

**Zadatak 1: Optimizacija kolonijom mravi**

- Najbolje rješenje svake instance prikazano listom koja sadrži permutaciju ogovarajućih gradova uz evaluaciju prijedrenog puta, za svaki kriterij zasebno
- Cilj je minimizirati duljinu prijedrenog puta u obilasku gradova svake instance
- Kriterij zaustavljanja: broj iteracija (inicijalno)
- Ponašanje mrava: za svakog mrava poziva se funkcija koja gradi rješenje
- Strategija osvježavanja feromona:
  - Isparavanje feromonskog traga
  - Samo određeni broj najboljih mrava ostavlja feromonski trag
- 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
- Implementacija algoritma u *dz3\_zad1.py*
- Rezultati spremljeni u *dz3\_ACO\_makespan\_solutions.txt*, *dz3\_ACO\_makespan.txt*, *dz3\_ACO\_traveltime\_solutions.txt*, *dz3\_ACO\_traveltime.txt*
- Broj optimalnih rješenja
  - Makespan: 33
  - Travel time: 5

**Zadatak 2: Optimizacija rojem čestica**

- Najbolje rješenje svake instance prikazano listom koja sadrži permutaciju ogovarajućih gradova uz evaluaciju prijedrenog puta, za svaki kriterij zasebno
- Cilj je minimizirati duljinu prijedrenog puta u obilasku gradova svake instance
- Početna rješenja generirana nasumično
- Svaka čestica ažurira najbolje globalno rješenje s obzirom na kriterij minimizacije
- Kriterij zaustavljanja: broj iteracija (inicijalno)

→ poboljšanje algoritma:

- 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
- Implementacija algoritma u *dz3\_zad2.py*
  - Rezultati spremljeni u *dz3\_PSO\_makespan\_solutions.txt*, *dz3\_PSO\_makespan.txt*, *dz3\_PSO\_traveltime\_solutions.txt*, *dz3\_PSO\_traveltime.txt*
  - Broj optimalnih rješenja Makespan: 0  
Traveltime: 1
  - Mijenjanje hiperparametara (težina inercije, broj čestica) i broj iteracija nije poboljšalo rezultat

### Zadatak 3: **Hibridni algoritam**

- Kombinacija Optimizacije kolonijom mravi i Tabu pretraživanja
- Tabu pretraživanje iz DZ2
- Početno rješenje svake instance je pohlepno rješenje iz prve zadaće
- 2-opt – zamijeni dva brida ako tom zamjenom dobivamo bolji makespan odnosno traveltime, ovisno o kriteriju
- TABU lista – promjena prethodno promijenjenih bridova zabranjena je 2 iteracije
- Nakon određenog broja iteracija ACO algoritma, primijeni Tabu pretraživanje
- Tabu pretraživanje istražuje susjedstvo trenutno najboljeg rješenja i nadalje ga uspoređuje sa globalno najboljim rješenjem kao i do sada
- Implementacija algoritma u *dz3\_zad3.py*
- Rezultati spremljeni u *dz3\_hybrid\_makespan\_solutions.txt*, *dz3\_hybrid\_makespan.txt*, *dz3\_hybrid\_traveltime\_solutions.txt*, *dz3\_hybrid\_traveltime.txt*
- Broj optimalnih rješenja
  - Makespan: 33
  - Travel time: 5