

# Problem minimalnog Štajnerovog drveta

Računarska inteligencija  
Matematički fakultet

Luka Kotorčević, Branislav Jovanović

24. septembar 2020.

## Sažetak

U ovom seminarskom radu prikazaćemo rešavanje problema minimalnog Štajnerovog stabla iz oblasti teorije grafova. Upoređićemo rezultate dobijene implementacijom grube sa algoritmom optimizovanim pomoću simuliranog kaljenja.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Implementacija grube sile</b>	<b>2</b>
<b>3</b>	<b>Optimizacija simuliranim kaljenjem</b>	<b>2</b>
<b>4</b>	<b>Rezultati</b>	<b>2</b>
<b>5</b>	<b>Literatura</b>	<b>6</b>

## 1 Uvod

Problem Štajnerovog drveta u grafu, ili minimalnog Štajnerovog drveta, nazvan je po Jakobu Štajneru. Ovaj problem se može formulirati na više načina, kojima je zajedničko da se traži najkraća veza između datih objekata.

Neka je dat neusmeren graf  $G$  sa nenegativnim težinama grana, kao i  $S$  koji je podskup svih čvorova. Čvorove podskupa  $S$  nazivamo terminalima. Štajnerovo drvo je zapravo minimalno razapinjuće stablo koje sadrži sve terminale tako da je zbir ivica minimalan. Još jedna razlika je u tome da Štajnerovo drvo može sadržati još neki dodatni čvor iz grafa (koji nije terminalni) ukoliko će putanja preko tog čvora smanjiti ukupan zbir ivica.

## 2 Implementacija grube sile

Gruba sila je algoritam eksponencijalne složenosti koji sistematično nabraja sve moguće kandidate za rešavanje problema i proverava da li svaki kandidat zadovoljava problem.

Postoji slučaj kada podskup  $S$  zapravo sadrži sve čvorove grafa i tada Štajnerovo drvo postaje minimalno razapinjuće stablo i rešava se u polinomijalnom vremenu. No, kada to nije slučaj, u pitanje je NP-težak problem.

Primenom grube sile ćemo proveriti sve moguće težine puteva između dva čvora koje treba spojiti i izabrati onaj koji je najmanje težine. Put najmanje težine nalazimo pomoću Dikstrinog algoritma.

## 3 Optimizacija simuliranim kaljenjem

Optimizacija simuliranim kaljenjem se zasniva na poboljšanju jednog rešenja tj. u svakom koraku ovog algoritma ćemo proveriti rešenja u okolini trenutnog i po potrebi ga ažurirati. Za uneti graf, nasumično ćemo izabrati početni terminal i preko dfs odnosno pretrage u dubinu nalazimo put do ostalih čvorova. Algoritam se ponavlja unapred zadati broj iteracija, pamti se najbolje rešenje odnosno svaki put kada se naiđe na bolje - ažurira se trenutno rešenje.

## 4 Rezultati

Test na manjem grafu od 12 grana:

e1, 1 , 2 , 7

e2, 2 , 3 , 1

e3, 3 , 4 , 1

e4, 4 , 5 , 1

e5, 5 , 6 , 10

e6, 6 , 1 , 6

e7, 2 , 6 , 5

e8, 3 , 5 , 3

e9, 4 , 6 , 4

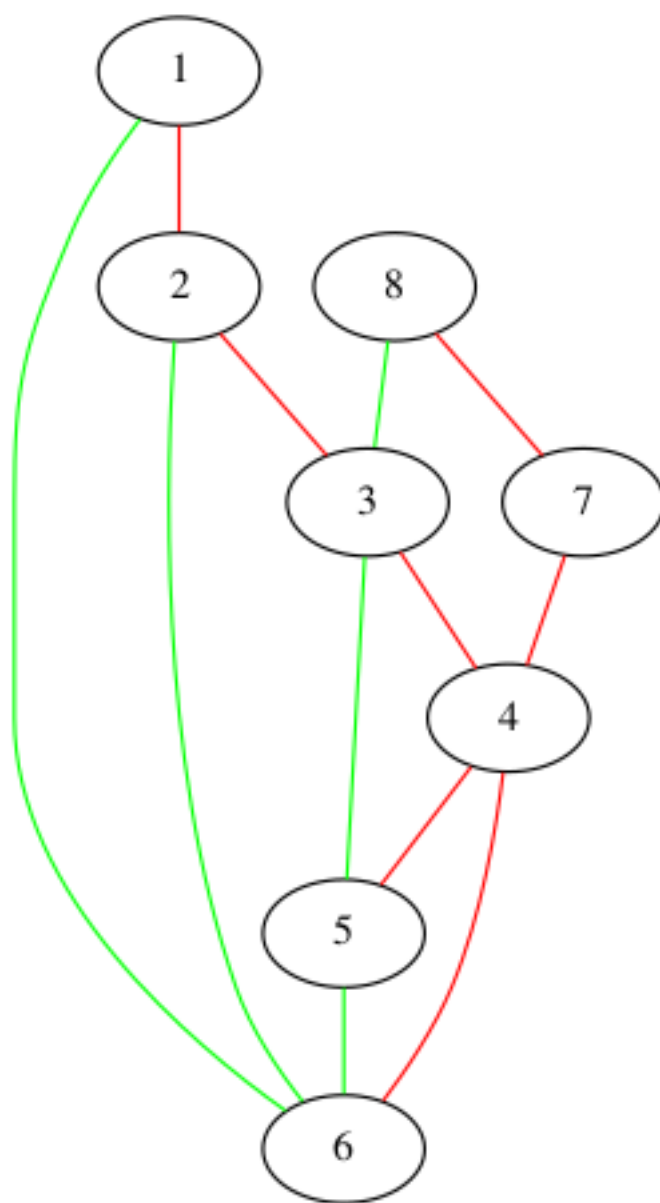
e10, 7 , 4 , 3

e11, 8 , 3 , 8

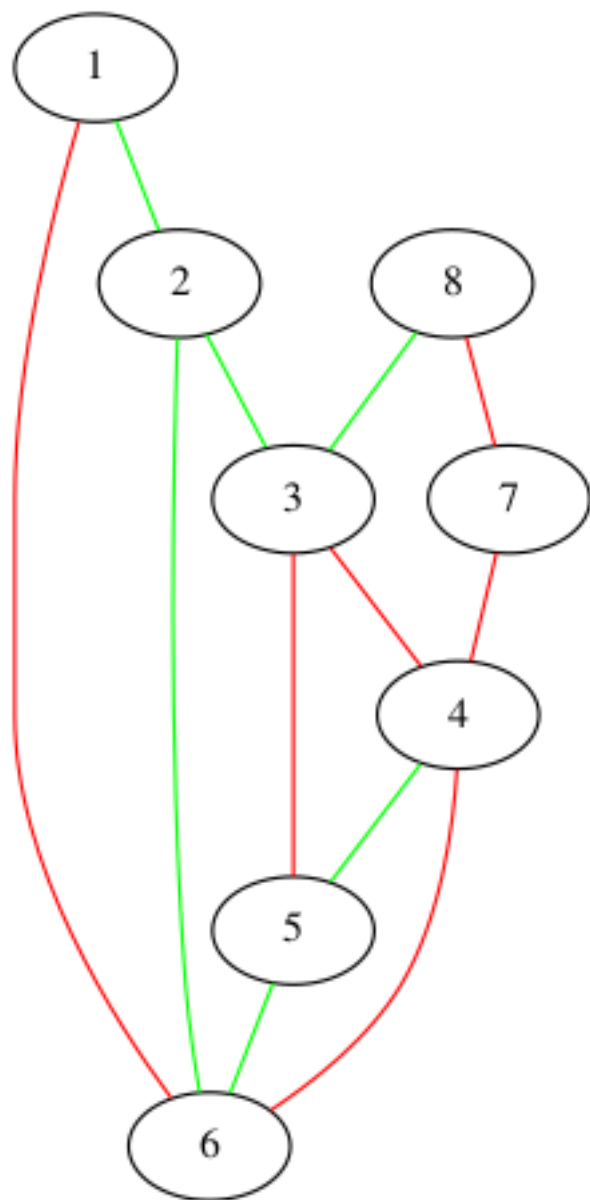
e12, 8 , 7 , 2

Terminali: 1, 3, 5, 6, 8

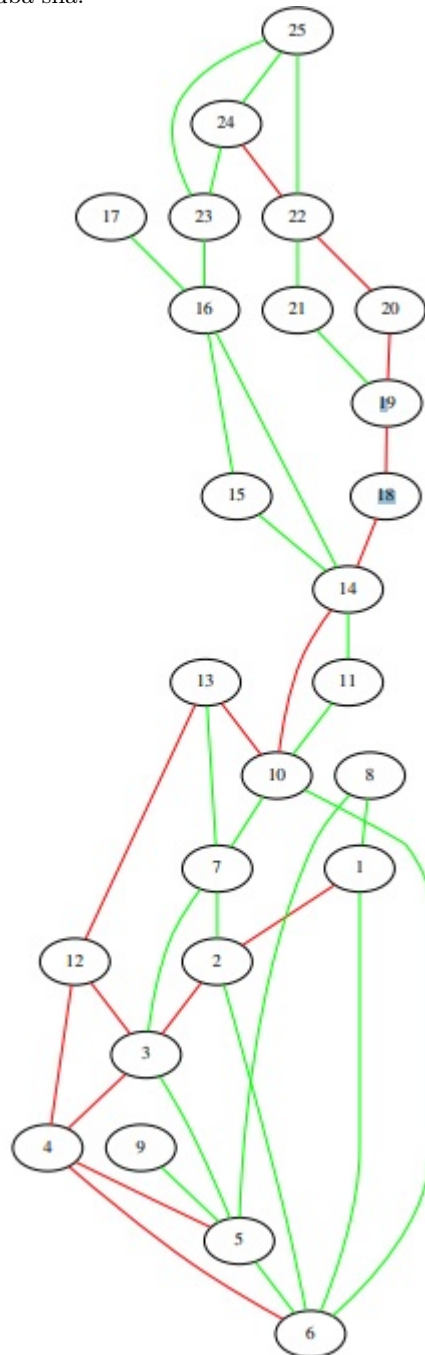
Rešenje: 19  
Gruba sila:



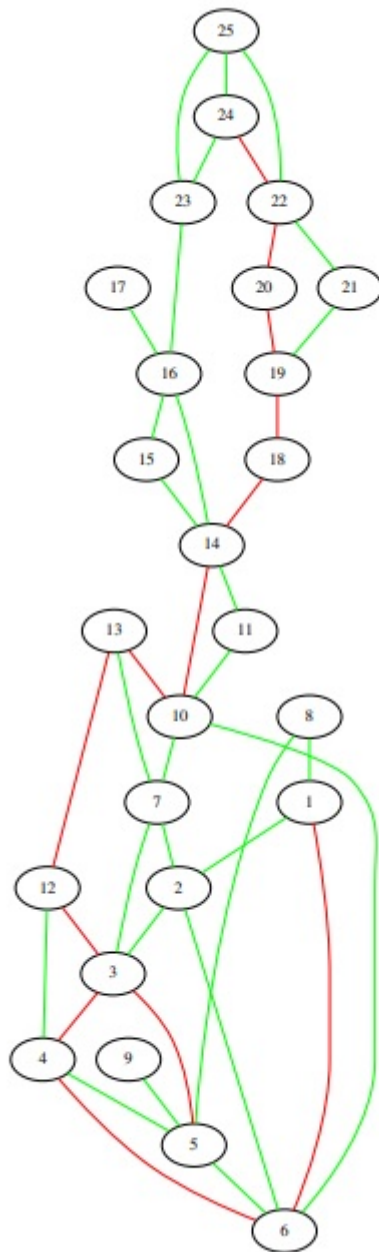
Simulirano kaljenje:



Terminali: 1, 3, 5, 6, 10, 24  
 Gruba sila:



Simulirano kaljenje:



## 5 Literatura

- Materijali sa vežbi iz Računarske inteligencije, dr Stefan Mišković
- [www.medium.com](https://www.medium.com)
- Wikipedia