

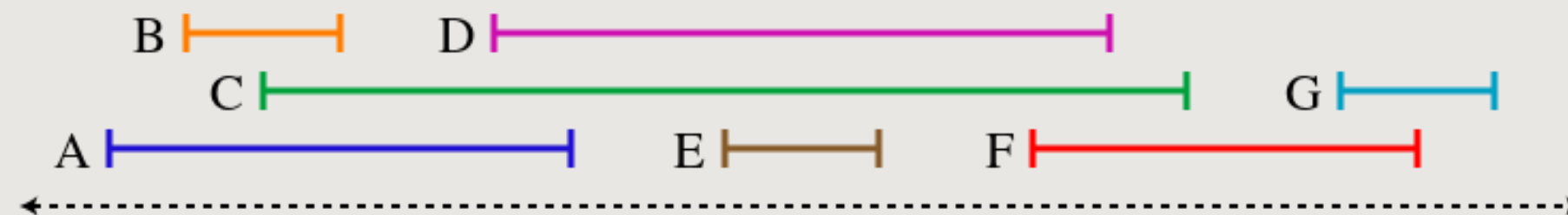
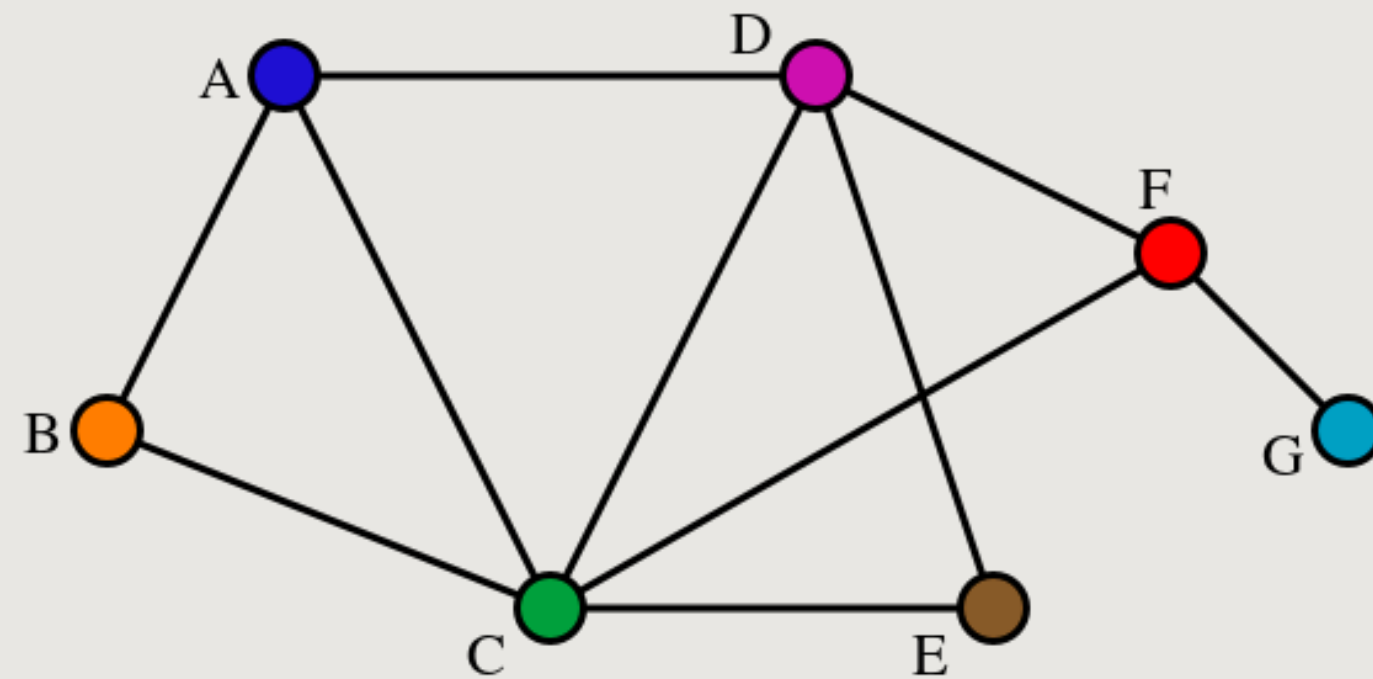


Minimum Interval Graph Completion

Mina Kovandžić 127/2019
Filip Ogrenjac 275/2019

Opis problema

Problem minimalnog kompletiranja grafa do intervalnog grafa (Minimum Interval Graph Completion)
bavi se grafovima i njihovim transformacijama u intervalne grafove.



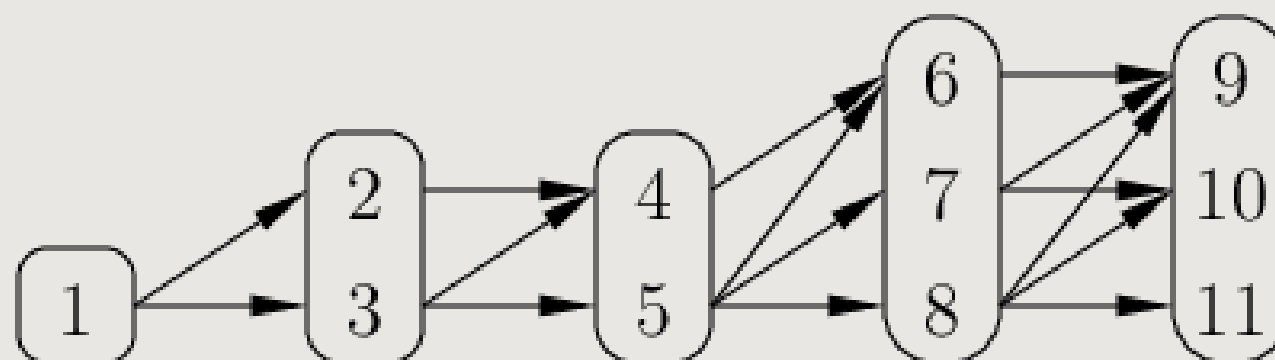
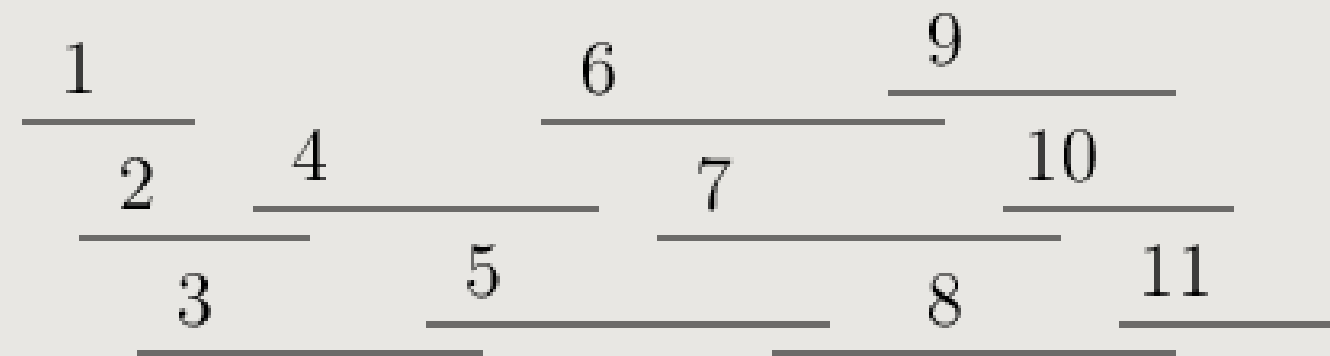
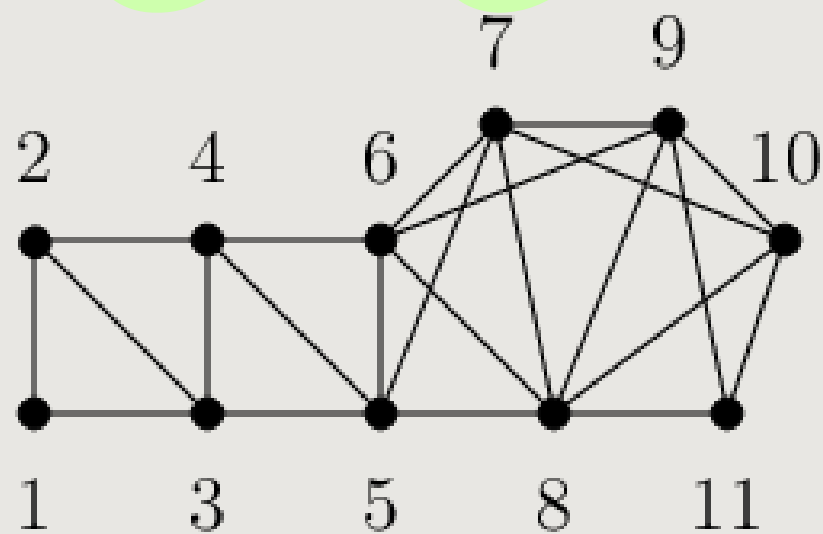
Definicija:

Graf $G=(V,E)$ transformisati u intervalni graf $G'=(V,E')$,
uz minimalno dodavanje grana,
tako da postane intervalni.

Mera problema:

Kardinalnost skupa grana E' u intervalnom grafu G'

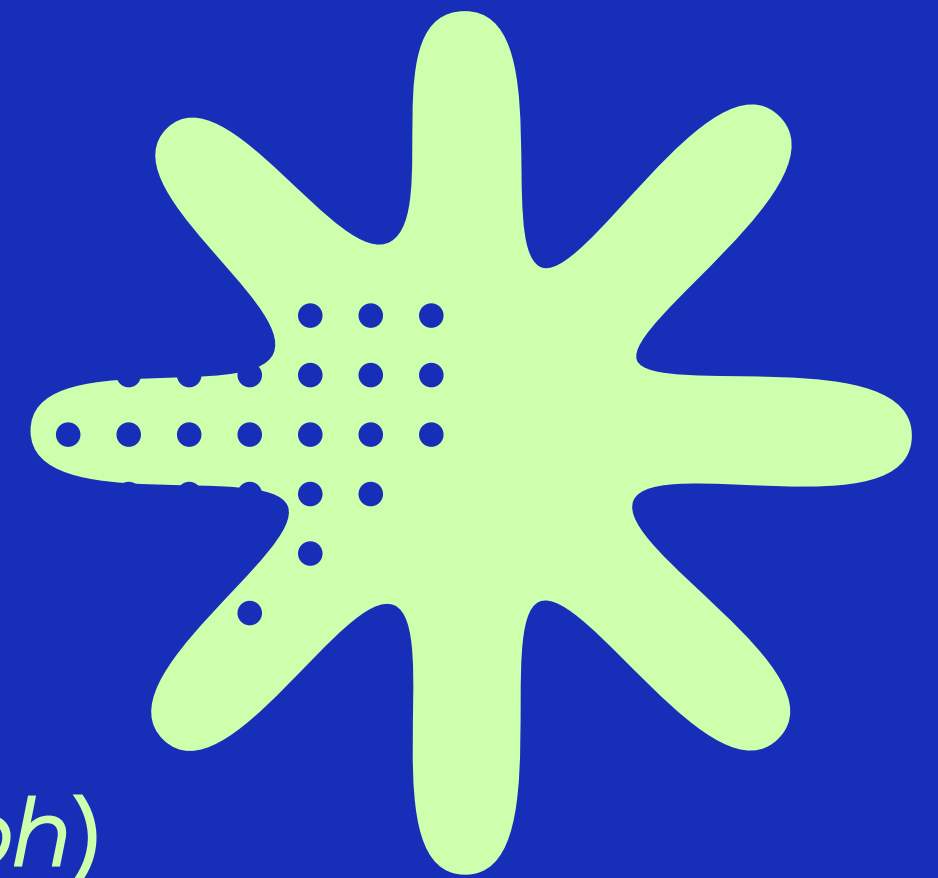
Ključne karakteristike intervalnog grafa



Preklapanje intervala:
Dva čvora su povezana
granom ako i samo ako se
njihovi intervali preklapaju



Ispitivanje intervalnosti



Graf je intervalni ako i samo ako je **tetivni** (*chordal graph*)
i ako je njegov komplement
graf uporedivosti (*comparability graph*).

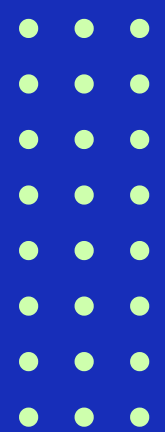
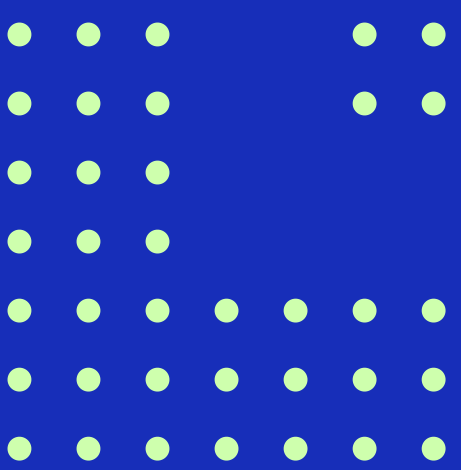
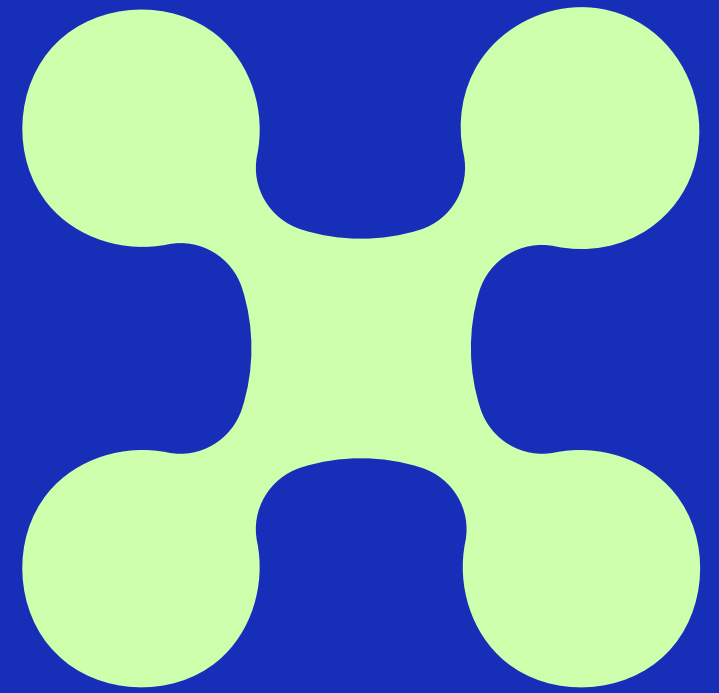
Za graf važi da je graf uporedivosti ako i samo ako je tetivni.

Alati: *SageMath* biblioteka za verifikaciju



Algoritmi za rešavanje problema

- Algoritam grube sile
- Simulirano kaljenje
- Simulirano kaljenje uz dodatak hlađenja
- Variable Neighborhood Search - VNS
- Genetski algoritmi



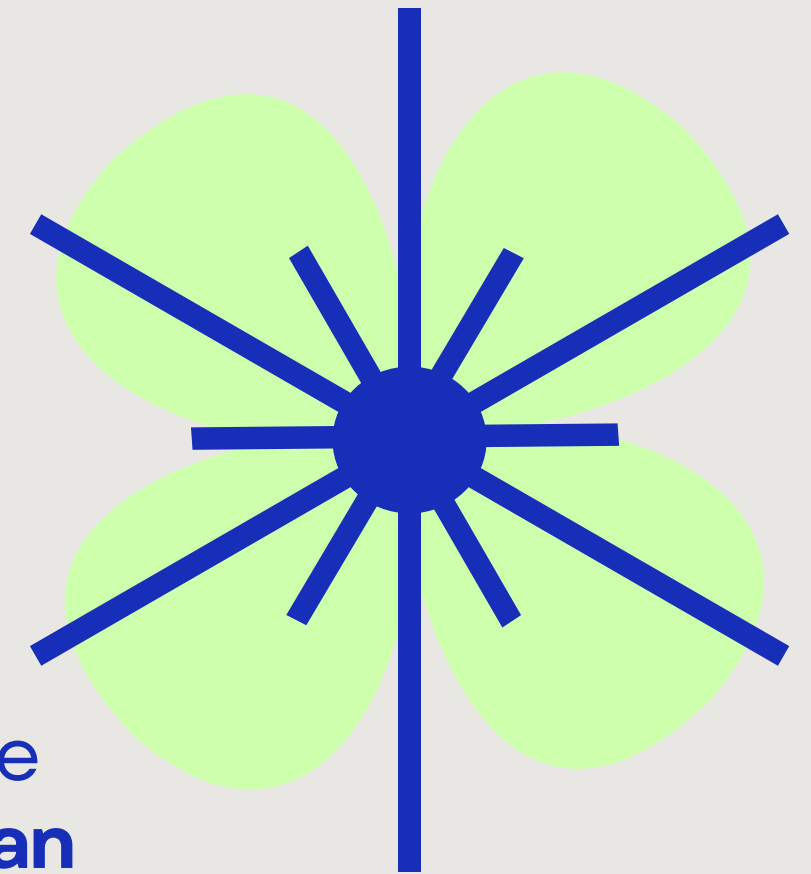
Brute Force algoritam

Ključni Koraci:

- Generisanje svih mogućih grana koje bi mogle biti dodate grafu.
- Korišćenje kombinatornih metoda za pronalaženje minimalnog skupa grana koje je potrebno dodati.
- Provera svih kombinacija nedostajućih grana i testiranje da li dodavanje svake od njih čini graf intervalnim.

Osobine:

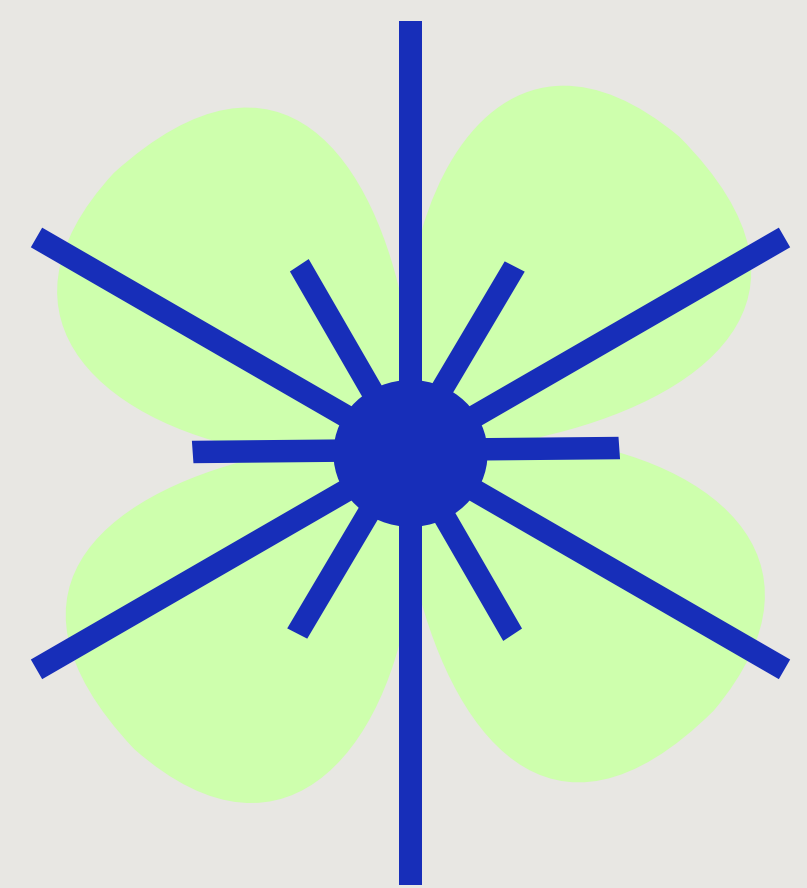
- Pruža tačno rešenje, ali može biti **veoma računski zahtevan** zbog eksponencijalnog broja kombinacija.
- Neefikasan za veće grafove sa velikim brojem nedostajućih grana.
- Testiran je samo na manjim grafovima.



Simulirano kaljenje

Glavna ideja:

- **Istraživanje i Eksploatacija:** Na početku, algoritam intenzivnije istražuje, prihvatajući čak i lošija rešenja, dok se kasnije fokusira na poboljšanje postojećih rešenja kako temperatura opada.
- **Temperatura:** U početnim iteracijama, temperatura je visoka, pa je veća verovatnoća da će lošija rešenja biti prihvaćena. Kako temperatura opada, algoritam postaje selektivniji i usmerava se ka finijem podešavanju najboljeg rešenja.
- **Lokalne i Globalne Pretrage:** Algoritam koristi lokalnu pretragu za poboljšanje trenutnog rešenja, dok globalna pretraga omogućava istraživanje novih rešenja, posebno u ranim fazama, čime se izbegava zaglavljivanje u lokalnim minimumima.
- **Manje izmene rešenja:** Nasumično dodaje ili uklanja granu iz trenutnog rešenja na osnovu adaptivne verovatnoće koja opada s vremenom, uzimajući u obzir sve moguće grane grafa, postojeće grane i trenutno rešenje.





Tehnike hlađenja

1. Multiplikativno Hlađenje:

- a. **Linearno Multiplikativno Hlađenje:** Temperatura se smanjuje linearno u odnosu na trenutnu iteraciju.
- b. **Logaritamsko Multiplikativno Hlađenje:** Temperatura opada s logaritamskom brzinom, što može omogućiti sporije smanjenje temperature u kasnijim iteracijama.

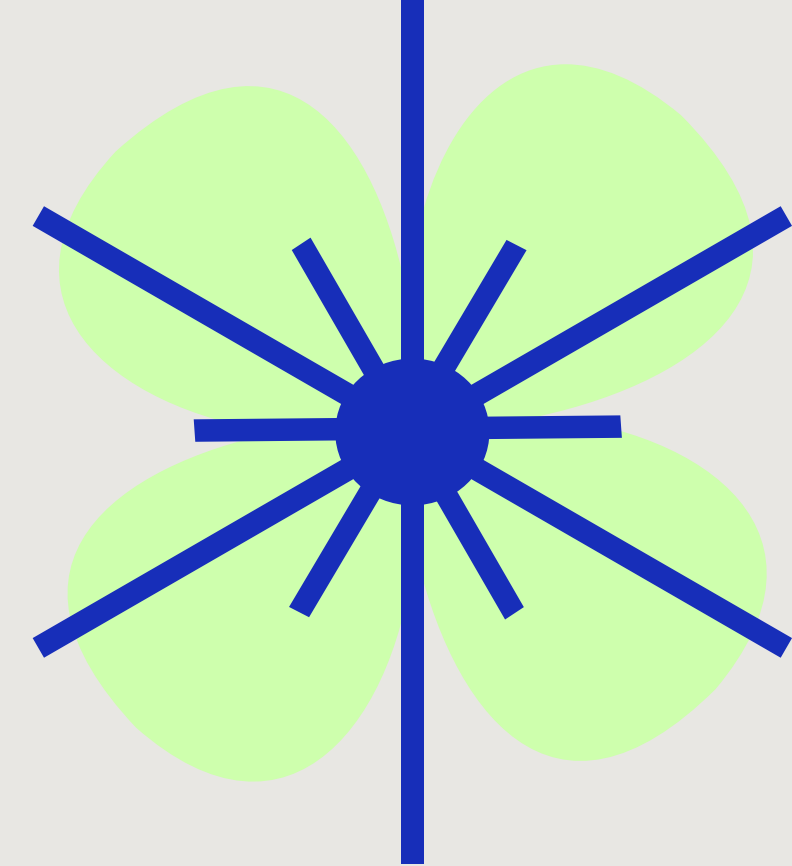


2. Aditivno Hlađenje:

- a. **Linearno Aditivno Hlađenje:** Temperatura opada linearno prema koraku do minimalne temperature.
- b. **Eksponencijalno Aditivno Hlađenje:** Ova metoda smanjuje temperaturu eksponencijalno, pružajući brže smanjenje u ranim fazama.



Variable Neighborhood Search VNS



Kombinacija lokalne pretrage i promene veličine područja pretrage.

Ključni koraci:

Shaking:

- Generisanje novih rešenja kroz nasumične promene (dodavanje/uklanjanje grana) na osnovu unapred definisanog broja promena **k**.
- Omogućava istraživanje različitih delova prostora rešenja.

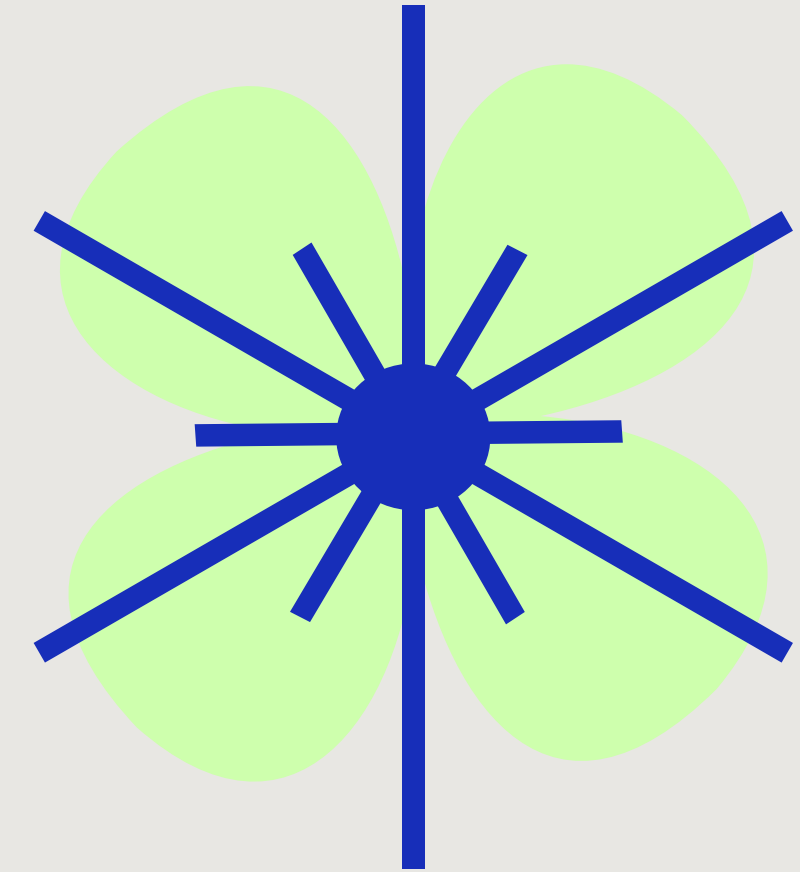
Lokalna pretraga:

- Primena algoritma najboljeg poboljšanja. (best improvement unbiased)
- Pokušaj dodavanja/uklanjanja grana i usvajanje boljih rešenja.
- Fino podešavanje rešenja unutar trenutne oblasti.

Iteracija:

- Ponavljanje procesa u okviru vremenskog ograničenja, istražujući veličine oblasti (k_{min} do k_{max}).
- Prihvatanje novog rešenja na osnovu poboljšanja ili verovatnoće.

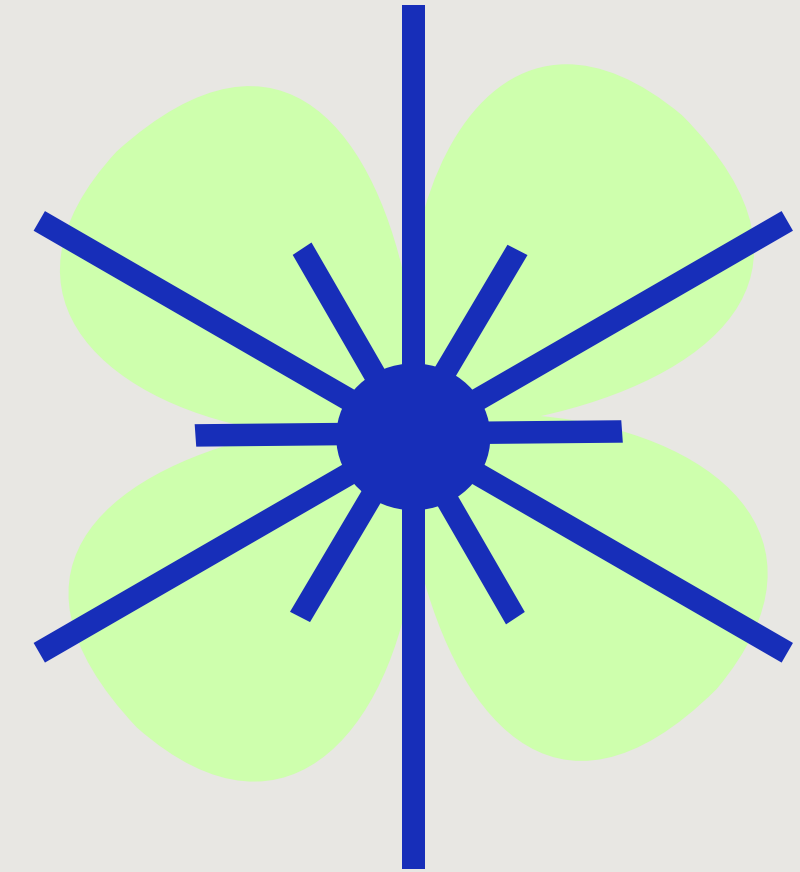
Genetski algoritmi



Detaljno objašnjenje jedinke:

- **Definicija:** Jedinka je lista ili skup grana koje se dodaju u grafu kako bi se formirao novi graf.
- **Različite kombinacije:** Svaka jedinka sadrži različit broj i kombinaciju grana koje se mogu dodati u originalni graf.
- **Cilj:** Pronaći jedinku koja dodaje tačno onoliko grana koliko je potrebno da graf postane intervalni graf, uz minimizaciju broja potrebnih grana.

Genetski algoritmi



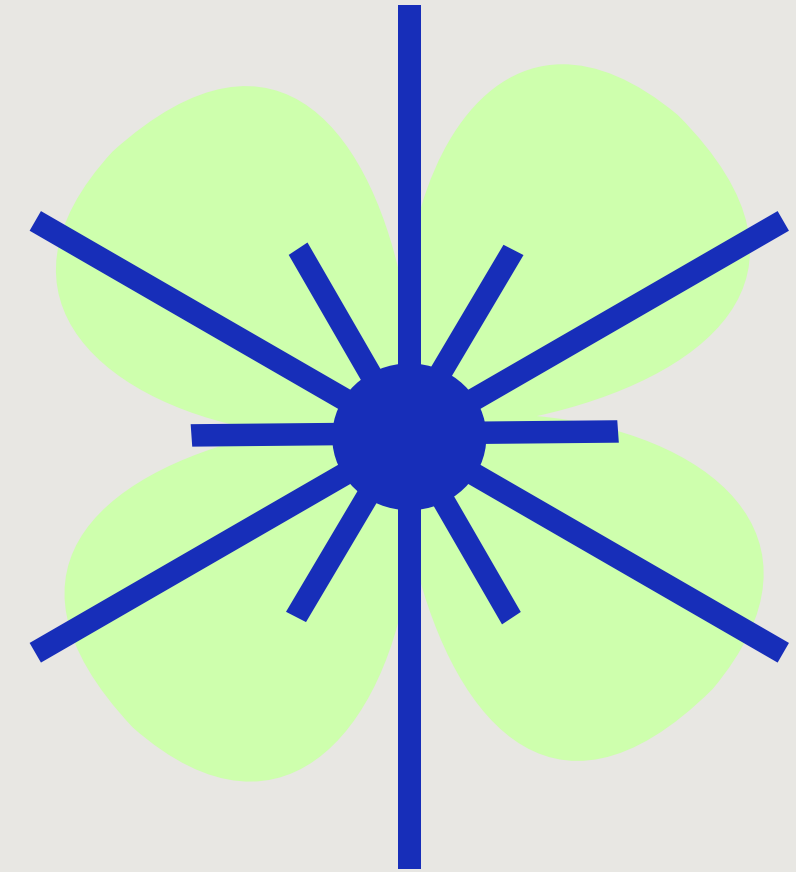
Fitness funkcija:

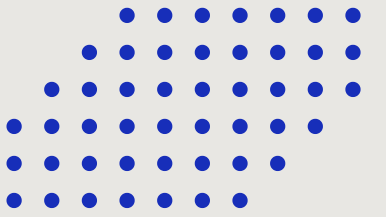
- **Merenje kvaliteta:** Kvalitet svake jedinke određen je fitness funkcijom koja meri kvalitet rešenja na osnovu toga da li dodate grane čine graf intervalnim.
- **Vrednost fitnessa:** Ako graf postane intervalni, fitness vrednost je *broj dodatih grana*, gde manje grana označava bolji rezultat.
- **Kazna za neuspeh:** Ako graf ne postane intervalni, rešenje se penalizuje beskonačnom vrednošću, čime se osigurava da se takvi rezultati ne koriste u sledećim generacijama.

Genetski algoritmi

Korišćene tehnike:

- **Selekcija:** Odabir najboljih jedinki za reprodukciju.
 - **Turnirska selekcija:** Izvodi se nasumičan odabir podskupa jedinki, koje se takmiče na osnovu fitness vrednosti kako bi se izabrali roditelji. Ovaj proces osigurava raznovrsnost.
 - **Ruletska selekcija:** Svaka jedinka dobija verovatnoću na osnovu fitness vrednosti, što favorizuje bolje jedinke pri odabiru roditelja.
- **Crossover (rekombinacija):** Kombinovanje gena različitih roditelja radi stvaranja novih rešenja.
- **Mutacija:** Uvođenje nasumičnih promena u jedinke za istraživanje novih mogućnosti.



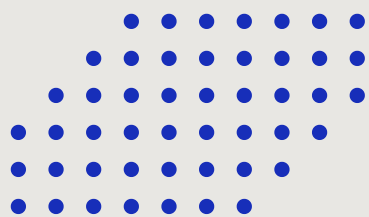
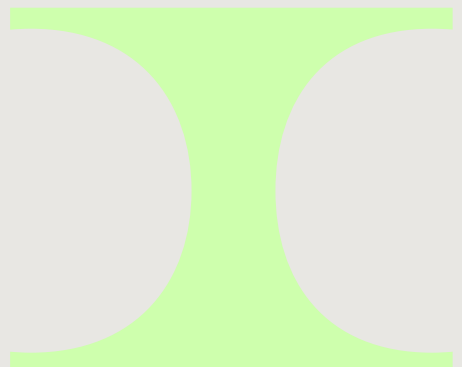


Analiza rezultata

Mali grafovi
(do 5 čvorova)

Nasumično generisani grafovi
(između 7 i 10 čvorova)

Veliki grafovi
(20, 30 i 50
čvorova)

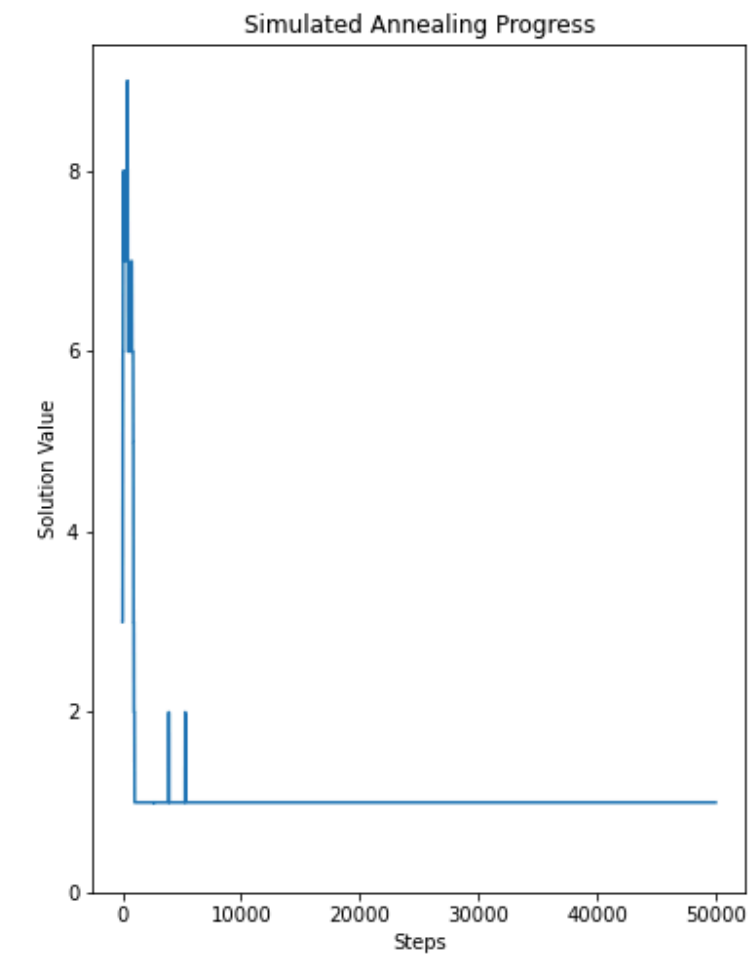
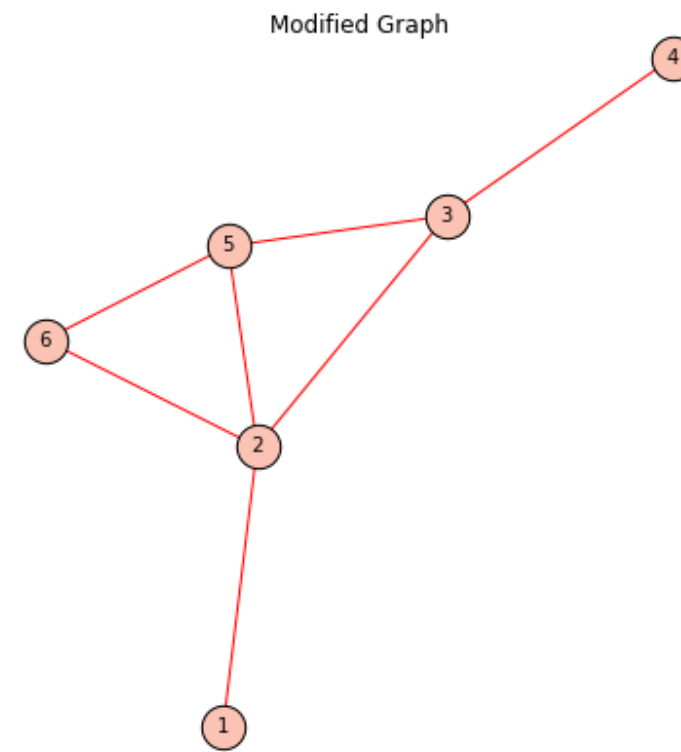
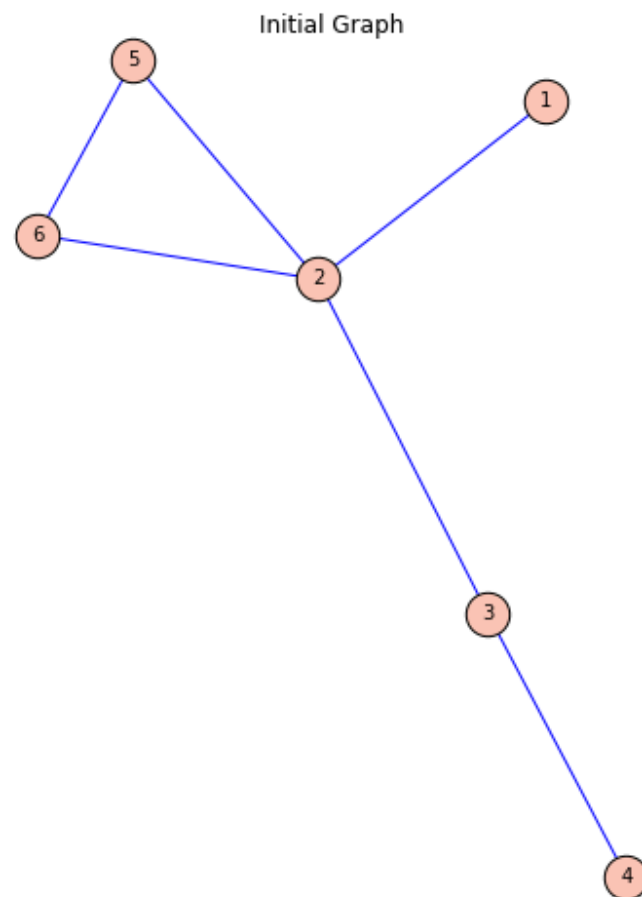


Mali grafovi

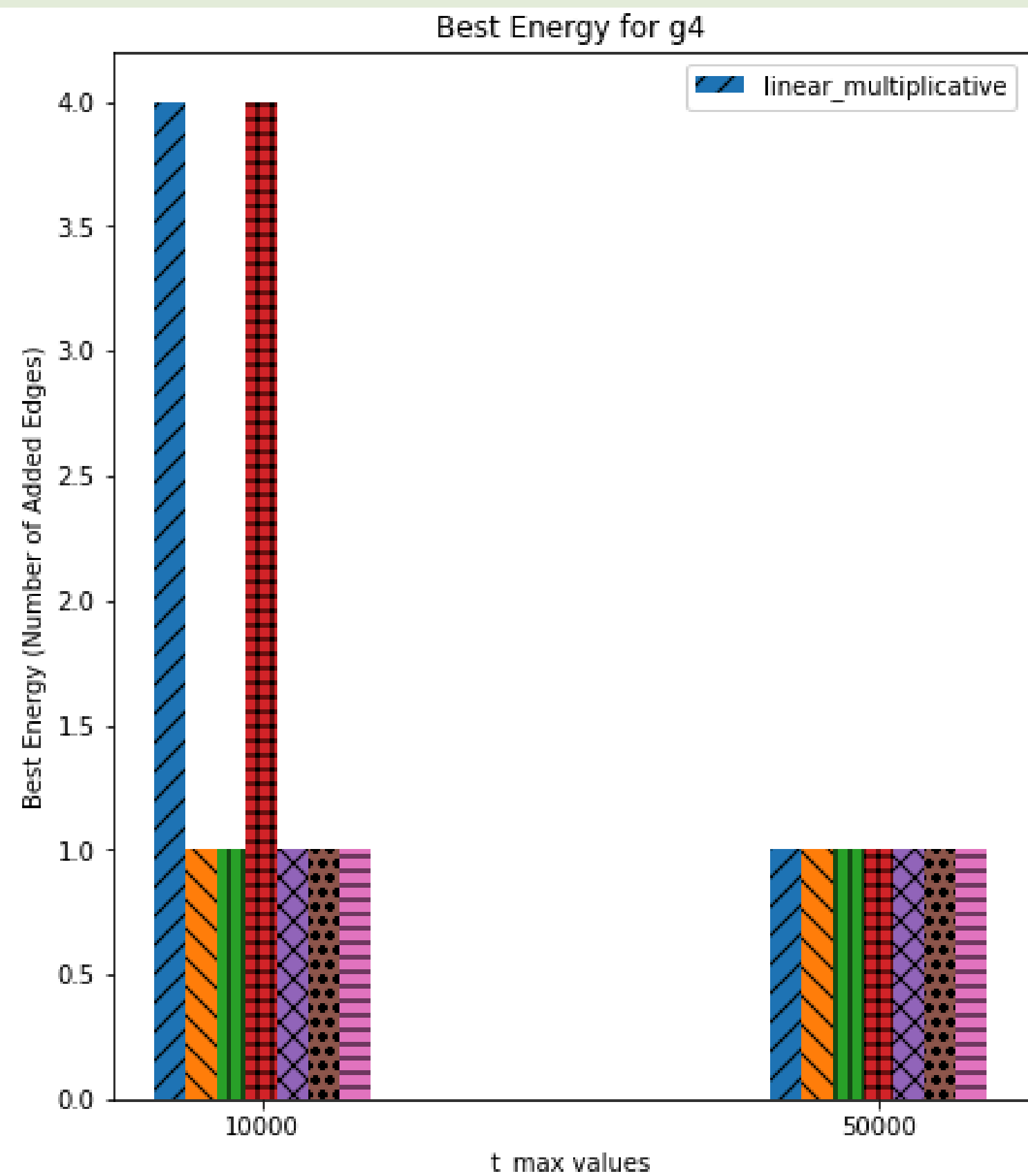
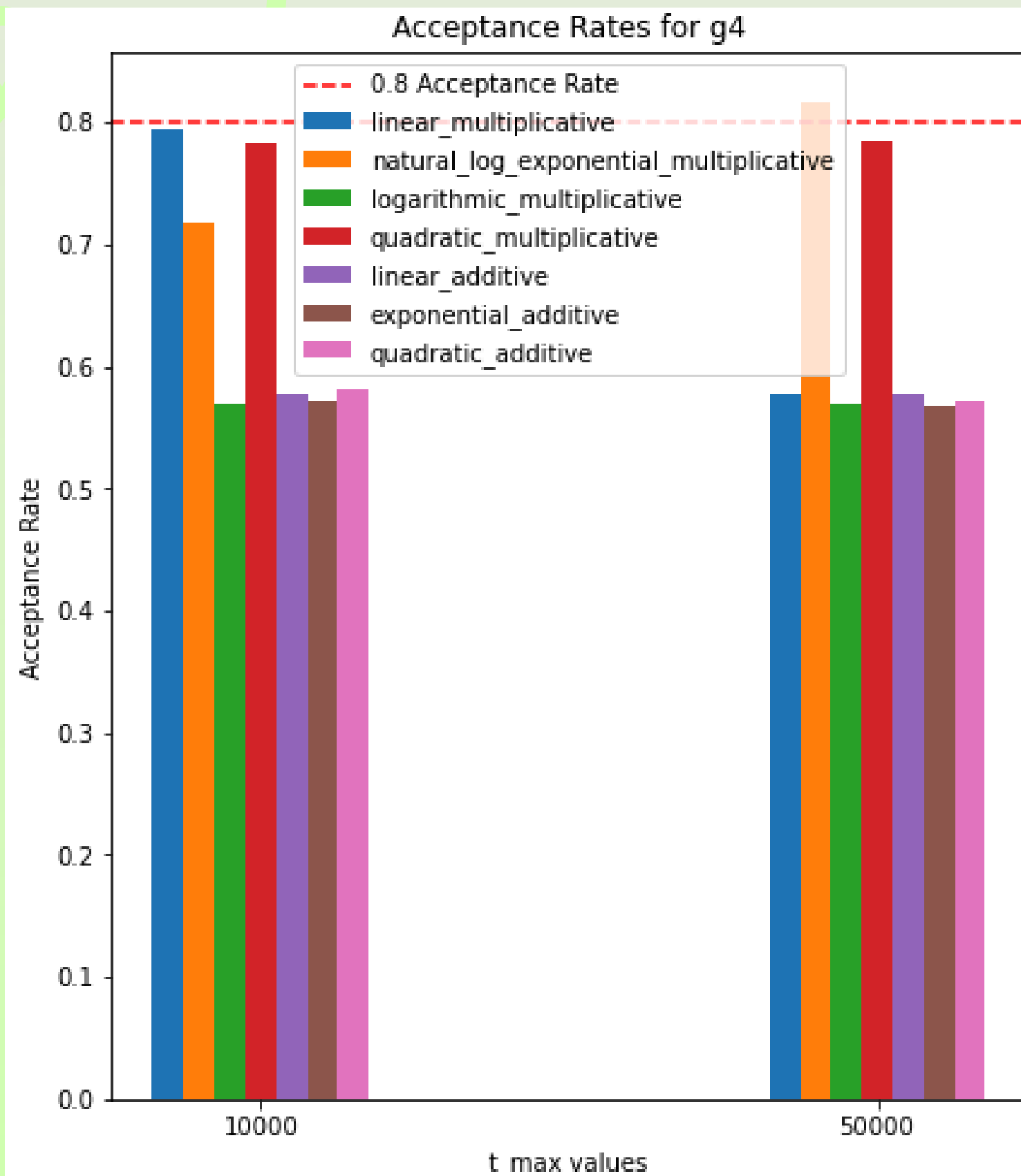
Naziv	V	E	BF	SA	SAC	VNS	GA
G1	4	4	1	1	1	1	1
G2	5	4	1	1	2	1	1
G3	5	5	1	1	1	1	1
G4	7	6	1	1	1	1	1

Naziv	V	E	BF	SA	SAC	VNS	GA
G1	4	4	0.0068s	9.3957s	8.63s	15s	0.1161s
G2	5	4	0.1064s	10.7620s	10.62s	15s	0.1262s
G3	5	5	0.1057s	10.5886s	9.72s	15s	0.1220s
G4	7	6	3.4807s	11.4943s	10.75s	15s	0.1309s

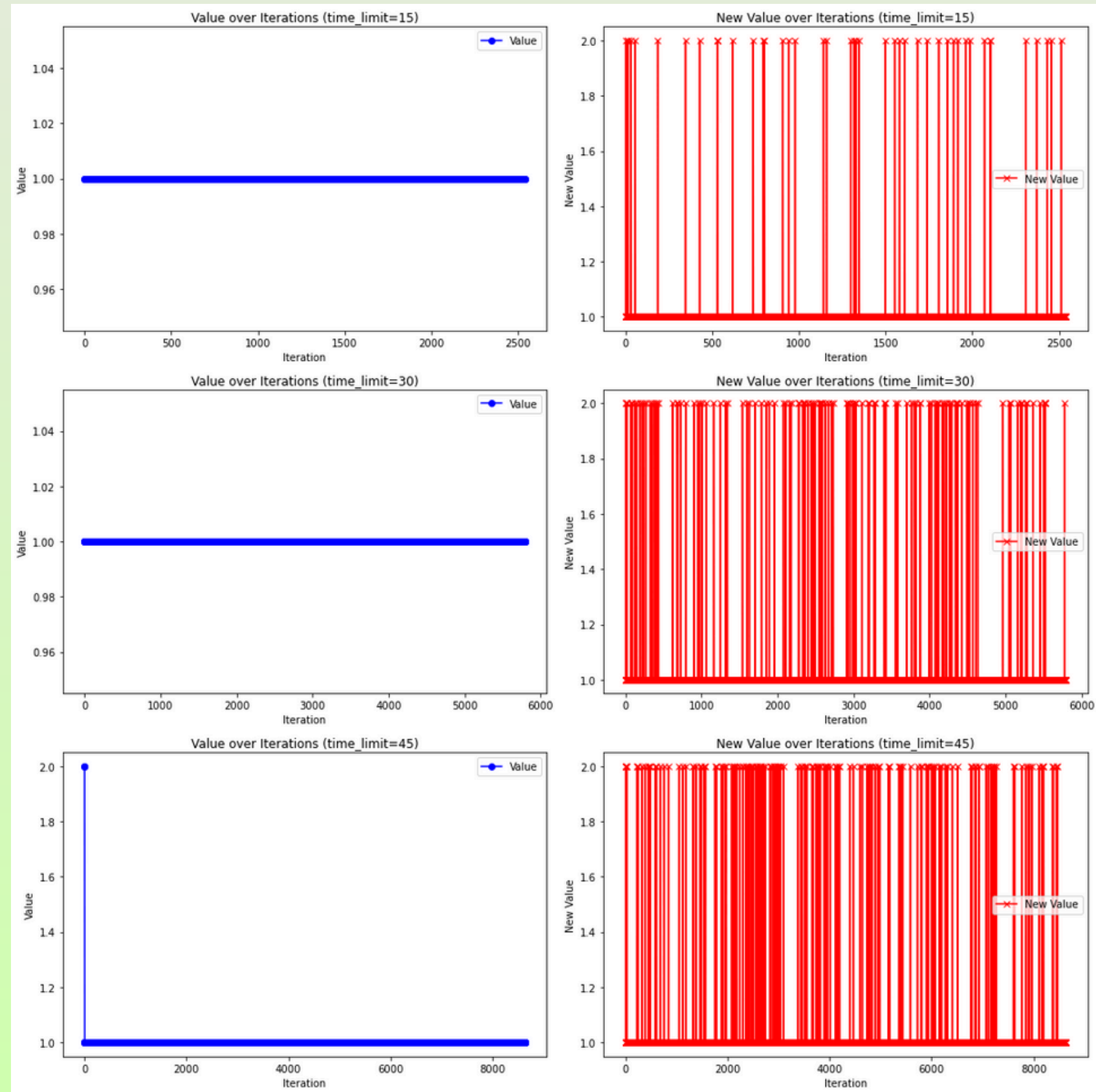
Mali grafovi



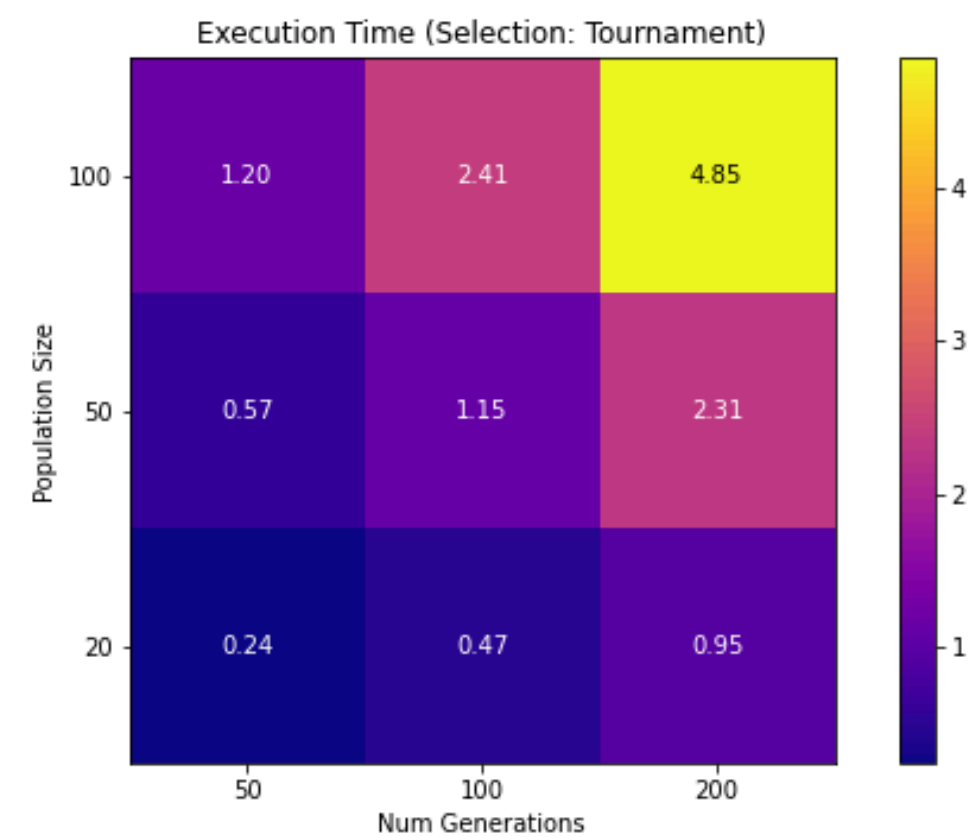
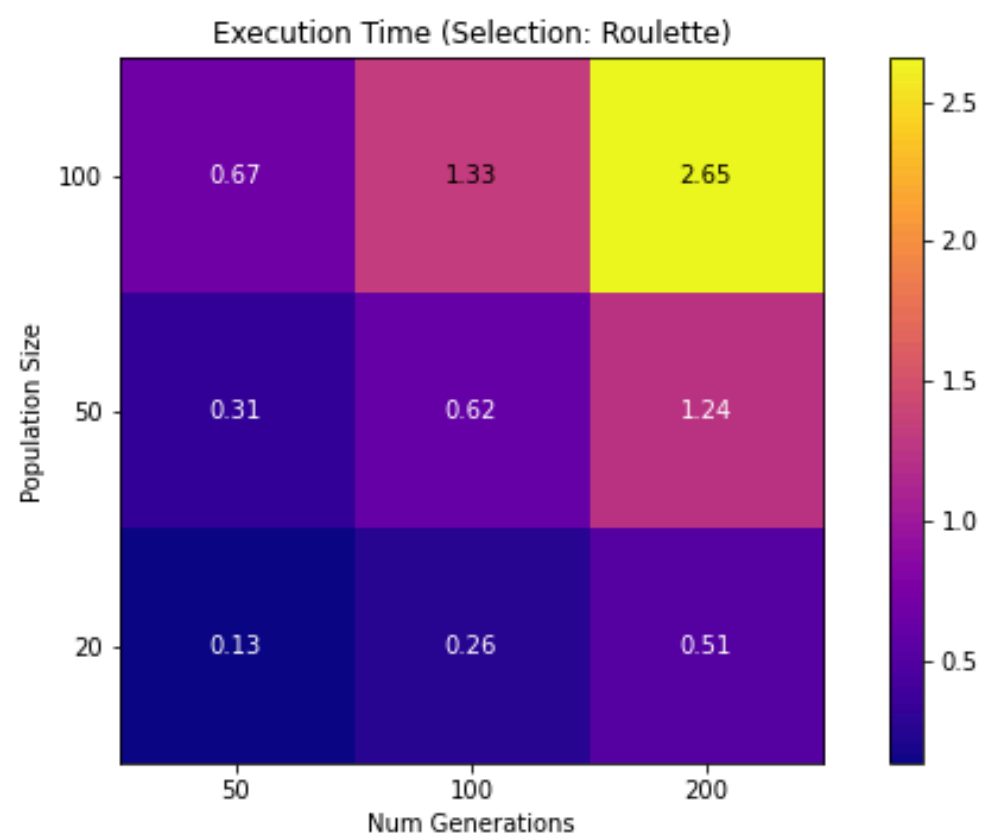
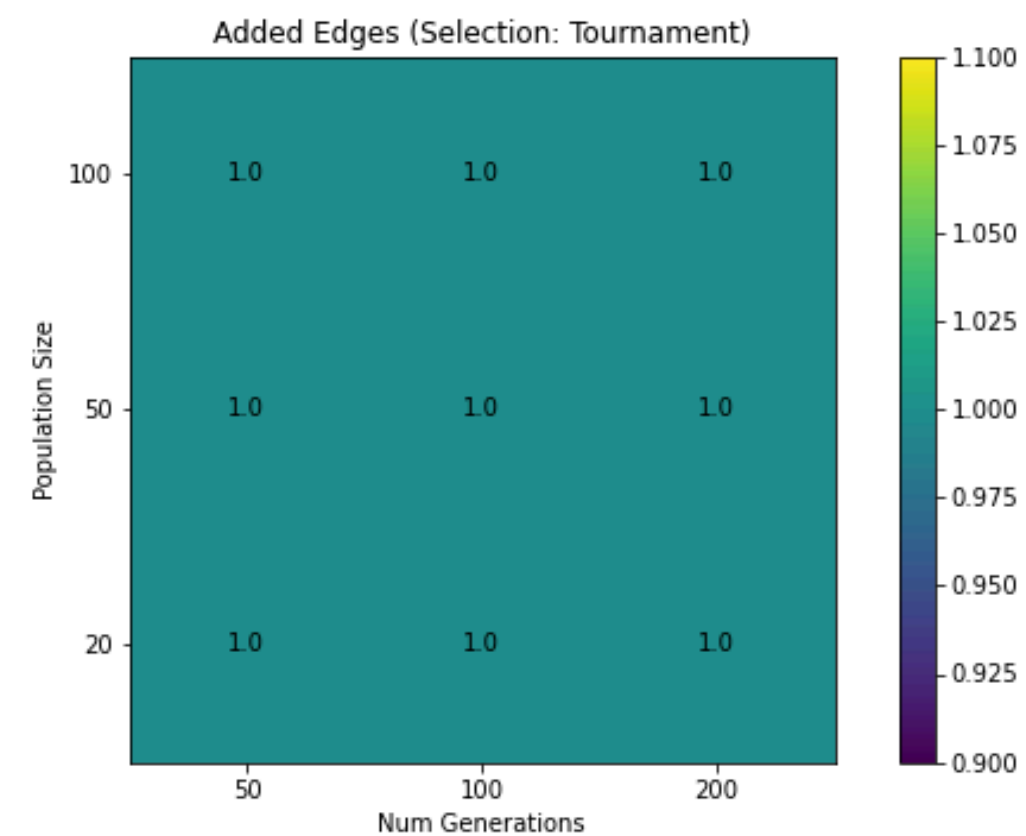
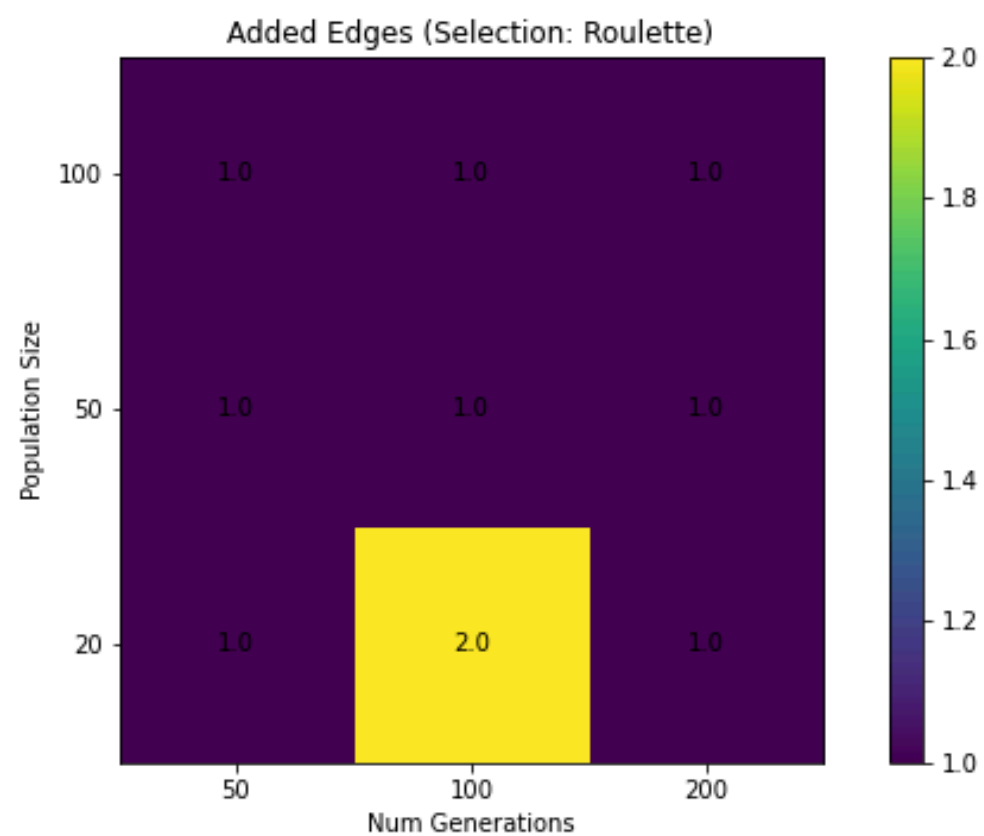
Mali grafovi



Mali grafovi



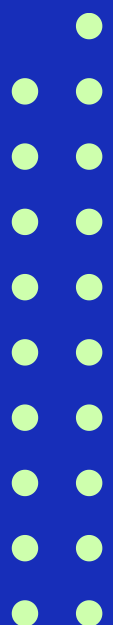
Mali grafovi



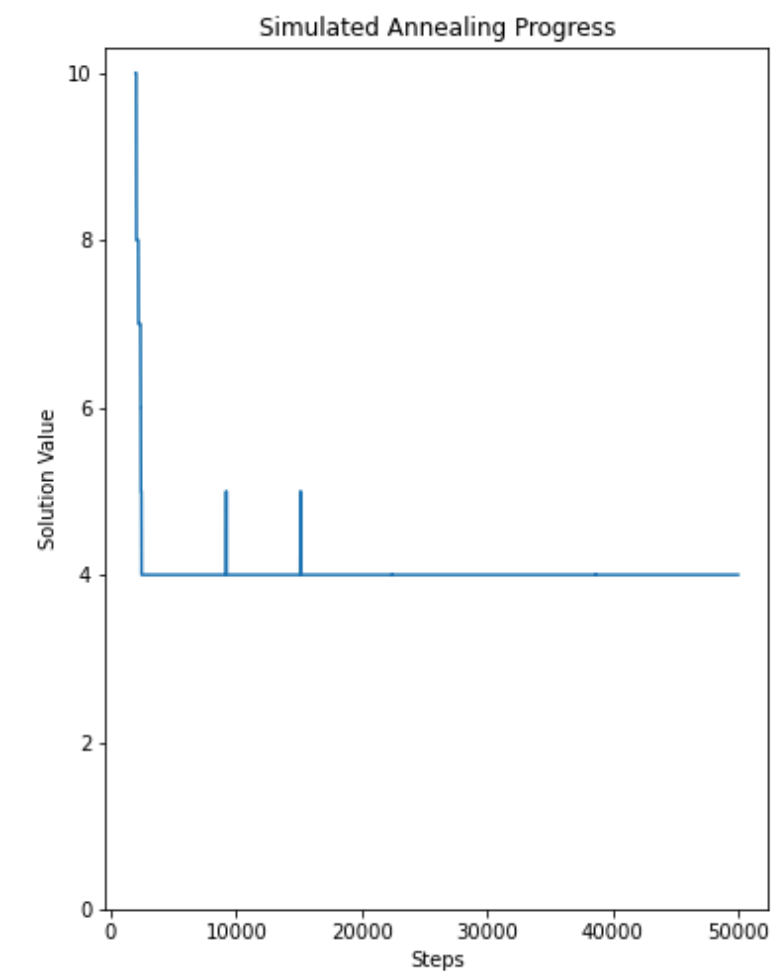
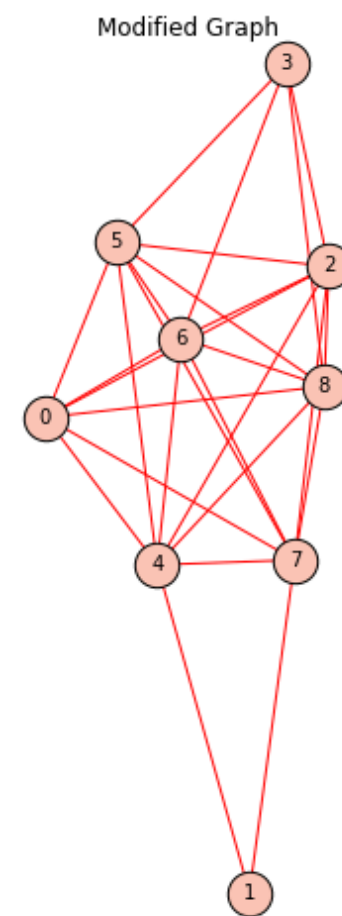
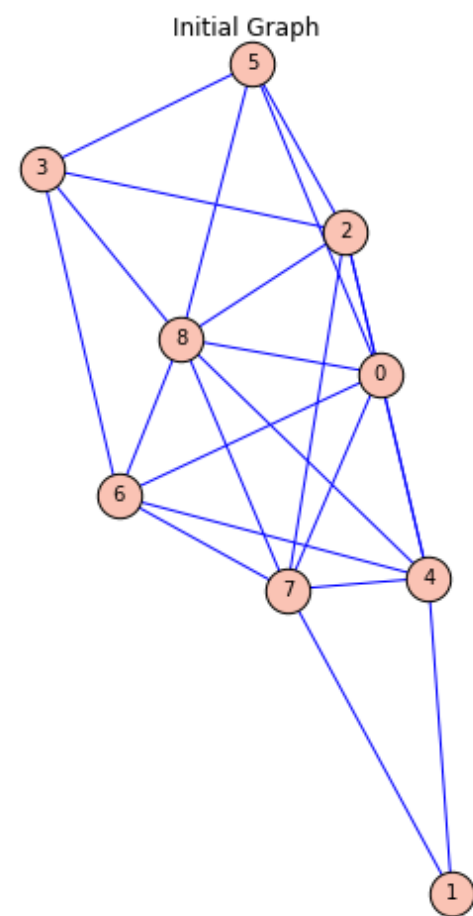
Nasumično generisani grafovi

V	E	SA	SAC	VNS	GA
6	7	3	4	3	3
7	12	2	3	2	2
7	15	1	1	1	1
8	15	2	4	2	2
8	16	2	4	2	2
8	19	/	/	/	/
9	21	5	8	5	5
9	21	6	7	5	5
9	21	6	9	6	6
9	23	4	6	3	3
10	20	-	17	-	7
10	26	-	12	7	7

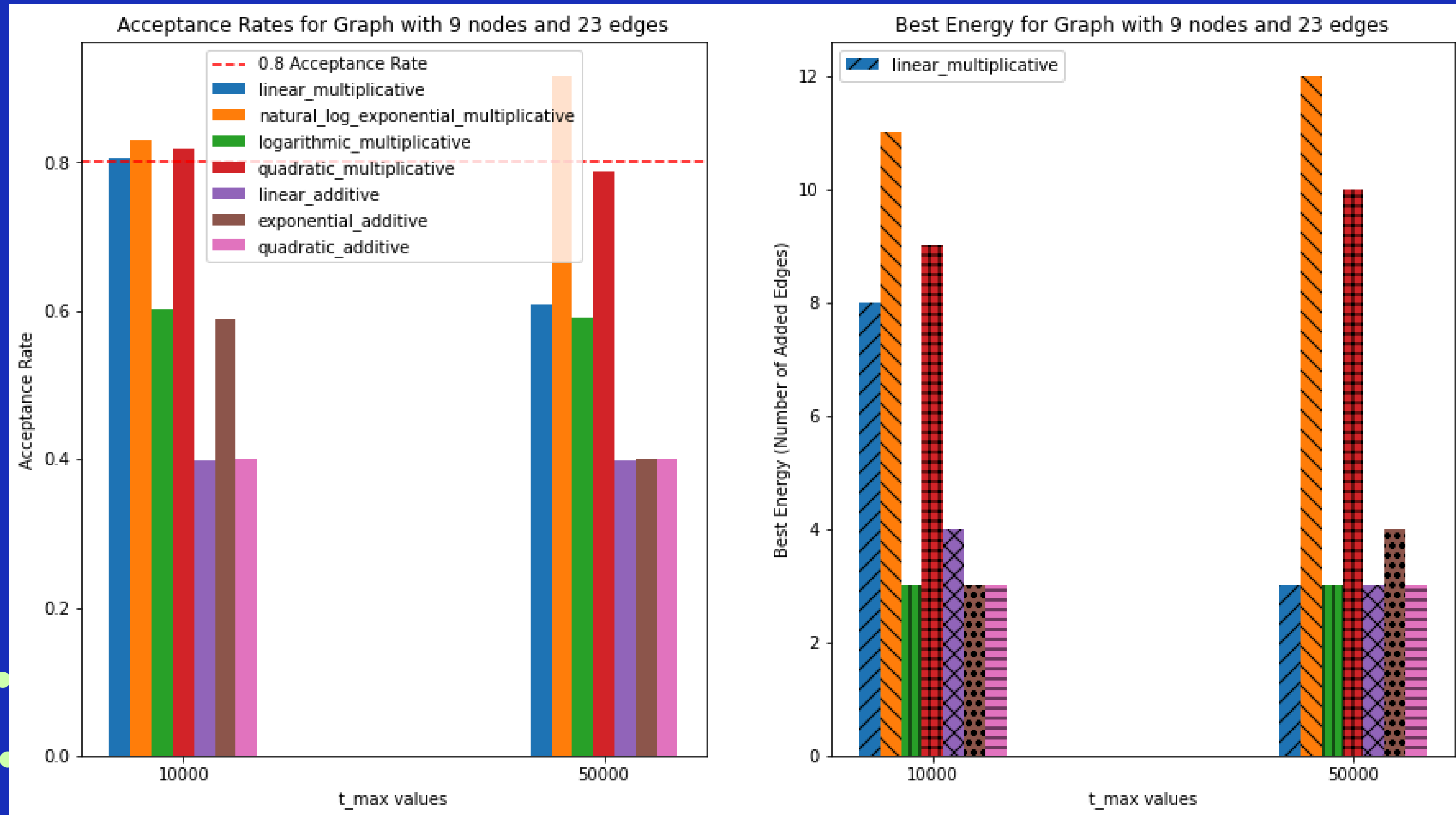
V	E	SA	SAC	VNS	GA
6	7	10.9590s	10.833s	15s	0.1375s
7	12	11.2987s	10.9148s	15s	0.1517s
7	15	14.1091s	12.2046s	15s	0.1895s
8	15	12.9164s	11.8133s	15s	0.3194s
8	16	13.6216s	12.3723s	15s	0.1488s
8	19	/	/	/	/
9	21	15.2925s	13.9151s	15s	0.3078s
9	21	14.7493s	13.7071s	15s	0.2043s
9	21	16.6224s	13.6s	45s	0.3398s
9	23	20.4286s	12.9857s	15s	0.3768s
10	20	15.2074s	14.006s	-	0.6427s
10	26	16.7679s	14.7433s	15s	0.3886s



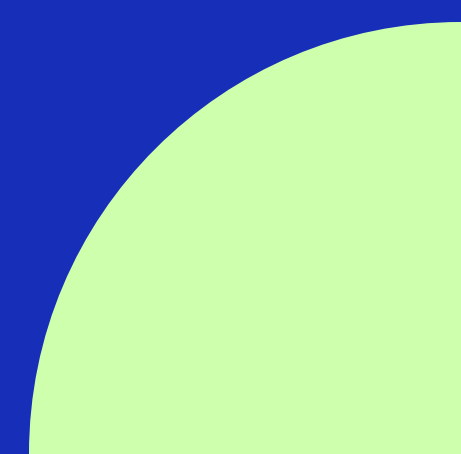
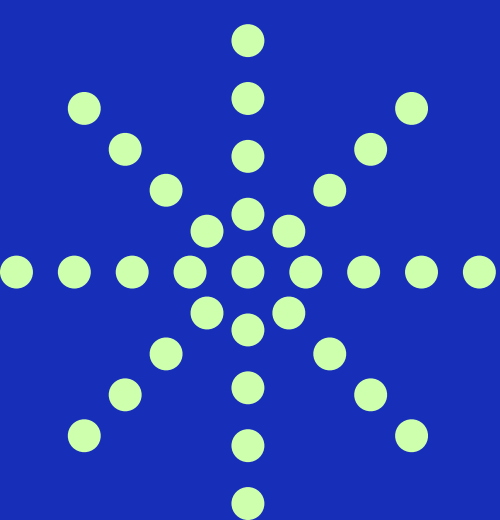
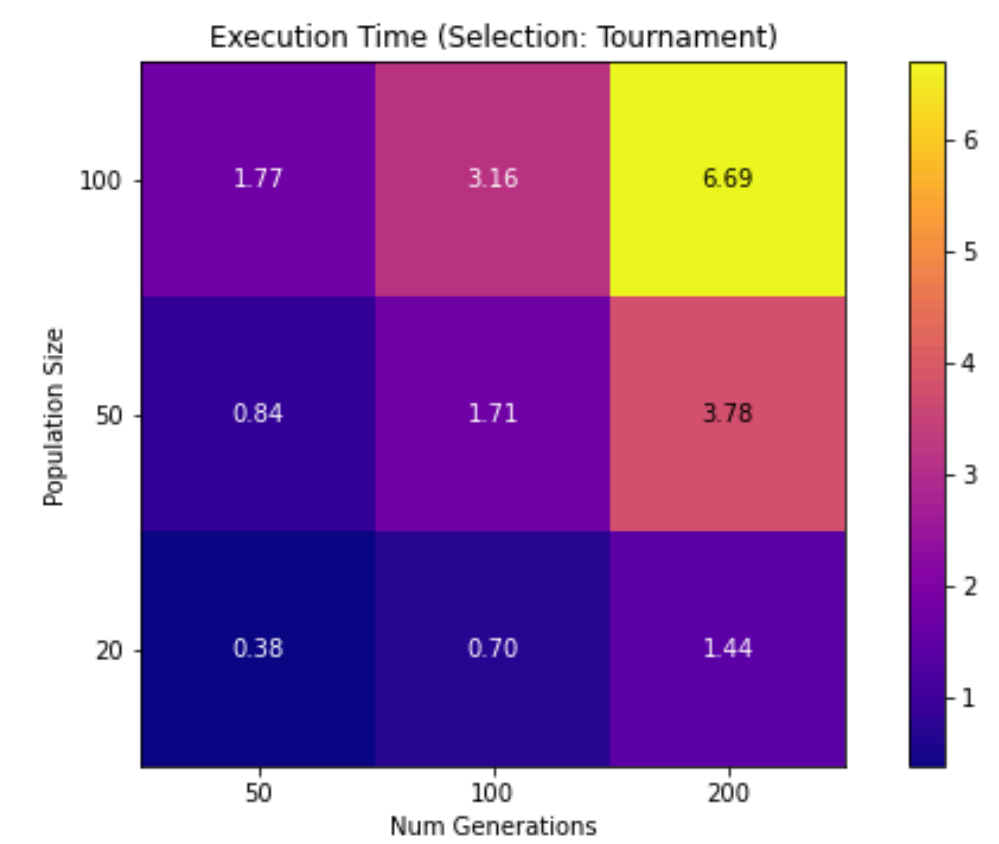
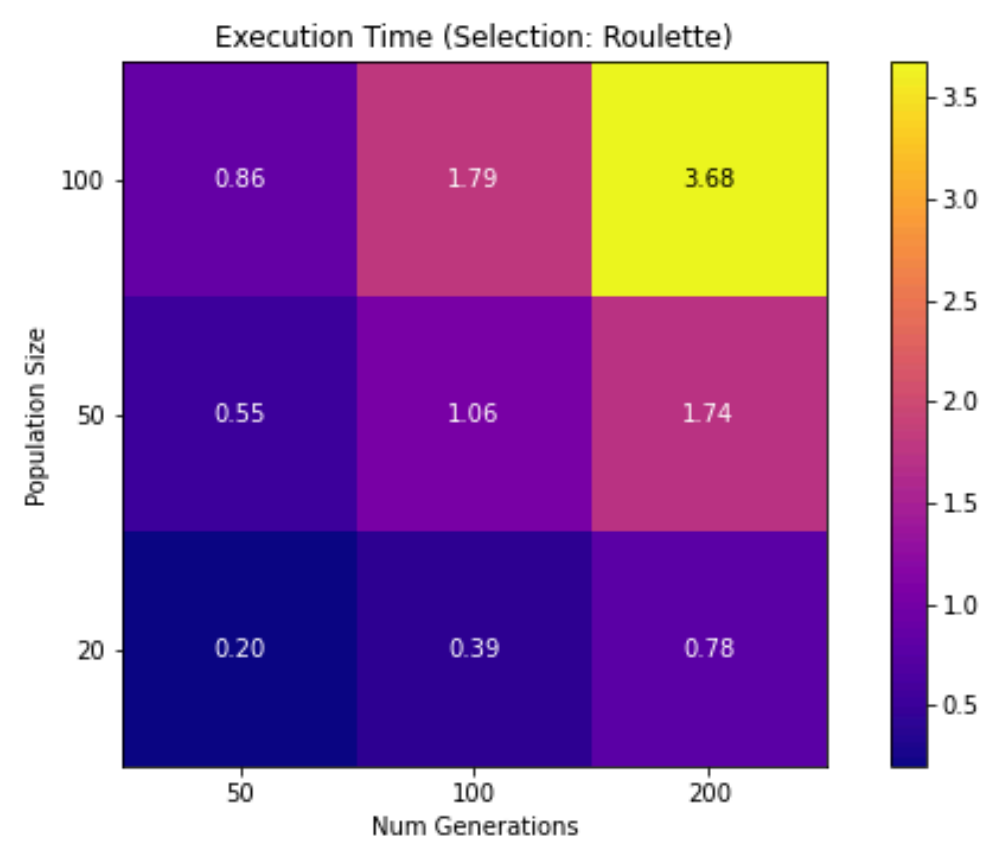
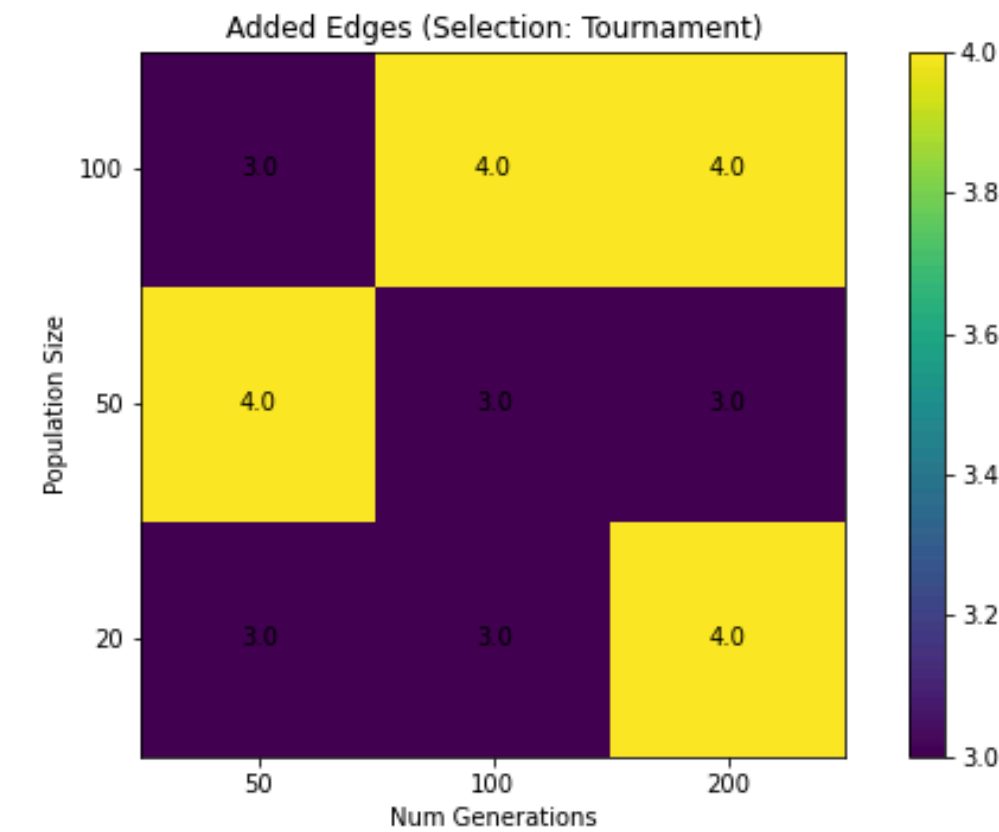
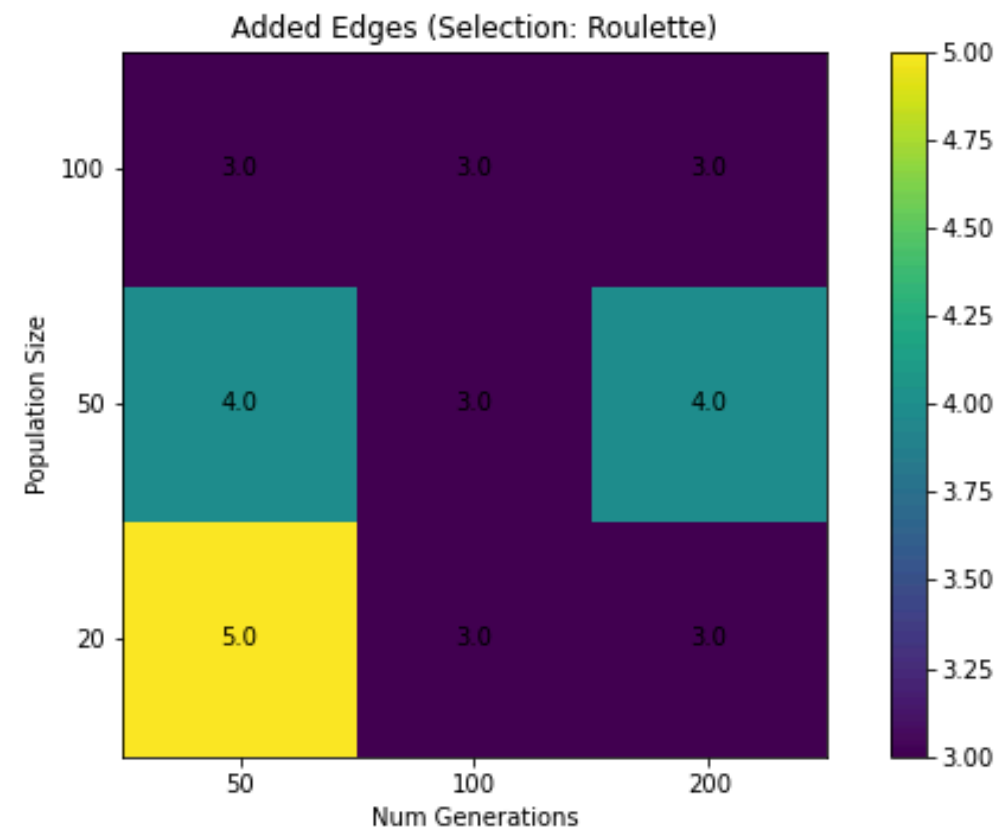
Nasumično generisani grafovi



Nasumično generisani grafovi



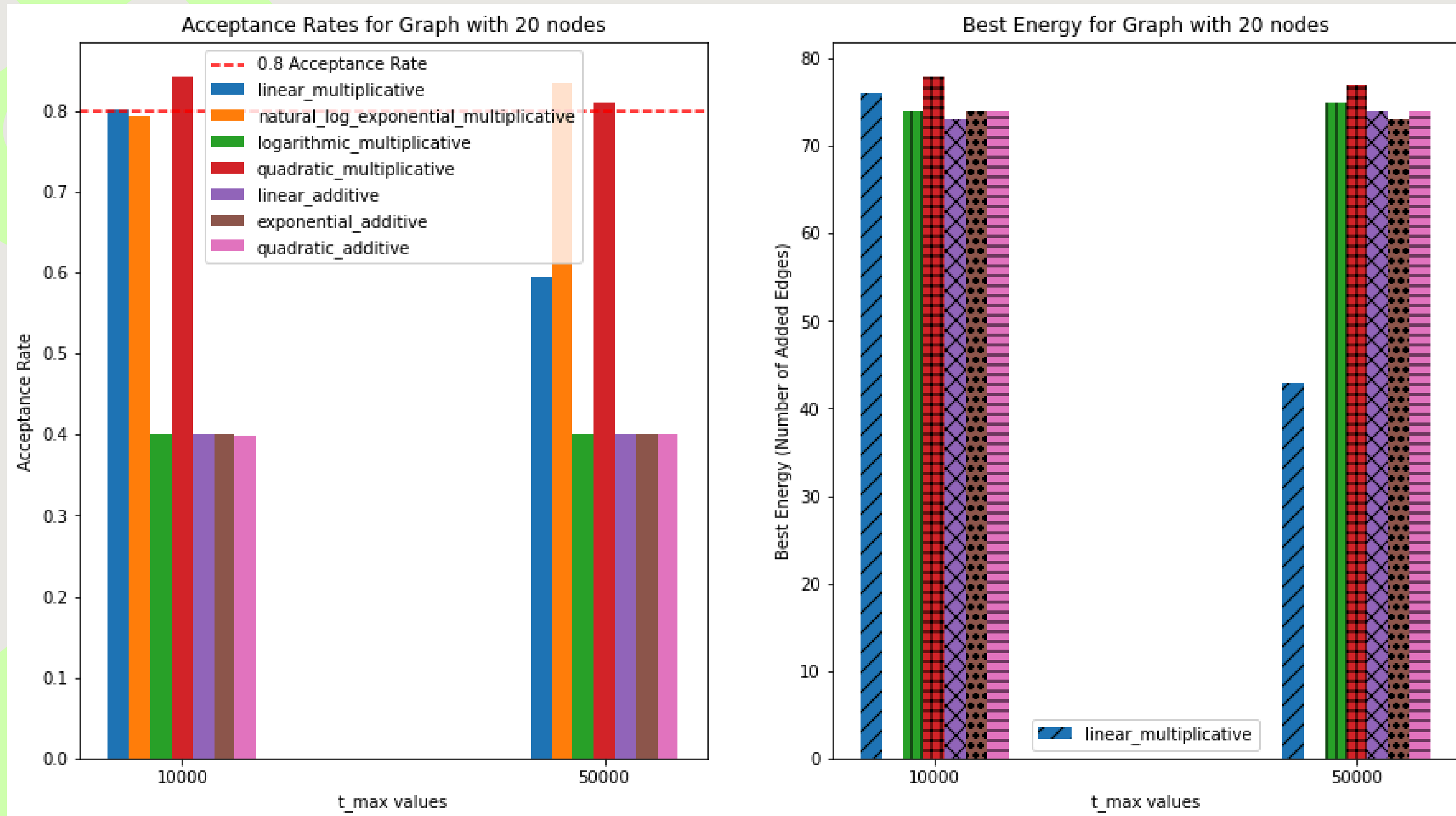
Nasumično generisani grafovi



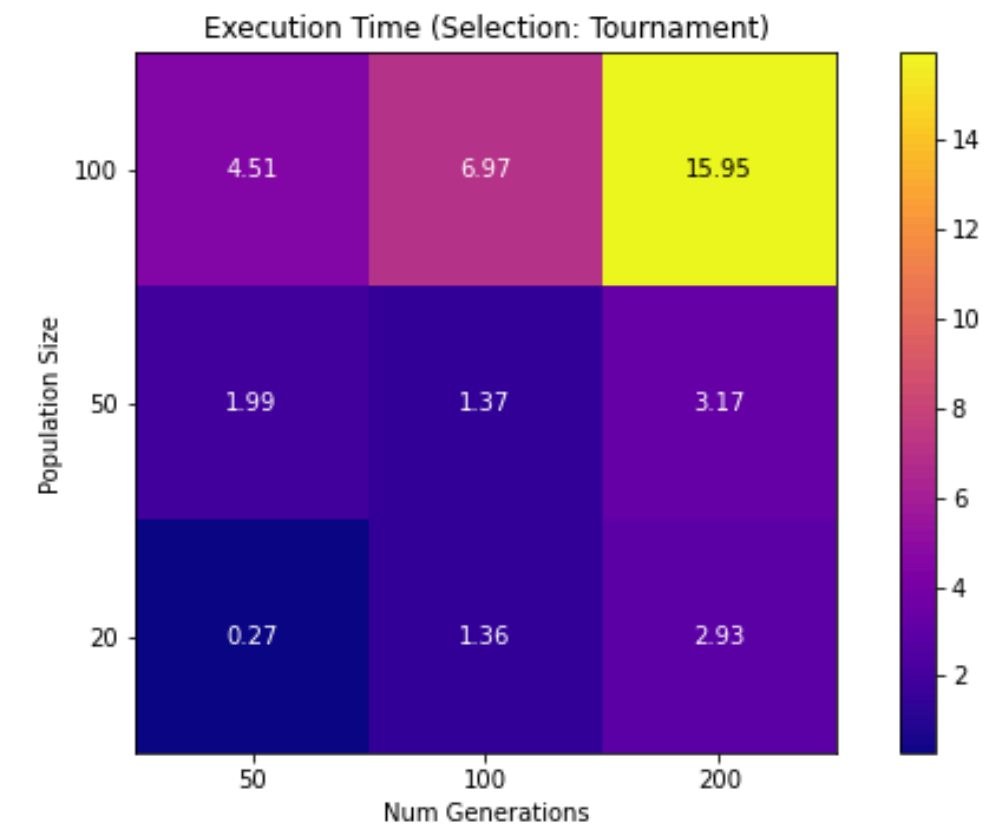
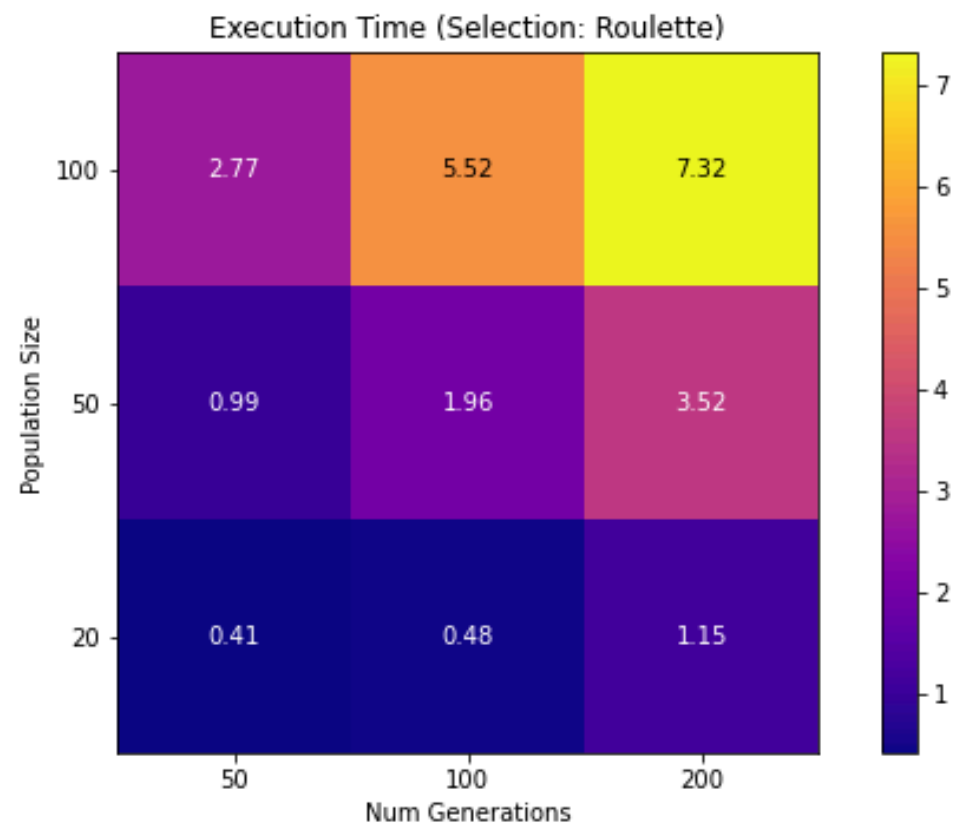
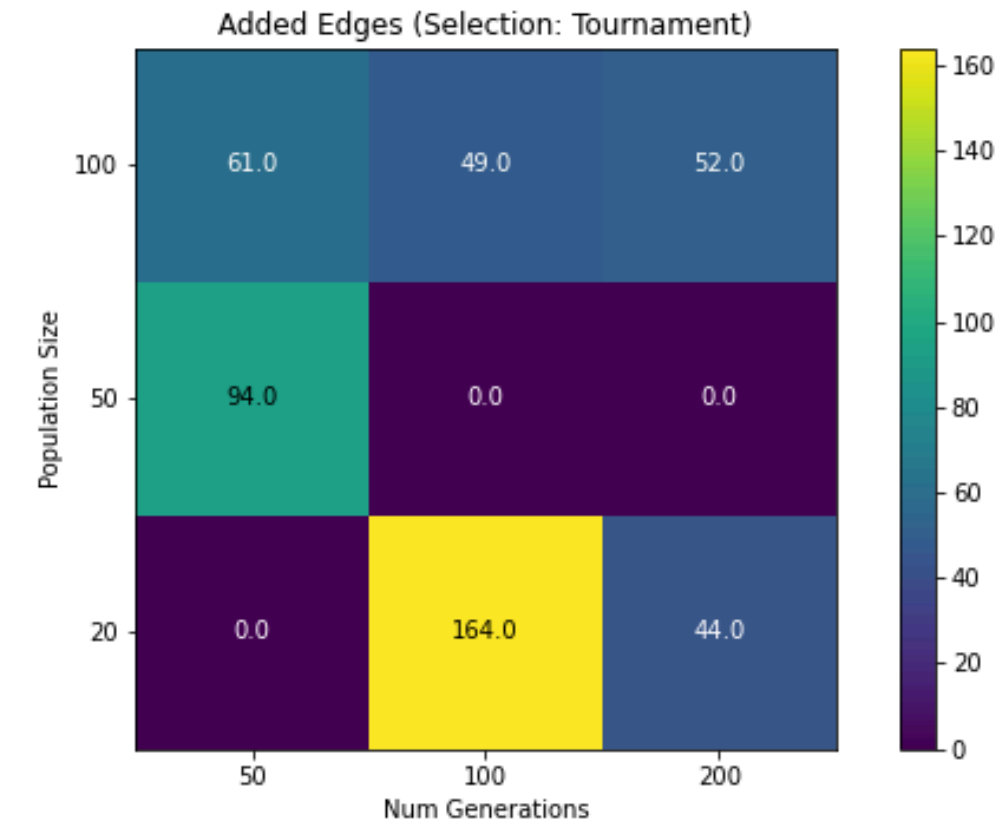
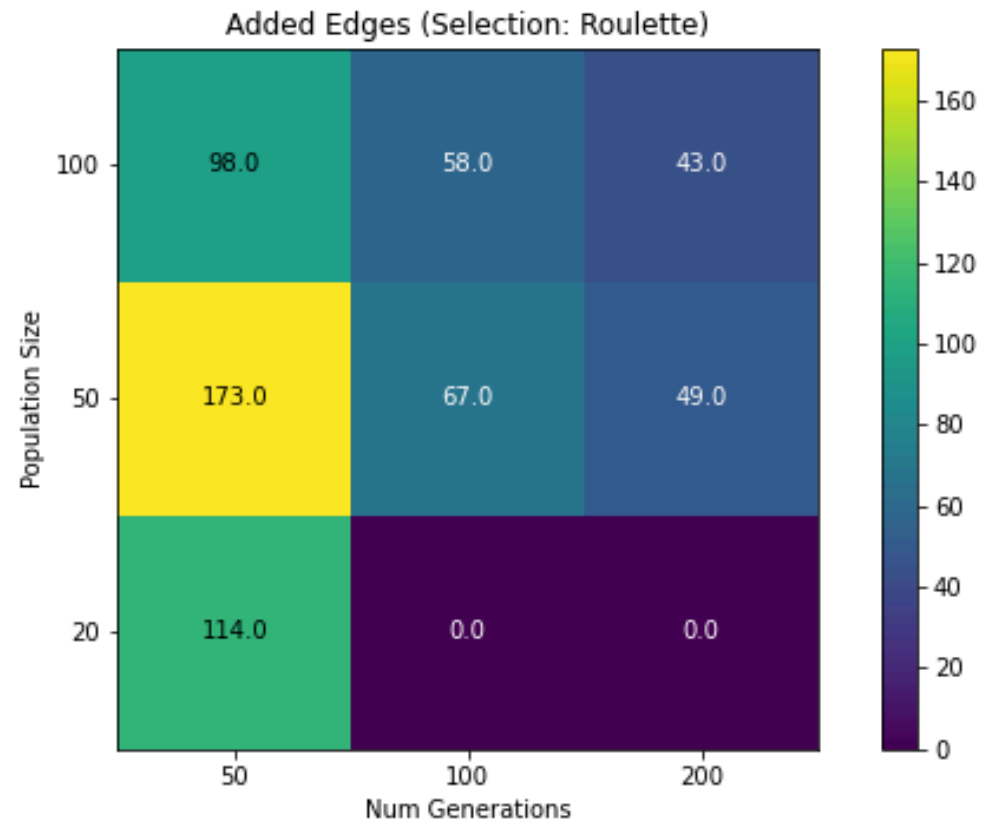
Veliki grafovi

Graph	Algorithm	Best Value	Execution Time
Graph with 20 nodes	Simulated Annealing	inf	40.284965
Graph with 20 nodes	Simulated Annealing - Cooling	43.0	54.108013
Graph with 20 nodes	VNS	inf	NaN
Graph with 20 nodes	Genetic Algorithm	43.0	7.323347
Graph with 30 nodes	Simulated Annealing	inf	73.846219
Graph with 30 nodes	Simulated Annealing - Cooling	138.0	109.957590
Graph with 30 nodes	VNS	inf	NaN
Graph with 30 nodes	Genetic Algorithm	157.0	7.235888
Graph with 50 nodes	Simulated Annealing	inf	215.980289
Graph with 50 nodes	Simulated Annealing - Cooling	403.0	299.023929
Graph with 50 nodes	VNS	inf	NaN
Graph with 50 nodes	Genetic Algorithm	878.0	36.100909

Veliki grafovi



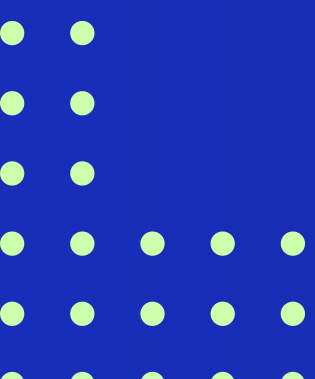
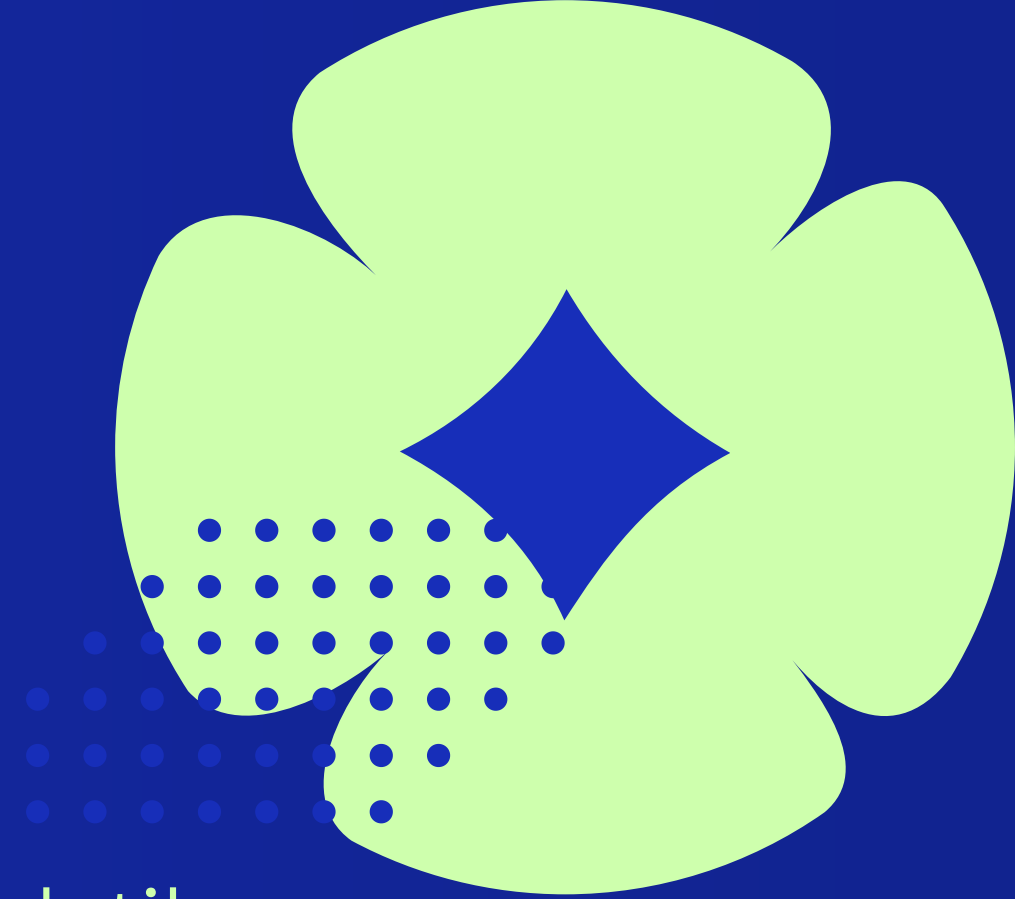
Veliki grafovi



Zaključak

Na osnovu analize podataka iz tabela, može se zaključiti da:

1. **Genetski algoritam** (GA) pruža najbolji balans između broja dodatih grana i vremena izvršavanja.
2. **VNS** se takođe ističe po minimalnom broju dodatih grana, ali sa dužim vremenima izvršavanja.
3. Algoritam grube sile nije efektivan na većim grafovima, što naglašava potrebu za naprednijim pristupima kao što su GA i VNS.
4. Tehnike hlađenja u algoritmu simuliranog kaljenja (SA, SAC) pokazuju varijacije u performansama, pri čemu viši t_{max} rezultira povoljnijim ishodima.



Hvala na
pažnji

