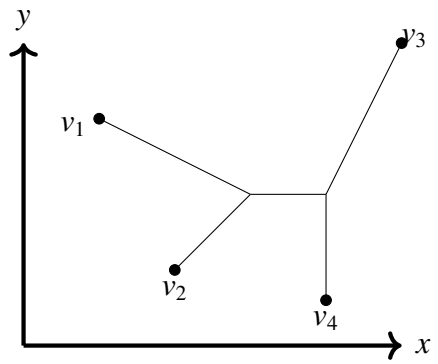
Koordinate i-te iteracije algoritma dane su sa:

$$x_k^{i+1} = \frac{\dfrac{\sum_{j:\{w_j,s_k\}\in E} u_j^{i+1}}{\sum_{j:\{w_j,s_k\}\in E} |\overline{w_j^i s_k^i}|}}{\dfrac{1}{\sum_{j:\{w_j,s_k\}\in E} |\overline{w_j^i s_k^i}|}} \tag{2.16}$$

$$y_k^{i+1} = \frac{\dfrac{\sum_{j:\{w_j,s_k\}\in E} t_j^{i+1}}{\sum_{j:\{w_j,s_k\}\in E} |\overline{w_j^i s_k^i}|}}{\dfrac{1}{\sum_{j:\{w_j,s_k\}\in E} |\overline{w_j^i s_k^i}|}}. \tag{2.17}$$

Ilustrirajmo rad algoritma na istom primjeru kao i u prethodnom potpoglavlju. Neka su dane terminalne točke $v_1 = (3, 5)$, $v_2 = (4, 3)$, $v_3 = (7, 6)$, $v_4 = (6, 2.6)$, te topologija sa dvije Steinerove točke kao na slici 2.9. Započinjemo algoritam sa Steinerovim točkama $s_1^0 = (5, 4)$, $s_2^0 = (6, 4)$, pa je ukupna duljina inicijalnog stabla prikazanog na slici 2.15 jednaka

$$|\overline{v_1 s_1}| + |\overline{v_2 s_1}| + |\overline{s_1 s_2}| + |\overline{v_3 s_2}| + |\overline{v_4 s_2}| = 8.2. \tag{2.18}$$



Slika 2.15: Inicijalno stablo.

Prvu iteraciju algoritma započinjemo tako da odredimo sume u 2.16. Za Steinerovu točku s_1 pripadne vrijednosti w_j iznose $w_j = (u_j, t_j)$ su $w_0^0 = v_1 = (3,5)$, $w_2^0 = v_2 = (3,5)$, $w_3^0 = s_2^0 = (6,4)$ te $u_1^1 = 3$, $u_2^1 = 4$, $u_3^1 = x_2^1$, pa u prvoj iteraciji dobijemo sljedeće vrijednosti:

$$x_1^1 = \frac{\frac{u_1^1}{|w_1^0 s_1^0|} + \frac{u_2^1}{|w_2^0 s_1^0|} + \frac{x_2^1}{|w_3^0 s_1^0|}}{\frac{1}{|w_1^0 s_1^0|} + \frac{1}{|w_2^0 s_1^0|} + \frac{1}{|w_3^0 s_1^0|}} = 1.935676771 + 0.4641835127 x_2^1, \tag{2.19}$$

$$y_1^1 = \frac{\frac{t_1^1}{|w_1^0 s_1^0|} + \frac{t_2^1}{|w_2^0 s_1^0|} + \frac{y_2^1}{|w_3^0 s_1^0|}}{\frac{1}{|w_1^0 s_1^0|} + \frac{1}{|w_2^0 s_1^0|} + \frac{1}{|w_3^0 s_1^0|}} = 2.022627817 + 0.4641835127 y_2^1. \tag{2.20}$$

Analogno, za drugu Steinerovu točku odredimo vrijednosti pripadnih w_i-jeva, te dobivamo sljedeće:

$$x_2^1 = \frac{\frac{u_1^1}{|w_1^0 s_2^0|} + \frac{u_2^1}{|w_2^0 s_2^0|} + \frac{x_1^1}{|w_3^0 s_2^0|}}{\frac{1}{|w_1^0 s_2^0|} + \frac{1}{|w_2^0 s_2^0|} + \frac{1}{|w_3^0 s_2^0|}} = 3.43104873 + 0.462641832 x_1^1, \tag{2.21}$$

$$y_2^1 = \frac{\frac{t_1^1}{|w_1^0 s_2^0|} + \frac{t_2^1}{|w_2^0 s_2^0|} + \frac{y_1^1}{|w_3^0 s_2^0|}}{\frac{1}{|w_1^0 s_2^0|} + \frac{1}{|w_2^0 s_2^0|} + \frac{1}{|w_3^0 s_2^0|}} = 2.10059028 + 0.462641832 y_1^1. \tag{2.22}$$

Dobiveni sustav linearnih jednadžbi određen sa 2.19, 2.20, 2.21, 2.22 daje sljedeća rješenja za tražene Steinerove točke $s_1^1 = (4.49324, 3.8175)$ te $s_1^1 = (5.50981, 3.86672)$. Ukupna duljina dobivenog minimalnog razapinjućeg stabla nakon jedne iteracije je 7.83776, što je manje od polazne duljine koja je iznosila 8.2.

2.3.4 Aproksimacijski algoritam

Za razliku od prethodno opisanih algoritama, u ovom potpoglavlju navodimo primjer algoritma koji daje približno rješenje za Steinerov problem u Euklidskoj ravnini. Koristit ćemo *Steiner Insertion Algorithm* (SIA). Pomoću navedenog algoritma u polazno stablo dodat ćemo Steinerove točke kako bismo smanjili ukupnu duljinu stabla.

Algoritam 6 Aproksimacijski algoritam pomoću SIA

Ulaz: Neka je zadan skup od n terminalnih točaka. Neka $s_k = (x_k, y_k)$ označava k-tu Steinerovu točku te neka su koordinate susjednih točaka od s_k dane sa $w_j = (u_j, t_j)$.

Izlaz: Steinerovo stablo.

1. Nađi minimalno razapinjuće stablo za dani skup terminalnih točaka.
2. Za svaki brid $\{x, y\}$ stabla pri čemu su x i y terminalne točke radi sljedeće:
   a) Nađi brid $\{y, z\}$ susjedan bridu $\{x, y\}$ između kojih je kut najmanji, pri čemu točka z može biti terminalna ili Steinerova.
   b) Ako je kut između bridova manji od 120° radi sljedeće:
      i. Umetni novu Steinerovu točku s u stablo pri čemu će s imati iste koordinate kao i točka y.
      ii. Izbaci bridove $\{x, y\}, \{y, z\}$ iz stabla.
      iii. Dodaj bridove $\{x, s\}, \{y, s\}, \{z, s\}$ u stablo.
3. Izbaci sve Steinerove točke stupnja 1.
4. Lokalno optimiziraj dobiveno Steinerovo stablo.

Navedeni algoritam daje Steinerovo stablo za dani skup terminalnih točaka te za razliku od Melzakova algoritma, nije potrebno odrediti topologiju traženog stabla. U zadnjem koraku algoritma vrši se optimizacija dodanih Steinerovih točaka pomoću pomoćnog algoritma kao što je *The Incremental Optimization Algorithm* iskazan u [3]. Svrha optimizacije u zadnjem koraku je točnije odrediti optimalne koordinate Steinerovih točaka.

2.3.5 Implementacija Smithovog i Aproksimacijskog algoritma

U prilogu diplomskog rada nalazi se implementacija Smithovog i Aproksimacijskog algoritma u programskom jeziku C++. Za automatizaciju izgradnje projekta korišten je softver otvorenog koda *CMake*.

Prema definiciji 1.0.1 stablo je uređen par skupa vrhova i skupa bridova, stoga u implementaciji koristimo klasu `Node` prikazanu na slici 2.16 za reprezentaciju svakog pojedinog vrha stabla u programu. Klasa `Node` sadrži podatke o x i y koordinatama vrha te indikator `steinerPoint` koji ukazuje prestavlja li vrh Steinerovu točku. U klasi su također implementirani konstruktori, operatori usporedbe, pridruživanja te pomoćni operator za ispis skupa vrhova.

```
class Node
{
public:
    double x, y;
    bool steinerPoint;

    Node();
    Node(double x_1, double y_1);
    Node(double x_1, double y_1, bool steiner);
    Node(const Node &p1);

    Node &operator=(const Node &v);
    friend bool operator==(const Node &a, const Node &b);
    friend bool operator!=(const Node &a, const Node &b);
    friend std::ostream &operator<<(std::ostream &os, const Node &e);
};
```

Slika 2.16: Klasa `Node` definirana u `node.h`.

Na analogan način za reprezentaciju svakog pojedinog brida u stablu koristimo klasu `Edge` prikazanu na slici 2.16. Klasa `Edge` sadrži podatke o vrhovima i duljini svakog pojedinog brida. Podaci o vrhovima pohranjeni su u varijablama `v1` i `v2` prethodno opisane klase `Node`. Slično kao i kod klase `Node` i za klasu `Edge` definirani su pripadni konstruktori, operatori pridruživanja i usporedbe te pomoćni operator za ispis klase.

Svi algoritmi korišteni u algoritmima za konstrukciju Steinerovog stabla implementirani su u datoteci `algorithms.h` prikazanoj na slici 2.17. Smithov algoritam 5 implementiran je u funkciji `numericalAlgorithm` koja kao ulazne argumente prima listu vrhova te listu bridova, odnosno polaznu topologiju stabla. Za rješavanje sustava linearnih jednadžbi koji dobijemo algoritmom implementirane su *Gauss-Jordanove transformacije* u funkciji `Gauss` koja kao argument prima matricu tj. sustav linearnih jednadžbi, a vraća pripadna rješenja sustava. Aproksimacijski algoritam 6 implementiran je u funkciji `aproxAlgorithm` koja kao argumente prima listu vrhova i listu bridova inicijalnog stabla te vraća listu vrhova i bridova dobivenog Steinerovog stabla. U implementaciji korišteni su

i pomoćni algoritmi: Kruskalov algoritam 1 implementiran u funkciji `Kruskal`, `SIAInsertion` algoritam za umetanje Steinerovih točaka te posljednje, optimizacija dobivenog stabla implementirana je u funkciji `SIAoptimization`.

```
std::pair<std::list<Node>,std::list<Edge>> numericalAlgorithm(
    std::list<Node> V, std::list<Edge> E);
std::vector<double> Gauss(std::vector<std::vector<double>> &matrica);
std::pair<std::list<Node>, std::list<Edge>> aproxAlgorithm(
    std::list<Node> V, std::list<Edge> E);
std::list<Edge> Kruskal(const std::list<Node> &V, std::list<Edge> &E);
void SIAInsertion(std::list<Node> &V, std::list<Edge> &E);
void SIAoptimization(std::list<Node> &V, std::list<Edge> &E);
```

Slika 2.17: Algoritmi definirani u datoteci `algorithms.h`.

```
#include "node.h"

class Edge
{
public:
    Node v1, v2;
    double length;

    Edge();
    Edge(Node v_1, Node v_2);
    Edge(const Edge &p1);

    Edge &operator=(const Edge &p1);
    friend bool operator==(const Edge &a, const Edge &b);
    friend bool operator!=(const Edge &a, const Edge &b);
    friend std::ostream &operator<<(std::ostream &os, const Edge &e);
};
```

Slika 2.18: Klasa `Edge` definirana u `Edge.h`.

Implementacijom algoritama 6 i 5 za primjer iz potpoglavlja 2.3.3 dobijemo sljedeće rezultate: Smithov algoritam nam daje Steinerovo stablo ukupne duljine 7.83776 dok aproksimacijski algoritam daje stablo ukupne duljine 7.92087. S obzizom na inicijalnu ukupnu duljinu od 8.2, oba algoritma daju dobra rješenja tj. stabla čija je ukupna duljina manja.

		Primjer iz potpoglavlja 2.3.3	Primjer sa 20 terminalnih točaka
Inicijalna duljina stabla		8.2	103.034
Numerički algoritam	Duljina dobivenog stabla	7.83776	93.2414
	Vrijeme izvršavanja	$77 \cdot 10^{-6}$ s	$12998 \cdot 10^{-6}$ s
Aproksimacijski algoritam	Duljina dobivenog stabla	7.92087	91.8187
	Vrijeme izvršavanja	$153 \cdot 10^{-6}$ s	$113526 \cdot 10^{-6}$ s

Tablica 2.1: Usporedba algoritama na primjeru sa 4 terminalne točke i primjeru sa 20 terminalnih točaka.

Također, zaključujemo da Smithov algoritam konstruira bolje rješenje od aproksimacijskog algoritma.

Kako bismo usporedili i vrijeme izvršavanja algoritama koristimo dodatni primjer stabla sa više terminalnih točaka. Primjer grafa dan je u datoteci `Example.txt` koje se nalaze u prilogu ovog rada. U tablici 2.1 prikazani su rezultati algoritama za oba primjera. Primjer sa 20 terminalnih točaka numerički algoritam rješava brže no sa manjom točnošću zato što on ovisi o zadanoj polaznoj topologiji, odnosno dobiveno Steinerovo stablo mora poštivati polaznu topologiju. Za razliku od numeričkog algoritma, aproksimacijski algoritam ne prima polaznu topologiju stabla kao ulazni argument pa broj i raspored Steinerovih točaka algoritam sam određuje bez ikakvih ograničenja vezanih uz topologiju. Kao što je vidljivo iz rezultata prikazanih u tablici 2.1, u oba slučaja aproksimacijski algoritam je sporiji od numeričkog. Vrijeme izvršavanja numeričkog algoritma ovisi i o broju iteracija koje postavimo. Rezultat za primjer iz potpoglavlja 2.3.3 dobiven je u jednoj iteraciji algoritma, dok je rezultat za primjer sa 20 terminalnih točaka dobiven nakon 30 iteracija. Povećanjem broja iteracija povećava se i vrijeme izvršavanja numeričkog algoritma. Zaključno, konstatirajmo da numerički algoritam daje egzaktno rješenje, no potreban je veliki broj iteracija pa i dulje vrijeme izvršavanja kako bismo dobili rezultat. Aproksimacijski algoritam daje približno rješenje te ne zahtijeva postavljanje polazne topologije koju mora poštivati i dobiveno Steinerovo stablo.

Poglavlje 3

Steinerov problem u mreži

U ovom poglavlju proučavat ćemo Steinerova stabla u mreži.

Problem minimalnog Steinerovog stabla u mreži Neka je dana povezana mreža $M = (\mathcal{G}, \ell)$ pri čemu je $\mathcal{G} = (V, E)$ graf, V skup vrhova, a E skup bridova te neka je $K \subseteq V$ skup terminalnih točaka. Potrebno je pronaći Steinerovo minimalno stablo za skup terminalnih točaka K u mreži M, tj. stablo T takvo da $\ell(T) = \min\{\ell(T') : T' \text{ je Steinerovo stablo za } K \text{ u mreži } M\}$, pri čemu duljinu $\ell(T')$ računamo kao zbroj težina uključenih bridova. Navedeno Steinerovo minimalno stablo može sadržavati i dodatne Steinerove točke iz skupa $V \setminus K$.

Na analogan način definiramo problem Steinerovog minimalnog stabla T u grafu $\mathcal{G}$ koji nije težinski, uz interpretaciju $l(T)$ kao broj bridova grafa T. Također, Steinerov problem u mreži može se shvatiti kao poopćenje Steinerovog problema u metričkom prostoru. Naime, primjerak problema u metričkom prostoru možemo pretvoriti u primjerak u mreži tako da točke iz prostora shvatimo kao vrhove grafa, a bridovima zadamo težine koje su jednake udaljenosti odgovarajućih točaka. Problem u mreži je općenitiji u smislu da težine bridova inače ne moraju poštivati aksiome metrike, npr. za te težine ne mora vrijediti nejednakost trokuta.

3.1 Egzaktni algoritmi

Steinerov problem u mreži bavi se proučavanjem samo konačnih objekata. U daljnjem razmatranju ograničimo se na konačan skup vrhova i bridova, tj. $|V| = n \in \mathbb{N}$ i $|E| = m \in \mathbb{N}$. Općenito gledajući problem se može jednostavno riješiti enumeracijom svih podskupova skupa bridova. Dovoljno je provjeriti da podskupovi čine Steinerovo stablo koje razapinje dan skup terminalnih točaka te odabrati najmanji takav skup koji predstavlja Steinerovo minimalno stablo. Kasnije ćemo pokazati da ne postoji polinomijalno složen algoritam

koji rješava Steinerov problem u grafovima pa ni Steinerov problem u mrežama. Steinerov problem u grafu možemo definirati na sljedeći način.

Neka je dan povezan graf $\mathcal{G} = (V, E)$ i skup terminalnih točaka $K \subseteq V$. Steinerovo minimalno stablo za K u grafu $\mathcal{G}$ je Steinerovo stablo T za koje vrijedi da je skup bridova minimalan, tj.$|E(T)|= \min\{|E(\tilde{T})|: \tilde{T}$ je Steinerovo stablo za K u $\mathcal{G}\}$.

3.1.1 Algoritam enumeracije

Lema 3.1.1. *Neka je $M = (\mathcal{G}, \ell)$ mreža te $\mathcal{G} = (V, E)$ pripadni graf i $K \subseteq V$ skup terminalnih točaka takav da vrijedi $|K|= k, k \in \mathbb{N}$. Tada Steinerovo stablo T za dani skup terminalnih točaka K sadrži najviše $k - 2$ Steinerovih točaka stupnja najmanje* 3.

Dokaz. Neka je $T = (V_T, E_T)$ Steinerovo stablo u mreži $M = (\mathcal{G}, \ell)$. Uvedimo sljedeće oznake:

$$S_2 = \{v \in V_T \mid d(v) = 2\},$$
$$S_3 = \{v \in V_T \mid d(v) \geq 3\}.$$

Steinerovo stablo T može sadržavati najviše $|V \setminus K|$ Steinerovih točaka po definiciji Steinerove točke. Također primjetimo da su svi listovi Steinerovog stabla točno svi vrhovi iz skupa terminalnih točaka K. Iz navedenog slijedi da vrijedi $|V_T|= k + |S_2|+|S_3|$. Kako je T stablo, vrijedi

$$\sum_{v \in V_T} d(v) = 2|E_T|= 2(|V_T|-1) = 2(k + |S_2|+|S_3|-1). \tag{3.1}$$

Također po definiciji skupova S_2 i S_3, slijedi

$$\sum_{v \in V_T} d(v) = \sum_{v \in K} d(v) + \sum_{v \in V_T \setminus K} d(v) \geq k + 2|S_2|+3|S_3|. \tag{3.2}$$

Pa iz 3.1 i 3.2 slijedi $|S_3|\leq k - 2$, odnosno tvrdnja leme. □

Promotrimo sada dva povezana vrha v i w, stupnja barem 3, u Steinerovom minimalnom stablu. Između ta dva vrha postoji put čiji su unutarnji vrhovi stupnja točno 2 u Steinerovom stablu te sigurno nisu terminalne točke. Dodatno, kako su to točke u Steinerovom stablu taj put je sigurno najkraći put u danoj mreži. Na prethodno opisani način možemo u danoj mreži $M = (V, E, \ell)$ izračunati sve najkraće puteve između vrhova u mreži M. Time dobivamo *mrežu potpune udaljenosti* $D(M)$. Mrežu potpune udaljenosti efektivno možemo dobiti primjenjujući Dijkstrin algoritam za nalaženje najkraćeg puta između dva vrha. Navedeno pokazuje inicijalnu ideju sljedećeg algoritma enumeracije za pronalaženje Steinerovog minimalnog stabla.

Algoritam 7 Algoritam enumeracije

Ulaz: Mreža $M = (V, E, \ell)$ i skup terminalnih točaka $K \subseteq V$.
Izlaz: Steinerovo minimalno stablo T za dani skup teminalnih točaka K.

1. Izračunaj mrežu udaljenosti $D(M) = (V, E_D, \ell_D)$ i spremi za svaki brid $\{v, w\}$ iz E_D najkraći put od vrha v do vrha w, u mreži M.

2. Za svaki podskup $S \subseteq V \setminus K$ takav da je $|S| \leq |K|-2$, izračunaj minimalno razapinjuće stablo za podmrežu mreže $D(M)$ induciranu sa $K \cup S$.

3. Odredi najmanje razapinjuće stablo dobiveno u koraku 2 te ga transformiraj u Steinerovo minimalno stablo.

Teorem 3.1.2. *Složenost algoritma enumeracije je* $O(n^2 \log n + nm + \min\{n^{k-2}, 2^{n-k}\} \cdot k^2)$.

Dokaz. Neka je $M = (V, E, \ell)$ mreža i $K \subseteq V$ skup terminalnih točaka. Mrežu udaljenosti $D(M) = (V, E_D, \ell_D)$, odnosno najkraći put između dva vrha možemo izračunati primjenom Dijkstrinog algoritma na svaki od vrhova, dakle n puta primjenimo algoritam.

Kako pomoću Dijkstrinog algoritma možemo izračunati najkraći put od pojedinog vrha u grafu do svih ostalih vrhova, u $O(n \log n + m)$ vremena. Slijedi da mrežu udaljenosti D iz prvog koraka algoritma enumeracije, možemo izračunati u $O(n^2 \log n + nm)$ vremena.

Iz leme 3.1.1 slijedi da za $S \subseteq V \setminus K$ takav da je $|S| \leq |K|-2$ vrijedi $|K \cup S| \leq 2k - 2$. U drugom koraku algoritma, potrebno je odrediti minimalno razapinjuće stablo za najviše $\min\{n^{k-2}, 2^{n-k}\}$ različitih skupova $K \cup S$ zato što vrijedi $\sum_{i=0}^{k-2} \binom{n-k}{i} \leq \min\{n^{k-2}, 2^{n-k}\}$. Koristeći Primov algoritam za nalaženje minimalnog razapinjućeg stabla, čija je složenost $O(k^2)$, imamo da je ukupna složenost drugog koraka algoritma enumeracije $O(\min\{n^{k-2}, 2^{n-k}\} \cdot k^2)$. Time dobivamo tvrdnju teorema. □

3.1.2 Dryfus-Wagner algoritam

Dryfus-Wagner algoritam je primjer dinamičkog programiranja. Algoritam se svodi na rekurziju koja računa duljinu Steinerovog minimalnog stabla za dane prave podskupove skupa terminalnih točaka. Detaljnije, algoritam prvo računa Steinerovo minimalno stablo za sve dvočlane podskupove skupa terminalnih točaka K pa u drugom koraku računa Steinerova minimalna stabla za sve tročlane podskupove skupa K i tako dalje. Algoritam se zaustavlja kada se nađe Stenierovo minimalno stablo za cijeli skup terminalnih točaka K.

Uvedimo sljedeću notaciju. Za podskup X skupa terminalnih točaka K i $v \in V \setminus X$ označimo sa $s(X \cup \{v\})$ duljinu Steinerovog minimalnog stabla za $X \cup \{v\}$, a sa $s_v(X \cup \{v\})$ duljinu Steinerovog minimalnog stabla za $X \cup \{v\}$ pri čemu vrh v je stupnja barem 2. Iz

prethodnog slijedi da vrijedi nejednakost: $s_v(X \cup \{v\}) \geq s(X \cup \{v\})$. Pokažimo sada da se označene vrijednosti mogu jednostavno i izračunati te će nam one biti osnova za kasniju formulaciju algoritma.

Lema 3.1.3. *Neka je $X \subseteq K$, $X \neq \emptyset$ i $v \in V \setminus X$ proizvoljan.*
Tada vrijedi sljedeće:

$$s_v(X \cup \{v\}) = \min_{\emptyset \neq X' \subset X} \{s(X' \cup \{v\}) + s(X \setminus X' \cup \{v\})\} \tag{3.3}$$

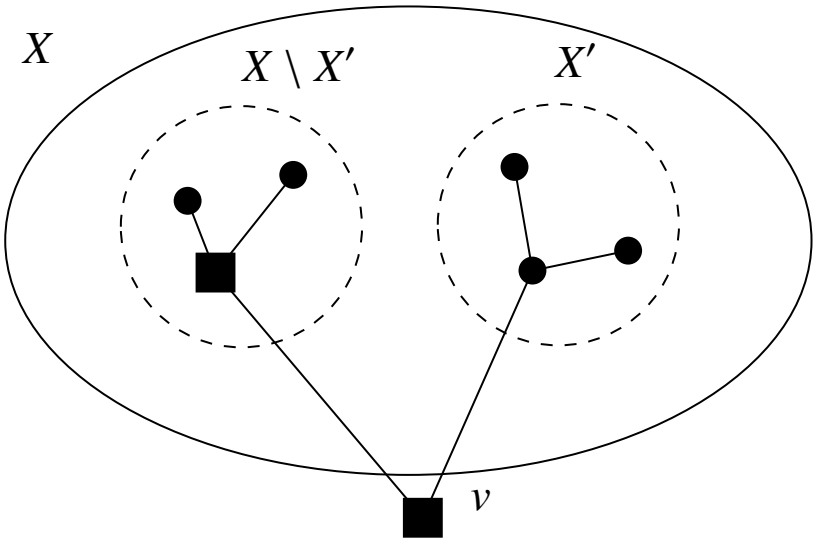
$$s(X \cup \{v\}) = \min\{\min_{w \in X} \{p(v, w) + s(X)\}, \min_{w \in V \setminus X} \{p(v, w) + s_w(X \cup \{w\})\}\} \tag{3.4}$$

pri čemu $p(v, w)$ označava duljinu najkraćeg puta između vrhova v i w.

Dokaz. Neka je $M = (V, E, \ell)$ mreža te K skup terminalnih točaka kao u prethodnom razmatranju. Neka je $X \subseteq K$ te $v \in V \setminus X$. Neka je T Steinerovo stablo za skup točaka $X \cup \{v\}$ čija je duljina $s_v(X \cup \{v\})$ pri čemu je vrh v stupnja barem 2.

Sada možemo stablo T podijeliti na dva podstabla T_1 i T_2 tako da T_1 bude Steinerovo stablo za skup vrhova $X' \cup \{v\}$ te T_2 Steinerovo stablo za skup $X \setminus X' \cup \{v\}$, pri čemu je X' neprazan pravi podskup od X, tj. za X' vrijedi $\emptyset \neq X' \subset X$. Ilustracija prethodno opisanog postupka prikazana je na slici 3.1, pri čemu su terminalne točke označene krugovima, a neterminalne kvadratima.

Sada minimizacijom po svim pravim podskupovima X' dobijemo 3.3.



Slika 3.1: Dekompozicija Steinerovog stabla na dva podstabla.

Preostaje za dokazati 3.4. U tu svhu neka je T, kao i prije, Steinerovo stablo za skup $X \cup \{v\}$ no vrh v ne mora biti stupnja barem 2. Za vrh v stabla T, neka P_v označava najduži

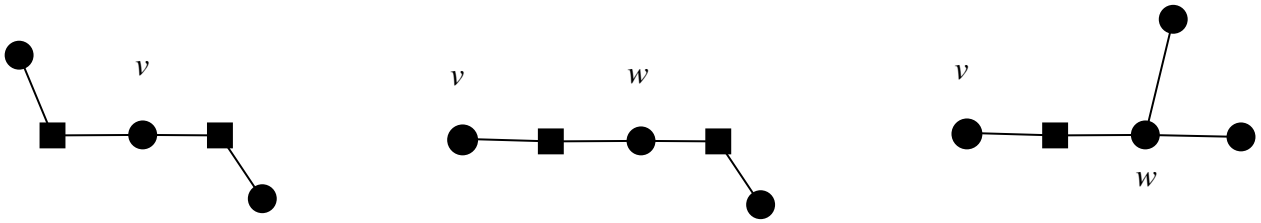
put u T iz vrha v pri čemu svi unutarnji vrhovi puta su stupnja 2 u stablu T te ne nalaze se u skupu K. Razlikujemo sljedeća tri slučaja koja su prikazana i na slici 3.2:

(1) Vrh v je stupnja 2 u stablu T. Tada je $s(X \cup \{v\}) = s_v(X \cup \{v\})$, odnosno tada je duljina puta P_v jednaka 0.

(2) Vrh v je list u stablu T i put P_v završava u vrhu $w \in X$. Tada je T unija Steinerovog stabla za skup vrhova X i najkraćeg puta od vrha v do w, tj. imamo $s(X \cup \{v\}) = p(v, w) + s(X)$.

(3) Vrh v je list u stablu T i put P_v završava u vrhu $w \notin X$. Tada je vrh w stupnja barem 3 u stablu T, odnosno stablo T je dobiveno kao unija Steinerovog minimalnog stabla za skup $X \cup \{w\}$ u kojemu vrh w ima stupanj barem 2 i najkraćeg puta od vrha v do vrha w. Time dolazimo do sljedećeg, $s(X \cup \{v\} = p(v, w) + s_w(X \cup \{w\})$.

Minimizacijom po gornjim slučajevima dolazimo do tvrdnje 3.4.



Slučaj 1 Slučaj 2 Slučaj 3

Slika 3.2: Tri slučaja u dokazu tvrdnje 3.4, leme 3.1.3.

□

Sada možemo pomoću prethodne leme iskazati Dreyfus-Wagnerov algoritam za Steinerovo minimalno stablo.

Algoritam 8 Dreyfus-Wagner algoritam

Ulaz: Mreža $M = (V, E, \ell)$ i skup terminalnih točaka $K \subseteq V$.

Izlaz: Duljina Steinerovog minimalnog stabla T za dani skup teminalnih točaka K.

$\forall v, w \in V$: izračunaj $p(v, w)$
$\forall \{x, y\} \subseteq K$: postavi $s(\{x, y\}) := p(x, y)$
for `i=2,...,k-1` **do**

  for $X \subseteq K$ t.d. $|X| = i$ i $v \in V \setminus X$ **do**
    $s_v(X \cup \{v\}) = \min_{\emptyset \neq X' \subset X} \{s(X' \cup \{v\}) + s(X \setminus X' \cup \{v\})\}$
  for $X \subseteq K$ t.d. $|X| = i$ i $v \in V \setminus X$ **do**
    $s(X \cup \{v\}) = \min\{\min_{w \in X} \{p(v, w) + s(X)\}, \min_{w \in V \setminus X} \{p(v, w) + s_w(X \cup \{w\})\}\}$

Teorem 3.1.4. *Dreyfus-Wagnerov algoritam računa duljinu Steinerovog minimalnog stabla u* $O(3^k n + 2^k n^2 + n^2 \log n + nm)$ *koraka.*

Dokaz. Korektnost algoritma proizlazi iz leme 3.1.3. Slično kao i kod teorema 3.1.2 složenost prva dva koraka algoritma je $O(n^2 \log n + nm)$. Navedeno proizlazi iz toga što za izračunavanje najkraćeg puta između vrhova v i w primjenimo Dijkstrin algoritam n, tj. za svaki od vrhova po jednom, sveukupno n puta.

Nadalje u rekurziji 3.3 leme 3.1.3 ukupan broj operacija je jednak kao i broj načina na koje možemo izabrati v, X' i X. Kako svaka terminalna točka skupa terminalnih točaka K pripada točno jednom od skupova X', $X \setminus X'$ i $K \setminus X$, ukupan broj koraka u rekurziji 3.3 je reda veličine $O(3^k n)$. Iz prethodnog zapravo vrijedi $\forall t \in K$ vrijedi $(t \in X') \oplus (t \in X \setminus X') \oplus (t \in K \setminus X)$.

Za svaki par (X, v) rekurzija iskazana u 3.4 obavi $O(n)$ operacija zbrajanja i usporedbi. Također broj parova je jednak broju načina na koje možemo izabrati podskup X i vrh $v \in V \setminus X$, dakle $O(2^k n)$ načina. Iz prethodnih razmatranja sumiranjem dolazimo do krajnjeg rezultata za složenost Dreyfu-Wagnerovog algoritma. □

Primjetimo da Dreyfus-Wagnerov algoritam iskazan u algoritmu 8 računa samo ukupnu duljinu Steinerovog minimalnog stabla. Kako bi došli i do same konstrukcije stabla potrebno je tijekom izračunavanja pamtiti skupove i vrhove za koje je izračunata minimalna vrijednost od $s(X \cup \{v\})$ i $s_v(X \cup \{v\})$.

3.2 Približni algoritmi

Kako je Steinerov problem u mreži $\mathcal{NP}$–težak problem, trenutno ne znamo za algoritam koji efikasno rješava navedeni problem, odnosno algoritam koji rješava problem u poli-

nomijalnom vremenu. Obzirom na složenost prethodno opisanih algoritama koji daju egzaktno rješenje, u ovom odjeljku proučavamo aproksimacijske algoritme koji kao rezultat ne moraju nužno dati optimalno rješenje, nego daju rješanje koje znatno ne odstupa od optimalnog rješenja te je njihova složenost manja.

Kao mjeru za kvalitetu aproksimacijskog algoritma koristit ćemo maksimalni omjer između rješenja koje vrati promatrani aproksimacijski algoritam i optimalnog rješenja pri čemu gledamo maksimum po svim dopustivim instancama problema. Taj omjer nazivamo i omjer performansi aproksimacijskog algoritma. U slučaju Steinerovog problema u mrežama, omjer performansi aproksimacijskog algoritma promatrat ćemo kroz omjer ukupne dužine Steinerovog stabla koje vrati aproksimacijski algoritam i dužine minimalnog Steinerovog stabla gdje promatramo maksimum po svim dopustivim instancama Steinerovog problema u mrežama.

3.2.1 Aproksimacijski algoritam s omjerom preformansi 2

Neka je $M = (V, E, \ell)$ mreža, K skup terminalnih točaka te ℓ pripadna težinska (nenegativna) funkcija kao i u prethodnim razmatranjima. Steinerovom problemu za mrežu M pridružujemo mrežu potpune udaljenosti $M_D = (K, E_D, \ell_D)$ na sljedeći način. Skup vrhova od M_D je pripadni skup teminalnih točaka K, skup bridova jednak je $E_D = \binom{K}{2}$ te težinska funkcija ℓ_D pridružuje svakom bridu iz $\{v_1, v_2\}E_D$ duljinu najkraćeg puta između vrhova v_1 i v_2. Označimo duljinu Steinerovog minimalnog stabla u M za dani skup terminalnih točaka K sa $smt(M)$.

Nadalje, iskažimo krucijalnu lemu za aproksimacijski algoritam koja daje vezu između duljine Steinerovog minimalnog stabla u mreži i minimalnog razapinjućeg stabla u pripadnoj mreži potpune udaljenosti.

Lema 3.2.1. *Neka je $M = (V, E, \ell)$ mreža te K skup terminalnih točaka. Za svako minimalno razapinjuće stablo T u mreži potpune udaljenosti M_D vrijedi*

$$\ell_D(T) \leq (2 - 2/k) \cdot smt(M) \tag{3.5}$$

pri čemu k označava kardinalnost skupa temrinalnih točaka, $k = |K|$.

Dokaz. Neka je S proizvoljno Steinerovo minimalno stablo u M. Promotrimo sada šetnju po bridovima stabla S sa vanjske strane u mreži M. Ova šetnja prolazi kroz sve terminalne točke i bridove točno 2 puta, pa je njezina duljina točno dvostruko dulja od duljine Steinerovog stabla S.

Neka je t broj listova u stablu S. Tada se šetnja W sastoji od $t \leq |K|$ puteva između susjednih listova u stablu S. Kada izdvojimo najduži takav put u šetnji W, duljina preostale šetnje je najviše $(1 - 1/t)$ puta duljina polazne šetnje W.

Sada primjetimo da pomoću prethodno konstruirane šetnje možemo konstruirati razapinjuće stablo u mreži M čija je duljina najviše $(1 - 1/t)$ puta duljina polazne šetnje W.

□

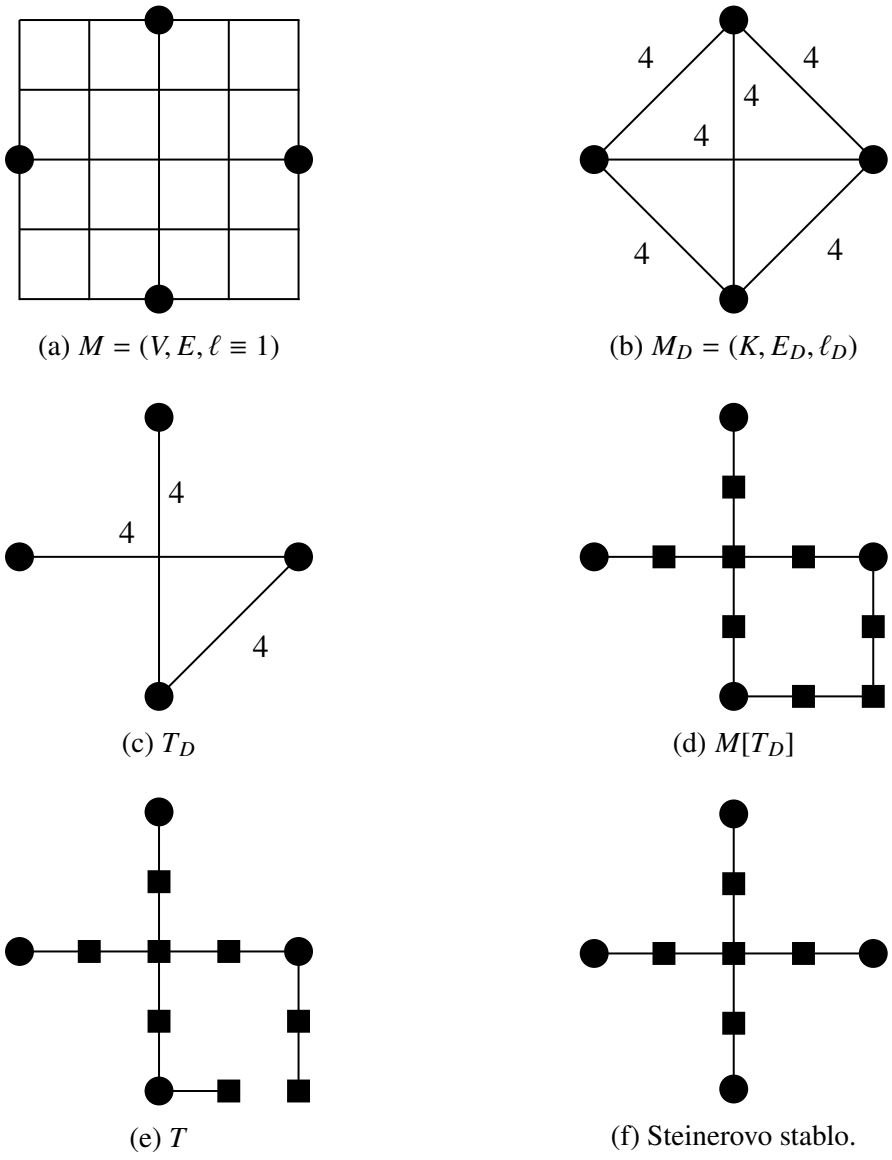
Iz prethodne leme vidimo da pomoću minimalnog razapinjućeg stabla mreže potpune udaljenosti možemo ocijeniti duljinu Steinerovog minimalnog stabla. Sada iskažimo aproksimacijski algoritam za računanje Steinerovog stabla u mreži.

Algoritam 9 Aproksimacijski MST algoritam

Ulaz: Mreža $M = (V, E, \ell)$ i skup terminalnih točaka $K \subseteq V$.
Izlaz: Steinerovo stablo S za dani skup teminalnih točaka K.

1. Izračunaj mrežu udaljenosti $M_D = (K, E_D, \ell_D)$.
2. Izračunaj minimalno razapinjuće stablo T_D za mrežu M_D.
3. Transformiraj stablo T_D u podmrežu $M[T_D]$ od M_D tako da zamijeniš svaki brid od T_D sa pripadnim najkraćim putem.
4. Izračunaj minimalno razapinjuće stablo T za podmrežu $M[T_D]$.
5. Preoblikuj stablo T u Steinerovo stablo za M tako da izbacimo sve listove koji nisu terminalne točke.

Ilustrirajmo rad prethodno opisanog algoritma na primjeru. Na slici 3.3 prikazan je rad algoritma 9 na jednostavnom primjeru pri čemu vrhovi kvadrata na slici označavju vrhove mreže M, a krugovi označavaju terminalne točke. Imamo jediničnu mrežu pa je duljina svih bridova u prikazanoj mreži sa slike 3.3a jednaka 1 te imamo 25 vrhova od čega su kružićima označene terminalne točke. Prvi korak algoritma prikazan je na slici 3.3b gdje je potrebno izračunati mrežu udaljenosti. Slika 3.3c prikazuje izračunato minimalno razapinjuće stablo T_D za mrežu M_D kao što je opisano u drugom koraku algoritma. Transformacija minimalnog razapinjućeg stabla iz trećeg koraka prikazana je na slici 3.3d. Kao što vidimo na slici 3.3d, u stablu imamo ciklus pa je potrebno odrediti minimalno razapinjuće stablo kao što je opisano u četvrtom koraku algoritma. Jedno od razapinjućih stabla, odnosno jedno od mogućih rješenja četvrtog koraka prikazano je na slici 3.3e. Konačno na kraju algoritma, u zadnjem koraku potrebno je još izbaciti listove koji nisu terminalne točke te je rezultat prikazan na slici 3.3f.



(a) $M = (V, E, \ell \equiv 1)$

(b) $M_D = (K, E_D, \ell_D)$

(c) T_D

(d) $M[T_D]$

(e) T

(f) Steinerovo stablo.

Slika 3.3: Primjer aproksimacijskog MST algoritma za Steinerovo stablo

Bibliografija

[1] D. Cieslik, *Steiner Minimal Trees*, Springer, 2010.

[2] X. Hu D. Du, *Steiner Tree Problems in Computer Communication Networks*, World Scientific Publishing Company, 2008.

[3] M. L. Overton D. R. Dreyer, *Two Heuristics for the Euclidean Steiner Tree Problem*, 2002.

[4] N. Emanet, *The Rectilinear Steiner Tree Problem - Sequential and Parallel Alorithms for the Rectilinear Steiner Tree Problem.*, Lambert Academic Publishing, 2010.

[5] J. Holby, *Variations on the E ariations on the Euclidean Steiner T uclidean Steiner Tree Problem and Algorithms oblem and Algorithms*, Rose-Hulman Undergraduate Mathematics Journal: Vol. 18 **18** (2017).

[6] F.K. Hwang, *The Steiner Tree Problem.*, 2012.

[7] H. Jürgen, *The Steiner Tree Problem.*, Springer, 2002.

[8] Z.A. Melzak, *On the Problem of Steiner*, Canadian Mathematical Bulletin **4** (1961).

[9] G. Soothill, *The Euclidean Steiner Problem*, 2010.

Sažetak

U ovom radu promatramo Steinerova stabla. Uvodno, iskazana je ukratko teorija grafova i neki rezultati iz teorije grafova potrebni za daljnje razmatranje Steinerovih stabla. Najprije promatramo Steinerova stabla u Euklidskom prostoru. Početno je iskazan Fermatov problem kao motivacija za proučavanje Steinerovih stabla te su dane dvije metode koje rješavaju Fermatov problem. Pomoću rješenja Fermatovog problema dolazimo i do prvog algoritma za Steinerov Euklidski problem. Dodatno iskazan je i numerički algoritam koji rješava Steinerov problem u Euklidskom prostoru. Posljednje proučavamo i Steinerov problem u mrežama. Navedeni su algoritmi za egzaktno i približno rješenje Steinerovog problema u mrežama.

Summary

In this thesis, we observe Steiner trees. In the introduction, we present fundamentals of graph theory and some results from graph theory required for further consideration of Steiner trees. Firstly, we observe Steiner's trees in Euclidean space. Fermat's problem is initially presented as a motivation for studying Steiner trees and two methods are given that solve Fermat's problem. Using the solution of Fermat's problem, we arrive at the first algorithm for Steiner's Euclidean problem. The numerical algorithm that solves the Steiner problem in Euclidean space is additionally presented. Lastly, we also study Steiner's problem in networks. Algorithms for the exact and approximate solution of the Steiner problem in networks are given.

Životopis

Rođen sam 11. ožujka 1997. godine u Zagrebu, gdje sam pohađao osnovnu i srednju školu. Maturirao sam 2015. godine i upisao preddiplomski studij matematike na Prirodoslovno–matematičkom fakultetu Sveučilišta u Zagrebu. Godine 2019. završio sam preddiplomski studij i stekao zvanje sveučilišnog prvostupnika matematike. Također iste godine sam upisao diplomski studij Računarstvo i matematika na Prirodoslovno–matematičkom fakultetu Sveučilišta u Zagrebu. Akademske godine 2018./2019. dobio sam rektorovu nagradu. Tijekom studija radio sam na raznim projektima u Supremumu consulting d.o.o. i Diverto d.o.o. Trenutno sam zaposlen u Ericsson Nikola Tesla d.d.