

Kapacitativni problem usmjeravanja vozila iz višebrojnih skladišta

M. Čulinović, D. Dolović, M. Hruška

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

15.1.2015.

Sadržaj

- Popis korištenih algoritama
- Generiranje početnih rješenja
- Opis rješenja
- Rezultati

Algoritmi

- Clark-Wright algoritam (napomena: u nastavku CW)
- Genetski algoritam
- Simulirano kaljenje - različite varijacije
 - za optimizaciju broja ciklusa
 - za cjelokupno rješenje
 - za raspoređivanje korisnika po ciklusima
- Lokalno pretraživanje

Početno rješenje

- generirana su početna rješenja tako da su:
 - sva skladišta otvorena
 - četiri skladišta otvorena, jedno zatvoreno
 - tri skladišta otvorena, dva zatvorena
- algoritam:
 - ponavljaj dok nisu svi korisnici pridruženi nekom skladištu
 - za svako skladište:
 - pridruži sljedećeg najbližeg korisnika tako dugo dok se ne prekorači kapacitet vozila - stvoren je ciklus
 - za svaki ciklus nađi optimalnu rutu pomoću *brute-force* algoritma kojim se generiraju sve permutacije (prihvatljivo jer ciklusi sadrže maksimalno 5 korisnika)
 - dodaj ciklus skladištu

Početna rješenja - zaključci

Analiza:

- najbolja rješenja dobivena kada su otvorena po tri skladišta
- najveći trošak otvaranja skladišta
- dobiven broj ciklusa: 25

Optimizacija - ideje:

- koristiti početna rješenja gdje su otvorena 3 skladišta
- optimizirati razmještaj korisnika po skladištima - grupiranje (*engl. clustering*)
- optimizirati broj ciklusa
- optimizirati razmještaj korisnika po ciklusima

Najbolje početno rješenje

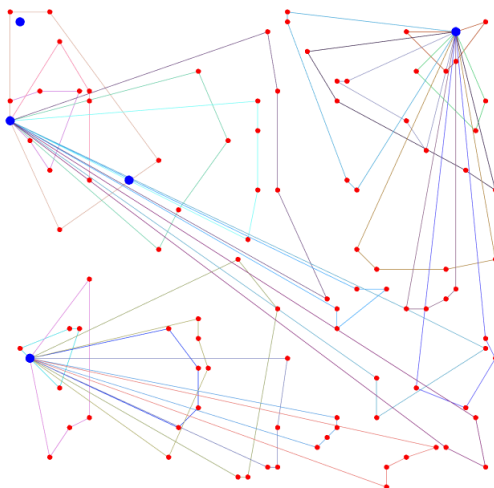


Figure 1: najbolje početno rješenje, trošak = 307528

Optimizacija

Četiri faze optimizacije

- ❶ *clustering* - raspoređivanje korisnika po skladištima
 - genetski algoritam za raspoređivanje
 - CW algoritam za stvaranje ciklusa u najboljem rješenju
- ❷ minimiziranje broja ciklusa
 - simulirano kaljenje - koristeći susjedstvo koje napravi zamjenu 2 korisnika iz različitih skladišta te kreira nove cikluse koristeći CW algoritam
- ❸ raspoređivanje korisnika po ciklusima
 - simulirano kaljenje - koristeći različita susjedstva, opisano kasnije
- ❹ lokalno pretraživanje
 - pokušava zamijeniti mjesta u konačnom rješenju svakom paru korisnika

Genetski algoritam

- generiranje početne populacije - u početnom rješenju dobivenom ranije opisanim postupkom desetorici nasumično odabranih korisnika promijeni skladište
- funkcija evaluacije - minimiziraj zbroj udaljenosti od skladišta do korisnika koji mu pripadaju
- elitizam - iz generacije u generaciju prenosi najbolju jedinku
- selekcija - veća vjerojatnost odabira jedinki s boljom dobrotom
- križanje u jednoj točki
- mutacija - desetorici korisnika dodijeli nasumično odabrano skladište
- kriterij zaustavljanja je broj iteracija
- stvori rute/cikluse na trenutnom razmještaju korisnika po skladištima koristeći CW algoritam

Clark-Wright algoritam

- algoritam se pokreće za svako skladište
- **Četiri koraka algoritma:**
 - 1 za svaki par korisnika (i,j) izračunaj uštedu koja se dobi ako se oni nalaze u istoj ruti:

$$s(i,j) = d(\text{skladiste}, i) + d(\text{skladiste}, j) - d(i,j) \quad (1)$$

- 2 sortiraj uštede od najveće prema najmanjoj te ih spremi u listu ušteta
- 3 za svaki element u listi ušteta pokušaj uključiti vezu (i,j) u ciklus; razlikujemo tri slučaja
 - ako ni i ni j nisu uključeni u neki ciklus, stvori novi ciklus i ubaci oba člana para
 - ako je samo jedan član para uključen u neki ciklus i on nije unutarnji u ruti, dodaj drugi član tom ciklusu
 - ako su oba člana para uključena u cikluse i oba su rubna u ruti spoji cikluse u jedan
- 4 za svakog korisnika koji nije uključen u neki ciklus u koraku 3, stvori novi ciklus duljine 1

Minimizacija broja ciklusa

- algoritam simuliranog kaljenje s jednim susjedstvom ranije opisanim
- nakon svakog pomaka pokušavaju se kreirati novi ciklusi kako bi se eventualno smanjio njihov ukupan broj
- analizirajući rezultate ovo susjedstvo se pokazalo uspješnim te se ukupan broj ciklusa/vozila uspije smanjiti s 26/25 dobivenih genetskim algritmom na 25/24

Simulirano kaljenje za raspoređivanje korisnika po ciklusima

- u ovom trenutku broj vozila/ciklusa svakog skladišta je fiksiran te je svaki korisnik već smješten u neki ciklus
- parametri: broj koraka, početna temperatura, faktor smanjivanja temperature
- dinamičko određivanje broja pokušaja te dovoljnog broja uspješnih pokušaja pri svakoj temperaturi u ovisnosti o broju korisnika

Susjedstvo

- 4 tipa susjedstva (svako s nekom vjerojatnošću):
 - odaberi po jednog korisnika iz dva različita skladišta te im zamijeni cikluse
 - odaberi dva korisnika iz istog skladišta te im zamijeni cikluse
 - iz 2 različita skladišta odaberi po 2 korisnika iz jednog ciklusa te im zamijeni cikluse
 - iz istog skladišta odaberi 2 korisnika iz jednog ciklusa i 2 iz drugog te im zamijeni cikluse
- za svaki ciklus ruta obilaska se stvara generirajući sve permutacije korisnika u ciklusu te se odabira optimalna; zbog malog broja korisnika u svakom ciklusu generiranje svih permutacija nije problem

Lokalno pretraživanje

- ponavljaj tako dugo dok je pronađen poboljšavajući potez
 - za svaki par korisnika pokušaj zamijeniti njihove cikluse
 - ako je potez poboljšavajući prihvati ga

Najbolje dobiveno rješenje

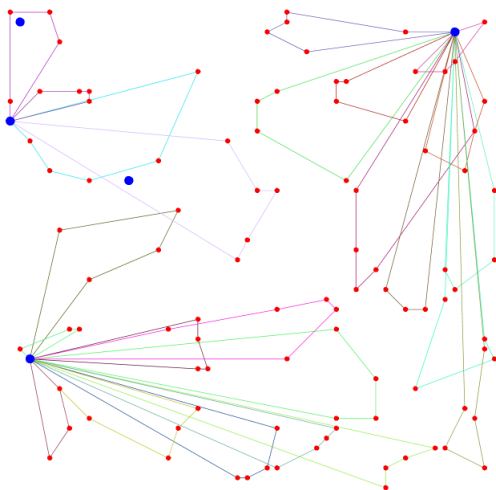


Figure 2: najbolje rješenje, trošak = 276498