

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Veronika Horvat

**Iskanje najcenejše poti v grafih preko  
polkolobarjev**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Polona Oblak

Ljubljana 2014



Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V računalništvu obstaja vrsta problemov, ki jih ne moremo reševati s klasičnimi algebraičnimi strukturami. Eden od njih je iskanje najcenejše poti v usmerjenem grafu. Za iskanje najcenejše poti v grafu lahko uporabimo Bellmanove enačbe. Če enačbe postavimo v model tropskega polkolobarja, dobimo enostaven algoritem nad matrikami z elementi v tropskem polkolobarju, ki nam določi najcenejšo pot v pripadajočem usmerjenem grafu.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisana Veronika Horvat, z vpisno številko **63110205**, sem avtorica diplomskega dela z naslovom:

*Iskanje najcenejše poti v grafih preko polkolobarjev*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom doc. dr. Polone Oblak,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 29. avgusta 2014

Podpis avtorja:





*Rada bi se zahvalila svoji mentorici Poloni Oblak za pomoč pri izbiri teme diplomskega dela, za vsak hiter odgovor po e-pošti in za vse korektne popravke, ki ste mi jih dali tekom pisanja naloge.*

*Prav tako bi se rada zahvalila svojim sošolcem Tilnu, Nini, Lukatu, Janu, Mancu in Mojci za pomoč, ki sem jo bila deležna tekom študija.*

*Iskrena hvala tudi moji družini za vso podporo in spodbujanje, ko sem ju najbolj potrebovala.*



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Tropska algebra</b>	<b>3</b>
<b>3</b>	<b>Reševanje predoločenih sistemov linearnih enačb</b>	<b>9</b>
<b>4</b>	<b>Iskanje najcenejše poti v grafu</b>	<b>17</b>
4.1	Kvazi inverz matrike . . . . .	17
4.2	Bellmanove enačbe v povezavi z iskanjem najcenejše poti v grafu . .	19
4.3	Bellmanov algoritem . . . . .	22
<b>5</b>	<b>Zaključek</b>	<b>29</b>



# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>TXT</b>	text file	besedilna datoteka
<b>SVD</b>	singular value decomposition	razcep singularnih vrednosti



# Povzetek

V diplomskem delu je predstavljen problem iskanja najcenejših poti v usmerjenem grafu s pomočjo Bellmanovega algoritma. Algoritem je različica Bellmanovih enačb z novostjo, da delo poteka nad matrikami in operacijami tropskega polkolobarja. Za utežen usmerjen graf cene povezav zapišemo v matriko  $A$ . Matrika ima kvazi inverz  $A^*$  nad tropskim polkolobarjem, s pomočjo katerega izračunamo minimalno rešitev sistema enačb, izraženega iz Bellmanovih enačb. Rešitev algoritma je vektor cen najcenejših poti do vseh vozlišč v grafu.

**Ključne besede:** Tropska algebra, kvazi inverz, Bellmanov algoritem, usmerjen graf.





# Abstract

The thesis presents the problem of finding the shortest path in a directed graph with the Bellman's algorithm. This algorithm a version of Bellman equation with the novelty that the we work with matrices and operations over tropical semiring. Given a weighted directed graph, its arc prices are written in a matrix  $A$ . The matrix has a quasi-inverse  $A^*$  over tropical semiring, which is used to compute the minimal solution of the system of equations expressed from the Bellman equation. The solution of the algorithm is a vector of prices that represents the prices of the shortest paths to all nodes in the graph.

**Keywords:** Tropical algebra, quasi inverse, Bellman's algorithm, directed graph.



# Poglavje 1

## Uvod

Do pred kratkim je delo razvijalcev zlasti na področjih fizike in uporabne matematike temeljilo na uporabi klasičnih algebraičnih struktur kot so grupe, obsegi ter kolobarji. V zadnjem času pa je prihajalo do vedno večjih potreb po uporabi novejših načinov reševanja problemov, saj običajne algebrske strukture ne vsebujejo primer-  
nih orodij za modeliranje in reševanje nekaterih problemov. Kot primer takšnega problema lahko podamo iskanje najcenejših poti v grafu, katerega bomo kasneje tudi podrobneje predstavili.

Diplomsko delo je sestavljeno iz treh poglavij. V prvem bo predstavljenih nekaj osnov tropske algebre, kot so osnovne operacije in njihove lastnosti. V drugem poglavju se bomo osredotočili na matrike nad realnimi števili in kako rešiti sistem enačb, če ta ima rešitev oziroma kako poiskati najboljši približek, če rešitev ne obstaja. Za računanje sistema, katerega rešitev ne obstaja bomo uporabili posplošen Moore-Penroseov inverz. Tako bomo vzpostavili povezavo s tropskim svetom, v katerem prav tako obstaja poseben kvazi inverz za reševanje sistemov linearnih enačb. Eden od problemov, ki ga rešuje takšen sistem je iskanje najcenejših poti v grafih, ki bo predstavljen v zadnjem poglavju.

Za iskanje takšne poti bomo uporabili Bellmanov algoritem, katerega smo tudi implementirali. Za lažje preverjanje algoritma smo spisali javanski program, ki na podlagi željenega števila vozlišč generira graf, katerega cene povezav so naključna števila med 1 in 9. Tako smo lahko na več grafih preverili delovanje algoritma. Prav

tako nas je zanimala časovna kompleksnost spisanega algoritma in kaj nanjo najbolj vpliva.

## Poglavje 2

# Tropska algebra

Za boljše razumevanje implementiranega algoritma bomo za začetek predstavili nekaj osnovnih pojmov sveta, v katerem smo problem najcenejših poti rešili. Tropska algebra, ki je poznana tudi pod imenom minus-plus algebra (zaradi definicije tropskega polkolobarja), je relativno novo področje v matematiki. Skozi leta je dobila vedno večji pomen v matematiki, predvsem v kombinatoriki, algebri in novejših področjih matematike. Pridevnik “tropska” so francoski matematiki podelili v čast brazilskemu kolegu Imre Simonu, ki je bil eden od pionirjev minus-plus algebre [4].

*Tropski polkolobar* je množica realnih števil z dodatnim elementom pozitivne neskončnosti

$$S = \mathbb{R} \cup \{+\infty\},$$

na kateri definiramo operaciji seštevanja in množenja:

$$x \oplus y = \min\{x, y\} \tag{2.1}$$

$$x \otimes y = x + y. \tag{2.2}$$

Tropska vsota (2.1) je minimum dveh števil, tropski zmnožek (2.2) pa njuna vsota.

Pri nekaterih problemih bi bilo bolj uporabno, če bi pri operaciji seštevanja namesto minimuma uporabljali maksimum. V takšnih primerih je tropski polkolobar

definiran kot  $P = \mathbb{R} \cup \{-\infty\}$ , operaciji seštevanja in množenja pa sta definirani kot:

$$x \oplus' y = \max\{x, y\}$$

$$x \otimes' y = x + y.$$

Zaradi procesa iskanja rešitve najcenejše poti smo se odločili, da bomo vedno uporabljali tropski polkolobar, katerega operacija seštevanja predstavlja minimum med izbranimi številoma.

Tukaj sta dva primera tropskega seštevanja in množenja:

$$4 \oplus 8 = 4$$

$$4 \otimes 8 = 12.$$

Kot vidimo smo v tropski aritmetiki osredotočeni le na seštevanje in množenje. Če imamo v realnem svetu podano enačbo  $3 + x = 10$  je ta enostavno rešljiva:

$$3 + x = 10$$

$$x = 10 - 7$$

$$x = 3.$$

Do problema pride, če želimo isto enačbo rešiti v svetu tropskih števil.

$$3 \oplus x = 10$$

$$\min\{3, x\} = 10.$$

Enačbo je nemogoče rešiti, saj bi moral biti  $x$  po eni strani enak 10, da bi ustrezal desni strani enačbe, po drugi strani pa bi moralo biti število 10 manjše od 3. Tukaj pride do očitnega protislovja.

Mnogo dobro poznanih lastnosti iz aritmetike velja tudi v tropski matematiki in jih je mogoče enostavno preveriti:

1. Seštevanje in množenje sta komutativna, kar pomeni, da velja:

$$x \oplus y = y \oplus x \quad \text{in} \quad x \otimes y = y \otimes x.$$

2. Velja, da je operacija množenja distributivna nad operacijo seštevanja:

$$x \otimes (y \oplus z) = x \otimes y \oplus x \otimes z \quad \text{in}$$

$$(x \oplus y) \otimes z = x \otimes z \oplus y \otimes z.$$

Ne smemo pozabiti, da ima operacija množenja prednost pred seštevanjem, zato bomo tukaj podali primer za lažje razumevanje distributivnosti in nadaljnjega seštevanja in množenja:

$$3 \otimes (6 \oplus 2) = 3 \otimes 2 = 5,$$

$$3 \otimes (6 \oplus 2) = 3 \otimes 6 \oplus 3 \otimes 2 = 9 \oplus 5 = 5.$$

3. Prav tako za obe operaciji velja zakon asociativnosti:

$$(x \oplus y) \oplus z = x \oplus (y \oplus z) \quad \text{in}$$

$$(x \otimes y) \otimes z = x \otimes (y \otimes z).$$

4. Obe operaciji imata svoj nevtralni element. To je poseben element, za katerega za vsak  $a \in S$  velja:

$$e \star a = a \star e = a.$$

Pri tem je  $\star$  operacija, za katero obstaja nevtralni element.

- (a) Ker za vsak  $a \in S$  velja  $x \oplus (+\infty) = x$ , je  $+\infty$  nevtralni element seštevanja. Odslej ga bomo prikazovali z znakom

$$\varepsilon = +\infty.$$

.

- (b) Ker za vsak  $a \in S$  velja  $x \otimes 0 = x$ , je 0 nevtralni element množenja. Tudi tega bomo odslej prikazovali z novim znakom

$$e = 0.$$

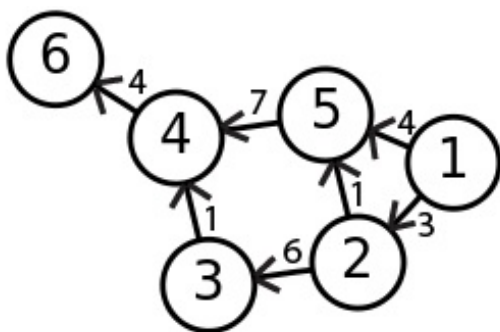
5. Operacija seštevanja je idempotentna. Torej za vsak  $a \in S$  velja:

$$a \oplus a = a$$

Ker preverjanje našega algoritma temelji na grafih, smo se odločili v nadaljevanju predstaviti nekaj osnovnih pojmov o grafih.

**Definicija 2.1** Usmerjen utežen graf  $G$  je definiran kot trojica  $(V, E, c)$ , kjer je  $V$  množica elementov, ki predstavljajo vozlišča,  $E$  nabor usmerjenih povezav med vozlišči in  $c$  množica cen med posameznimi povezavami.

Kljub temu, da je možno med dvema vozliščema imeti več povezav  $(u, v)$ , ki imajo isto začetno ( $u$ ) in končno ( $v$ ) vozlišče, se bomo mi ukvarjali izključno z grafi, v katerih je med vozliščema največ ena povezava  $(u, v)$ . To pa še ne pomeni, da ne more obstajati povezava  $(v, u)$ . Prav tako bomo za vsako povezavo podali določeno ceno  $c_{uv}$  in na podlagi teh izračunali najcenejše poti do določenih vozlišč. Primer enostavnega grafa je prikazan na sliki 2.1.



Slika 2.1: Graf

Vsak graf z  $n$  vozlišči lahko zapišemo z matriko velikosti  $n \times n$ . Za lažjo nadaljnjo razlago smo jo poimenovali matrika  $A$ . Vsak element matrike predstavlja ceno povezave med dvema vozliščema. Element  $A_{ij}$  je cena povezave med vozliščem  $i$  in vozliščem  $j$ . Če med dvema vozliščema ni povezave, je vrednost elementa na tem mestu  $\varepsilon$ . Za vse diagonalne elemente  $A_{ii}$ , katerih pripadajoča povezava ima isto začetno in končno točko, smo se odločili vedno podati ceno  $e$ . Na primer, matrika katere elementi so določeni z grafom na sliki 2.1 je enaka:



$$A = \begin{pmatrix} e & 3 & \varepsilon & \varepsilon & 4 & \varepsilon \\ \varepsilon & e & 6 & \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon & e & 1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & 4 \\ \varepsilon & \varepsilon & \varepsilon & 7 & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \end{pmatrix}$$

Definirajmo še tropsko seštevanje in množenje na matrikah. Če imamo podani matriki  $A$  in  $B$  z elementi iz  $S$ , definiramo njuno vsoto kot matriko, katere elementi so minimum istoležnih elementov matrik  $A$  in  $B$ .

$$(A \oplus B)_{ij} = A_{ij} \oplus B_{ij}$$

To je možno le, če sta  $A = (A_{ij})$  in  $B = (B_{ij})$  enake velikosti.

Primer seštevanja:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \oplus \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} \min\{1, 5\} & \min\{2, 6\} \\ \min\{3, 7\} & \min\{4, 8\} \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Matrično množenje matrik  $A$  in  $B$  je možno le, če je število stolpcev matrike  $A$  enako številu vrstic matrike  $B$  (torej, velikost  $A$  je  $m \times n$ ,  $B$  pa  $n \times r$ ) [1]. Definiramo:

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^n A_{ik} \otimes B_{kj}$$

Primer množenja:

$$\begin{aligned} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \otimes \begin{pmatrix} 5 & \varepsilon \\ e & \varepsilon \end{pmatrix} &= \begin{pmatrix} \min\{1+5, 2+e\} & \min\{1+\varepsilon, 2+\varepsilon\} \\ \min\{3+5, 4+e\} & \min\{3+\varepsilon, 4+\varepsilon\} \end{pmatrix} \\ &= \begin{pmatrix} \min\{6, 2\} & \min\{\varepsilon, \varepsilon\} \\ \min\{8, 4\} & \min\{\varepsilon, \varepsilon\} \end{pmatrix} = \begin{pmatrix} 2 & \varepsilon \\ 4 & \varepsilon \end{pmatrix} \end{aligned}$$

Matrika

$$I_n = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \in \mathbb{R}^{n \times n} \quad (2.3)$$

predstavlja nevtralni element za množenje matrik nad  $\mathbb{R}$ .

Velja:

$$A \cdot I_n = A \text{ in } I_m \cdot A = A, \text{ za vse } A \in \mathbb{R}^{m \times n}$$

Prav tako obstaja identična matrika v množici tropskih matrik, katere elementi na diagonali so enaki  $e$ , ostali pa  $\varepsilon$ :

$$I = \begin{pmatrix} e & \varepsilon & \varepsilon & \dots & \varepsilon \\ \varepsilon & e & \varepsilon & \dots & \varepsilon \\ \varepsilon & \varepsilon & e & \dots & \varepsilon \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varepsilon & \varepsilon & \varepsilon & \dots & e \end{pmatrix}$$

## Poglavje 3

# Reševanje predoločenih sistemov linearnih enačb

Za trenutek se vrnimo v svet realnih števil in osredotočimo na reševanje sistema linearnih enačb za katerega je značilno, da ima lahko eno, nobene ali neskončno možnih rešitev. Podanih imamo  $m$  enačb z  $n$  neznankami:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\dots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m. \end{aligned} \tag{3.1}$$

Tako napisan sistem (3.1) lahko zapišemo v matrični obliki:

$$Ax = b, \tag{3.2}$$

kjer je  $A$  matrika sistema,  $x$  vektor neznank in  $b$  vektor desne strani enačb.

Poleg matrike  $A$  potrebujemo tudi razširjeno matriko  $\tilde{A}$ . To dobimo tako, da na desni strani matrike  $A$  dodamo vektor  $b$ :

$$\tilde{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ & \cdots & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{pmatrix}$$

**Izrek 3.1** *Sistem linearnih enačb (3.2) je rešljiv natanko tedaj, ko je rang matrike  $A$  enak rang matrike  $\tilde{A}$ .*

Eden od načinov kako ugotoviti rang matrike je, da poiščemo vrstično kanonično formo za matriko s pomočjo Gaussove eliminacije, kot bomo to naredili pri primeru 3.1. Rang je enak številu pivotov.

**Izrek 3.2** *Če je sistem  $Ax = b$  rešljiv, je njegova rešitev enaka  $x = Gb$  natanko tedaj, ko je  $G$  posplošen inverz matrike  $A$ . To je matrika za katero velja:  $AGA = A$ .*

*Dokaz.* Pokazali bomo le, da za posplošen inverz  $G$  velja, da je  $Gb$  rešitev sistema  $Ax = b$ . Velja  $AGA = A$  in želimo dokazati, da je  $A(Gb) = b$ . Ker je sistem rešljiv, obstaja takšen  $y$ , da velja:  $Ay = b$

$$A(Gb) = A(G(Ay)) = Ay = b.$$

□

V primeru, da je matrika  $A$  obrnljiva, obstaja takšna matrika  $A^{-1}$ , da velja:

$$A \cdot A^{-1} = A^{-1} \cdot A = I.$$

Potem je rešitev enačbe (3.2) enaka:

$$x = A^{-1}b.$$

Obrnljive so lahko le kvadratne matrike. To so matrike, katere število stolpcev je enako številu vrstic.

**Primer 3.1** *Podan imamo sistem treh linearnih enačb z dvema neznankama*

$$\begin{aligned} -2x + 3y &= 5 \\ x - 2y &= -4 \\ \frac{1}{2}x + y &= 4, \end{aligned} \tag{3.3}$$

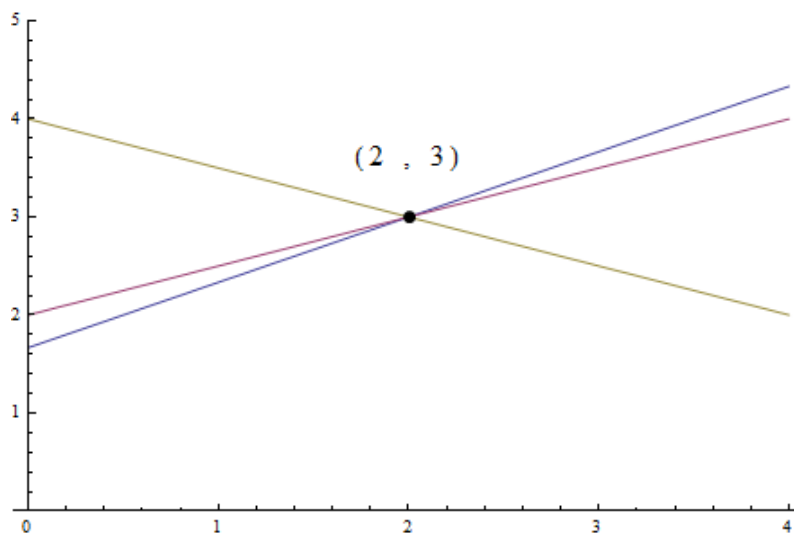
zapisan v matrični obliki:

$$A = \begin{pmatrix} 1 & -2 \\ -2 & 3 \\ \frac{1}{2} & 1 \end{pmatrix} \text{ in } b = \begin{pmatrix} -4 \\ 5 \\ 4 \end{pmatrix}.$$

Na razširjeni matriki naredimo Gaussovo eliminacijo in tako dobimo rešitev:  $x = 2$ ,  $y = 3$ :

$$\tilde{A} = \left( \begin{array}{cc|c} 1 & -2 & -4 \\ -2 & 3 & 5 \\ \frac{1}{2} & 1 & 4 \end{array} \right) \rightarrow \left( \begin{array}{cc|c} 1 & -2 & -4 \\ 0 & 1 & -3 \\ 0 & 2 & 6 \end{array} \right) \rightarrow \left( \begin{array}{cc|c} 1 & -2 & -4 \\ 0 & -1 & -3 \\ 0 & 0 & 0 \end{array} \right) \rightarrow \left( \begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{array} \right)$$

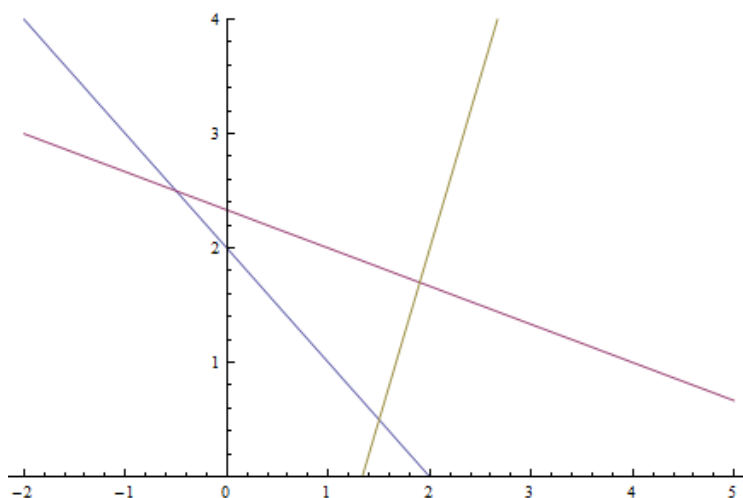
Kot smo omenili že v izreku 3.1 ima sistem rešitev, če sta ranga matrik  $A$  in  $\tilde{A}$  enaka. V tem primeru sta oba ranga enaka 2. Da obstaja rešitev lahko vidimo tudi na grafu na sliki 3.1, saj se vse premice sekajo v isti točki, katere koordinate so enake rešitvi sistema.



Slika 3.1: Graf treh premic, ki se sekajo v točki, ki je rešitev sistema enačb (3.3).

**Primer 3.2** Tudi tokrat imamo podan sistem treh enačb z dvema neznankama

$$\begin{aligned} x + y &= 2 \\ x + 3y &= 7 \\ 3x - y &= 4 \end{aligned} \tag{3.4}$$



Slika 3.2: Graf treh premic, ki se ne sekajo v skupni točki.

Če prikažemo tri premice na grafu na sliki 3.2, lahko vidimo, da se ne sekajo v skupni točki. Iz tega lahko predvidevamo, da rešitev ne obstaja. Tudi po izračunu rangov matrik lahko vidimo, da rešitve ni, saj je rang matrike  $A$  enak 2, matrike  $\tilde{A}$  pa 3.

V takšnih primerih se lahko osredotočimo na iskanje takšnega  $x$ , da bo  $Ax$  kar najbolj podoben  $b$ . Iščemo torej takšen  $x$ , da bo dolžina  $\|Ax - b\|$  najmanjša. Tega dobimo s pomočjo Moore-Penroseovega posplošenega inverza  $A^+$ :

$$x = A^+b. \tag{3.5}$$

**Definicija 3.1** Moore-Penroseov posplošeni inverz je poseben posplošen inverz matrike  $A$ , ki zadostuje naslednjim štirim lastnostim:

1.  $AA^+A = A$
2.  $A^+AA^+ = A^+$
3.  $(AA^+)^T = AA^+$
4.  $(A^+A)^T = A^+A$

Izračunamo ga s pomočjo matrik, dobljenimi pri singularnem razcepu matrike  $A$ .

**Izrek 3.3** (SVD [3]) Vsako matriko  $A$  velikosti  $n \times m$  lahko izrazimo kot produkt matrik  $U$ ,  $\Sigma$  in  $V^T$  tako, da velja:

$$A = U\Sigma V^T. \quad (3.6)$$

1.  $U$  je ortogonalna matrika velikosti  $n \times n$  z levimi singularnimi vektorji, tj. lastnimi vektorji matrike  $A^T A$
2.  $V$  je ortogonalna matrika velikosti  $m \times m$  z desnimi singularnimi vektorji, tj. lastnimi vektorji matrike  $AA^T$

3. diagonalna matrika  $\Sigma = \begin{pmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \sigma_3 & & \\ & & & \ddots & \\ & & & & \sigma_n \end{pmatrix}$  velikosti  $n \times m$ , katere elemente  $\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_n$  imenujemo singularne vrednosti.

Moore-Penroseov inverz matrike  $\Sigma$  je enak:

$$\Sigma^+ = \begin{pmatrix} \sigma_1^+ & & & & \\ & \sigma_2^+ & & & \\ & & \sigma_3^+ & & \\ & & & \ddots & \\ & & & & \sigma_n^+ \end{pmatrix},$$

kjer je

$$\sigma_i^+ = \begin{cases} 1/\sigma_i, & \sigma_i \neq 0, \\ 0, & \sigma_i = 0. \end{cases}$$

**Izrek 3.4** Moore-Penroseov inverz poljubne matrike  $A = U\Sigma V^T$  velikosti  $n \times m$  je enak

$$A^+ = V\Sigma^+U^T \quad (3.7)$$

Primer 3.2 nima rešitev. Če pa želimo poiskati najboljši približek  $x$  in  $y$  po metodi najmanjših kvadratov, potem se lotimo problema tako, da poiščemo matrike  $V$ ,  $U$  in  $\Sigma$  s pomočjo singularnega razcepa. Matrika  $V$  je sestavljena iz normiranih lastnih vektorjev  $v_1, v_2, \dots, v_m$ , ki pripadajo lastnim vrednostim  $\sigma_1, \sigma_2, \dots, \sigma_m$  matrike  $A^T A$ . Podobno dobimo matriko  $U$ , le da namesto matrike  $A^T A$  uporabimo matriko  $AA^T$ . Diagonalni elementi matrike  $\Sigma$  so  $\sqrt{\sigma_1}, \sqrt{\sigma_2}, \dots, \sqrt{\sigma_{\min\{n,m\}}}$ . Nato izračunamo  $\Sigma^+$  in s pomočjo enačbe 3.7 dobimo Moore-Penroseov inverz, s katerim po enačbi 3.5 poiščemo rešitev. V nadaljevanju bo prikazan potek reševanja omenjenega primera. Iz sistema (3.4) zapišemo

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 3 \\ 3 & -1 \end{pmatrix} \text{ in } b = \begin{pmatrix} 2 \\ 7 \\ 4 \end{pmatrix}.$$

Najprej izračunamo matriko  $V$ . Matrika

$$A^T A = \begin{pmatrix} 1 & 1 & 3 \\ 1 & 3 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 3 \\ 3 & -1 \end{pmatrix} = \begin{pmatrix} 11 & 1 \\ 1 & 11 \end{pmatrix}$$

ima karakteristični polinom enak  $(\lambda - 12)(\lambda - 10)$ , torej sta njeni lastni vrednosti enaki  $\lambda_1 = 12$  in  $\lambda_2 = 10$ . Za pripadajoča lastna vektorja lahko izberemo  $v_1 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$

in  $v_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ . Če ju normiramo ju lahko sestavimo v matriko:

$$V = \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

Po istem postopku izračunamo še matriko  $U$ :

$$AA^T = \begin{pmatrix} 1 & 1 \\ 1 & 3 \\ 3 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 3 \\ 1 & 3 & -1 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 2 \\ 4 & 10 & 0 \\ 2 & 0 & 10 \end{pmatrix}.$$



Tudi tokrat izračunamo karakteristični polinom, ki je enak  $\lambda(\lambda - 12)(\lambda - 10)$ . Iz njega dobimo tri lastne vrednosti  $\lambda_1 = 12$ ,  $\lambda_2 = 10$  in  $\lambda_3 = 0$ , s pomočjo katerih izberemo tri pripadajoče lastne vektorje  $u_1 = \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix}$ ,  $u_2 = \begin{pmatrix} 0 \\ -1 \\ 2 \end{pmatrix}$  in  $u_3 = \begin{pmatrix} -5 \\ 2 \\ 1 \end{pmatrix}$ .

Tudi te normiramo in iz njih sestavimo matriko:

$$U = \begin{pmatrix} -\frac{1}{\sqrt{6}} & 0 & -\frac{5}{\sqrt{30}} \\ -\frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{30}} \\ -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{30}} \end{pmatrix}.$$

Nato iz singularnih vrednosti matrike  $A$  sestavimo diagonalno matriko:

$$\Sigma = \begin{pmatrix} \sqrt{12} & 0 \\ 0 & \sqrt{10} \\ 0 & 0 \end{pmatrix}.$$

Ker nas zanima ali smo pravilno izračunali singularni razcep, po enačbi 3.6 zmnožimo  $U\Sigma V^T$ , za katero ugotovimo, da je enaka matriki  $A$ . Potem, ko smo se prepričali o pravilnosti matrik  $V$ ,  $U$  in  $\Sigma$ , izračunamo:

$$\Sigma^+ = \begin{pmatrix} \frac{1}{\sqrt{12}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{10}} & 0 \end{pmatrix}.$$

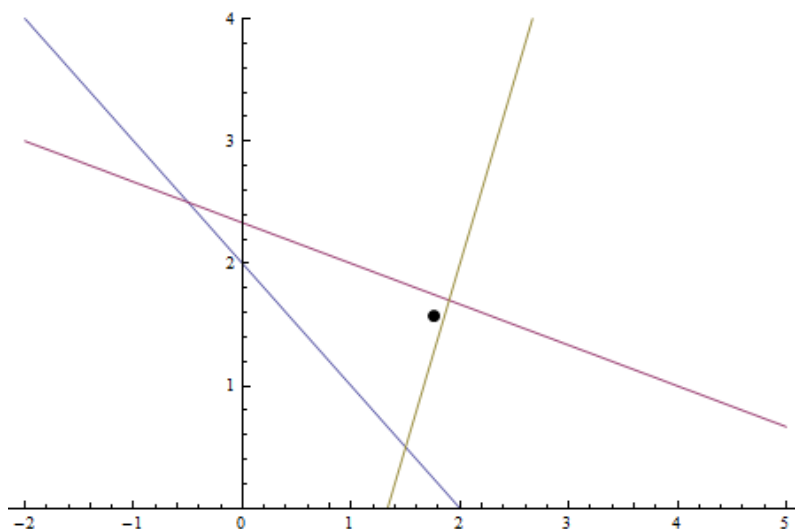
Tako imamo dovolj podatkov, da izračunamo Moore-Penroseov inverz

$$A^+ = \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{12}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{10}} & 0 \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ -\frac{5}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{1}{\sqrt{30}} \end{pmatrix} = \begin{pmatrix} 0.0833 & 0.0667 & -0.2833 \\ 0.0833 & 0.2667 & -0.1167 \end{pmatrix}$$

in rešimo sistem (3.4) po enačbi (3.5) kot

$$x = A^+b = \begin{pmatrix} 0.0833 & 0.0667 & -0.2833 \\ 0.0833 & 0.2667 & -0.1167 \end{pmatrix} \begin{pmatrix} 2 \\ 7 \\ 4 \end{pmatrix} = \begin{pmatrix} 1.7667 \\ 1.5667 \end{pmatrix}.$$

Rešitev sistema je torej  $x = \begin{pmatrix} 1.7667 \\ 1.5667 \end{pmatrix}$ . V grafu na sliki 3.3 jo lahko prikažemo kot točko, ki po metodi najmanjših kvadratov najboljše aproksimira premice, podane z enačbami (3.4).



Slika 3.3: Graf treh premic brez skupnega presečišča s točko, ki je od treh premic najmanj oddaljena.

Tako se izračuna rešitev, če sta vektorja  $x$  in  $b$  stolpca, vendar je prav tako mogoče sistem rešiti s podanimi vrsticami. Iz enačbe

$$x^T A = b^T,$$

izračunamo  $x^T$ :

$$x^T = b^T A^+.$$

V tropskem svetu obstaja veliko posplošenih inverzov. Eden izmed njih je  $A^*$ , ki ga bomo predstavili v poglavju 4.1. Reši nek sistem tropskih enačb. Oba približka  $A^*b$  in  $A^+b$  rešitev linearnega sistema  $Ax = b$ , ki ju dobimo z  $A^*$  v tropskem svetu ali  $A^+$  v realnem svetu sta najbližja vektorju  $b$ .

# Poglavje 4

## Iskanje najcenejše poti v grafu

V tem poglavju se vrnimo v svet tropske algebre in se osredotočimo na iskanje najcenejše poti v grafu. Kot je prikazano v prejšnjem poglavju, če enačba  $Ax = b$  v realnem svetu nima rešitve, potrebujemo le Moore-Penroseov inverz za izračun  $x$ , da se  $Ax$  karseda najbolj prilega  $b$ . Tudi v tropskem svetu obstaja podoben kvazi inverz  $A^*$ , s pomočjo katerega lahko rešimo sistem enačb oblike:

$$y = y \otimes A \oplus b^T.$$

Rešitev, ki jo dobimo s pomočjo kvazi inverza je v primeru, da je  $A$  matrika cen povezav v grafu enaka vektorju najcenejših poti. Da to drži, bomo dokazali v poglavju 4.2.

### 4.1 Kvazi inverz matrike

Naj bo  $A$  kvadratna matrika velikosti  $n \times n$  z elementi v polkolobarju  $(S, \oplus, \otimes)$ . Za poljuben  $k \in \mathbb{N}$  je  $A^k$   $k$ -ta potenca matrike  $A$ , torej  $A \otimes A \otimes \dots \otimes A$  ( $k$  krat). Matrika  $A^{(k)}$  je definirana kot

$$A^{(k)} = I \oplus A \oplus A^2 \oplus \dots \oplus A^k, \tag{4.1}$$

kjer je  $I$  identična matrika [2].

**Trditev 4.0.1** *Velja  $A^{(k)} = (I \oplus A)^k$ .*

*Dokaz.* Ker je  $a \oplus a = \min\{a, a\} = a$ , je tudi  $A \oplus A = A$ . Dokaz z indukcijo:

1. Trditev velja za  $k = 1$ :

$$A^{(1)} = (I \oplus A)^1$$

2. Če trditev velja za  $k$ , velja tudi za  $k + 1$ :

$$\begin{aligned} (I \oplus A)^{(k+1)} &= (I \oplus A)^k \otimes (I \oplus A) = A^{(k)} \otimes (I \oplus A) = \\ &\stackrel{\text{def(4.1)}}{=} (I \oplus A \oplus \dots \oplus A^k) \otimes (I \oplus A) = \\ &= I \oplus \dots \oplus A^k \oplus A \oplus A^2 \oplus \dots \oplus A^{k+1} = \\ &\stackrel{\text{idempotentnost}}{=} I \oplus A \oplus \dots \oplus A^{k+1} \stackrel{\text{def(4.1)}}{=} A^{(k+1)} \end{aligned}$$

□

**Definicija 4.1** *Kvazi inverz  $A^*$  matrike  $A$  je limita*

$$A^* = \lim_{k \rightarrow \infty} A^{(k)},$$

*če ta obstaja.*

**Trditev 4.0.2** *V tropih velja:*

$$e \oplus a \oplus a^2 \oplus \dots \oplus a^k = \min\{e, a, a^2, \dots, a^k\}.$$

*Če je  $a$  poljubno nenegativno število, trditev vedno velja.*

Naslednje trditve ne bomo dokazovali v celoti.

**Trditev 4.0.3** *Če so uteži vseh ciklov v grafu  $G(A)$  nenegativna števila, limita  $A^*$  obstaja in velja*

$$A^* = \lim_{k \rightarrow \infty} A^{(k)} = A^{(n-1)} = A^{(n)} = \dots,$$

*kjer je  $n$  število vozlišč v grafu. Zato:*

$$A^* = I \oplus A \oplus \dots \oplus A^{n-1}. \quad (4.2)$$

*Velja tudi, da inverz izpolnjuje matrični enačbi:*

$$A^* = I \oplus A \otimes A^* = I \oplus A^* \otimes A.$$

*Dokaz.*

$$\begin{aligned} I \oplus A \otimes A^* &\stackrel{\text{enakost(4.2)}}{=} I \oplus A \otimes (I \oplus A \oplus \cdots \oplus A^{n-1}) \\ &= I \oplus A \oplus A^2 \oplus \cdots \oplus A^n = A^{(n)} = A^{(n-1)} = A^*. \end{aligned}$$

Drugo enakost dokažemo podobno. □

## 4.2 Bellmanove enačbe v povezavi z iskanjem najcenejše poti v grafu

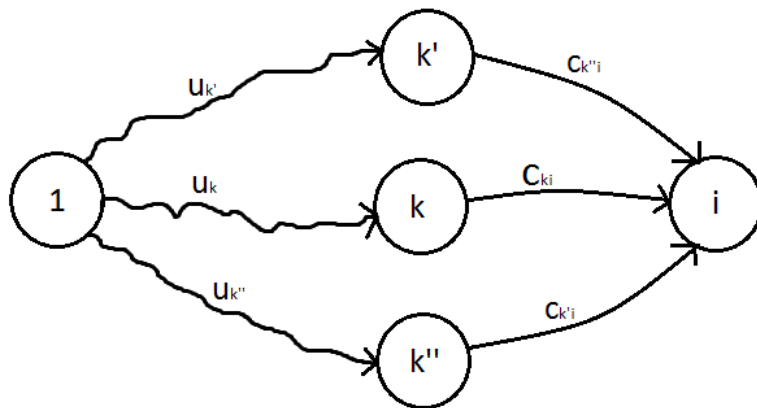
Podan imamo graf  $G = [V, E, c]$ , kjer je  $V$  množica vozlišč,  $E$  množica povezav in  $c$  množica cen teh povezav. Zanima nas, kako bi iz prvega vozlišča najceneje prišli do vseh ostalih vozlišč. Za izhodišče bi lahko izbrali poljubno vozlišče v grafu, vendar smo se odločili, da bomo vedno izhajali iz prvega. Kot primer iz realnega sveta lahko podamo potnika, ki želi iz neke železniške postaje najceneje pripotovati do večih krajev. Vozlišča v grafu so tukaj železniške postaje, povezave so železniške proge, cene pa cena vozovnice od enega postajališča do drugega. Eden od načinov, kako to izračunati, je s pomočjo Bellmanovih enačb:

$$u_i = \begin{cases} 0, & \text{če je } i = 1 \\ \min_{k \neq i} \{u_k + c_{ki}\}, & \text{če je } i \neq 1 \end{cases} \quad (4.3)$$

za  $i = 1, 2, \dots, n$ . Pri tem  $u_k$  predstavlja ceno poti od začetnega vozlišča do neposrednega predhodnika izbranega končnega vozlišča.  $c_{ki}$  pa predstavlja ceno povezave med predhodnim in končnim vozliščem. Pomen obeh je grafično prikazan na sliki 4.1.

Prej napisane Bellmanove enačbe lahko nekoliko drugače zapišemo tudi v svetu trojske algebre. Nahajamo se v polkolobarju  $(S, \oplus, \otimes)$ , kjer sta operaciji  $\oplus$  in  $\otimes$  definirani kot je zapisano v enakostih (2.1) in (2.2). Prav tako ne smemo pozabiti na nevtralna elementa seštevanja in množenja  $\varepsilon$  ter  $e$ . Tako enačbe (4.3) zapišemo kot

$$y^{t+1} = \begin{cases} b^T, & \text{če je } t = 1, \\ y^t \otimes A \oplus b^T, & \text{če je } t \neq 1, \end{cases} \quad (4.4)$$

Slika 4.1: Grafični prikaz cen  $u_k$  in  $c_{ki}$ .

za  $t = 1, 2, \dots, n$ .

Z vektorjem  $b^T$  velikosti  $n$  podamo, iz katerega vozlišča želimo izračunati najcenejše poti. V primeru, da je začetno vozlišče prvo vozlišče v grafu, je vektor enak  $b^T = [e \ \varepsilon \ \dots \ \varepsilon]$ . Matrika  $A$  velikosti  $n \times n$  predstavlja cene povezav med posameznimi vozlišči. Bellmanove enačbe lahko pretvorimo v algoritem, ki ga imenujemo Bellmanov algoritem. Za njega je značilno iterativno izvajanje. Ob prvem obhodu je  $y$  enak vektorju  $b^T$ , nato pa se ob vsakem obhodu starim cenam poti prištejejo cene povezav do naslednjega vozlišča iz matrike  $A$ . Med vsemi cenami se izbere tista, ki je najmanjša.

Če poračunamo prvih nekaj členov algoritma, pridemo do novega zanimivega spoznanja:

$$\begin{aligned}
 y^1 &= b^T \\
 y^2 &= b^T \otimes A \oplus b^T \\
 y^3 &= y^2 \otimes A \oplus b^T = (b^T \otimes A \oplus b^T) \otimes A \oplus b^T = \\
 &= b^T \otimes A^2 \oplus b^T \otimes A \oplus b^T = b^T \otimes (I \oplus A \oplus A^2) \\
 &\vdots \\
 y^t &= b^T \otimes (I \oplus A \oplus \dots \oplus A^t).
 \end{aligned} \tag{4.5}$$

Iz enakosti (4.5) je razvidno, da konvergiranje Bellmanovega algoritma enačimo s konvergiranjem vrste:  $I \oplus A \oplus \dots \oplus A^t \oplus \dots$ . Ker imamo v grafu  $n$  vozlišč lahko predvidevamo, da bo algoritem konvergiral v največ  $n - 1$  iteracijah, saj v  $n - 1$  korakih pregleda vseh  $n$  vozlišč. Torej je  $y^n$  vektor najcenejših poti v grafu. Sistem Bellmanovih enačb lahko zapišemo tudi v matrični obliki:

$$y = y \otimes A \oplus b^T. \quad (4.6)$$

Ker je  $b^T \otimes A^* \otimes A \oplus b^T = b^T \otimes (A^* \otimes A \oplus I)$ , ustreza  $b \otimes A^*$  rešitvi enačbe (4.6). Zato po trditvi 4.0.3 Bellmanov algoritem konvergira k:

$$y = b^T \otimes A^*,$$

kjer je  $y$  vrstica,  $b$  pa stolpec.

Pri urejenosti v tropskem kolobarju velja še več:  $b^T \otimes A^*$  ni samo rešitev enačbe (4.6), ampak je tudi minimalna rešitev sistema glede na  $\preceq$ . Pri tem s  $\preceq$  označimo linearno urejenost v tropskem polkolobarju. Definiramo jo s predpisom

$$a \preceq b \Leftrightarrow \exists c : b = a \oplus c = \min\{a, c\}. \quad (4.7)$$

Da bi se prepričali, da je  $b^T \otimes A^*$  minimalna rešitev, v enačbo (4.6) namesto  $y$  na desni strani vstavimo  $y = y \otimes A \oplus b^T$ . Sledi:

$$\begin{aligned} y &= (y \otimes A \oplus b^T) \otimes A \oplus b^T = \\ &= y \otimes A^2 \oplus b^T \otimes (I \oplus A) \end{aligned}$$

Če postopek ponavljamo  $k$ -krat dobimo:

$$y = y \otimes A^k \oplus b^T \otimes A^{(k-1)}$$

Tako za  $k \geq n$  dobimo:

$$y = y \otimes A^k \oplus b^T \otimes A^* = \min\{y \otimes A^k, b^T \otimes A^*\}$$

Po (4.7) velja  $b^T \otimes A^* \preceq y$ , zato je  $b^T \otimes A^*$  minimalna rešitev sistema.

### 4.3 Bellmanov algoritem

Iskanje najcenejše poti deluje po načelu dinamičnega programiranja. To je metoda, ki pri reševanju sistematično preglejuje vse možne poti in zato tudi pride do optimalne rešitve [5]. Osnova dinamičnega programiranja so Bellmanove enačbe iz katerih lahko zapišemo Bellmanov algoritem.

---

#### Algoritem 1 Bellmanov algoritem

---

Določitev prve vrstice matrike  $A^*$  ali dokaz, da  $A^{(K)} \neq A^*$

$$y^0 = b^T = (e, \varepsilon, \dots, \varepsilon)$$

$$t = 1$$

**loop**

▷  $y^t$  je trenutna rešitev

$$y^{t+1} = y^t \otimes A \oplus b^T$$

**if**  $y^{t+1} = y^t$  **then**

▷ algoritem se konča

$$y^t = b^T \otimes A^*$$

break

**if**  $y^{t+1} \neq y^t \wedge k \leq K$  **then**

$$t = t + 1$$

**if**  $y^{t+1} \neq y^t \wedge k = K$  **then**

▷ prekini z računanjem:  $A^* \neq A^{(K)}$

break

---

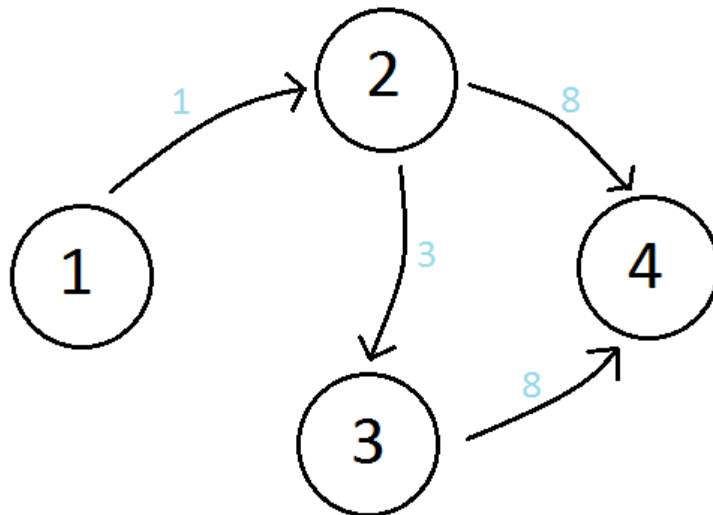
Zgoraj je napisana pseudokoda Bellmanovega algoritma (Algoritem 1), katerega smo v Matlabu implementirali. Da smo lahko preverili pravilnost le tega, smo v jeziku Java napisali enostaven program, ki glede na vpisano število vozlišč sestavi graf, katerega cene povezav so naključna števila med 1 in 9. Tako nastalo matriko cen  $A$  zapiše v txt datoteko, ki se v Matlabu prebere in na podlagi te se izvede Bellmanov algoritem. Primer takšne matrike je matrika

$$A = \begin{bmatrix} e & 1 & \varepsilon & \varepsilon \\ \varepsilon & e & 3 & 8 \\ \varepsilon & \varepsilon & e & 8 \\ \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix},$$

iz katere nastane graf, prikazan na sliki 4.2. Za boljše razumevanje algoritma bomo



na roke po algoritmu 1 izračunali ceno najcenejših poti za prej omenjen graf. Kot začetno vozlišče smo si izbrali prvo, zato bo vektor  $b^T = [e \ \varepsilon \ \varepsilon \ \varepsilon]$ . Ker imamo štiri vozlišča vemo, da se bo algoritem končal po največ treh ponovitvah.



Slika 4.2: Graf, ki nastane iz matrike na sliki A.

$$t = 1 \quad y^1 = b^T = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

$$t = 2 \quad y^2 = b^T \otimes A \oplus b^T = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} e & 1 & \varepsilon & \varepsilon \\ \varepsilon & e & 3 & 8 \\ \varepsilon & \varepsilon & e & 8 \\ \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \oplus \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} =$$

$$= \begin{bmatrix} e & 1 & \varepsilon & \varepsilon \end{bmatrix}$$

$$t = 3 \quad y^3 = y^2 \otimes A \oplus b^T = \begin{bmatrix} e & 1 & \varepsilon & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} e & 1 & \varepsilon & \varepsilon \\ \varepsilon & e & 3 & 8 \\ \varepsilon & \varepsilon & e & 8 \\ \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \oplus \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} =$$

$$= \begin{bmatrix} e & 1 & 4 & 9 \end{bmatrix}$$

$$t = 4 \quad y^4 = y^3 \otimes A \oplus b^T = \begin{bmatrix} e & 1 & 4 & 9 \end{bmatrix} \otimes \begin{bmatrix} e & 1 & \varepsilon & \varepsilon \\ \varepsilon & e & 3 & 8 \\ \varepsilon & \varepsilon & e & 8 \\ \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \oplus \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} =$$

$$= \begin{bmatrix} e & 1 & 4 & 9 \end{bmatrix}$$

Vektor, ki nastane po končanem algoritmu je enak  $y = \begin{bmatrix} e & 1 & 4 & 9 \end{bmatrix}$ . Iz tega lahko razberemo, da je cena najcenejše poti od začetnega do končnega vozlišča enaka 9. Iz vektorja lahko razberemo najcenejše poti do vseh vozlišč grafa in ne samo do končnega. Kljub temu, da je najcenejša pot do končnega vozlišča po vozliščih 1, 2 in 4, lahko vidimo, da je najcenejša pot do 3. vozlišča 1, 2 in 3.

### 4.3.1 Časovna kompleksnost

Predvidevamo, da je kompleksnost vsake operacije  $\oplus$  in  $\otimes$  enaka  $\mathcal{O}(1)$ .

**Trditev 4.0.4** Časovna zahtevnost Bellmanovega algoritma je  $\mathcal{O}(KM)$ , kjer je  $M$  število elementov, različnih od  $\varepsilon$ ,  $K$  pa število iteracij.

Dokaz. V vsaki iteraciji zmnožimo matriko z vektorjem ( $n^2$  operacij  $\oplus$  in  $\otimes$ , kjer

je  $n$  velikost matrike in vektorja) ter seštejemo dva vektorja ( $n$  operacij  $\oplus$ ), zato je kompleksnost takrat enaka  $\mathcal{O}(Kn^2)$ . V primeru, da se  $M$  elementov matrike  $A$  razlikuje od  $\varepsilon$ , je  $M \ll n^2$ , se kompleksnost zniža na  $\mathcal{O}(KM)$ .  $\square$

```

1  %the new times(+)
2  function num = multiplication(A,B)
3  -     n = size(A);
4  -     m = size(B);
5  -     if n(1) == 1 && m(1) == 1           %če sta parametra samo 2 števili
6  -         num = A+B;
7  -     else
8  -         for i=1:n(1)                   %če sta parametra matriki
9  -             for j=1:m(2)
10 -                 a = A(i,:)' ;
11 -                 b = B(:,j);
12 -                 c = a+b;
13 -                 num(i,j) = min(c);
14 -             end
15 -         end
16 -     end
17 - end

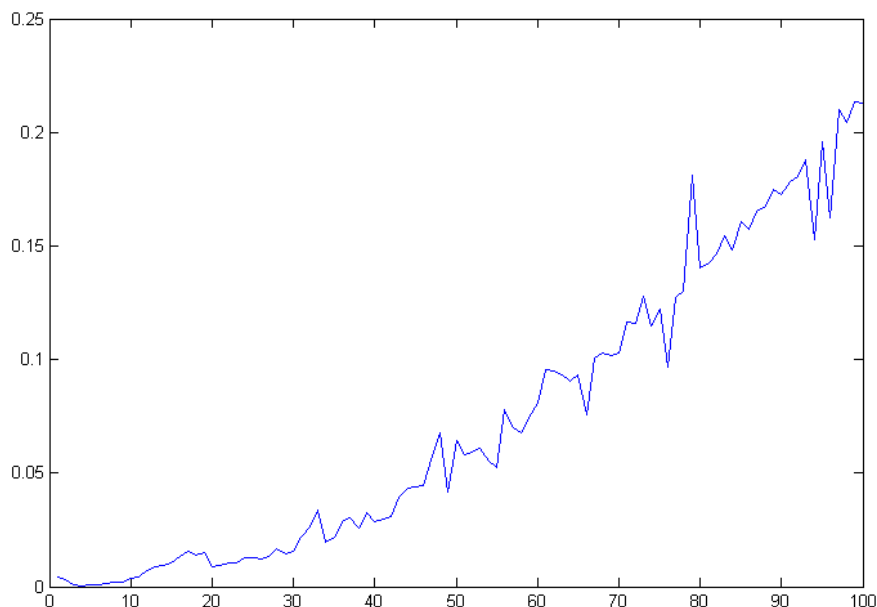
```

Slika 4.3: Funkcija tropskega množenja.

Do problema pri implementiranem algoritmu pride pri računanju osnovnih operacij  $\oplus$  in  $\otimes$ . Kot piše pred trditvijo 4.0.4, naj bi bila časovna kompleksnost obeh operacij  $\mathcal{O}(1)$ . V našem primeru za operacijo  $\oplus$  ta velja, pri  $\otimes$  pa se pojavijo težave. Večino seštevanja in množenja je matričnega, zato sta obe operaciji nekoliko zahtevnejši, kot pa če bi se izvajali nad enostavnimi realnimi števili. Funkcija množenja je prikazana na sliki 4.3. Kot vidimo vsebuje dve for zanki, zaradi katerih njena časovna zahtevnost ni enaka  $\mathcal{O}(1)$ .

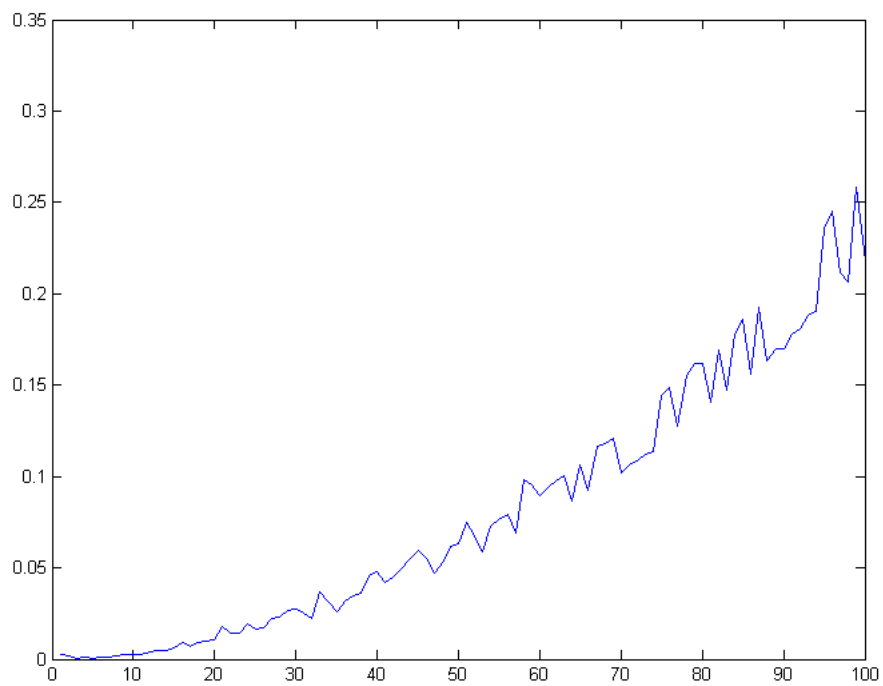
Da bi videli, kakšna je razlika med kompleksnostjo množenja in seštevanja, smo konstruirali matriko  $A$  velikosti  $10 \times 10$ . Izvedli smo operaciji seštevanja in množenja tako, da smo matriko  $A$  sešteli oziroma pomnožili samo s sabo in izmerili čas trajanja obeh funkcij. Rezultati so pokazali, da se za seštevanje porabi 0.000019 s v primerjavi z množenjem, ki se izvaja 0.001519 s. Množenje torej porabi skoraj 80-krat več časa kot seštevanje.

Zanimala nas je tudi časovna kompleksnost algoritma glede na število vozlišč grafa, zato smo zgenerirali 100 naključnih matrik in na njih preverili algoritem. Vsaka naslednja je bila za ena večja od prejšnje. Rezultati so prikazani na grafu na sliki 4.4.



Slika 4.4: Časovna kompleksnost algoritma glede na število vozlišč grafa.

Več kot očitno je, da se trajanje programa z večanjem števila vozlišč veča, saj se sorazmerno veča tudi število iteracij v algoritmu. Vseeno pa krivulja na glafu 4.4 ni monotona, verjetno zaradi naključnih matrik, ki so nastale. Te se razlikujejo v številu  $\varepsilon$ . Za matrike, iz katerih je nastal graf na sliki 4.4 je značilno, da ne obstajajo povezave v vozlišča, katerih številke so manjše od izhodišča povezave. Zato je v matrikah veliko več  $\varepsilon$  kot bi jih bilo drugače. Za primerjavo in ilustracijo trditve 4.0.4 smo še enkrat zgenerirali 100 naključnih matrik, v katerih smo pustili možnost povratnih povezav in nad njimi izvedli Bellmanov algoritem. Časovna zahtevnost teh je prikazana na sliki 4.5. Razlika ni drastična, vendar lahko vidimo, da se trajanje algoritma v prvem grafu komaj dvigne nad 0.2 s, v drugem pa je maksimum krivulje nad 0.25 s. Torej lahko predvidevamo, da se z večanjem števil, ki so različna od  $\varepsilon$ , veča tudi čas trajanja algoritma.



Slika 4.5: Časovna kompleksnost algoritma glede na število vozlišč grafa s povratnimi povezavami.



# Poglavje 5

## Zaključek

V diplomskem delu smo najprej podrobneje spoznali svet tropske algebre. Pogledali smo si na novo definirane operacije in njihove lastnosti. Prav tako smo opisali značilnosti usmerjenih grafov in kako so povezani z matrikami. V nadaljevanju smo se prestavili v svet realnih števil, v katerem smo predstavili sisteme linearnih enačb, katere smo zapisali tudi v matrični obliki. Ti lahko imajo eno, več ali nič rešitev. V primeru, da rešitev ne obstaja, smo želeli poiskati takšno, ki se rešitvi najbolj približa. Izračunali smo jo s pomočjo Moore-Penroseovega inverza in tako vzpostavili povezavo s tropsko algebro. Tudi v tej obstajajo sistemi linearnih enačb, ki jih rešimo s pomočjo podobnega inverza. Ta se imenuje kvazi inverz. Nato smo se osredotočili na glavni problem iskanja najcenejših poti v grafu. Ta je enaka minimalni rešitvi sistema  $y = y \otimes A \oplus b^T$ . Poiskali smo jo s pomočjo Bellmanovega algoritma, ki smo ga tudi implementirali. Enega izmed testiranih primerov smo tudi zapisali v delo in na njem razložili, kaj nam takšna rešitev sploh pove. Zanimala nas je tudi časovna kompleksnost takšnega algoritma in kaj vse nanajo vpliva.

Opazili smo, da implementiran algoritem deluje nad grafi, ki vsebujejo tudi negativne povezave pod pogojem, da ne vsebuje negativnih ciklov. Edini problem, ki smo ga opazili je, da je implementiran algoritem relativno počasen zaradi na novo napisane funkcije  $\otimes$ .





# Literatura

- [1] François Baccelli, Guy Cohen, Geert Jan Olsder, and Jean-Pierre Quadrat. *Synchronization and linearity*, volume 3. Wiley New York, 1992.
- [2] Michel Gondran and Michel Minoux. *Graphs, Dioids and Semirings: New Models and Algorithms (Operations Research/Computer Science Interfaces Series)*. Springer Publishing Company, Incorporated, 1 edition, 2008.
- [3] Dan Kalman. A singularly valuable decomposition: The svd of a matrix. *College Math Journal*, 27:2–23, 1996.
- [4] David Speyer and Bernd Sturmfels. Tropical mathematics. *arXiv preprint math/0408099*, 2004.
- [5] prosta enciklopedija Wikipedija. Dinamično programiranje. [http://sl.wikipedia.org/wiki/Dinami%C4%8Dno\\_programiranje/](http://sl.wikipedia.org/wiki/Dinami%C4%8Dno_programiranje/). Dostop: 26.8.2014.