



## Predaja zadaće

Rok za predaju zadaće je srijeda 22.11.2023. do 12:00 sati.

U zadaćama iz Heurističkih algoritama promatrat ćemo inačicu dobro poznatog problema trgovačkog putnika - Problem trgovačkog putnika s vremenskim prozorom (engl. Travelling salesman problem with time window - TSPTW). Rješavanje problema trgovačkog putnika podrazumijeva pronalazak rute u kojoj su sva dana mjesta posjećena točno jedan put, da se vratilo u početnu točku i da se pri tom minimizirao neki kriterij optimizacije (trajanje putovanja, udaljenost...). U problemu koji ćemo mi promatrati, TSPTW, postoji dodatan uvjet kojim je dan vremenski interval u kojem je moguće posjetiti određeno mjesto, te izvan tog intervala obilazak nije moguć, tj. potrebno je pričekati do trenutka kad je obilazak moguć te nakon toga nastaviti dalje put.

Prilikom rješavanja TSPTW, najčešće se koriste dva kriterija optimizacije: vrijeme potrebno za putovanje (engl. travel time) i ukupno potrebno vrijeme za obilazak rute (engl. makespan). Kod prvog kriterija optimizacije u obzir se uzima samo vrijeme potrebno za putovanje (isti kriterij optimizacije kao originalni TSP), dok kod drugog kriterija je potrebno uračunati i vrijeme čekanja (kad se na neku lokaciju dođe izvan vremenskog intervala u kojem je moguće napraviti obilazak). Napomena: uočite da iako prvi kriterij optimizacije odgovara onom kod originalnog TSP-a to ne znači da je i rješenje isto, jer zbog vremenskog okvira u kojem neko mjesto može biti posjećeno, postoje rute koje nisu dopustive, npr. moguće je da neki čvor nije moguće posjetiti nakon nekog drugog jer je vremenski interval posjeta prvom završen prije nego što je drugom počeo.

Instance problema koje ćemo rješavati u sklopu zadaća dostupne su na web stranici: <https://lopez-ibanez.eu/tsptw-instances>. Osim instanci (Benchmark Instances dio na stranici) na stranici se može pronaći opis instanci, popis literature koji može u nekim trenucima dobro doći, te najbolja poznata rješenja za oba kriterija koja ćemo koristiti prilikom optimizacije. Prilikom rješavanja zadataka treba koristiti sve dostupne instance.

**Zadatak 1.** Ovaj zadatak u zadaći se sastoji od definiranja stvari koje će se koristiti u svim zadaćama. Za početak je potrebno:

- implementirati parser kojim će se moći u program učitavati instance problema koje će se zatim rješavati određenim pristupom,
- implementirati parser kojim će se moći učitati najbolja poznata rješenja za sve instance za oba kriterija,
- implementirati metodu kojom će se provjeravati je li rješenje dopustivo,
- implementirati metodu kojom će se usporediti je li proslijeđeno rješenje jednako najboljem poznatom.

S obzirom da se radi o inačici TSP problema, prikaz rješenja se sam po sebi nameće - prikaz permutacijama. Kod određivanja ruta poznato je da se kreće iz mjesta 0



(skladište) te se u njemu i završava. Zbog toga u rješenju nije potrebno pisati 0 na početku i na kraju. Odnosno ako imamo mjesta 1, 2, 3, 4 i 5 za obići i obilazimo ih na sljedeći način:  $0 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 0$ , rješenje će nam biti zapisano kao 2 1 5 3 4.

U skladu s tim potrebno je napraviti metodu koja će raditi ispis rješenja u tekstualnu datoteku na način da napiše o kojoj se instanci radi, koje je rješenje, koju vrijednost postiže za kriterij *travel time*, a koju za kriterij *makespan*. Kako bi se mogle izračunavati vrijednosti za ove kriterije, potrebno je definirati i implementirati odgovarajuće funkcije/metode. (potrebno pozvati kod svakog pristupa/zadatka u ovoj i narednim zadaćama)

Dodatno, napravite metodu koja će za ostvarena rješenja napraviti tekstualnu datoteku (za svaki kriterij posebno) koja će se sastojati od sljedećih stupaca:

- naziv instance,
- apsolutno odstupanje,
- relativno odstupanje<sup>1</sup>,
- DA ili NE ovisno o tome je li se pronašlo najbolje poznato rješenje.

(potrebno pozvati kod svakog pristupa/zadatka u ovoj i narednim zadaćama)

**Zadatak 2.** Definirajte pohlepni pristup za rješavanje ovog problema. Definirajte pohlepni korak i argumentirajte je li on statička ili dinamička heuristika. Implementirajte svoje rješenje, a konačne rezultate zapišite kako je predviđeno Zadatkom 1.

**Zadatak 3.** Definirajte lokalno pretraživanje za dani problem. Definirajte barem dva različita načina za dobivanje susjedstva. Implementirajte svoje rješenje i usporedite dobivena konačna rješenja prilikom korištenja različitih susjedstva. Implementirajte tehniku odabira susjeda: najbolji poboljšavajući susjed i prvi poboljšavajući susjed. Rješenja zapišite kako je predviđeno Zadatkom 1 i to za oba definirana susjedstva.

Napomena: podzadatke koje treba raspisati predajte u pdf fileu. Rješenja koja dobijete ne moraju biti optimalna na kraju, bitno je da se vidi na koji način ste implementirali svoja predložena rješenja.

---

<sup>1</sup>Ako je  $s$   $x$  označeno optimalno rješenje, a  $s$   $x^*$  aproksimacija tog rješenja, apsolutno odstupanje se računa kao:  $\Delta x = |x - x^*|$ , a relativno odstupanje kao  $\delta x = \frac{\Delta x}{|x|}$ .