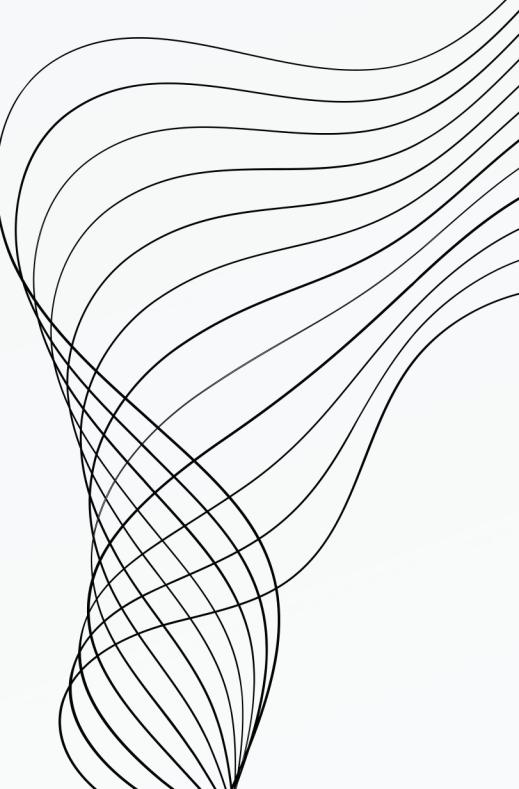




AI OPERATIONS RESEARCH

# MAXIMUM BETWEENNESS

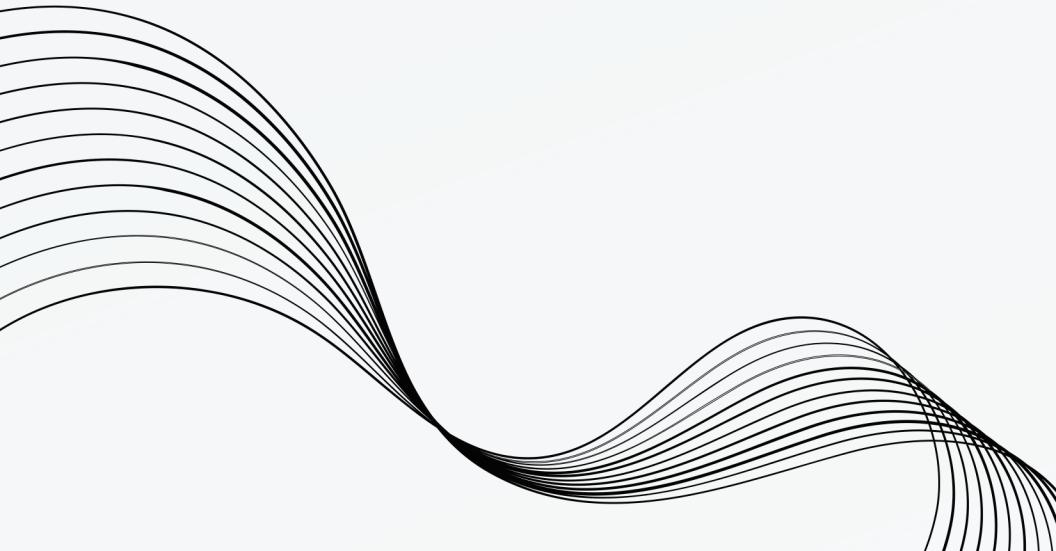


VUKAŠIN MARKOVIĆ

# OPIS PROBLEMA



- > Postoji skup elemenata **A** i kolekcija uređenih trojki  $(a, b, c)$ , pri čemu su **a**, **b** i **c** različiti elementi iz skupa **A**.
- > Cilj je pronaći jednoznačnu funkciju **f** koja preslikava elemente iz skupa **A** u opseg od 1 do veličine skupa **A**, tako da se maksimalizuje broj trojki u kojima se poštuje određeni redosled, tj. ili  $f(a) < f(b) < f(c)$ , ili  $f(c) < f(b) < f(a)$ .



# ALGORITMI

**01**

GRUBA SILA

**02**

POHLEPNI ALGORITMI

**03**

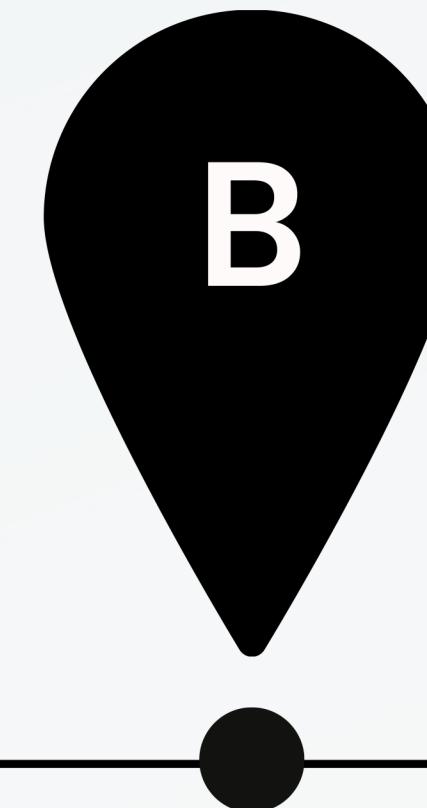
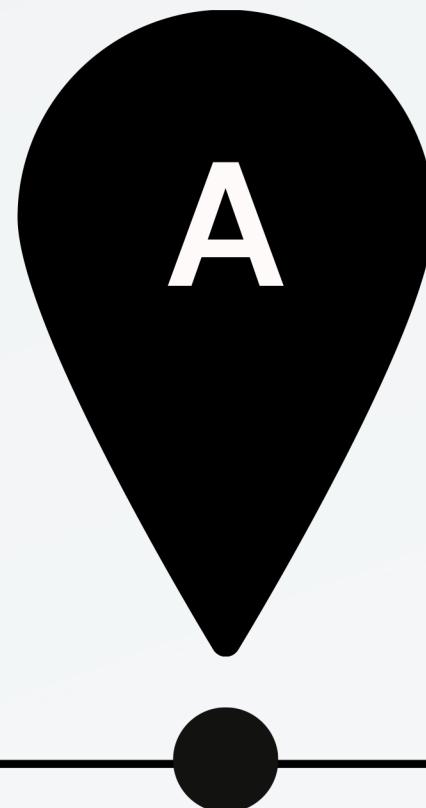
SIMULIRANO KALJENJE

**04**

GENETSKI ALGORITAM

# GRUBA SILA

Brute-force algoritam obezbeđuje uporednu analizu optimizacionih metoda. On efikasno obrađuje male instance problema i daje optimalna rešenja kako bi se mogla proveriti ispravnost optimizacionih pristupa.



## BRUTE FORCE

- daje optimalna rešenja
- Efikasno obradjuje male instance problema

## BRUTE FORCE BNB

- daje optimalna rešenja
- Efikasno obradjuje male instance problema
- odbacuje suvišne grane pretrage

```
ostatak = C_size - i
if (current_betweenness + ostatak <= max_betweenness):
    break
```

# POHLEPNI ALGORITMI

Kada je primarni cilj optimizacija vremenske složenosti rešavanja problema, koriste se pohlepni algoritmi. Oni omogućavaju brzo izvršavanje, ali ne garantuju optimalno rešenje.

U implementaciji algoritma generiše se okolina trenutnog rešenja i iz nje bira sledeće (generisanje okoline utiče na brzinu izvršavanja algoritma).



**LOCAL SEARCH  
FIRST IMPROVEMENT**

Sledeće rešenje je prvo rešenje iz okoline koje je bolje od trenutnog.



**LOCAL SEARCH  
BEST IMPROVEMENT**

Sledeće rešenje je najbolje rešenje iz okoline koje je bolje od trenutnog.

# SIMULIRANO KALJENJE

Ključna ideja algoritma je simulacija procesa hlađenja metala.

**Šta se dešava kada je metal usijan?**

**Šta se dešava kada se metal hlađi?**

Ova tehnika je korisna kod velikih i složenih prostora pretrage jer ne zahteva iscrpnu pretragu celokupnog prostora pretrage i ne generiše celu okolinu trenutnog rešenja, već ga iterativno unapređuje.

Prednosti:

- Efikasno pretraživanje prostora
- Prilagodljivost temperature
- Verovatnosno prihvatanje rešenja – Metropolisov kriterijum
- Jednostavna implementacija

Mane:

- Neophodna opreznost prilikom podešavanja parametara
- Nije garantovan globalni optimum

# SIMULIRANO KALJENJE

## Metropolisov kriterijum

- Omogućava algoritmu da prihvati lošija rešenja sa određenom verovatnoćom, time se sprečava zaglavljivanje u lokalnim optimumima.
- Direktno je proporcionalan temperaturi hlađenja

$e_{next}$ =	<i>rešenje koje razmatramo za sledeće</i>
$e_{curr}$ =	<i>rešenja koje imamo za trenutno</i>
$\Delta E$ =	$e_{next} - e_{curr}$
$T$ =	<i>temperatura</i>

- Ako je razlika  $\Delta E$  pozitivna – **imamo poboljšanje**
  - $e_{curr}$  postaje  $e_{next}$
- Ako je razlika  $\Delta E$  negativna – **nemamo poboljšanje**
  - verovatnoća prihvatanja lošijeg rešenja je izražena Metropolisovim kriterijumom odnosno rešenje će biti prihvaćeno sa verovatnoćom --->  $e^{\frac{\Delta E}{T}}$

# SIMULIRANO KALJENJE

**Primer:**

a. Za negativnu vrednost  $\Delta E$  i visoku vrednost  $T$ :

- $\Delta E = -10$  (na primer,  $e_{next}$  lošije za 10 jedinica u odnosu na  $e_{curr}$ )
- $T = 100$  (visoka temperatura)
- $e^{(-10 / 100)} = e^{(-0.1)} \approx 0.9048$

Zaključak: Verovatnoća prihvatanja  $e_{next}$  je oko 0.9048, što znači da je vrlo verovatno da će  $e_{next}$  biti prihvaćen.

b. Za negativnu vrednost  $\Delta E$  i nisku vrednost  $T$ :

- $\Delta E = -10$  ( $e_{next}$  lošije za 10 jedinica u odnosu na  $e_{curr}$ ) kao u primeru pod 'a'
- $T = 10$  (niska temperatura)
- $e^{(-10 / 10)} = e^{(-1)} \approx 0.3679$

Zaključak: Verovatnoća prihvatanja  $e_{next}$  je oko 0.3679, što znači da je manja verovatnoća da će  $e_{next}$  biti prihvaćen.

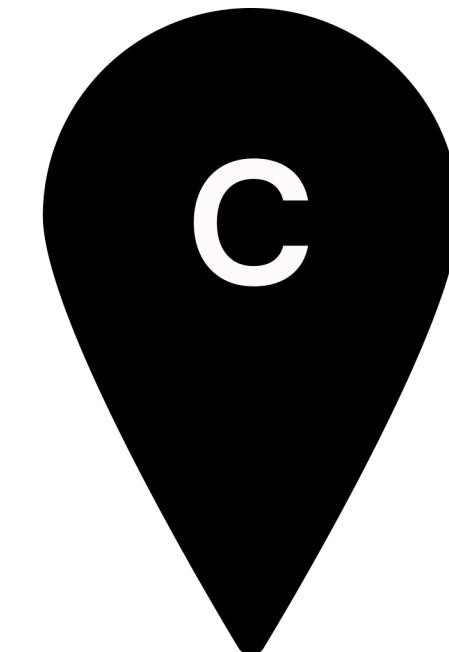
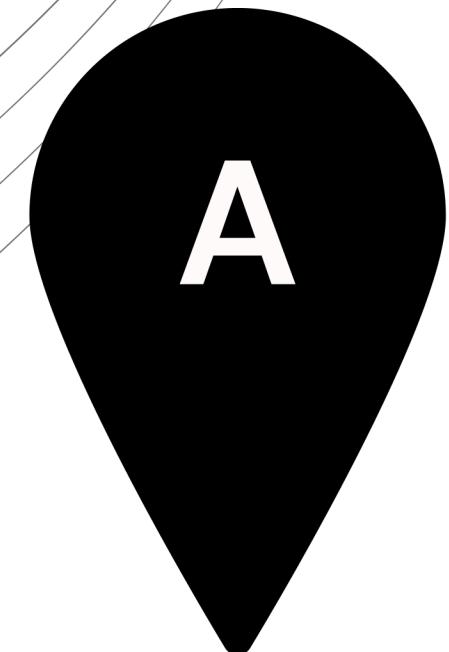
$$e^{\frac{\Delta E}{T}}$$

**Metropolisov kriterijum**

$e_{next} =$	rešenje koje razmatramo za sledeće
$e_{curr} =$	rešenja koje imamo za trenutno
$\Delta E =$	$e_{next} - e_{curr}$
$T =$	temperatura

Ovo omogućava algoritmu da istraži različite delove prostorarešenja na početku, a zatim se fokusira na poboljšavanje kvaliteta rešenja kako pretraga napreduje.

# SIMULIRANO KALJENJE



## SIMULATED ANNEALING

linearna strategija hlađenja

## SIMULATED ANNEALING COOLING STRATEGY

- geometrijska
- korenska
- logaritamska

## SIMULATED ANNEALING ADAPTIVE

dinamički određuje hlađenje temperature  
Algoritam prati brojače prihvaćenih i ispitanih rešenja, koristeći ih da bi svaka iteracija petlje definisala trenutnu stopu prihvatanja novih rešenja.

## SIMULATED ANNEALING WITH INTENSIVE SEARCH

"intenzivna pretraga" ukazuje na to da se algoritam trudi da pažljivije istraži prostor rešenja kako bi pronašao što bolje rešenje.

"generate\_neighbor\_with\_heuristic"

# GENETSKI ALGORITMI

Genetski algoritmi su metaheurističke tehnike inspirisane procesom evolucije u prirodi.

Ovi algoritmi koriste princip selekcije, ukrštanja i mutacije kako bi pretražili prostor rešenja i pronašli optimalna ili suboptimalna rešenja za različite probleme.

Implementacija genetskog algoritma uključuje stvaranje početne populacije, evaluaciju rešenja, selekciju roditelja, ukrštanje između roditelja i primenu mutacije. Kroz iteracije generacija, algoritam evoluira populaciju ka željenom rešenju.

- Početna populacija – Slučajno generisane permutacije skupa A
- Selekcija roditelja – ruletska
- Ukrštanje – jednopoziciono ukrštanje sa određenom verovatnoćom
- Mutacija – zamena dva nasumično odabrana gena svakog rešenja u trenutnoj populaciji sa određenom verovatnoćom
- Najbolje rešenje u poslednjoj generaciji ne mora biti najbolje u čitavom prostoru pretrage

# REZULTATI

A	C	BF	time [sec]	BF_BNB	time [sec]	LSFI	time [sec]	LSBI	time [sec]
5	4	3	0	3	0.001	3	0	2	0
7	14	10	0.038	10	0.033	10	0.001	10	0.001
10	20	14	18.811	14	15.096	11	0.003	10	0.003
25	50	-	too long	-	too long	38	0.068	36	0.106
50	100	-	too long	-	too long	81	1.766	74	2.092
100	200	-	too long	-	too long	155	35.654	159	54.253
125	250	-	too long	-	too long	191	88.907	198	173.201
150	300	-	too long	-	too long	242	299.812	233	396.027
175	350	-	too long	-	too long	283	610.709	277	810.542

$$e^{\frac{\Delta E}{T}}$$

# REZULTATI

A	C	SA	time [sec]	SACS geometry	time [sec]	SACS sqrt	time [sec]	SACS log	time [sec]
5	4	3	0.029	3	0.025	3	0.058	3	0.053
7	14	10	0.033	10	0.032	10	0.052	10	0.059
10	20	14	0.027	14	0.028	13	0.07	12	0.069
25	50	40	0.065	38	0.07	32	0.216	28	0.217
50	100	80	0.205	79	0.202	52	0.597	52	0.594
100	200	152	0.669	148	0.65	99	2.004	90	2.181
125	250	171	1.021	178	0.995	118	3.05	110	3.184
150	300	205	1.393	222	1.463	140	4.255	124	4.925
175	350	254	1.844	247	2.122	158	5.944	147	5.696

$$e^{\frac{\Delta E}{T}}$$

# REZULTATI

A	C	SA adaptive	time [sec]	SA with intensive search	time [sec]	Genetic algorithm	time [sec]
5	4	2	0.053	3	0.197	3	0.541
7	14	6	0.061	10	0.206	10	0.896
10	20	5	0.069	14	0.272	14	1.258
25	50	18	0.209	45	0.86	29	3.752
50	100	26	0.615	90	2.465	55	11.259
100	200	84	2.186	174	8.98	86	39.942
125	250	79	3.079	207	12.627	104	60.23
150	300	106	4.427	254	19.042	130	83.224
175	350	118	5.693	297	25.537	144	111.108

$$e^{\frac{\Delta E}{T}}$$



**HVALANA**  
**PAŽNJI**