

Zadatak 1: Genetički algoritam

- Najbolje rješenje svake instance prikazano listom koja sadrži permutaciju ogovarajućih gradova uz evaluaciju prijađenog puta, za svaki kriterij zasebno
- Cilj je minimizirati duljinu prijađenog puta u obilasku gradova svake instance
- Kriterij zaustavljanja: broj iteracija (inicijalno)
- Ideja za poboljšanje algoritma – brže izvršavanje
 - Ako se određeni broj iteracija isto rješenje pokazuje kao trenutno najbolje, uzmi to rješenje kao optimalno za danu instancu
 - Ako određeni broj iteracija nije pronađeno bolje rješenje od prethodno najboljeg (svako iduće je veće od njega), uzmi to (najmanje) rješenje
 - *mutationRate* = 0.05
- Implementacija algoritma u *dz4_zad1.py*

Eliminacijski

- U svakom koraku mijenja se samo jedna jedinka koja može postati roditelj
- Strategija odabira roditelja: k-turnirska selekcija, za k=3
- Strategija reprodukcije -> operator križanja: slijedno križanje
- operator mutacije – mutacija zamjenom
- Rezultati spremljeni u *GA_elim_makespan_solutions.py*,
GA_elim_makespan.py, *GA_elim_traveltime_solutions.py*,
GA_elim_traveltime.py
- Broj dopustivih rješenja:
 - Makespan: 6
 - Travel time: 4
- Broj optimalnih rješenja:
 - Makespan: 0
 - Travel time: 0

Generacijski

- Ciklički postupak smjene generacije
- k-turnirska selekcija, $k=3$
- operator križanja: slijedno križanje
- operator mutacije – mutacija zamjenom
- Rezultati spremljeni u *GA_gen_makespan_solutions.py*,
GA_gen_makespan.py, *GA_gen_traveltime_solutions.py*,
GA_gen_traveltime.py
- Broj dopustivih rješenja:
 - Makespan: 6
 - Travel time: 6
- Broj optimalnih rješenja:
 - Makespan: 0
 - Travel time: 0

+ elitizam

- svojstvo algoritma da ne može izgubiti najbolje pronađeno rješenje odnosno najbolje rješenje/a iz trenutne populacije uvijek prolazi u sljedeću generaciju
- Rezultati spremljeni u *GA_gen_elit_makespan_solutions.py*,
GA_gen_elit_makespan.py, *GA_gen_elit_traveltime_solutions.py*,
GA_gen_elit_traveltime.py
- Broj dopustivih rješenja:
 - Makespan: 5
 - Travel time: 6
- Broj optimalnih rješenja:
 - Makespan: 0
 - Travel time: 0

Zadatak 2: Usporedba implementiranih algoritama

Algoritam	OPTIMALNA RJEŠENJA			DOPUSTIVA RJEŠENJA	
	Makespan	Travel time		Makespan	Travel time
Pohlepni algoritam	1	0		65	4
Lokalno pretraživanje (2-opt)	34	0		57	4
Lokalno pretraživanje (random swap)	10	1		22	23
Simulirano kaljenje (SA)	2	2		6	4
Tabu pretraživanje	55	0		66	4
Optimizacija kolonijom mravi (ACO)	33	5		51	52
Optimizacija rojem čestica (PSO)	0	1		3	5
Hibridni algoritam (ACO + TABU)	33	5		52	52
Eliminacijski genetički algoritam	0	0		6	4
Generacijski genetički algoritam	0	0		6	6
Generacijski genetički algoritam sa elitizmom	0	0		5	6

Najveći broj optimalnih rješenja postignut je Tabu pretraživanjem odnosno ACO algoritmom. Ako u obzir uzmemo i broj postignutih dopustivih rješenja, hibridni algoritam također daje umjereno dobre rezultate. Generalno gledano, kod svih algoritama nije bio potreban jako veliki broj iteracija, odnosno to nije dovelo do boljih rezultata. Izmjenjivanje ostalih pripadnih hiperparametara poboljšalo je neke od algoritama.