



## Predaja zadaće

Rok za predaju zadaće je utorak 15.11.2022.

U ovoj zadaći ćemo promatrati problem raspoređivanja vozila s kapacitetima (engl. *Capacitated Vehicle Routing Problem* - CVRP). U problemu je zadano  $n$  kupaca te je dostupno  $K$  vozila kapaciteta  $Q$  koja kupcima dostavljaju robu iz skladišta. U inačici problema koju ćemo promatrati u ovoj zadaći, postoji samo jedno skladište. Svaki od  $n$  kupaca ima točno određen zahtjev za količinom robe. Cilj je svim kupcima dostaviti traženu robu iz skladišta sa što manje vozila tako da duljina ukupno prijeđenih puteva bude minimalna. Vozilo kada jednom krene iz skladišta može obilaziti gradove sve dok ne potroši svu robu koju je povezo. Nakon što potroši robu, treba se vratiti natrag u skladište.

U folderu instance su dani testni primjeri (instance problema) u obliku datoteka s `.vrp` ekstenzijom za problem raspoređivanja vozila s kapacitetima na kojima ćete testirati svoje implementacije. U svakoj instanci su dostupne sljedeće informacije:

- *NAME* - naziv filea
- *COMMENT* - iz kojeg skupa problema dolazi
- *TYPE* - tip problema, kod nas su svi CVRP (*capacitated vehicle routing problem*)
- *DIMENSION* - broj kupaca + 1 skladište
- *EDGE\_WEIGHT\_TYPE* - tip udaljenosti kojom treba izračunati udaljenost između čvorova (lokacije kupaca i skladišta su zadane preko koordinata)
- *CAPACITY* - sadrži informaciju koliki je kapacitet svakog dostupnog vozila
- *NODE\_COORD\_SECTION* - dio u kojem su napisane koordinate skladišta i kupaca; svaki redak ispod (ima ih *DIMENSION*) kao prvi element ima redni broj lokacije (1. je uvijek skladište, ostalo su kupci), a kao druga dva elementa koordinate skladišta ili kupca
- *DEMAND\_SECTION* - dio u kojem su opisane potrebe svakog kupca za jedinicama robe; svaki redak ispod (ima ih *DIMENSION*) kao prvi element ima redni broj lokacije (1. je uvijek skladište, ostale su kupci), a drugi element predstavlja količinu robe koju određeni kupac zahtijeva (uočite da kraj skladišta uvijek stoji demand 0)
- *DEPOT\_SECTION* - označava kojom je koordinatom označeno skladište (u svim primjerima će biti 1) i ispod toga se nalaze -1 i EOF koji označava kraj instance



Rješenje problema treba zapisati u sljedećem obliku:

- za svako iskorišteno vozilo je potrebno ispisati jedan redak u obliku:  
Vozilo #(rbr\_vozila): kupac\_1, kupac\_2, ..., kupac\_n  
gdje su kupac\_1, kupac\_2 itd. redni brojevi lokacija na kojima se nalaze kupci koje vozilo obide na svojoj ruti.
- nakon redaka koji predstavljaju vozila i njihove rute, treba pisati još jedan redak u obliku:  
Trošak (ukupna\_duljina\_svih\_prijeđenih\_ruta)

**Zadatak 1.** 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 u datoteku rjesenjaGreedy.txt tako da prije svakog rješenja piše naziv instance, zatim ukupna dobivena duljina puta za tu instancu te apsolutno i relativno odstupanje <sup>1</sup> od optimalnog (ili najboljeg poznatog) rješenja.

Npr.

X-n101-k25 27600 9 0.00032619

A-n33-k6 742 0 0 itd.

Na kraju svakog retka napišite i DA ako ste dobili optimalno (ili najbolje poznato) rješenje (informacija o najboljem rješenju se nalazi u excel fileu opisInstanci.xlsx) ili NE ako se vaše rješenje razlikuje od optimalnog (ili najboljeg poznatog) rješenja.

**Zadatak 2.** Definirajte lokalno pretraživanje za dani problem. Potrebno je definirati na koji način će se prikazati stanje, kako će se definirati susjedstvo i koja je funkcija cilja. 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 u datoteke rjesenjaLS1.txt i rjesenjaLS2.txt u obliku kao i kod pohlepnog pretraživanja za prvo i drugo definirano susjedstvo.

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  $\delta x = \frac{\Delta x}{|x|}$ .