Matematički fakultet
Univerziteta u Beogradu

Seminarski rad u okviru kursa
Naučno izračunavanje

Profesor: dr Mladen Nikolić
Asistent: dr Stefan Mišković

Tabu pretraga za lokacijski problem ograničenih kapaciteta sa jednostrukim alokacijama

Marija Mijailović 1093/2017
mi14199@alas.matf.bg.ac.rs

Miroslav Mišljenović 1110/2018
mr12260@alas.matf.bg.ac.rs

Beograd, septembar 2019.

1. Prost lokacijski problem ograničenih kapaciteta sa jednostrukim alokacijama

Prost lokacijski problem ograničenih kapaciteta sa jednostrukim alokacijama (eng. single-source capacitated facility location problem – SSCFLP) bavi se problematikom dodeljivanja korisnika određenim resursima. Svaki korisnik ima svoje zahteve (eng. demands), dok resurse karakteriše kapacitet, kao i fiksna cena korišćenja pojedinačnog resursa, odnosno cena uspostavljanja tog resursa. Poznata je i matrica cena, tj. matrica koja govori o tome koliko korisnika košta korišćenje nekog resursa. Svaki korisnik mora biti pridružen tačno jednom resursu, dok jedan resurs može opsluživati više korisnika. Zadatak je minimizovati ukupnu cenu korišćenja takvog sistema raspoređivanja korisnika i uspostvaljanja resursa.

2. Tabu pretraga

Tabu pretraga je metaheuristika koja se zasniva na poboljšavanju vrednosti tekućeg rešenja. Za razliku od lokalne pretrage, ovde se uvodi skup zabranjenih poteza T, koji sužava izbor novog rešenja. U svakoj iteraciji se skup T može ažurirati na različite načine. Na početku algoritma se proizvoljno ili na neki drugi način generiše početno rešenje i izračunava vrednost njegove funkcije cilja. Vrednost najboljeg rešenja se najpre inicijalizuje na vrednost početnog. Zatim se algoritam ponavlja kroz nekoliko iteracija. U svakom koraku se razmatra rešenje u okolini trenutnog. Ukoliko je vrednost njegove funkcije cilja bolja od vrednosti funkcije cilja trenutnog rešenja, ažurira se trenutno rešenje.

Po potrebi se ažurira i vrednost najboljeg dostignutog rešenja. Algoritam se ponavlja dok nije ispunjen kriterijum zaustavljanja. Kriterijum zaustavljanja može biti, na primer, dostignut maksimalan broj iteracija, dostignut maksimalan broj ponavljanja najboljeg rešenja, ukupno vreme izvršavanja, itd.

Tabu pretraga se može prikazati sledećim pseudokodom:

Algoritam Tabu pretraga

1: Generisati početno rešenje s
2: Inicijalizovati vrednost najboljeg rešenja $f^* \leftarrow f(s)$
3: Inicijalizovati skup zabranjenih poteza $T \leftarrow \emptyset$
4: **while** nije ispunjen kriterijum zaustavljanja **do**
5: Izabrati proizvoljno rešenje s' u skupu $N(s) \setminus T$
6: **if** $f(s') < f(s)$ ili drugačiji, podesno definisani, kriterijum prihvatanja **then**
7: $s \leftarrow s'$
8: **end if**
9: **if** $f(s') < f^*$ **then**
10: $f^* \leftarrow f(s')$
11: **end if**
12: Ažurirati skup T
13: **end while**
14: Ispisati vrednost rešenja f^*

Memorija se kod tabu pretrage vezuje za dužinu čuvanja rešenja u skupu T. U opštem slučaju, ona može biti kratkoročna, srednjoročna i dugoročna.

Kratkoročna memorija se implementira kao lista nedavno razmatranih rešenja. Kada se potencijalno rešenje pojavi u tabu listi, ne može se ponovo razmatrati u glavnom delu algoritma, sve dok se to rešenje ne izbaci iz tabu liste, po nekom unapred dogovorenom kriterijumu.

Srednjoročna memorija predstavlja pravila intenziviranja namenjena pristranosti prema obećavajućim područjima prostora pretrage.

Dugoročna memorija sadrži pravila diverzifikacije tj. pravila koja pretražuju nove regione (kada se pretraga zaglavi "na platou" ili u lokalnom optimumu, ova pravila određuju kako se tabu lista prazni ili osvežava).

3. Implementacija algoritma

Okosnica našeg seminarskog rada se sastoji, u osnovi, od koda sa vežbi iz oblasti lokalne pretrage, uz izmene fukcije "solutionValue" i dodatak funkcije "tabuSearch".

U glavnom delu programa imamo spoljašnju petlju, gde se skup uspostavljenih resursa inicijalizuje na slučajan način.

"SolutionValue" je funkcija koja izračunava vrednost funkcije cilja za zadatu konfiguraciju, tj. uspostavljene resurse. Tu je implementirana unutrašnja petlja za naš program, tako što se skup korisnika permutuje na slučajan način, za koren broja korisnika. Ovo je urađeno u cilju postizanja što boljih rezultata. Ispostavilo se da su dobijena mnogo bolje rešenja, nego bez pomenute unutrašnje petlje i permutacije korisnika.

Funkcija "tabuSearch" sadrži implementaciju tabu listu bezirane na kratkoročnoj memoriji.

4. Rezultati

Algoritam smo testirali na 22 test primera, preuzetih sa sajta:
http://people.brunel.ac.uk/~mastjjb/jeb/orlib/capinfo.html

Neki od test primera su sadržali greške, tako da nisu mogla da se dobiju zadovoljiva rešenja, jer su neki zahtevi korisnika bili veći nego kapaciteti resursa. Te podatke smo korigovali kako bi se dobilo zadovoljivo rešenje, stoga su dobijeni različiti rezultati od optimalnih. Optimalni rezultati dati su na linku:

http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/capopt.txt

File	Best value	Optimal value
cap101	796648.4374999998	796648.437
cap102	854704.1999999998	854704.200
cap103	893782.1124999999	893782.112
cap104	928941.7499999999	928941.750
cap111	793439.5625	826124.713 *
cap112	852180.375	901377.213 *
cap113	897194.0750000001	970567.750 *
cap114	940356.2625000001	1063356.488 *
cap121	793439.5625000001	793439.563
cap122	851495.3250000001	852524.625 ***
cap123	894447.8250000002	895302.325 ***
cap124	928941.7499999999	946051.325 ***
cap131	793792.4624999999	793439.562
cap132	851495.3250000001	851495.325
cap133	893251.5125000001	893076.712
cap134	928941.75	928941.750
cap41	936220.8499999999	1040444.375 *
cap43	1010641.4499999998	1153000.450 *
cap51	1014038.45	1025208.225 ***
cap61	932615.7499999997	932615.750
cap71	932615.7499999998	932615.750
cap72	977799.3999999998	977799.400

* - izmena u podacima
*** - dobijeno bolje rešenje od optimalnog

Takođe, testirali smo algoritam na primerima od 5x5 do 10x10 redom. Optimalna rešenja za te test primere smo dobili pomoću algoritma grube sile, a našim algoritmom dobili smo poklapanja sa tim rešenjima.

File	Dimension	BruteForce	TabuSearch
cap131	5x5	43122.387500000004	43122.3875
cap101	6x6	55061.30000000001	55061.299999999996
cap111	7x7	83410.35	83410.35
cap121	8x8	97278.2875	97278.28749999999
cap41	9x9	103121.675	103121.67499999999
cap51	10x10	127208.85000000002	127208.84999999999

5. Literatura

- Izvorni kod: https://github.com/miroslav-misljenovic/Tabu-search-heuristic-for-the-Single-Source-Facility-Location-Problem
- Sadržaji kursa Naučnog izračunavanja: http://ni.matf.bg.ac.rs/
- Tabu pretraga: https://en.wikipedia.org/wiki/Tabu_search
- Test primeri: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/