Kapacitativni problem usmjeravanja vozila iz višebrojnih skladišta

Krešimir Baksa, Mihael Šafarić i Matija Šantl

Heurističke metode optimizacije
Fakultet elektrotehnike i računarstva

Zagreb, siječanj 2015.

Uvod

Zadano:

- usmjereni težinski graf
- skladišta, njihov položaj, kapacitet i pripadajući troškovi otvaranja skladišta
- korisnici, njihov položaj i potražnja
- početni trošak i kapacitet vozila

Varijable:

- broj skladišta
- broj vozila
- obilasci vozila

Optimizacijski kriteriji:

- minimizirati ukupni trošak otvaranja skladišta i ruta te ukupni trošak odabranih ruta

Pohlepni algoritam

Pohlepni algoritam nakon učitavanja podataka razvrstava korisnike po skladištima na način da u trenutnom koraku minimizira trošak trenutnog korisnika do njemu najbližeg iz skladište u koje ga pokušava staviti uz dodatak troška povratka od tog korisnika do skladišta. Ako skladište u tom trenutku nema ni jednog korisnika, cijeni se dodaje trošak otvaranja skladišta.

To radi tako dugo dok ne razvrsta sve korisnike.

Mutacija

Nakon dobivanja početnog rješenja, napravi se *POPULACIJA* broj kopija tog rješenja, te se nad svakim od njih radi se mutacija. Za slučajno odabrana skladišta X i Y, slučajnim odabirom uzmi korisnika Z iz skladišta X te ga probaj staviti u skladište Y ako time ne narušavamo kapacitet skladišta Y.

Ukoliko je mutirano rješenje bolje od trenutnog, ono se zamjenjuje u populaciji.

Taj se postupak ponavlja *ITERACIJA* broj puta.

Lokalna pretraga

Nakon što smo dobili *POPULACIJA* broj mutiranih rješenja, nad svakim od njih se pokreće postupak lokalne pretrage.

Postupak lokalne pretrage, odabire par korisnika X i Y, te zamjenjuje njihova skladišta. Ako je rješenje ne narušava kapacitet ni jednog skladišta, spremamo ga u red te nastavljamo lokalnu pretragu sa sljedećim rješenjem iz reda.

Postupak radimo tako dugo dok ima rješenja u redu ili napravimo *limit* broj iteracija.

Početno rješenje

```
1  dok ima nereazvrstanih korisnika X:
2    za svako skladiste Y:
3      za svakog korisnika Z u skladistu Y:
4
5        cijena = udaljenost(X, Z) + udaljenost(X, Y)
6
7        ako je skladiste Y prazno:
8          cijena += cijena otvaranja skladista Y
9
10       ako je  cijena najmanja do sad:
11         W = Y
12
13       stavi korisnika X u skladiste W
```

Mutacija

```
1  ponovi ITERACIJA broj puta:
2    za svako rjesenje R iz populacije:
3      slucajno odaberi skladiste X
4      slucajno odaberi skladiste Y
5      slucajno odaberi korisnika A iz skladista X
6
7      ako prebacivanje A iz X u Y narusava ogranicenja:
8        ponovi odabir skladista i korisinka
9      inace:
10       prebaci korisnika A iz X u Y
11       spremi izmjenjeno rjesenje
```

Lokalna pretraga

```
1  dodaj rjesenje X u red Q
2
3  ponovi LIMIT broj puta ili dok ne konvergira:
4    rjesenje R = prvo rjesenje iz reda Q
5    za svaki par korisnika (A, B):
6      ako zamjenom skladista korisnika A i B u R ne
           narusavao ogranicenja:
7        R' = zamjeni skladista korisnika A i B
8
9        ako je R' bolje od X:
10         X = R'
11
12       ako je trenutno R' bolje R:
13         dodaj R' u Q
```

Fitness

```
1  fitness = 0
2  za svako skladiste X:
3    ako ima korisnika u X:
4      fitness += cijena otvaranja skladista X
5
6      pohlepnim algoritmom napravi potrebne rute R[] za
           sve korisnike iz X
7      za svaku rutu R‘:
8        grubom silom izracunaj najkraci Hamiltonov ciklus
             , C
9        cijena += obilazak ciklusa C
10       cijena += cijena automobila * broj ruta
```

Rješenje

```
1  A = pocetno rjesenje
2  P = niz od POPULACIJA kopija od A
3
4  ponovi ITERACIJA broj puta:
5    za svako rjesenje R iz P:
6      mutiraj(R)
7
8  za svako rjesenjeR  iz P:
9    lokalna_pretraga(R)
10
11 izdvoji najbolje rjesenje iz P
```

Ostvareno rješenje

Najbolje ostvareno rješenje iznosi 315983.