Univerzitet u Beogradu

Matematički fakultet

Ana Simićević

Lokacijski problemi na mrežama

diplomski (master) rad

Beograd

2012.

Mentor: **Doc. dr Zorica Stanimirović**

Matematički fakultet u Beogradu

Član komisije: **prof. dr Đorđe Dugošija**

Matematički fakultet u Beogradu

Član komisije: **prof. dr Milan Dražić**

Matematički fakultet u Beogradu

Datum odbrane: ________________________________

Lokacijski problemi na mrežama

Rezime

U ovom radu opisani su Lokacijski problemi na mrežama. Ovi NP-teški problemi nalaze veliku primenu u praksi. Modeli optimizacije na grafovima i mrežama predstavljaju značajnu oblast interesovanja u operacionim istraživanjima i primenjenoj matematici. U poslednjih par decenija raste broj radova koji se u toku jedne godine objavi na temu mrežnih optimizacionih modela. Razlog tome su teorijski zanimljivi problemi koji nastaju pri formulisanju matematičkih modela za rešavanje važnih praktičnih zadataka. Pored onih koji se odnose na realne mreže, kao što su putne, električne, telekomunikacione, računarske i druge, postoji i čitav niz drugih problema koji se mogu formulisati kao zadaci optimizacije na grafu ili mreži.

Ključne reči: lokacijski problemi, grafovi i mreže, metaheuristike, kombinatorna optimizacija

Location problems on network

Abstract

This review describes Location problems on networks. These NP-hard problems are widely used in practice. Models of optimization on graphs and networks represent a significant area of interest in operational research and applied mathematics. In the past few decades, there is a growing number of works revealed during one year on the subject of network optimization models. This is due to the theoretically interesting problems which arise during formulation of mathematical models for solving important practical problems. Apart from those relating to real networks, such as roads, electricity, telecommunication, computer and others, there is a number of other problems which can be formulated as optimization tasks on the graph or network.

Keywords: location problems, graphs and networks, metaheuristics, combinatorial optimization.

Predgovor

Rad se sastoji od pet poglavlja. U uvodnom poglavlju date su osnovne informacije o lokacijskim problemima. U drugom poglavlju opisane su osobine grafova i mreža. U trećem poglavlju opisane su metode za rešavanje lokacijskih problema. Pregled i opis sedam lokacijskih problema na mrežama dat je u četvrtom poglavlju. U zaključku je naglašen predmet, značaj i raznovrsnost matoda za rešavanje lokacijskih problema.

Želela bih da se zahvalim:

Mentoru doc. dr Zorici Stanimirović na rukovođenju prilikom izrade ovog rada, podršci, razumevanju i više nego korisnim sugestijama.

Članovima komisije prof. dr Đorđu Dugošiji i prof. dr Milanu Dražiću na pažljivom čitanju rukopisa.

Svojoj porodici, prijateljima i kolektivu Univerziteta Singidunum bez čije bi podrške izrada ovog rada bila neuporedivo teža.

Beograd, 2012.

Kandidat

Ana Simićević

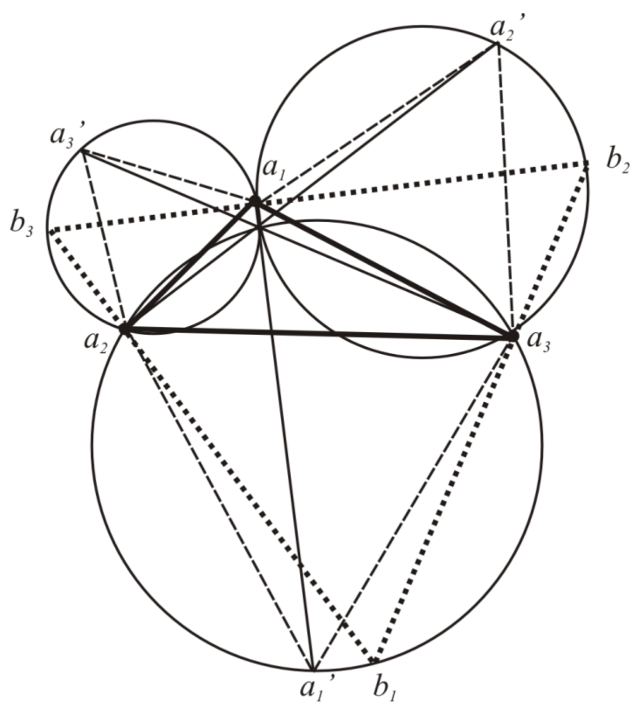
1.Uvod

1.1 Lokacijski problemi

Teorija lokacijskih problema bavi se zadacima izbora jedne ili više lokacija za objekte koji pružaju određene usluge u prostoru određene dimenzionalnosti. Prilikom izbora lokacije vodi se računa o interakcijama između samih lokacija i prostorno raspoređenih korisnika. Interakcije mogu predstavljati transporte roba ili ljudi, fizičke veze kao što su saobraćajnice, cevovodi, kablovi, komunikacije i sl. Priroda ovih interakcija u velikoj meri je uslovljena prirodom objekata koji se lociraju. Veliki broj lokacijskih problema svrstava se u NP-teške probleme kombinatorne optimizacije ([Gar79], [Cre97], [Dre02]), pa je dobijanje optimalnih rešenja limitirano dimenzijom problema. NP-teški problemi predstavljaju kombinatorne probleme za čije rešavanje nisu poznati algoritmi s vremenom računanja koje je zavisno od veličine problema, tj. čija se složenost može izraziti polinomnom funkcijom (npr. linearnom, kvadratnom, kubnom,...). U tom smislu, poželjno je razvijati i implementirati razne heuristike koje mogu da podrže dobijanje rešenja i u slučaju obimnih i kompleksnih lokacijskih problema kakvi se u realnosti sreću.

1.1.1. Istorijski razvoj teorije lokacijskih problema

Značaj lokacijskih problema potvrđuje činjenica da se prvi pisani trag o lokacijskim problemima sreće još u Bibliji. Prvi korak, s matematičke tačke gledišta, u rešavanju ovih problema načinio je francuski matematičar *Pjer de Ferma* (1601-1665). On je postavio prvi lokacijski problem: Za zadate tri proizvoljne tačke u ravni, locirati četvrtu tačku, tako da je suma njenih rastojanja do preostale tri tačke minimalna ili drugačije rečeno naći tačku trougla čiji je zbir rastojanja do temena minimalan. *Toričeli* (1608-1647) je prvi primetio da se krugovi opisani oko jednakostraničnih trouglova, konstruisanih nad stranicama zadatog trougla, seku u optimalnoj tački (Slika 1.).



Slika 1.

Praktični značaj lokacijskih problema prvi je opisao *Alfred Weber* (1868-1958) u svojoj poznatoj knjizi [Web09], koju je objavio početkom dvadesetog veka. On je ilustrovao problem minimizacije troškova transporta u industriji postavljajući pitanje: Gde izgraditi industrijsko postrojenje tako da cena transporta bude minimalna. *Weber* je optimalnu lokaciju našao konstruktivnim putem i predložio isti postupak za rešavanje problema dimenzije veće od 3.

Ako su (x_i, y_i) koordinate n fiksiranih tačaka (lokacije potrošača), w_i *i=1, ..., n* odgovarajuće težine (cene transporta do datog potrošača po jedinici dužine), *Weber-ov* problem nalaženja optimalne lokacije skladišta (x^*, y^*) može biti formulisan na sledeći način:

$$\min_{(x,y)} W(x,y) = \sum_{i=1}^{n} w_i d_i(x,y)$$

gde je $d_i(x,y) = \sqrt{(x-x_i)^2 - (y-y_i)^2}$ Eukidsko rastojanje tačaka (x,y) i (x_i, y_i).

Razvijene su brojne i raznovrsne metode i modeli za rešavanje *Weber-ovog* problema i raznih njegovih modifikacija koje su predmet proučavanja i do današnjih dana, ([Dea85], [Lov88], [Dre02]).

1.1.2. Klasifikacija lokacijskih problema

Lokacijski problemi mogu se klasifikovati na različite načine u zavisnosti od principa klasifikacije. Bilo je više pokušaja u literaturi da se izvrši stroga klasifikacija svih lokacijskih problema [Bra89], ali se ti pokušaji nisu održali jer se uvek može naći lokacijski problem koji ne pripada ni jednoj od unapred određenih klasa.

Nekoliko najčešćih podela lokacijskih problema su:

1) Prema prostoru u kome se donosi odluka, lokacijski problemi se dele na kontinualne, diskretne i mrežne lokacijske probleme. Ako novi objekti mogu biti locirani u ravni (R2) ili prostoru (R3), govori se o kontinualnim problemima: polje promenljivih je kontinum, odnosno dopustivi skup ima beskonačno mnogo tačaka. U diskretnim lokacijskim problemima postoji utvrđena lista mogućih lokacija, pa se zadatak svodi na izbor jedne ili više lokacija iz konačnog, odnosno diskretnog skupa mogućih lokacija. Kod mrežnih lokacijskih problema koristi se matematička struktura: težinski graf ili mreža, pa se zadatak izbora lokacije svodi na izbor jednog ili više čvorova mreže (diskretni mrežni problemi) ili tačaka na granama mreže (kontinualni mrežni problemi). Dakle mrežni modeli imaju elemente i kontinualnih i diskretnih problema.

2) Pema obliku funkcije cilja, lokacijski problemi se dele na min-sum i min-max modele. U prvoj grupi, funkcijom cilja se minimizuje težinski zbir svih rastojanja novih i fiksnih objekata. Na taj način se favorizuju "prosečni" korisnici, a zanemaruju udaljeni ili izolovani. Drugi tip funkcije cilja ravnopravno tretira sve korisnike tako što se nalaze nove lokacije koje minimizuju maksimalno rastojanje između postojećih i nepoznatih objekata. Primer ovakvih problema je lokacija vatrogasnih brigada, gde i udaljeni stanovnici imaju ista prava kao i oni koji se nalaze u gradovima.

3) Prema broju objekata koje treba otvoriti, lokacijski problemi se dele na endogene i egzogene probleme. U endogenim problemima broj novih objekata je unapred zadat. Nasuprot njima su egzogeni problemi, gde je broj novih objekata nepoznata veličina i njena vrednost se dobija kao rezultat optimizacije. Primeri egzogenih modela su: prost lokacijski problem (Simple-plant location problem ili Uncapacitated facility location problem) i problem pokrivanja skupa.

4) Karakteristike na osnovu kojih se takođe mogu praviti razlike među lokacijskim problemima su: matematička priroda kriterijuma relevantnih za problem (deterministička, nedeterministička), planski horizont (statički problem, dinamički problem), procedure za rešavanje lokacijskih problema (intuitivni pristup, egzaktni algoritmi, heuristike, složene procedure, simulacija, ekspertni sistemi), broj kriterijuma optimalnosti relevantnih za problem (jedan, više).

1.1.3. Merenje rastojanja u lokacijskim problemima

Među ulazne podatke za rešavanje lokacijskih problema spadaju rastojanja između korisnika koji zahtevaju uslugu i čvorova u kojima je moguće locirati objekte. Pod "rastojanjem" se pored fizičkog rastojanja može podrazumevati i vreme putovanja, troškovi putovanja i sl.

U lokacijskim problemima je potrebno odrediti rastojanje između dve tačke u posmatranom prostoru na osnovu poznavanja njihovih koordinata. U tu svrhu se koriste l_p metrike gde je p realni broj takav da je $1 \le p \le \infty$. Ove metrike se u opštem slučaju definišu nad tačkama prostora R^n (tj. skupa svih uređenih n-torki realnih brojeva).

Neka su u prostoru R^n date dve tačke $A = (x_1^A, ..., x_j^A, ..., x_n^A)$ i $B = (x_1^B, ..., x_j^B, ..., x_n^B)$ Rastojanje između tačaka A i B u l_p metrici za $1 \le p \le \infty$ se računa po obrascu:

$$d_{l_p}(A,B)=\sum_{j=1}^{n}\sqrt[p]{|x_j^A-x_j^B|^p}$$

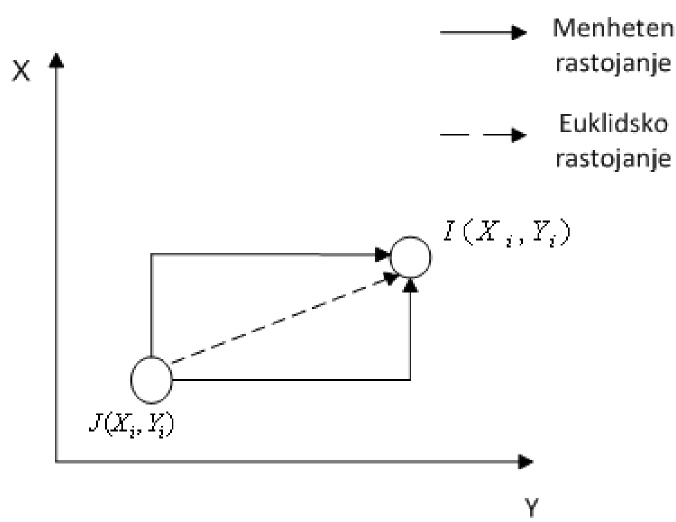
$$d_{l_\infty}(A,B)=\lim_{p\to\infty} d_{l_p}(A,B)$$

Mada teorijski ima beskonačno mnogo metrika l_p tipa, najznačajnije su i u praksi se najviše koriste sledeće metrike:

$$d_{l_1}(A,B)=\sum_{j=1}^{n}|x_j^A-x_j^B|$$ l_1 (pravougaona) metrika

$$d_{l_2}(A,B)=\sqrt{\sum_{j=1}^{n}(x_j^A-x_j^B)^2}$$ l_2 (Euklidova) metrika

$$d_{l_\infty}(A,B)=\max|x_j^A-x_j^B|$$ l_∞ (Čebiševljeva) metrika



Slika 2.

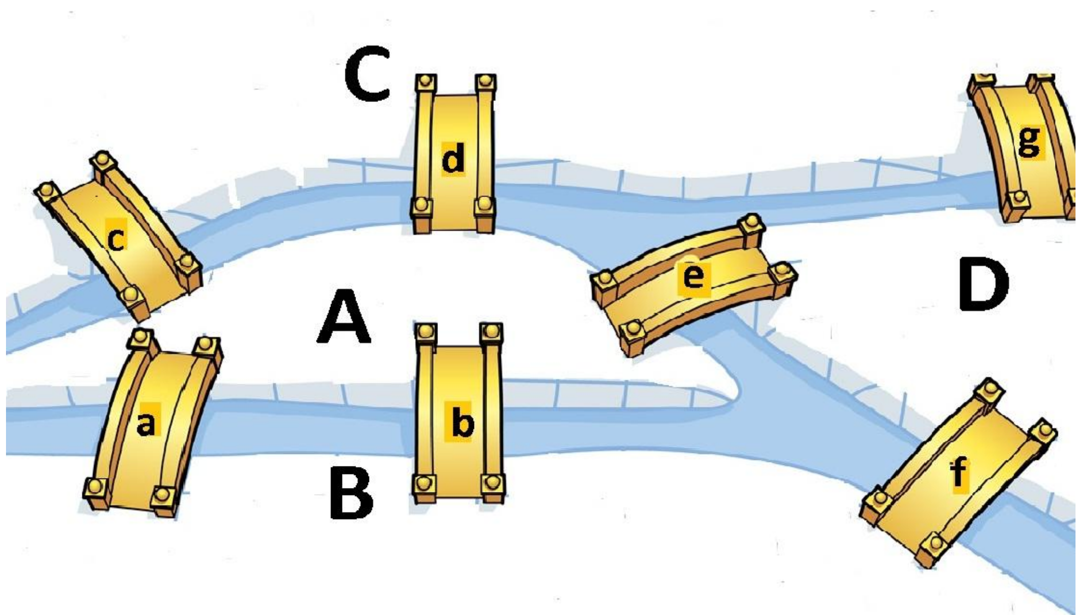
Izbor l_p metrike zavisi od više faktora, od kojih su najznačajniji:

1. Priroda problema: korišćenje određene l_p metrike daje manje ili više dobru aproksimaciju realnog rastojanja u zavisnosti od prirode prostora u kome se tačke nalaze. Na primer, ako je moguće kretati se pravolinijski između dve tačke, tačno rastojanje između njih se dobija Euklidovom metrikom (Slika 2.), ali u gradovima u kojima su ulice pod pravim uglom, dužina puta između dve tačke najbolje će se aproksimirati pravougaonom metrikom. Ovo rastojanje se naziva i *Menhetn* rastojanje (Slika 2.) jer podseća na rastojanja u ovoj njujorškoj četvrti u kojoj su ulice i avenije pod pravim uglom.

2. Računska složenost: ako je neki model bitno lakše rešiti primenom određene metrike, a preciznost dobijenog rezultata nije od presudnog značaja (tj. ako se manja odstupanja mogu tolerisati), tada će ta metrika biti upotrebljena.

2.Grafovi i mreže

Prvi problem i njegovo rešenje u teoriji grafova jeste rad *Leonarda Ojlera* pod nazivom Sedam mostova Kenigsberga (Slika 3.), objavljen 1736. godine, ali se začetkom teorije grafova smatra problem četiri boje koji je izložio *Frensis Gutri* 1852. godine. On postavlja pitanje da li je moguće obojiti zemlje na geografskoj karti sa samo četiri boje, a da se ne pojave četiri susedne zemlje obojene istom bojom. Tokom pokušaja rešavanja ovog problema otkrivene su mnoge teoreme i postavljeni mnogi teoretski pojmovi i koncepti. Problem su rešili *Kenet Apel i Volfgang Heken*, [Ape77] 1976. godine. Njihov dokaz je veoma komplikovan i oslanja se na teorijske rezultate niza matematičara i značajan rad računara.



Slika 3. Dijagram Keningsburških mostova

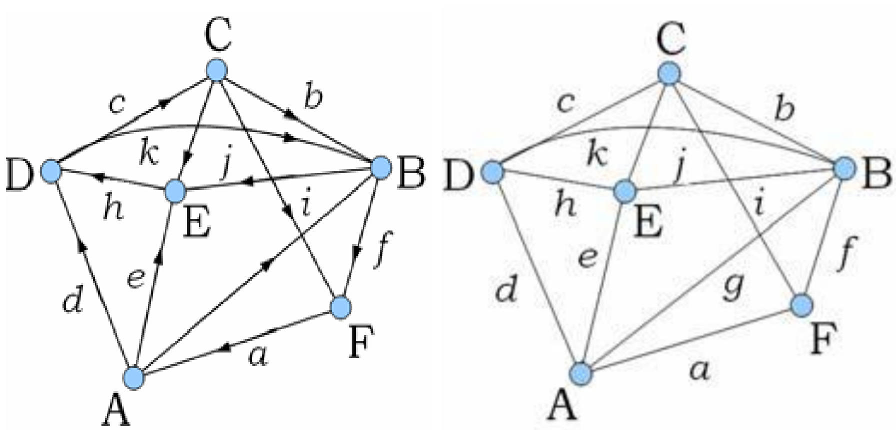
Mnogi lokacijski problemi mogu se predstaviti i rešiti primenom rezultata teorije grafova i mreža. Graf odnosno mreža su matematički koncepti koji se često koriste u modeliranju lokacijskih problema. U narednim podsekcijama biće izložene definicije i osobine grafova.

2.1 Osnovni pojmovi

Graf je uređen par G=(X,U) , gde je X konačan neprazan skup i U skup dvoelementnih podskupova skupa X. Elementi skupa X se nazivaju čvorovi, dok se elementi skupa U nazivaju grane.

Digraf je uređen par G=(X,U), gde je X konačan neprazan skup i U skup uređenih parova elemenata skupa X. Ovi uređeni parovi se nazivaju orjentisane grane.

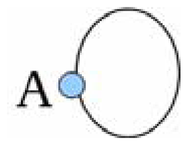
Graf ili digraf se predstavlja geometrijskom figurom sastavljenom od tačaka i linija koje spajaju pojedine parove tačaka.Tačke predstavljaju čvorove grafa dok linije predstavljaju grane grafa ili orjentisane grane digrafa. U prvom slučaju linije su neorjentisane, dok se u drugom slučaju linije orjentišu (Slika 4.).



Graf (X,U) Digraf(X,U)

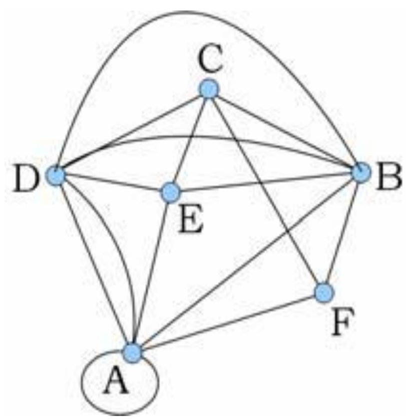
Slika 4. Graf i digraf

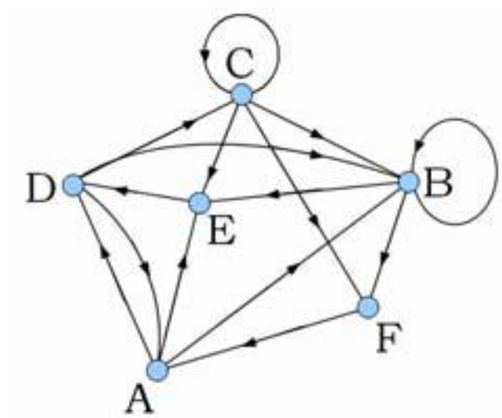
Grana oblika {a,a}, odnosno (a,a) naziva se petlja.Na Slici 5. ona spaja čvor A sa samim sobom.



Slika 5. Petlja

Graf sa samo jednim vrhom se naziva trivijalni graf, inače je netrivijalan. Za graf kažemo da je prost ukoliko nema petlje i nikoje dve grane ne spajaju isti par čvorova. Ako su u grafovima dva čvora spojena sa više različitih grana, ukuljučujući i višestruke petlje, onda takve grafove nazivamo multigrafovima ili multidigrafovima (Slika 6.).





Multigraf ***Multidigraf***

Slika 6.

Za dva čvora grafa kažemo da su susedna ako su spojena granom.Dva susedna čvora su krajnje tačke svake grane koja ih spaja.Ako je neki čvor jedna od krajnjih tačaka izvesne grane kaže se da su čvor i grana incidentni ili susedni, odnosno da se ta grana *stiče* u ovom čvoru.Dve grane su susedne ako imaju zajednički čvor. Kažemo da su čvor i grana incidentni ili susedni ako grana počinje odnosno završava se u tom čvoru.

Stepen čvora x predstavlja broj susednih čvorova za čvor x, odnosno broj grana koje se stiču u tom čvoru.

Ako u nekom digrafu grana u spaja čvorove xi i xj i orjentisana je od xi ka xj, kaže se da grana u izlazi iz čvora xi i ulazi u čvor xj.Ulazni stepen čvora je jednak broju grana koje ulaze u taj čvor, a izlazni stepen je jednak broju grana koje izlaze.Petlja se smatra i ulaznom i izlaznom granom za odgovarajući čvor.

Definicija 1. Neka je dat graf (ili digraf) G=(X,U). Graf oblika H=(Y,T), gde su $Y \subset X$ i $T = U \cap Y \times Y$ (T je podskup skupa U i sadrži sve one parove iz U koji su obrazovani samo od elemenata skupa Y), se naziva podgrafom grafa (podgrafom digrafa) G, obrazovanim (indukovanim) skupom čvorova Y.

Definicija 2. Put dužine k u digrafu je svaki niz grana u_1,u_2, ..., u_k koji ima sledeće osobine:

1. Grana u_1 polazi iz proizvoljnog čvora digrafa

2. Grana u_i počinje u onom čvoru u kojem se završava grana u_{i-1}

Elementarni put je put koji prolazi najviše jedanput kroz svaki čvor grafa. Put koji se završava u istom čvoru u kom i počinje naziva se kružni ili zatvoreni put.

U grafu se svaka grana može shvatiti kao dvostrano orjentisana. Put nije definisan samo nizom grana , već se za svaku granu koja ulazi u posmatrani put mora naznačiti njena orjentacija na tom putu. Za grafove, put dužine k se definiše kao naizmenični niz čvorova x_i i grana u_i oblika x_1, u_1, x_2, u_2, ..., x_k, u_k, x_{k+1} , pri čemu je za i=1,2,..., k čvor x_i početni čvor, a x_{i+1} krajni čvor grane u_i.

Dužina puta u grafu (digrafu) je po definiciji jednaka broju grana koje se nalaze u putu. Rastojanje čvorova x i y u grafu se definiše kao dužina najkraćeg puta koji povezuje ta dva čvora.

Definicija 3. Graf je povezan ako se proizvoljna dva njegova čvora mogu povezati putem. Ako postoje čvorovi koji se ne mogu povezati putem, graf je nepovezan.

Neka su d_1, d_2, ..., d_n stepeni čvorova x_1, x_2, ..., x_n u grafu bez petlji koji ima m grana. Ako saberemo sve stepene čvorova, dobijamo dvostruki broj grana, pa važi relacija $d_1 + . . . + d_n = 2m$.

Teorema 1. Broj čvorova neparnog stepena u konačnom grafu bez petlji je paran.

Prethodna teorema u literaturi je poznata kao Lema o rukovanju [Cve96], jer se može interpretirati na sledeći način: U svakom društvu broj osoba koje su se rukovale neparan broj puta je paran broj. Ovde broj ososba koje su se rukovale predstavlja čvorove grafa.

Definicija 4. Graf se naziva regularnim stepena r ako je $d_1 = d_2 = \ldots = d_n = r$

Posebno su interesanti regularni grafovi stepena dva.

Definicija 5. Konačan povezan regularan graf stepena dva zove se kontura. Kontura koja sadrži sve čvorove grafa naziva se Hamiltonova kontura.

Definicija 6. Regularni grafovi sa n čvorova stepena n - 1 nazivaju se potpuni grafovi.Potpuni graf sa n čvorova označava se sa K_n.

Definicija 7. Dva grafa su izomorfna ako postoji uzajamno jednoznačno preslikavanje skupa njihovih čvorova (iz jednog u drugi) koje održava osobinu susednosti čvorova.

U mnogobrojnim zadacima modeliranja lokacijskih problema, čvorovi predstavljaju potencijalne lokacije za objekat (objekte) koji želimo locirati i/ili korisničke punktove, a grane grafa odgovaraju saobraćajnicama, komunikacionim linijama, električnim vodovima, itd.

Ukoliko se svakoj grani $(i,j) \in L$ pridruži jedan ili više skalara koji predstavljaju njenu dužinu, kapacitet, itd. i svakom čvoru $j \in N$ pridruži jedan ili više skalara koji predstavljaju njegov prioritet, kapacite, itd., onda se takav graf u literature naziva težinski graf ili mreža. Najčešće je u modeliranju lokacijskih problema mrežom svakoj grani $(i,j) \in L$ pridružen jedan nenegativni skalar c_{ij} koji je parametarski reprezent rastojanja (ili vremena putovanja, troškova putovanja, itd.) između dva čvora povezana granom. Svakom čvoru $j \in N$ pridružen je jedan nenegativni skalar V_j, težinski koeficijent čvora, koji predstavlja njegov prioritet: što je veće V_j, čvor $j \in N$ je većeg prioriteta. Mrežu, za koju važi da je $c_{ij} = c_{ji}$ za svako $(i,j) \in L$, zovemo neorijentisanom ili simetričnom. U praksi su posebno interesantne orijentisane tj. nesimetrične mreže, kod kojih može važiti $c_{ij} \neq c_{ji}$. Mreža je orijentisana ako su grane u njoj određene parovima čvorova (i,j).

Za rešavanje lokacijskih problema na mreži, koja se može predstaviti u ravni, razvijaju se različiti modeli. Opšti mrežni lokacijski problem se predstavlja u sledećem obliku: odrediti jedan ili više čvorova koji imaju najbolje karakteristike u zadatom smislu. U ovakvim modelima se koristi pojam rastojanja između dva čvora mreže. To rastojanje je po definiciji dužina najkraćeg puta (vremena putovanja, troškova putovanja, itd.) između posmatranih čvorova.

U literaturi su dobro poznati algoritmi za određivanje najkraćih puteva (algoritmi Dijkstra-e [Dij59], Bellman-a [Blm58], Dantzig- a, Floyd-a, itd.). Najkraće rastojanje između čvorova $i \in N$ i $j \in N$ označićemo sa d(i,j), a matricu najkraćih vremena putovanja (rastojanja) između svaka dva čvora $i \in N$ i $j \in N$ sa $D=[d(i,j)]_{nxn}$. Pri računanju dužine puta, u nesimetričnim mrežama, važno je koji je početni, a koji krajnji čvor puta. Prema tome, za jedan par čvorova mogu postojati dva rastojanja.

2.2. Pretraživanje grafova

U mnogim algoritmima zahteva se pretraživanje grafova po nekom utvrđenom principu. Na primer, može se početi od proizvoljnog čvora u obilazak njegovih susednih čvorova, zatim suseda njihovih suseda i tako redom dok ne obiđemo sve čvorove grafa. U tom procesu, uvek ćemo znati koji deo čvorova je pretražen, a u svakom narednom koraku se određuje koji će sledeći čvor biti prosleđen. Prioritet izbora potencijalnih kandidata se reguliše uvođenjem nekih ograničenja.

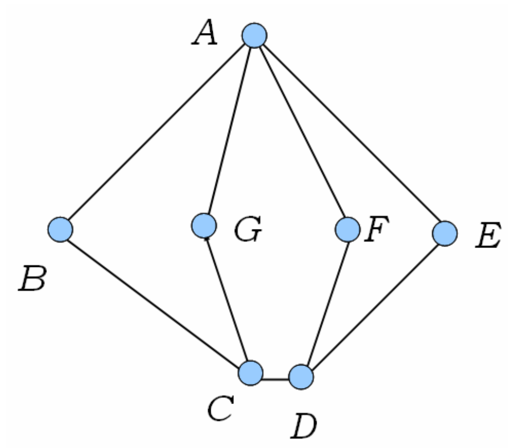
Postoje tri vrste pretraživanja grafova:

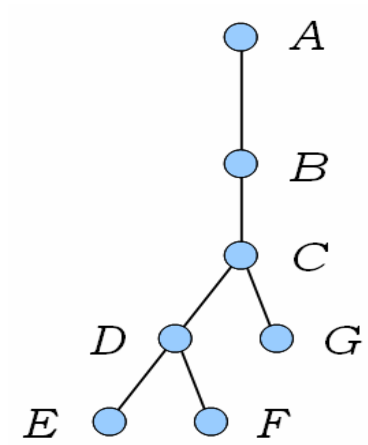
- Pretraživanje u dubinu
- Pretraživanje u širinu
- Kombinacija pretraživanja u dubinu i širinu

Za ove pretrage je karakteristično da se obilaze svi čvorovi grafa kao i grane, i to svaka po jedanput u oba smera.

Pretraga u dubinu:

Neka je *X* skup svih čvorova, a Y skup već prosleđenih čvorova. Označimo sa y čvor koji je poslednji prosleđen. Izbor narednog čvora za prosleđivanje se vrši na sledeći način: prvo se odredi da li među ne prosleđenim čvorovima, tj. čvorovima skupa X/Y postoji čvor koji je sudedan sa y, i ako postoji tada se među potencijalnim kandidatima bira bilo koji od njih (prosleđivanje unapred). U suprotnom, umesto čvora y se posmatra njegov prethodnik i za njega se ponavlja isti postupak kao i za čvor y (prosleđivanje unazad). Ukoliko ja graf povezan, ovim postupkom će svi čvorovi grafa biti prosleđeni. Kod nepovezanih grafova isti postupak se može primeniti na onim podgrafovima koji su povezani (Slika 7.).



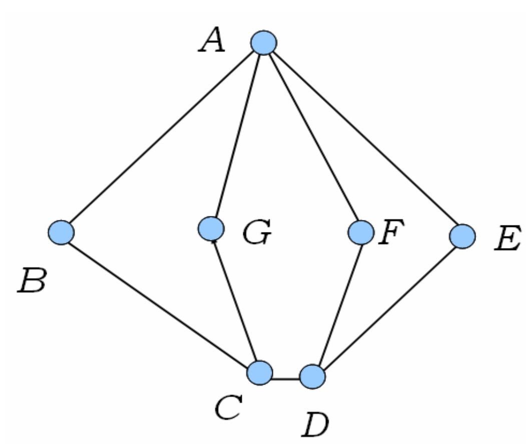


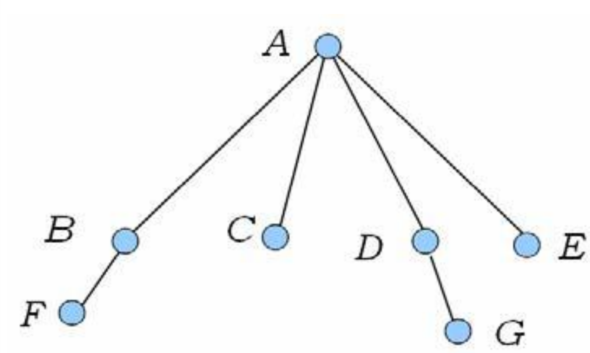
Graf

Stablo pretrage u dubinu po koracima A-B-C-D G-C-B -E-D-F-D-C-A

Slika 7.Pretraga u dubinu

Neka je *X* skup svih čvorova, a Y skup već prosleđenih čvorova. Neka je skup Q skup svih ne prosleđenih čvorova kojima je prioritet prosleđivanja ranije određen. Prosleđivanje čvorova je sada komplikovanije, jer se za izbor sledećeg kandidata mora posmatrati i skup Q, koji tretiramo kao red za čekanje (prvo se prosleđuje čvor koji je najduže u redu). Na početku pretrage skup Y je prazan i njemu se proizvoljno dodeljuje bilo koji čvor iz grafa. Iz skupa Q se bira čvor koji je najduže u redu i on se prosleđuje skupu Y, npr. čvor x. Skupu *Q* se na kraju reda unose čvorovi koji su susedni sa *x* a koji se već ne nalaze u skupu *Q* nisu ranije prosleđeni u *Y* (Slika 8.).





Graf

Stablo pretrage u širinu (po koracima A-B-C-D-E-F-G)

Slika 8.Pretraga u širinu

3. Metode za rešavanje lokacijskih problema

Metode za rešavanje lokacijskih problema se mogu podeliti na egzaktne i heurističke (približne). Egzaktne metode su najpoželjnije. One obezbeđuju optimano rešenje i verifikuju njegovu optimalnost, ali najčešće daju loše rezultate na problemima velikih dimenzija. U tim slučajevima pribegava se primeni heurističkih metoda. Heuristike ne garantuju nalaženje optimalnog rešenja problema, ali dobre heuristike rešavaju problem u velikom broju slučajeva i daju suboptimalna rešenja koja su bliska optimalnom rešenju. Neretko se dobijaju optimalna rešenja, ali se optimalnost ne može dokazati. Pri rešavanju problema kombinatorne optimizacije treba imati u vidu činjenicu da ne postoji metoda za njihovo rešavanje koja je univerzalna i koja daje najbolje rezultate na svim problemima. Zato se često vrše kombinacije raznih metoda i dobijaju hibridni algoritmi.

3.1. Egzaktne metode za rešavanje lokacijskih problema

3.1.1. Metoda odsecanja

Metodu odsecanja (***Cutting plane method***) za rešavanje problema celobrojnog programiranja prvi je predložio *Gomory,* [Gom63]. Nedostatak ove metode je spora konvergencija. Iz tog razloga ova metoda nije bila aktuelna dugo vremena u literaturi. Uvođenjem i razvojem teorije poliedara (polyherdal theory) došlo je do ponovnog oživljavanja ove metode, koja danas rešava široki skup problema.

Ako se u problemu celobrojnog programiranja izuzme uslov celobrojnosti, dobijamo problem linearnog programiranja (relaksacioni problem) koji se rešava simpleks metodom. Ako je dobijeno rešenje ovakve relaksacije celobrojno, onda je i rešenje problema celobrojno. U suprotnom, ideja u rešavanju problema je formiranje jedne linearne jednačine koju zadovoljava svako dopustivo polazno rešenje, a ne zadovoljava necelobrojno rešenje relaksacionog problema. Kažemo da ova linearna jednačina odseca nezadovoljavajuće nelinearno rešenje zbog čega se često naziva odsecajuća nejednakost ili rez. Ukoliko dopišemo ovu jednačinu ograničenjima polaznog problema, dobijamo zadatak ekvivalentan polaznom. Optimalno rešenje njegove relaksacije dobijene izostavljenjem uslova celobrojnosti, ako postoji, biće različito od rešenja prethodne relaksacije, pa je ili optimalno za polazni problem ili se postupak odsecanja može nastaviti. Na taj način, nastaju metode odsecanaja koje se razlikuju po načinu konstrukcije odsecajuće nejednačine i relaksacija.

Posmatramo *Gomory-evu* metodu odsecanja:
Relaksacija problema se izvodi tako što se uslov celobrojnosti rešenja izostavlja. Pretpostavimo da je linearna relaksacija rešena primalnom ili dualnom simpleks metodom do završnog oblika:

$$f - f_0 = +\sum_{j\in N} r_j x_j$$
$$y_{i_0} = x_i + \sum_{j\in N} y_{ij} x_j$$

Pri tome je *B* skup indeksa baz̆ičnih, a *N* skup indeksa nebazičnih promenljivih, $y_{i_0} \geq 0, i \in B$ i $r_j \geq 0$. Ukoliko su svi $y_{i_0}, i \in B$ celobrojni, odgovarajuće bazično rešenje je optimalno rešenje polaznog problema. Pretpostavimo da je neko y_{i_0} necelobrojno.
Označimo sa Φ(a) = a – [a] razlomljeni deo realnog broja a. Tada je $y_{ij} = \left[y_{ij} \right] + \phi(y_{ij})$, pa i-tu jednačinu možemo napisati u obliku:

$$\sum_{j\in N} \phi(y_{ij}) x_j - \phi(y_{i0}) = -x_i - \sum_{j\in N} \left[y_{ij} \right] x_j + \left[y_{i0} \right]$$

Kako je za svaki realan broj a, Φ(a) ∈ [0, 1), leva strana jednakosti je veća od -1, za svako dopustivo rešenje polaznog problema, a desna strana je celobrojna. Dakle, sledi:

$$\sum_{j\in N} \phi(y_{ij}) x_j - \phi(y_{i0}) \geq 0$$

Dobijena nejednakost predstavlje Gomory-ev rez.

Ukoliko se na spisak ograničenja dopiše ograničenje

$$-\phi(y_{i0}) = \sum_{j\in N} \phi(y_{ij}) x_j + s$$

gde je s izravnavajuća celobrojna nenegativna promenljiva, relaksacioni problem se može dalje rešavati dualnom simpleks metodom, a zatim po potrebi generisati novi rez.
Ukoliko neka od relaksacija ima prazan dopustiv skup, polazni problem nema dopustivih tačaka.

3.1.2. Metoda grananja i ograničavanja

Metoda grananja i ograničavanja (***Branch and Bound***), bazira se na principu “podeli pa vladaj“. Prvi su je predložili *Land A.H. i Doig A.G.* 1960 godine [Lan60]. Kako je početni problem suviše složen da bi bio direktno rešen, deli se na manje i manje delove koji se mogu rešiti. Deljenje prostora rešenja obavlja se ograničavanjem vrednosti pojedinih varijabli, dok se procenom maksimalne vrednosti rešenja u pojedinom delu mogu odbaciti delovi prostora rešenja koji daje najbolje vrednosti funkcije cilja. Loša osobina metode je nepolomijalnost, dok je dobra osobina što se u praksi ova metoda završava u dopustivoj tački koja je blizu rešenja.

Do trenutno najboljeg rešenja se može doći neposrednom proverom ili nekim heurističkim postupcima u toku algoritma. Metoda grananja i ograničavanja sastoji se od tri osnovna koraka: grananje (*branch*), ograničavanje (*bound*) i procenjivanje (*fathoming*).

Posmatrajmo problem:

$$\min f(x),\quad x \in X \qquad (P)$$

X – diskretan skup

Problem (P) razlažemo na potprobleme sa ciljem da se neki od potproblema neće ni rešavati ako se ustanovi da nemaju bolje rešenje od trenutno najboljeg rešenja. Grananje problema P na potprobleme se može vršiti tako što se skup X pokrije unijom svojih delova X_k $k \in K$ i problem P zameni skupom potproblema:

$$\min f(x)\,,\ x \in X_k \qquad (P_k)$$

Optimalno rešenje polaznog problema je optimalno za bar jedan problem sa spiska. Razlaganje skupa X na podskupove je pogodno zato što skupovi mogu biti dovoljno sitni ili laki za rešavanje, pa se vrednost funkcije na njima može lakše oceniti. U slučaju da je potproblem težak za rešavanje, možemo ga olakšati izostavljanjem nekih ograničenja (umesto funkcije f(x) posmatramo funkciju g(x) u kojoj je neko ograničenje izostavljeno) ili proširivanjem njegovog dopustivog skupa X_k na neki skup Y_k.Na taj način dobijamo problem:

$$\min g(x)\ , x \in Y_k \qquad (Q_k)$$

koji nazivamo relaksacioni problem.

Optimalna vrednost relaksacionog problema Q_k nije veća od optimalne vrednosti problema P_k, tj. važi $v(Q_k) \leq v(P_k)$

U idealnom slučaju optimalne vrednosti su jednake i početni problem je rešen. Ako problem nije rešen, dobijamo ocenu odozdo $m(P_k)= v(Q_k)$ njegove optimalne vrednosti. Neka je, npr. trenutno najbolje pronađeno rešenje M ili je najbolje rešenje ograničeno sa M:

$$m(P_k)\geq M$$

U skupu X_k nema rešenja koje je bolje od onog koje smo dobili kao tekući rekord zato što je:

$$v(P_k)\geq m(P_k)\geq M\geq v(P)$$

U trenutku kada procenimo da problem neće dati bolja rešenja od tekućih, možemo preći na sledeći potproblem, a tekući problem eliminisati sa spiska potproblema.

U svakom grananju generiše se samo konačan broj potproblema, i ako se oni nikada ne ponavljaju, algoritam se posle konačno mnogo iteracija završava nalaženjem optimalnih vrednosti i rešenja polaznog problema.

Ovaj postupak se najlakše može prikazati preko grafa čiji su čvorovi dobijeni grananjem. Dobijeni graf nazivamo drvetom pretraživanja u kome je koren polazni problem a listovi najsitniji problemi (problemi koji ne mogu dati bolje rešenje daljim grananjem).

Pri izboru problema za grananje, postoje dve osnovne taktike:

- Grananje u širinu
- Grananje u dubinu (LIFO strategija, grana se prvo problem koji je poslednji ušao na spisak)

U početnoj fazi poželjnije je koristiti grananje u dubinu, jer na brži način dolazimo do dopustivog rešenja, dok je u kasnijoj fazi bolja taktika grananja u širinu, jer daje bolja odsecanja.

3.1.3 Metoda grananja i odsecanja

Metoda grananja i odsecanja (***Brunch and Cut method***) je veoma uspešna pri rešavanju problema kombinatorne optimizacije. Ovom metodom se teorijski dobijaju optimalna rešenja. Ona kombinuje prednosti metode grananja i ograničavanja i *Gomory-eve* metode odsecanja. Najpoznatiji brunch-and-cut algoritmi se koriste za rešavanje problema p-medijane ([Ces03], [Ces05]), hab lokacijskih problema [Lab03], problema rutiranja (The Location-Routing Problem, LRP), [Bel10], problema trgovačkog putnika (Traveling salesman problem, TSP). Dva rada koja opisuju primene ove metode na problem trgovačkog putnika su radovi *Grotschel-a i Holland-a,* [Gro91] i *Padberg-a i Rinaldi-a* , [Pad91]. Kod teških problema metoda grananja i odsecanja se koristi sa heuristikom ili metaheuristikom da bi se obezbedila dobra (po mogućnosti) optimalna rešenja i da bi se ukazalo koliko daleko je rešenje. Razmotrimo ovu metodu na jednostavnom primeru:

Problem celobrojnog programiranja:

$$\min \; z = -6x_1 - 5x_2 \qquad (1)$$

$$\begin{aligned} 3x_1 + x_2 &\le 11 \\ -x_1 + 2x_2 &\le 5 \\ x_1, x_2 &\ge 0 \end{aligned}$$

je prikazan na Slici 9. Označene su celobrojne dopustive tačke. LP relaksacija je dobijena ignorisanjem uslova celobrojnosti. Brunch and cut algoritam prvo rešava LP relaksaciju davajući tačke $\left(2\frac{3}{7}, 3\frac{5}{7}\right)$ sa vrednošću $-33\frac{1}{7}$. Sada imamo izbor da LP relaksaciju ili poboljšamo tako što ćemo dodati ograničenje $x_1 + x_2 \le 5$, ili da problem podelimo na dva, razdvajanjem promenljivih. Ako se algoritam razdvoji na x_1, dobijaju se dva nova problema:

$$\min \; z = -6x_1 - 5x_2 \qquad (2)$$

$$\begin{aligned} 3x_1 + x_2 &\le 11 \\ -x_1 + 2x_2 &\le 5 \\ \mathbf{x_1} &\mathbf{\ge 3} \\ x_1, x_2 &\ge 0 \end{aligned}$$

$$\min \; z = -6x_1 - 5x_2 \qquad (3)$$

$$\begin{aligned} 3x_1 + x_2 &\le 11 \\ -x_1 + 2x_2 &\le 5 \\ \mathbf{x_2} &\mathbf{\le 2} \\ x_1, x_2 &\ge 0 \end{aligned}$$

Najbolje rešenje početnog problema biće bolje rešenje od ova dva potproblema. Rešenje za LP relaksaciju za problem (2) je (3, 2) sa vrednošću -28. Ovo rešenje je celobrojno, tako da rešava problem (2) i postaje aktuelno najbolje dopustivo rešenje. LP relaksacija za problem (3) daje optimalno rešenje (2, 3.5) sa vrednošću -29.5 koje nije celobrojno tako da ne rešava problem (3) i mora se dalje razmatrati. Predpostavimo da algoritam koristi metodu odsecanja (*cutting plane)* i dodaje nejednačinu $2x_1 + x_2 \le 7$ problemu (*3*). Ovo je važeća (ispravna) nejednačina, i zadovoljena je za svaku celobrojnu dopustivu tačku iz problema (3). Dalje, nejednakost nije zadovoljena za (2, 3.5) tako da je to odsecajuća ravan. Rezultirajući problem je:

min $z = -6x_1 - 5x_2$ (4)

$3x_1 + x_2 \leq 11$

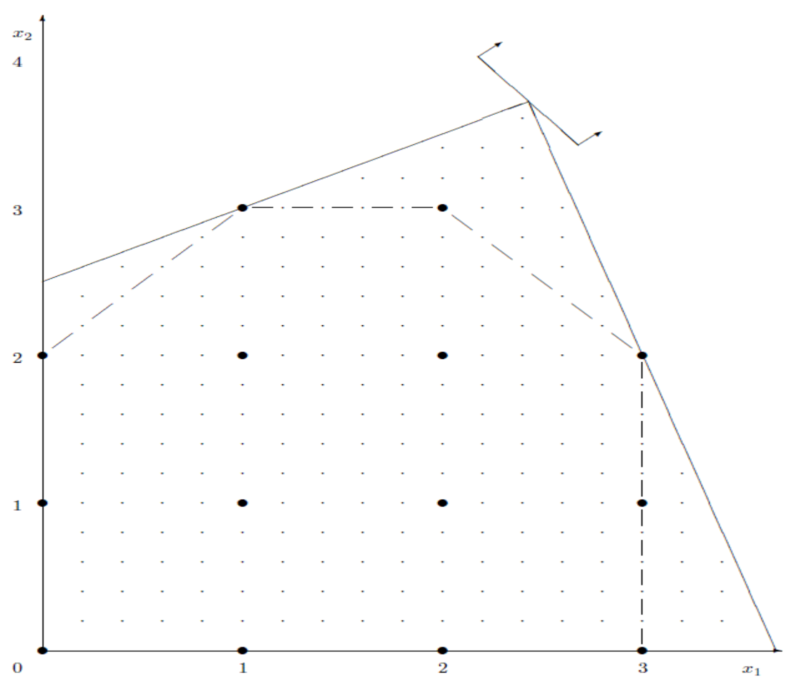
$-x_1 + 2x_2 \leq 5$

$x_2 \leq 2$

$\mathbf{2x_1 + x_2 \leq 7}$

$x_1, x_2 \geq 0$

LP relaksacija problema (4) ima optimalno resenje (1.8, 3.4) sa vrednošću -27.8. Primetimo da je optimalna vrednost ove modifikovane relaksacije veća nego vrednost aktuelnog rešenja. Vrednost optimalnog celobrojnog rešenja za drugi potproblem mora da bude veliko barem koliko i vrednost relaksacije. Aktuelno rešenje (3,2) je bolje nego bilo koje dopustivo celobrojno rešenje problema (4), odnosno rešenje je početnog problema.



Slika 9. Dvodimenzionalni problem celobrojnog programiranja

3.1.3 Metoda implicitnog prebrojavanja

Metoda implicitnog prebrojavanja predstavlja specijalan slučaj metode grananja i ograničavanja. Primenjuje se na problem oblika:

$min\ f(x_1, x_2, \ldots, x_n)$
$g_i(x_1, x_2, \ldots, x_n) \leq 0,\ \ i=1,2,\ldots,m,$
$x_j \in K_j,\ \ j=1,2,\ldots,n$

pri čemu su K_j, $j=1,2,\ldots,n$ konačni skupovi. Grananje se vrši fiksiranjem mogućih vrednosti izabrane promenljive x_j iz skupa K_j, tako da svakom problemu sa spiska odgovara delimično rešenje, kod kojeg neke promenljive x_{j1}, x_{j2}, ..., x_{jk} imaju fiksiranu vrednost, dok su ostale slobodne (slobodne promenljive obeležavamo sa *). Umesto relaksacija, za donju ocenu optimalne vrednosti problema sa spiska, koristi se neposredna ocena.

Posmatrajmo sledeći primer:

$$
\begin{aligned}
&min\ \ f = -4x_1 - 3x_2 - 3x_3 - 2x_4 \\
&g_1 = -2x_1 - 2x_2 + x_3 + 4x_4 \leq 5 \\
&g_2 = 2x_1 + 3x_2 - 2x_3 + 5x_4 \leq 7 \\
&g_3 = x_1 + 4x_2 + 5x_3 - x_4 \leq 6 \\
&x_1, x_2, x_3, x_4 \in \{0,1\}
\end{aligned}
$$

Donja ocena za funkciju cilja je $\underline{f} = -4\cdot1 - 3\cdot1 - 3\cdot1 - 2\cdot1 = -12$

Testiramo da li je ovo rešenje dopustivo tako što računamo donje ocene za donja ograničenja $\underline{g}_i$ funkcija g_i

$$
\begin{aligned}
&\underline{g}_1 \geq 2\cdot0 - 2\cdot1 + 1\cdot0 + 4\cdot0 = -2, -2 \leq 5 \\
&\underline{g}_2 \geq 2\cdot0 + 3\cdot0 - 2\cdot1 + 5\cdot0 = -2, -2 \leq 7 \\
&\underline{g}_3 \geq 1\cdot0 + 4\cdot0 + 5\cdot0 - 1\cdot1 = -1, -1 \leq 6
\end{aligned}
$$

Početni problem je prošao test nedopustivosti (najmanje moguće vrednosti početnih ograničenja se mogu zadovoljiti za neke vrednosti x_i) i možemo ga podeliti na dva potproblema:

1. potproblem: za $x_1 = 1$

2. potproblem: za $x_1 = 0$

U zavisnosti od vrednosti promenljive x_1 funkcija cilja ima vrednosti $\underline{f} = -12$ i $\underline{f} = -8$

Ponovo proveravamo dopustivost rešenja:

$$\underline{g}_1 \geq -2 \cdot 1 - 2 \cdot 1 + 1 \cdot 0 + 4 \cdot 0 = 0, 0 \leq 5$$

$$\underline{g}_2 \geq 2 \cdot 1 + 3 \cdot 0 - 2 \cdot 1 + 5 \cdot 0 = 0, 0 \leq 7$$

$$\underline{g}_3 \geq 1 \cdot 1 + 4 \cdot 0 + 5 \cdot 0 - 1 \cdot 1 = 0, 0 \leq 6$$

Rešenje za $x_1 = 1$ je dopustivo. Delimo ovaj problem na dva potproblema:

1.1. za $x_2 = 1$

1.2. za $x_2 = 0$

U zavisnosti od vrednosti promenljive x_2 funkcija cilja uzima vrednosti $\underline{f} = -12$ i $\underline{f} = -9$

Ponovo proveravamo dopustivost rešenja:

$$\underline{g}_1 \geq 2 \cdot 1 - 2 \cdot 1 + 1 \cdot 0 + 4 \cdot 0 = 0, 0 \leq 5$$

$$\underline{g}_2 \geq 2 \cdot 1 + 3 \cdot 1 - 2 \cdot 1 + 5 \cdot 0 = 3, 3 \leq 7$$

$$\underline{g}_3 \geq 1 \cdot 1 + 4 \cdot 1 + 5 \cdot 0 - 1 \cdot 1 = 4, 4 \leq 6$$

Rešenje za $x_1 = 1$, $x_2 = 1$je dopustivo. Delimo ovaj problem na dva nova potproblema:

1.1.1. za $x_3 = 1$

1.1.2. za $x_3 = 0$

U zavisnosti od vrednosti promenljivex_3, funkcija cilja uzima vrednosti $\underline{f} = -12$ i $\underline{f} = -9$.

Proveravamo dopustivost rešenja za promenljivu x_3 :

$$\underline{g}_1 \geq 2 \cdot 1 - 2 \cdot 1 + 1 \cdot 1 + 4 \cdot 0 = 1, 1 \leq 5$$

$$\underline{g}_2 \geq 2 \cdot 1 + 3 \cdot 1 - 2 \cdot 1 + 5 \cdot 0 = 3, 3 \leq 7$$

$$\underline{g}_3 \geq 1 \cdot 1 + 4 \cdot 1 + 5 \cdot 1 - 1 \cdot 1 = 9, 9 \geq 6!!!$$

Za $x_1 = 1$, $x_2 = 1, x_3 = 1$ rešenje nije dopustivo, pa ga odbacujemo i proveravamo da li je skup $x_1 = 1$, $x_2 = 1$, $x_3 = 0$ dopustiv:

$$\underline{g}_1 \geq 2\cdot 1 - 2\cdot 1 + 1\cdot 0 + 4\cdot 0 = 0, 0 \leq 5$$

$$\underline{g}_2 \geq 2\cdot 1 + 3\cdot 1 - 2\cdot 0 + 5\cdot 0 = 5, 5 \leq 7$$

$$\underline{g}_3 \geq 1\cdot 1 + 4\cdot 1 + 5\cdot 0 - 1\cdot 1 = 4, 4 \leq 6$$

Kako je skup dopustiv , delimo problem na dva nova slučaja:

1.1.2.1 za $x_4 = 1$

1.1.2.2 za $x_4 = 0$

U zavisnosti od vrednosti promenljive x_4 funkcija cilja uzima vrednosti $\underline{f} = -9$ i $\underline{f} = -7$

Proveravamo dopustivost rešenja za promenljivu x_4:

$$\underline{g}_1 \geq 2\cdot 1 - 2\cdot 1 + 1\cdot 0 + 4\cdot 1 = 4, 4 \leq 5$$

$$\underline{g}_2 \geq 2\cdot 1 + 3\cdot 1 - 2\cdot 0 + 5\cdot 1 = 10, 10 \geq 7!!!$$

$$\underline{g}_3 \geq 1\cdot 1 + 4\cdot 1 + 5\cdot 0 - 1\cdot 1 = 4, 4 \leq 6$$

Rešenje za $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 1$ nije dopustivo. Skidamo ga sa spiska i proveravamo da li je skup $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$ dopustiv.

$$\underline{g}_1 \geq 2\cdot 1 - 2\cdot 1 + 1\cdot 0 + 4\cdot 0 = 0, 0 \leq 5$$

$$\underline{g}_2 \geq 2\cdot 1 + 3\cdot 1 - 2\cdot 0 + 5\cdot 0 = 5, 5 \leq 7$$

$$\underline{g}_3 \geq 1\cdot 1 + 4\cdot 1 + 5\cdot 0 - 1\cdot 0 = 5, 5 \leq 6$$

Ovaj skup je dopustiv i daje vrednost funkcije cilja $\underline{f} = -7$.

Nastavljamo traženje optimalnog rešenja, i nakon određnog broja koraka, zaključujemo da su dva optimalna rešenja $(1,1,0,0)$ i $(1,0,1,0)$, a optimalna vrednost funkcije cilja je -7.

3.2. Heuristike za rešavanje lokacijskih problema

Veliki broj standardnih problema optimizacije su NP-teški i NP-kompletni problemi. Definicija i osobine NP-teških i NP-kompletnih problema dati su u poglavlju 4.1. Poznati algoritmi za rešavanje NP-teških problema su eksponencijalne i faktorijelne složenosti, zbog čega za rešavanje ovih problema koristimo metode čija je tačnost eksperimentalno potvrđena. Neke od tih metoda su heuristički algoritmi.

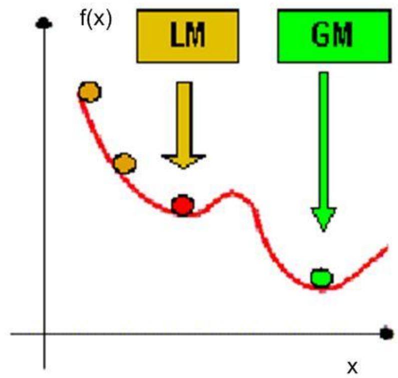
Reč heuristika potiče od starogrčke reči *heuriskein* što znači ’’naći’’ ili ’’otkriti’’. Heuristički algoritmi su nastali eksperimentisanjem u cilju dobijanja zadovoljavajućeg rešenja. Mana algoritama je što nisu jednoznačno određeni. Heuristike su namenjene za rešavanje posebnih vrsta problema optimizacije, poštujući pri tome svojstva i specifičnosti ovih problema. Takođe, kod problema kod kojih postoje egzaktni algoritmi polinomske složenosti, rešavanje problema heuristikama često ne vodi ka zadovoljavajućem rešenju.

Postoji čitav niz osobina koje bi trebalo da neka heuristika zadovolji. Potrebno je da bude jednostavna, da bi se brzo izvršavala i da ne menja drastično svoje ponašanje za male promene paramatra problema. Potrebno je da je korisnik u mogućnosti da utiče na proces dobijanja rešenaja, tj. da ima mogućnost interaktivnog rada. Takođe, potrebno je da poseduje mogućnost generisanja većeg broja dobrih rešenja, da bi korisnik mogao izabrati najprihvatljivije od njih.

Princip rada heurističkih algoritama je sledeći. Prvo je potrebno definisati prostor rešenja (koji se često naziva i *pretraživački prostor*) za dati problem. Zatim se odabere početno rešenje, pa se nakon toga prostor rešenja pretražuje u potrazi za što boljim rešenjima. Razlika među heurističkim algoritmima je u tome na koji način realizuju to pretraživanje.

3.2.1. Lokalno pretraživanje

Lokalno pretraživanje (***Local search, LS***) je heuristički pristup na kome se baziraju kompleksne metaheuristike zasnovane na pojedinačnom rešenju. Često se primenjuje u kontinualnoj i diskretnoj optimizaciji. Lokalno pretraživanje nastoji pronaći optimalno rešenje smanjujući prostor pretraživanja. U lokalnom pretraživanju za početno rešenje bira se proizvoljna tačka iz prostora dopustivih rešenja. Svakom elementu x iz prostora dopustivih rešenja X pridružuje se neki podskup od X koji se naziva “okolina“ od x, a njeni članovi su „susedi“ od x. U svakoj iteraciji, pretražuje se okolina trenutnog rešenja i u njoj nalazi, prema nekom kriterijumu, sused koji predstavlja sledeće rešenje. Ukoliko se sused ne može pronaći, pretraživanje staje i za aproksimaciju optimalnog rešenja uzima se ono za koje je vrednost funkcije cilja najmanja. Osnovni nedostatak lokalnog pretraživanja je što se zaustavlja pri nailasku prvog lokalnog minimuma koji ne mora biti globalni minimum, što dosta zavisi od izbora početnog rešenja (Slika 10.).



Slika 10. Lokalni i globalni minimum

Opšta procedura metode lokalnog pretraživanja može se opisati na sledeći način:

Inicijalizacija: *Izabrati početno rešenje* $x_1 \in X$;

$x^* = x_1, f^* = f(x_1)$

Iterativni korak: $n = 1, 2, \ldots$

U okolini $N(x_n)$ *trenutnog rešenja naći sledeće rešenje* x_{n+1} *, prema nekom kriterijumu izbora.*

Ako je $f(x_{n+1}) < f^*$ *, tada* $x^* = x_{n+1}$ *i* $f^* = f(x_{n+1})$

Kraj: *Ako kriterijum izbora nije zadovoljen ni za jednog suseda iz okoline* $N(x_n)$*, ili je zadovoljen neki drugi kriterijum zaustavljanja, metoda staje, a* x^* *se uzima za aproksimaciju optimalnog rešenja.*

Efikasnost ove metode zavisi od strukture okolina koje se koriste, kao i od kriterijuma koji se koristi u svakoj iteraciji za izbor sledećeg rešenja.Pri definisanju okoline, poželjno je da budu zadovoljeni sledeći uslovi:

- Okolina svake tačke treba da je simetrična
- Okolina nije ni suviše velika, ni suviše mala
- Polazeći od proizvoljne tačke prostora X, nizom uzastopnih pomaka možemo doći do bilo koje druge tačke ovog prostora
- Pomak treba da obezbedi što jednostavnije i brže generisanje susednih rešenja

3.2.2 Simulirano kaljenje

Simulirano kaljenje (Simulated annealing, SA) je opšta heuristika za rešavanje problema kombinatorne optimizacije, a ime je dobila zbog analogije sa kaljenjem metala iz čije se simulacije i razvila. Metodu su nezavisno jedan od drugog opisali *Scott Kirkpatrick, C. Daniel Gelatt i Mario P. Vecchi* 1983. godine, [Kir83] i *Vlado Černý* 1985 godine, [Čer85]. Metoda pokušava pronaći globalni optimum, ili mu se bar približiti, u zadatom prostoru konfiguracija.

Ova heuristika se u opštem obliku može prikazati na sledeći način:

Inicijalizacija: *Izabrati početno rešenje* $x_1 \in X$

$x^* = x_1, f^* = f(x_1)$

Iterativni korak: $n = 1, 2, \ldots$

Naći na slučajan način x u okolini $N(x_n)$ *trenutnog rešenja* x_n.

Ako je $f(x) \leq f(x_n)$, *tada* $x_{n+1} = x$

Ako je $f(x) < f^*$, *tada* $x^* = x$, $f^* = f(x)$

Ako je $f(x) > f(x_n)$, izabrati slučajan broj p uniformno raspodeljen na intervalu [0,1]

Ako je $p \leq p_n$, *tada* $x_{n+1} = x$

Ako je $p > p_n$, *tada* $x_{n+1} = x$

Kraj: *Ako je zadovoljen kriterijum zaustavljanja niz iterativnih koraka se zaustavlja, a* x^* *se uzima za aproksimaciju optimalnog rešenja.*

Vrednost p_n predstavlja verovatnoću prihvatanja pogoršanja funkcije cilja u iteraciji *n*. U osnovnj verziji simuliranog kaljenja ova verovatnoća je jednaka:

$$p_n = e^{(-\frac{f(x) - f(x_n)}{t_n})}$$

po analogiji sa procesom kaljenja, gde je t_n, vrednost temperature u iteraciji n. Temperatura je pozitivan kontrolni parametar, čije su vrednosti zadate nizom $t_1, t_2, \ldots$, takvim da je $t_1 \geq t_2 \geq \ldots$ i $\lim_{n \to \infty} t_n = 0$. Ovakav niz se naziva šema hlađenja i definiše način i brzinu smanjivanja temperature tokom iteracija.

3.2.3 Pohlepni algoritam

Jedan od najjednostavnijih heurističkih algoritama je pohlepni ili konstruktvini algoritam (***Greedy Algorithm***). Pohlepni algoritam je znatno brži od ostalih algoritama, ali ne mora uvek naći optimalno rešenje, jer nas u svakom trenutku usmerava na trenutno najbolje rešenje, ne uzimajući u obzir da nas to ne mora voditi ka globalnom optimumu. Traženje optimalnog rešenja može biti eksponencijalne složenosti, a pohlepni algoritam smanjujući prostor pretraživanja, smanjuje i samu složenost. Ovakav pristup vodi do rešenja koje sigurno predstavlja donju granicu globalnog maksimuma, ukoliko se traži maksimum, odnosno gornju granicu globalnog minimuma, ukoliko se traži minimum. Kako se pohlepni algoritam brzo izvodi i daje granicu optimalnog rešenja, on nam omogućava da primenom nekih drugih algoritama smanjimo prostor pretrage rešenja. Dodatna prednost pohlepne strategije je da, čak i kada ne pronađe optimalno rešenje, često vodi na novu strategiju razvoja algoritma koja rezultira efikasnijim rešenjem ili tehnikom koja brzo pronalazi dobru aproksimaciju rešenja.

Princip pohlepne metode je jednostavan: pronađi najbolje trenutno rešenje i idi za njim. Postoje mnogi algoritmi koji su zasnovani na pohlepnom pristupu, pa kada govorimo o pohlepnom algoritmu, ustvari govorimo o široj klasi algoritama.

Primer 1. Trgovac koji koristi pohlepni algoritam

Jednostavan problem na kome možemo ilustrovati primenu pohlepnog algoritma je vraćanje kusura prilikom kupovine. „Pohlepnost“ pristupa se ogleda u tome što se prvo bira novčić koji ima najveću vrednost:

Trgovac treba da vrati mušteriji kusur od 62 dinara, a na raspolaganju su mu novčanice od 50, 20, 10, 5 i 1 dinara. Trgovac će vratiti jednu novčanicu od 50 dinara, jednu od 10 dinara i 2 novčanice od dinar. Ovaj algoritam vraća tačan iznos uz najkraću moguću listu novčanica.

Algoritam: izabere se najveća novčanica koja ne prelazi ukupnu sumu (50 dinara), stavlja se na listu za vraćanje, oduzme se vraćena vrednost od ukupne kako bi se izračunao ostatak za vraćanje (12 dinara), zatim se bira sledeća novčanica koja ne prelazi ostatak koji treba vratiti (10 dinara), dodaje se na listu za vraćanje i računa novi ostatak za vraćanje (2 dinara) itd, sve dok se ne vrati tačan iznos.

Ako trgovac treba da vrati 15 dinara, a ima novčanice od 11 , 5 i 1 dinar, on će pohlepnim algoritmom vratiti ostatak jednom novčanicom od 11 dinara i pomoću četiri novčanice od 1 dinar,iako bi mu bilo brže vratiti ostatak sa tri novčanice od 5 dinara.

Dakle, strategija pohlepnog pristupa je u primeru sa 62 dinara dovela do najefikasnijeg rešenja problema vraćanja novca i to zahvaljujući specijalnim svojstvima raspoloživih novčanica, dok u primeru sa 15 dinara pohlepni algoritam nije optimalan za dati skup novčanica.

U mnogim problemima pohlepni algoritmi mogu biti pogrešan izbor strategije. Na primer u partiji šaha, pohlepan algoritam će uvek odigrati onaj potez koji izgleda najbolje u datom trenutku, ne obazirući se na njegove posledice. Ukoliko najveću vrednost za igrača u datoj situaciji ima potez u kome on uzima protivničkog lovca, on će to i učiniti, iako taj potez daje protivniku mogućnost da matira igrača. Međutim, postoje problemi u kojima je dokazano da pohlepni algoritam daje optimalno rešenje. Jedan od tih problema je i problem rasporeda zadataka. Pitanje je kako rasporediti zadatke, a da prosečno vreme završavanja zadatka bude minimalno.

3.3 Metaheuristički algoritmi za rešavanje lokacijskih problema

Za razliku od klasičnih heurističkih metoda ideja metaheurističkih metoda je da se one uz odeđene modifikacije mogu primenjivati na širu klasu problema kombinatorne optimizacije. Metaheurističke metode pretažuju skup dopustivih rešenja u cilju nalaženja što boljeg rešenja. U ovom pretraživanju dopušteno je kretanje ka lošijem rešenju od trenutnog, proširivanje skupa na kome tražimo rešenje nedopustivim elementima ili traženje rešenja kombinovanjem postojećih.

Problema kombinatorne optimizacije se može definisati kao:

$$\min f(x), \quad x \in X,$$

gde prostor dopustivih rešenja X ima konačno mnogo elemenata. Konkretna implementacija metaheuristika za rešavanje problema kombinatorne optimizacije zavisi od prirode svakog pojednačnog problema. Metaheuristike koje se najčešće koriste su: tabu pretraživanje, metoda promenljivih okolina, genetski algoritmi, kao i brojni metaheuristički algoritmi inspirisani prirodom kao što su mravlje kolonije, neuronske mreže, metoda kolonija pčela. Pregled metaheuristika sa primenama dat je u [Glo03].

3.3.1. Tabu pretraživanje

Tabu pretraživanje (***Tabu search, TS***) se bazira na principu lokalnog prtraživanja, i koristi kao svoju najvažniju komponentu tzv. adaptivnu memoriju, tj. pamćenje nekih podataka o prethodnim fazama procesa pretraživanja, koji utiču na izbor sledećih tačaka u ovom procesu.

Koncept tabu pretraživanje je prvi predložio *Fred Glover* u svojim radovima ([Glo86], [Glo90]). Tabu pretraživanje koristi memorijske strukture i kada pronađe potencijalno rešenje, označi ga kao tabu (zabranjeno) kako se više ne bi vraćao na njega. Sva rešenja označena kao tabu pamte se u takozvanoj tabu listi. Tabu vreme je broj iteracije za vreme kojih je stavka sadržana u tabu listi zabranjena.

Algoritam tabu retraživanja radi na sledeći način: u svakoj iteraciji algoritma računamo promene vrednosti funkcije cilja za sve moguće poteze iz okoline trenutnog rešenja i biramo najbolji potez. Moguće je da čak i najbolji potez ne poboljšava funkciju cilja. Na primer ukoliko se trenutno nalazimo u lokalnom optimumu, svi potezi pogoršavaju funkciju cilja. Očigledno je da se kod ovakve optimizacijske procedure mogu pojaviti ciklična kruženja poteza, odnosno, može doći do situacije da se određeni sled poteza stalno ponavlja (npr. ukoliko smo iteraciju ranije bili u lokalnom minimumu i nakon što je algoritam odabrao potez koji ga je najmanje udaljio od lokalnog minimuma, sledeći najbolji potez nas ponovo vraća u isti lokalni minimum). Kako se to ne bi dogodilo, uvodimo listu zabranjenih poteza koja definiše poteze koji nisu dopušteni. Ta lista je najčešće realizovana na principu FIFO (*first- in- first- out*). Kako se koji potez prihvati, stavlja se u tabu listu. Taj potez kroz određeni broj iteracija nije dopušteno izvoditi, čime će se sprečiti ciklično ponavljanje poteza. Naravno, potrebno je da tabu lista bude odgovarajuće veličine. Međutim, budući da se u listu stavljaju samo potezi, a ne i celokupni kontekst u kojem je taj potez zabranjen, moguće je da zabranjivanje tog poteza, nakon što se algoritam nađe u nekom drugom rešenju, spreči istraživanje moguće zanimljivih rešenja kojim bi taj potez vodio. Stoga se uvodi i aspiracijska funkcija pomoću koje se poništava tabu status nekog poteza i omogućava da se on prihvati iako se nalazi u listi zabranjenih poteza. Klasična implementacija aspiracijske funkcije poništava potezu tabu status, ukoliko taj potez vodi do rešenja koje je bolje od dosada najboljeg pronađenog rešenja. Predloženo da aspiracijska funkcija ne poklanja pažnju vrednosti funkcije cilja, već da vrši ulogu diversifikacije rešenja. Smatramo da potez zadovoljava aspiracijsku funkciju ukoliko dodeljuje obe jedinke na lokacije koje one nisu okupirale unutar zadnjih *t* iteracija, gde je *t* kontrolni parameter algoritma.

Kod tabu pretraživanja, veličina liste zabranjenih poteza je od velike važnosti. Naime, ukoliko je lista premala, može doći do cikličnog ponavljanja poteza, dok u slučaju da je ona prevelika, zanimljivi potezi mogu biti odbačeni, što povećava broj iteracija za dobijanje kvalitetnijih rešenja.

Veličina tabu liste zavisi od veličine problema koji se rešava i s njom stoji u proporcionalnom odnosu (što je veći problem, potrebna je i veća tabu lista). Međutim, pojavljivanje cikličnih ponavljanja poteza nije obavezno u direktnoj vezi sa veličinom tabu liste. *Taillard* je u svom radu [Tai90] predložio da veličina tabu liste ne bude konstantna već da se slučajno menja unutar nekog intervala sa određenom frekvencijom.

Ako uvedemo sledeću notaciju:

x= *trenutno rešenje*
x*= *najbolje pronađeno rešenje*
f*= *vrednost funcije u tački x**
N(x)= *okolina od x*
TL= *tabu lista*
TV= *tabu vreme,*

algoritam tabu pretraživanja se može predstaviti na sledeći način:

Inicijalizacija:

odaberi početno rešenje x_0

postavi: x= x_0, *x*=*x_0, *f*= f(*x_0*), TL=*$\emptyset$*;*
Pretraga:
dok (uslov_zaustavljanja nije zadovoljen) radi
{
odaberi najbolje dozvoljeno rešenje x' iz N(x);
postavi x= x';
ako je f(x) < f, onda postavi f* = f(x), x* = x;*
stavi x u TL na TV iteracija;
}
vrati rešenje x;*

Najčešći uslovi zaustavljanja su:
• nakon fiksnog broja iteracija (ili fiksnog vremena procesora),
• nakon određenog broja iteracija bez napretka u minimiziranju funkcije,
• nakon što funkcija dostigne neku unapred definisanu vrednost.
Kao uslov zaustavljanja mogu se koristiti i sve kombinacije gore navedenih uslova.

3.3.2 Genetski algoritmi

Osnove genetskih algoritama (**Genetic algorithms, GA**) prvi je postavio *Holland* u svojoj knjizi [Hll75] 1975. godine. *Holland* je GA razvio radi razvoja sistema veštačke inteligencije koji oponašaju modele prilagođavanja i radi proučavanja procesa prilagođavanja kod prirodnih sistema. Genetski algoritam je metoda optimizacije koja imitira proces genetske evolucije. Analogija evolucije kao prirodnog procesa i genetskog algoritma kao metode optimizacije, ogleda se u procesu selekcije i genetskim operatorima.

Postoji veliki broj radova o genetskim algoritmima. Neki od njih su: [Djo75], [Gol89], [Yur94], [Mic96], [Mit96], [Bä00a] i [Bä00b]. O genetskim algoritmima se može naći i u domaćoj literaturi: [Cve96], [Fil98], [Kra00], [Toš04].

Genetski algoritam radi nad populacijom jedinki, najčeće veličine između 10 i 200. Početna populacija je obično slučajno generisana, što obezbeđuje raznovrsnost genetskog materijala. Svaka jedinka je predstavljena genetskim kodom nad određenom konačnom azbukom, najčešće binarnom. Svakoj jedinki populacije se dodeljuje funkcija prilagođenosti (fitness function) koja određuje kvalitet date jedinke tj. odgovarajućeg rešenja. Zatim se primenjuje operator selekcije – uzimaju se bolje jedinke, po nekom postupku odabira i formira nova generacija. Nakon toga neki članovi populacije su podvrgnuti uticajima genetskih operatora: ukrštanja i mutacije.

Selekcijom se favorizuju natprosečno prilagođene jedinke i njihovi natprosečno prilagođeni geni. Slabije prilgođene jedinke i geni imaju slabiju šansu za reprodukciju i veća je verovatnoća da izumru.

Operator ukrštanja služi za razmenu genetskog materijala između jedinki. Ukrštanjem se vrši rekombinacija gena i time razmenjuje genetski material. Na ovaj način se omogućuje da i slabije prilagođene jedinke u kombinaciji sa nekim dobro prilagođenim genima proizvedu dobro prilagođene jedinke.

Višestrukom primenom selekcije i ukrštanja moguće je gubljenje genetskog materijala, što sprečava mutacija. Ona vrši slučajnu promenu određenog gena, sa datom malom verovatnoćom p_m . Na taj način se vraća izgubljeni genetski materijal u populaciju čime se sprečava preuranjena konvergencije GA u lokalnom ekstremu.

Prost genetski algoritam predstavlja osnovnu varijantu genetskog algoritma i sastoji se od proste rulet selekcije, jednopozicionog ukrštanja i proste mutacije. Najčešći problem u primeni prostog genetskog algoritma je preuranjena konvergencija.

Šematski prikaz osnovne varijante genetskog algoritma je:

Unošenje_Ulaznih_Podataka();
Generisanje_Početne_Populacije();
while not *Kriterijum_Zaustavljanja_GA() do*

for *i=1 to Npop do*
obj[i] = Funkcija_Cilja(i);
endfor
Funkcija_Prilagođenosti();
Selekcija();
Ukrštanje();
Mutacija();

endwhile
Štampanje_Izlaznih_Podataka();

Šematski zapis genetskog algoritma

Genetski algoritmi su u osnovi stohastičke metode pretrage dopustivog prostora rešenja, tako da mogu raditi beskonačno dugo, ukoliko im ne nametnemo kriterijum zaustavljanja.
Postoji nekoliko kriterijuma zaustavljanja GA:

- maksimalni broj generacija,
- sličnost jedinki u populaciji,
- najbolja jedinka je ponovljena maksimalni broj puta,
- algoritam je dostigao optimalno rešenje (ukoliko je ono unapred poznato),
- dokazana optimalnost najbolje jedinke (ukoliko je to moguće),
- ograničeno vreme izvršavanja GA,
- prekid od strane korisnika itd.

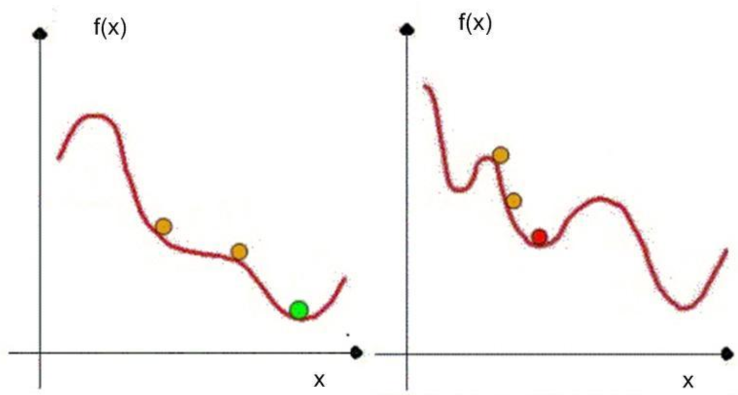
Svaki od navedenih kriterijuma ima dobre i loše aspekte, tako da se u praksi najbolje pokazalo njihovo kombinovanje, jer se tako smanjuje mogućnost loše procene prekida GA.

Iako je teorijski moguće primeniti koncept GA na svaki lokacijski problem, GA nije podjednako uspešan za sve probleme. Kod problema sa mnogo ograničenja mogu nastati teškoće u pronalaženju adekvatnog načina kodiranja. Ispostavilo se da je genetski algoritam najefikasniji za rešavanje problema sa malo ograničenja i relativno složenom funkcijom cilja [Ree93]. Pomoću GA uspešno se rešavaju: problem p-medijane [Dre03], [Cor04], problem p-centra [Mih03], hab lokacijski problemi [AbH98], [AbH99], [Ern04a], prost lokacijski problem [Kra00], problem dizajniranja mreže [Kra02], problem selekcije indeksa [Kra03].

3.3.3 Metoda promenljivih okolina

Metodu promenljivih okolina **(Variable Neighborhood Search, VNS)** predložili su *Mladenović* i *Hansen*, u radovima ([Mla95], [Mla97]). Metoda promenljivih okolina se zasniva na lokalnom pretraživanju, pri čemu se u svakoj iteraciji može vršiti prestruktuiranje okoline trenutnog rešenja.

Osnovna ideja metode je sistematska promena okolina unutar lokalnog pretraživanja (Slika 11.). Neophodno je uvesti više okolina, bilo promenom metrike u odnosu na koju se defniše okolina, bilo povećavanjem rastojanja u odnosu na istu metriku. U trenutku kada se dođe do nekog lokalnog optimuma, vrši se slučajno pomeranje u tekućoj okolini do nekog rešenja (koje može biti i loše). Zatim se iz tog rešenja lokalnim pretraživanjem pokuša pronalaženje nekog drugog lokalnog ekstremuma. Ovim pomeranjem (do rešenja koje se može nalaziti relativno daleko od trenutnog) postiže se sistematično pretraživanje prostora rešenja i sprečava konvergencija metode ka lošijem lokalnom ekstremumu. Takođe, u slučajevima kada pomeranje nije dovelo do boljeg rešenja, zadržavanje u trenutno najboljem rešenju smanjuje mogućnost nepotrebnog širenja pretraživanja na nove oblasti prostora dopustivih rešenja. Metoda se primenjuje za rešavanje problema p-medijane ([Han97], [Cra03], [Cra04]), prostog lokacijskog problema [Han07], problema veštačke inteligencije. Kriterijumi zaustavljanja ove metode su slični kao i kriterijumi zaustavljanja u slučaju tabu pretraživanja. Više o ovovj metodi može se naći u ([Mla95, Mla97, Han99, Han01, Dav06, Kov08,]).



Slika 11. Pretraživanje po različitim okolinama

VNS metaheuristika zasnovana je na sledećim činjenicama:

- lokalni minimum u odnosu na jednu okolinu ne mora biti i lokalni minimum u odnosu na neku drugu okolinu;
- globalni minimum je lokalni minimum u odnosu na sve okoline;
- za većinu problema lokalni minimumi u odnosu na razne okoline su međusobno bliski.

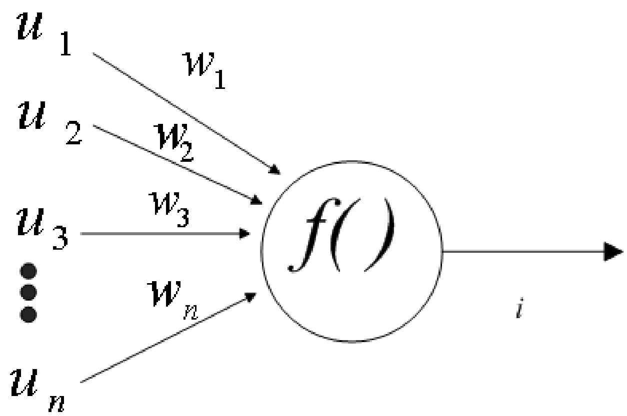
Ove tri činjenice mogu se iskoristiti na tri različita načina: deterministički, stohastički ili kombinovano (videti [Dav06]). Ukoliko iskoristimo determnističku varijantu dobijamo metodu promenljivog spusta (***Variable Neighborhood Descent, VND).*** Ona se sastoji u tome da se izabere k_{max} okolina, N_k, k=1, 2, ... , k_{max}, odredi početno rešenje x i startuje procedura lokalnog pretraživanja u odnosu na svaku od okolina, počev od izabranog rešenja x. Ukoliko je $k_{max} = 1$ reč je o običnom lokalnom pretraživanju. Ukoliko iskoristimo stohastičku definiciju metode koraci su jednostavniji zbog toga što ne postoji proces lokalnog pretraživanja. Metoda se sastoji u sistematskoj promeni okolina i izboru jednog slučajnog rešenja u svakoj od okolina. Koraci odlučivanja bazirani su na tom jednom slučajnom rešenju. Metoda koja se na ovaj način dobija naziva se redukovana metoda promenljivih okolina (***Reduced Variable Neighborhood Search, RVNS*).** Ova metoda je korisna kod primera velikih dimenzija, jer se izbegava složena i dugotrajna metoda lokalnog pretraživanja. Kombinacijom prethodna dva principa, tj. sistematskom (determinističkom) promenom okolina, slučajnim izborom početnog rešenja u tekućoj okolini i primenom metode lokalnog pretraživanja počev od slučajnog rešenja dobija se osnovna metoda promenljivih okolina (***Basic Variable Neighborhood Search, VNS***). Ovo je najrasprostranjenija varijanta metode promenljivih okolina jer obezbeđuje više preduslova za dobijanje kvalitetnijih konačnih rešenja.

3.3.4 Neuronske mreže

Neuronska mreža (***Neural network***) je sistem sastavljen od više jednostavnih procesora (jedinica, neurona) [Fan90]. Svaki od njih ima lokalnu memoriju u kojoj pamti podatke koje obrađuje. Ove jedinice su povezane komunikacionim kanalima (vezama). Podaci koji se ovim kanalima razmenjuju su obično numerički. Jedinice obrađuju samo svoje lokalne podatke i ulaze koje primaju preko konekcije. Ograničenja lokalnih operatora se mogu otkloniti tokom treninga.

Postoje dve kategorije neuronskih mreža: veštačke i biološke neuronske mreže. Predstavnik bioloških neuronskih mreža je nervni sistem živih bića. Veštačke neuronske mreže su po strukturi, funkciji i obradi informacija slične biološkim neuronskim mrežama, ali se radi o veštačkim tvorevinama. Neuronska mreža u računarskim naukama predstavlja veoma povezanu mrežu elemenata koji obrađuju podatke. One su sposobne da daju rešenja problema koji se tradicionalnim pristupima teško rešavaju, kao što su govor i prepoznavanje oblika. Jedna od važnijih osobina neuronskih mreža je njihova sposobnost da uče na ograničenom skupu primera.

Veštački neuroni imaju jednostavnu strukturu i slične funkcije kao i biološki neuroni. Telo neurona se naziva čvor ili jedinica (Slika 12).

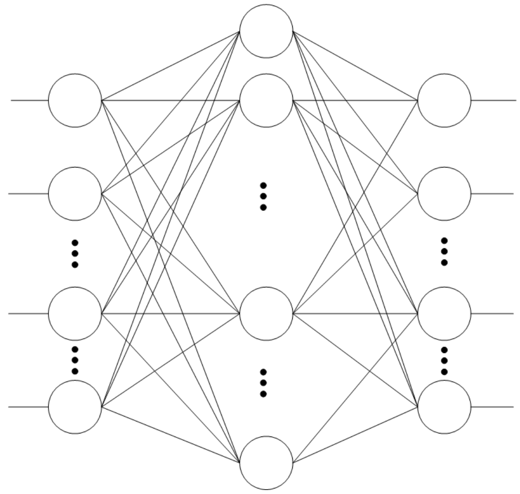


Slika 12. Telo neurona

$u_{1..n}$– ulazni podaci
$w_{1..n}$ – težinski koeficijenti
f( ) – aktivaciona funkcija
i – izlazni podatak

Neuronsku mrežu čine: arhitektura (topologija) mreže, odnosno šema vezivanja neurona, prenosna funkcija neurona, zakoni učenja.

Arhitekturu veštačke neuronske mreže karakteriše specifično uređenje i povezivanje neurona u obliku mreže (Slika 13), [Mac03]. Po arhitekturi, neuronske mreže se mogu razlikovati prema broju neuronskih slojeva. Obično svaki sloj prima ulaze iz prethodnog sloja, a svoje izlaze šalje narednom sloju. Prvi sloj se naziva “ulazni”, poslednji je “izlazni”, ostali slojevi se obično nazivaju “skrivenim” slojevima. Jedna od najčešćih arhitektura neuronskih mreža je mreža sa tri sloja. Prvi sloj (ulazni) je jedini sloj koji prima signale iz okruženja. Prvi sloj prenosi signale sledećem sloju (skriveni sloj) koji obrađuje ove podatke i izdvaja osobine i šeme iz primljenih signala. Podaci koji se smatraju važnim se upućuju izlaznom sloju, poslednjem sloju mreže. Na izlazima neurona trećeg sloja se dobijaju konačni rezultati obrade. Složenije neuronske mreže mogu imati više skrivenih slojeva, povratne petlje i elemente za odlaganje vremena, koji su dizajnirani da omoguće što efikasnije odvajanje važnih osobina ili šema sa ulaznog nivoa.



Slika 13. Neuronska mreža

Postoji veliki broj različitih realizacija neuronskih mreža, a samim tim postoji i mnogo podela. Neuronske mreže možemo klasifikovati prema:

- broju slojeva (jednoslojne, višeslojne)
- vrsti veza između neurona (slojevite, potpuno povezane, celularne)
- vrsti obučavanja neuronskih mreža (nadgledano, delimično nadgledano, nenadgledano)
- prema smeru prostiranja informacija (nepovratne (feedforward), povratne(feedback))
- prema vrsti podataka (analogne, diskretne)

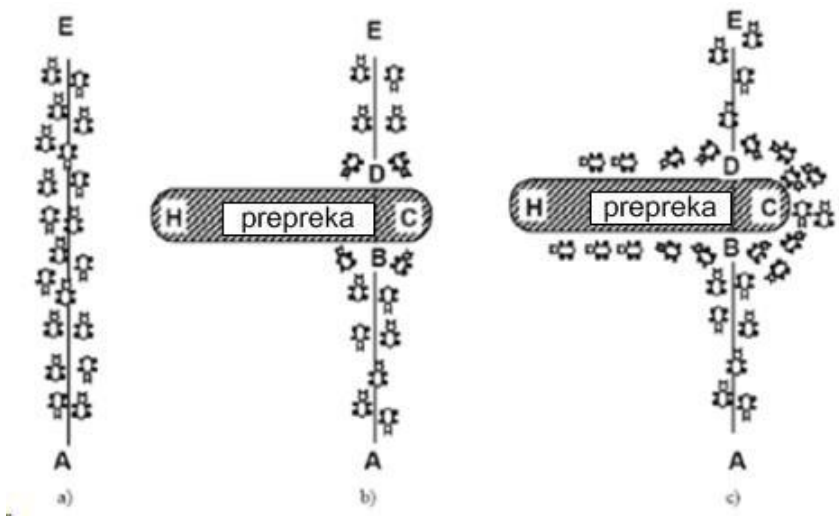
Primenu neuronskih mreža je moguće podeliti na tri karakteristične oblasti, [Ili99] :

- procesiranje senzorskih informacija (prepoznavanje oblika, rukopisa, govora)
- analiza podataka (analiza medicinskih testova, električnih kola, analiziranje podataka pri pirolizi i spektroskopiji)
- kontrola upravljanja (upravljanje sistemima, upravljanje proizvodnim procesima, upravljanje robotima)

Neuronske mreže mogu biti kako od koristi pri akustičnom modeliranju ulaznog govornog signala i njegovih osobina, tako i pri klasifkaciji ili analizi strukture govora i drugim mogućim oblastima govora. Konkretnu primenu neuronskih mreža predstavljaju dva programa: program za prepoznavanje ćiriličnih slova OCR i program za obučavanje neuronskih mreža ANN.

3.3.5 Mravlji algoritmi

Mravlji sistemi (***Ant systems)*** predstavljaju algoritme za rešavanje kompleksnih problema kombinatorne optimizacije koji su nastali praćenjem kolektivnih ponašanja životinja poput mrava. Ove algoritme prvi je opisao *Marco Dorigo*, u radu [Dor96]. Način na koji mravi uspevaju u nalaženju najkraćeg mogućeg puta između izvora hrane i njihovog gnezda, podseća na probem trgovačkog putnika. Lučenjem feromona u procesu prikupljanja hrane, jedinke daju informacije ostalim članovima kolonije. Sprovedeni su eksperimenti u kojima je mravlje gnezdo povezano sa izvorom hrane pomoću dva mosta čija se dužina razlikuje. Utvrđeno je da će se nakon izvesnog vremena većina mrava kretati kraćim putem. Razlog tome je što mrav odlučuje kojim će se putem kretati u zavisnosti od količine feromona koju oseća na određenom putu. Kako se feromoni više osećaju na kraćem putu, zbog veće gustine, većina mrava će odlučiti da se kreće upravo kraćim putem. Pri formiranju heurističkog algoritma, koji je inspirisan ovom idejom, veštački mravi pretražuju dopustivi prostor i formiraju rute, a određeni mehanizmi (poput isparavanja feromona) sprečavaju prebrzu konvergenciju ovakvih procesa. Eksperimenti sa preprekom (Slika 14) pokazuju da mravi uspevaju veoma brzo da otkriju kraći put. Pretpostavlja se da je izboru svakog puta od strane mrava pridružena određena verovatnoća koja je utoliko veća ukoliko je jači feromon duž tog puta. Na početku ekspereimenta sa preprekom mravi se kreću duž najkraćeg puta između staništa i izvora hrane. Potom se mravima postavlja prepreka na njihovom putu što uslovljava da mravi moraju da skrenu levo ili desno. Otprilike polovina mrava skreće levo i otprilike polovina mrava skreće desno. Mravi koji su skrenuli desno stižu za kraće vreme da uzmu hranu i da se vrate u stanište, da ponovo odu po hranu, da se ponovo vrate, itd. Samim tim, količina feromona duž desnog puta i verovatnoća izbora ovog puta se povećava. Za relativno kratko vreme mravi su u stanju da otkriju kraći put.



Slika 14.

Najznačajniji veštački sistemi razvijeni na inteligenciji roja su mravlji sistem i optimizacija kolonijom mrava. Optimizacija kolonijom mrava predstavlja novu heuristiku razvijenu sa ciljem da se pomoću nje rešavaju složeni problemi kombinatorne optimizacije. Osnovna razlika između mravljeg sistema i optimizacije kolonijom mrava ogleda se u načinu na koji se izračunava verovatnoća da će mrav *k* koji je lociran u čvoru *i* da izabere čvor *j*.

Model koji je *Debebourg* predložio pretpostavlja da je količina feromona na putanji proporcionalna broju mrava koji su prošli tom putanjom. Neka je A_n i B_n broj mrava koji se kreću duž putanja A i B, respektivno, od prvih n mrava koji su napustili mravinjak. Verovatnoća da će (n+1)-i mrav izabrati putanju A je data sa:

$$p_{n+1} = \frac{(K + A_N)^V}{(K + A_N)^V + (K + B_N)^V}$$

Parametar K definiše mogućnost izbora neobeležene staze (za veće vrednosti potrebna je jača koncentracija feromona da izbor ne bi bio slučajan), dok vrednost V određuje stepen nelinearnosti ovog izbora.

Lučić i Teodorović, [Luč02] su modifikovali optimizaciju kolonijom mrava i predlozili **fazi mravlji sistem** kombinujući optimizaciju kolonijom mrava sa fazi logikom. Osnovnu modifikaciju predstavlja način izračunavanja verovatnoća odlaska mrava u sledeći čvor. Oni su svakom potencijalnom izboru novog čvora dodelili određene koristi koje ostvaruje mrav. Veštački mrav koji treba da donese odluku uzima u obzir vidljivost i količine feromona deponovane po pojedinim granama. Fazi mravlji sistem je zasnovan na ideji da mrav doživljava određena rastojanja kao kratka, srednja, ili duga. Takođe, deponovanu kolicinu feromona mrav vidi kao malu, srednju ili veliku.

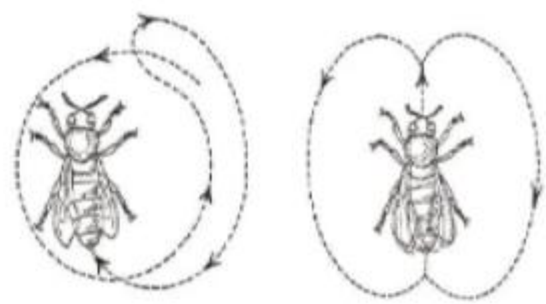
Pretpostavlja se da veštački mrav *k* koji se nalazi u čvoru *i* računa "korist" u^k_{ij} koje može ostvariti ukoliko poseti cvor *j*. U zavisnosti od percepiranog rastojanja i percepirane količine deponovanog feromona mrav ostvaruje veće ili manje koristi ukoliko poseti određeni čvor. Mrav doživljava ove koristi kao male, srednje, velike.

Algoritam aproksimativnog rezonovanja za izračunavanje koristi koje ima mrav pri izboru sledeće grane, traži da rastojanje bude što kraće i količina feromona što veća.

Algoritam aproksimativnog rezonovanja može da zameni izraze za izračunavanje verovatnoće izbora. Na ovaj način je moguće izračunavati koristi i u slučajevima kada su neke od ulaznih veličina poznate samo aproksimativno. Lokalna pravila za promenu feromona i globalna pravila za promenu feromona su u slučaju fazi mravljeg sistema ista kao i u slučaju optimizacije kolonijom mrava.

3.3.6 Optimizacije metodom "rojeva pčela"

Pčelinji ples je jedno od najintrigantnijih društvenih ponašanja kod insekata. On predstavlja formu koju pčele radilice koriste da bi regrutovale druge pčele iz roja da ih prate do mesta sa resursima. Kada se pčela vrati u košnicu sa nektarom koji je dovoljno hranljiv da bi ostale pčele otišle do njegovog izvora, ona izvede karakterističan ples na zidu košnice kako bi podelila informaciju o lokaciji tog izvora sa ostalim pčelama. Ovaj ples sadrži dve bitne informacije: o pravcu i udaljenosti izvora hrane. Pčela izvodi kružni ples kojim samo signalizira udaljenost hrane, ukoliko se izvor nalazi blizu košnice(na udaljenosti manjoj od 50m). Ukoliko je udaljenost velika (preko 150m), pčela izvodi ples u obliku osmice koji u sebi sadrži i informaciju o udaljenosti hrane. Ugao pod kojim se pčela kreće u toku uspravnog dela osmice predstavlja ugao pod kojim se nalazi hrana u odnosu na Sunce. Za srednje udaljenosti, pčela koristi kombinacije ova dva plesa da bi prenela informacije.



Slika15. Pčelinji plesovi: kružni ples(levo), ples u obliku spljoštene osmice(desno)

Ovako opisan način na koji pčele prikupljaju hranu može biti upotrebljen u algoritmima za optimizaciju (pronalaženje najboljeg izvora hrane), distribuiranu podelu zadataka (regrutacija pčela u zavisnosti od kvaliteta izvora hrane), itd.

Osnovni algoritam za optimizaciju koji se zasniva na ponašanju pčela (**Bees Algorithm, BA**) oslanja se na koncept centralnog plesnog podijuma, kako bi se odredila najpogodnija lokacija, dok direktna komunikacija između jedinki roja ne postoji. Regrutacija može biti deterministička, na osnovu funkcije prilagođenosti svake lokacije, ili stohastička, kada vrednosti ove funkcije određuju verovatnoću da pčela bude regrutovana. Algoritam se izvršava u iteracijama i zaustavlja se kada se pronađe rešenje u okviru neke margine greške ili kada se dostigne maksimalan broj iteracija.

Drugi značajan algoritam za optimizaciju (**Artificial Bee Colony, ABC**) podržava tehniku centalnog podijuma, ali predlaže da pčele imaju različite uloge u okviru roja. Postoje specijalne "pčele izviđači" koje se šalju na slučajnu putanju u potrazi za hranom kada se postojeći izvor hrane istroši i "pčele sakupljači" koje se regrutuju na osnovu funkcije prilagođenosti izvora hrane.

Nijedna metaheuristička metoda ne garantuje optimalnost dobijenog rešenja. Čak je izuzetno teško predvideti njegovo odstupanje od optimalnog rešenja. Kako ne postoji univerzalna metoda koja daje najbolja rešenja za sve probleme, često se vrše kombinacije raznih metoda i dobijaju hibridni algoritmi.

Najčešće se kombinuju genetski algoritmi i tabu pretraživanje. Kombinacija genetskog algoritma i lokalnog pretraživanja poznata je pod nazivom memetski algoritam (**Memetic Algorithm**, **MA**) koji je naziv dobio po kulturološkom genu *„mem"* koji prenosi informacije iz jednog uma u drugi. Za razliku od gena, *„mem"* se ne prenosi razmnožavanjem. Memetski algoritmi imaju svojstvo da dobro pretražuju i iskorišćavaju prostor rešenja, ali su naročito osetljivi na odnos poremećaja informacije i njenog očuvanja prilikom mutacije: da bi algoritam izašao iz lokalnog optimuma i da bi se postigla stabilnost algoritma, potrebno je taj odnos uravnotežiti. Takođe moguće je kombinovati metaheurističe metode sa egzaktnim metodama. Hibridizacija GA sa egaktnom metodom Branch-and-Cut-and-Price je data u [Lju04].

The Greedy Randomized Adaptive Search Procedure, **GRASP** je kombinacija pohlepne heuristike i lokalnog pretraživanja. Rešenje se sastavlja korak po korak uzimajući u svakoj iteraciji, slučajnim odabirom, komponentu rešenja iz ograničene liste kandidata. Ograničena lista kandidata se sastoji od α najboljih elemenata rangiranih prema vrednosti pohlepne funkcije, tj. prema kvalitetu rešenja koje bi mogli postići. Ukoliko je $\alpha=1$, dobijamo osnovni pohlepni algoritam, a ukoliko je α jednak broju komponenti dobijamo slučajno pretraživanje. Algoritam je osetljiv na parametar α i najčešće se algoritam menja iz iteracije u iteraciju algoritma (adaptivna promene parametara).

GRASP metoda predstavlja iterativan proces gde se svaka iteracija sastoji od dve faze: faze izgradnje rešenja i faze lokalnog pretraživanja. Najbolje nađeno rešenje se zadržava kao rezultat. U prvoj fazi, konstruišemo rešenje iterativnim putem, element po element. U svakoj iteraciji izgradnje, izbor sledećeg elementa se određuje tako da se slože svi elementi u listi kandidata s obzirom na pohlepnu funkciju koja meri profit pri izboru svakog elementa.

Heuristika je adaptivna, jer se profit povezan sa izborom svakog elementa ponovno računa prilikom svake iteracije u fazi izgradnje kako bi se uzele u obzir promene nastale izborom prethodnog elementa. Slučajnost je uvedena tako što se među najboljim kandidatima slučajno bira element, koji najčešće nije i najbolji. Rešenja generisana u fazi izgradnje uglavnom nisu lokalno optimalna, s obzirom na definciju okoline. Zato je potrebno primeniti lokalno pretraživanje kako bi se poboljšalo izgrađeno rešenje. Iako ovako pretraživanje može zahtevati eksponencijalno vreme ukoliko se kreće iz proizvoljne početne tačke, empirijski se efikasnost pretraživanja značajno poboljšava kako se početno rešenje poboljšava.

Korišćenjem prilagođenih struktura podataka i pažljivom implementacijom, može se stvoriti efikasna faza izgradnje koja daje dobra početna rešenja za lokalno pretraživanje.

4. Lokacijski problemi na mrežama

Iako se nekoliko objavljenih radova vezanih za lociranje proizvodnog pogona i skladišta baziranih na mreži, pojavilo u istraživačkoj literaturi 50-tih godina, ozbiljna pažnja usmerena ka mrežnim lokacijskim problemima pojavila se tek sredinom 60-tih godina. Hakimi je u radovima ([Hak64],[Hak65]), istraživao minimalno težinsko rastojanje lociranja p objekata na n čvorova mreže i taj problem je nazvao problem p-medijane. Iako nije predložio metod koji rešava problem p-medijane, dokazao je postojanje najmanje jednog optimalnog rešenja koje svih p objekata locira pojedinačno na čvorove mreže. Njegov doprinos je bio sličan *Dantzig-ovom* u linearnom programiranju, po tome što je smanjio skup optimalnih rešenja sa potencijalno beskonačnog do konačnog skupa. Ovaj rezultat se obično naziva "Hakimijeva teorema" ili "osobina čvorova".

Slično modelima optimizacije za rešavanje lokacijskih problema u ravni, koriste se različiti modeli za rešavanje problema lokacije na mrežama, koje je dosta proučavao *Klincewicz* [Kli91]. U ovim modelima se koristi pojam rastojanja (prosečno vreme putovanja, prosečni transportni troškovi) između dva čvora mreže, koje je po definiciji dužina najkraćeg puta između posmatranih čvorova. Pri računanju dužine puta, u nesimetričnim mrežama, važno je koji je početni, a koji krajnji čvor puta. Prema tome, za jedan par čvorova mogu postojati dva rastojanja.

Mrežni lokacijski problemi predstavljaju kombinaciju diskretnih i kontinualnih problema. Mogu imati diskretna rešenja koja se nalaze u čvorovima grafa ili neprekidna rešenja koja se nalaze na granama grafa. Najpoznatiji diskretni mrežni lokacijski problemi su: problemi maksimalnog pokrivanja (maximal covering problems), problemi *p*-centra (*p*-center problems), *p*-medijane (*p*-median problems), problemi fiksnih troškova (fixed charge problem), lokacije habova (hub location problems), maksimalne sume (maxisum problem) itd.

Čvorovima se pripisuju brojevi koji se koriste kao težinski koeficijenti pri računanju težinskih rastojanja. Kao kriterijum može da se koristi težinsko rastojanje od novog objekta do njemu najudaljenijeg čvora ili zbir težinskih rastojanja od novog objekta do čvorova mreže. U prvom slučaju radi se o problemima tipa *min-max* koji se u teoriji grafova nazivaju problemi određivanja centra grafa. U drugom slučaju u pitanju su problemi tipa *min-sum*, odnosno problemi određivanja medijane grafa.

4.1 NP–teški problem

Teorija NP-kompletnih problema je utemeljena u radovima [Cook71] i [Krp72]. Tokom 70-tih godina je dokazano da više od 300 problema pripada toj klasi.

Algoritam je *polinomske složenosti* po vremenu izvršavanja, ako je vremenske složenosti najviše $O(n^k)$, za neku konstantu *k*.

Problem pripada klasi složenosti **P**, i nazivamo ga problemom *polinomske složenosti*, ako ga je moguće rešiti u opštem slučaju, nekim od poznatih algoritama polinomske složenosti.

Problem pripada klasi **NP**, i nazivamo ga problemom *nedeterminističke polinomske složenosti*, ako se rešenje datog problema može verifikovati algoritmom polinomske složenosti. Preciznije, za unapred dato rešenje se utvrđuje da li su ispunjeni svi uslovi problema. Pri tome je potrebno da on bude zadat kao problem odlučivanja, da bi odgovor bio i formalno (matematički) korektan.

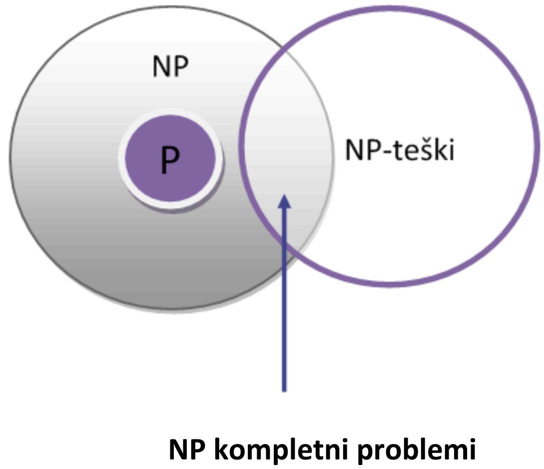
Ključna razlika između klasa *P* i *NP* je u tome što se problemi iz klase P mogu efektivno rešiti, dok se u slučaju problema iz klase NP može samo verifikovati rešenje u polinomskom vremenu izvršavanja. Takođe, jasno je da svaki problem iz klase *P* pripada i klasi *NP,* odnosno da je $P \subset NP$, ali postavlja se pitanje da li je *P* prava potklasa klase *NP*.

Postoje problemi koji ne pripadaju ni klasi P ni klasi NP. Kod takvih problema možemo da konstruišemo eksponencijalni algritam i da pokažemo da se on ne može izbeći.

Problem odlučivanja nazivamo *NP-kompletnim* ako:

- pripada klasi NP;
- svi ostali NP problemi se mogu algoritmom polinomske složenosti svesti na dati problem.

Problem koji nije problem odlučivanja, a čije rešavanje nije jednostavnije od rešavanja nekog NP-kompletnog problema, naziva se NP-težak [Gar79, Gib85]. Problem je NP-težak ako i samo ako postoji ekvivalentan problem odlučivosti koji je NP-kompletan. NP-teški problemi predstavljaju kombinatorne probleme za čije rešavanje nisu poznati algoritmi s vremenom računanja koje zavisi od veličine problema, tj. čija se složenost može izraziti polinomnom funkcijom (npr. linearnom, kvadratnom, kubnom,...). Iako nikad nije dokazano, pretpostavlja se da odnos skupova NP, P, NP-teških i NP-kompletnih problema izgleda kao na Slici 16.



Slika 16. Grafički prikaz odnosa skupova NP, P, NP-teških i NP-kompletnih problema

4.1 Prost lokacijski problem

Prost lokacijski problem (***Simple Plant Location Problem, SPLP*****),** takođe poznat kao problem lokacije nekapacitativnog objekta (***Uncapacitated facility location problem ,UFLP*****),** je osnovni predstavnik ove klase problema i mnogi lokacijski problemi se mogu formulisati pomoću njega. Sličan je problemu p-medijane, osim toga što su mnogi objekti koji treba da se lociraju u sastavu problema. Zadatak je minimizovati sumu fiksnih troškova i troškova transporta, a pri tome ispoštovati zahteve skupa klijenata.

Iako je *Balinski* [Bal65] prvi formulisao problem, napustio je formulu, jer je njegov kolega *Gomory* demonstrirao mali kontra primer. *Morris* [Mor78] , *Ra Velle i Swain*, [RaV70] su pokazali kroz eksperimente da su problem p-medijane i prost lokacijski problem "celobrojno orjentisani", to jest njihove relaksacije linearnog programiranja teže celobrojnim ili skoro celobrojnim rešenjima koja zahtevaju nekoliko grananja u proceduri grananja i ograničavanja. *Problem lokacije proizvodnog pogona sa ograničenim kapacitetima* je indentičan prostom lokacijskom problemu sa dodatkom ograničavanja kapaciteta na objekte. Dodatna ograničenja uništavaju osobinu kojom su svi zahtevi korisnika zadovoljeni iz jednog objekta.

Krarup [Krr83] je dokazao da prost lokacijski problem (SPLP) pripada klasi NP kompletnih problema.

4.1.1 Formulacija problema

Neka je dat skup $I=\{1,\ldots,m\}$ potencijalnih lokacija snabdevača i skup $J=\{1,\ldots,n\}$ korisnika. Označimo sa f_i fiksne troškove postavljanja snabdevača na potencijalnu lokaciju $i\in I$. Neka je b_j količina robe koju zahteva svaki korisnik $j\in J$, uz transportne troškove snabdevanja sa *i-te* lokacije c_{ij}(po jedinici robe). Potrebno je rasporediti snabdevače na neke od tih lokacija tako da ukupni troškovi budu minimalni. Pri tome u razmatranje ulaze fiksni troškovi postavljanja snabdevača i promenljivi troškovi transporta do svakog korisnika (videti [Bej93]).

Problem se matematički formuliše na sledeći način:

$$\min\left(\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}+\sum_{i=1}^{m}f_i y_i\right)$$

Uz ograničenja:

$\sum_{i=1}^{m}x_{ij}=1$, *za svakog korisnika* $j\in J$

$0\le x_{ij}\le y_i$, *za svakog korisnika* $j\in J$ *i snabdevača* $i\in I$

$y_i\in\{0,1\}$, *za svakog snabdevača* $i\in I$

$x_{ij}\ge 0$, *za svakog korisnika* $j\in J$ *i snabdevača* $i\in I$

Promenljive x_{ij} označavaju količinu robe koju snabdevač *i* isporučuje korisniku *j*. Binarna promenljiva y_i uzima vrednost 1 ako je na potencijalnoj lokaciji *i* uspostavljen snabdevač ili 0 u suprotnom.

Datom problemu odgovara sledeći realni zadatak opisan u [Kra00]: U nekom naselju postoji *n* stambenih zgrada, i *m* potencijalnih lokacija za prodavnice (ili trafike, domove zdravlja, benzinske pumpe, kafane, servisne službe,...). Potrebno je u datom naselju optimalno rasporediti prodavnice na neke od potencijalnih lokacija, tako da budu blizu stanovnicima naselja, ali uzimajući u obzir i troškove zakupa poslovnog prostora prodavnica. Najčešće su dati uslovi suprotstavljeni, pa lokacije u centru naselja koje su relativno najbliže svim stanovnicima, imaju visoke troškove zakupa poslovnog prostora, a lokacije na periferiji sa niskim troškovima zakupa, zahtevaju velike transportne troškove.

4.1.2 Metode za rešavanje SPLP-a

Iako je prost lokacijski problem NP težak, u specijalnim slučajevima ga je moguće rešiti u polinomskom vremenu izvršavanja. Egzaktne metode za rešavanje SPLP-a su detaljno prikazane u radovima [Aqe90], [Ryu92] i [DSi96].

Najefikasniji algoritam za rešavanje SPLP-a dugo vremena je bio *Dualoc* algoritam, čiji je autor *Erlenkotter*, [Erl78]. Osnova ovog algoritma je dualna formulacija pridruženog problema linearnog programiranja u skraćenoj formi, koji doprinosi jednostavnoj implementaciji dualne metode penjanja i odgovarajuće dualne metode poravnanja. Ukoliko rešenje dobijeno primenom te dve metode nije optimalno, primenjuje se metoda grananja i ograničavanja. Na kraju izvršavanja algoritma dobija se optimalno rešenje. Postoje dve varijante *Dualoc* algoritma:

- Puna implementacija koja koristi metodu grananja-i-ograničavanja (BnB) za dobijanje optimalnog rešenja;
- Redukovana metoda koja sadrži samo metodu penjanja (dual ascent) i metodu poravnanja (dual adjustment). Kako redukovana metoda ne sadrži metodu grananja i ograničavanja vreme izvršavanja joj je vrlo kratko, ali daje samo suboptimalno rešenje.

Neke od uspešnih metoda za rešavanje SPLP-a su: genetski algoritmi [Kra99], [Kra01], [Fil00], metoda promenljivih okolina (Variable neighborhood search) [Han07], tabu pretraživanje (tabu search) [Sun06], simulirano kaljenje (simulated annealing) [Yig03], Lagranževa relaksacija (Lagrangean relaxation) [Cor06].

U praksi se pokazalo da najbolje rezultate za rešavanje prostog lokacijskog problema daju dualne metode ([Erl78], [Sim89], [Hlm95]). Dualne metode rešavaju dati problem korišćenjem metode grananja i ograničavanja i u opštem slučaju su eksponencijalne složenosti. Vrlo brzo rešavaju SPLP instance manje dimenzije, dok u slučaju SPLP instanci velikih dimenzija, brzo dolaze do optimalnog rešenja, ako je broj suboptimalnih rešenja relativno mali. Samo u slučaju problema vrlo velikih dimenzija, sa malim brojem neperspektivnih potencijalnih lokacija i velikim brojem suboptimalnih rešenja, vreme izvršavanja je ekstremno veliko.

U radu [Tch88] prikazan je pokušaj hibridizacije dualnih metoda i Add-heuristike. Pokazano je da je vreme izvršavanja hibridizacije obično vrlo kratko, iako se dobijaju suboptimalna rešenja koja mogu biti lošijeg kvaliteta kod instanci veće dimenzije. Još jedna varijanta hibridizacije Add heuristike i genetskog algoritma opisana je u radu [Kra98a].

Rešavanje problema (SPLP) pomoću genetskih algoritama detaljno je opisano u [Kra00]. Izabrano je binarno kodiranje potencijalnih lokacija snabdevača. Svaki bit u genetskom kodu jedinke, čija je vrednost 1 označava da je na datoj lokaciji postavljen snabdevač, a vrednost 0 da nije. Genetski kod jedinke je alociran kao niz 32-bitnih reči, radi bržeg izvršavanja, pri primeni genetskih operatora.

Za m=5, genetski kod jedinke **01011|000000010000000010001110101**
1. - 5. 6. - 32.
označava da su na lokacijama 2, 4 i 5 postavljeni snabdevači dok na lokacijama 1 i 3 nisu postavljeni snbdevači.Bitovi 6-32 se ignorišu.

Iz genetskog koda nalazi se niz indikatora y_i (i=1, 2, ... , m) koji prikazuju da li je na odgovarajućoj potencijalnoj lokaciji postavljen snabdevač ili nije. Pošto za snabdevača na određenoj lokaciji ne postoji ograničenje u kapacitetu lokacije i pošto je poznato na kojim su sve lokacijama postavljeni snabdevači a na kojima nisu (niz y_i), svaki korisnik može da izabere sebi najbližeg snabdevača. U slučaju kada su snabdevači fiksirani, minimizacijom troškova za svakog korisnika ujedno se dobija i minimum ukupnih troškova. Za rešavanje datog problema primenjena je selekcija bazirana na rangu (rank-based selection) sa linearnim smanjenjem rangova.

Pri izvršavanju GA implementacije, uniformno ukrštanje je dalo nešto bolje rezultate u odnosu na ostale šeme ukrštanja, iako razlike u performansama, pri poređenju sa jednopozicionim, dvopozicionim i višepozicionim ukrštanjem nisu bile velike. Prosta mutacija je implementirana pomoću Gausove (normalne) raspodele, radi bržeg izvršavanja. Nivo mutacije zavisi od veličine problema.

Za razliku od ukrštanja, gde se variranjem parametara performanse menjaju relativno malo, pri variranju vrednosti nivoa mutacije, se dobijaju drastično drugačiji rezultati, naročito za SPLP instance veće dimenzije.

Genetski algoritmi predstavljaju robustan način za rešavanje SPLP-a, pa su zbog toga razlike izvršavanja na malim i velikim SPLP instancama mnogo manje nego kod dualnih metoda. Uspešno se rešavaju čak i instance sa velikim brojem približnih rešenja bliskih optimalnom. Rekombinacijom jedinki sa vrednostima bliskim optimalnom rešenju, vrlo lako se poboljšavaju date vrednosti i postupak konvergira brzo prema optimalnom rešenju, [Fil06].

4.2 Problem p-medijane

U slučaju problema p medijane potrebno je locirati jedan ili više objekata na mreži, tako da se minimizira prosečno rastojanje (ili prosečno vreme putovanja, prosečni transportni troškovi) od objekta do korisnika ili od korisnika do objekta. Problem p-medijane je naročito značajan za transportne sisteme, s obzirom da se ova grupa problema javlja prilikom projektovanja različitih distributivnih sistema.

Hakimi je u [Hak64] pokazao da postoji najmanje jedan skup p snabdevača u čvorovima mreže, što znači da p optimalnih lokacija objekata u mreži moraju da se nalaze isključivo u čvorovima mreže. Potrebno je ispitati samo lokacije koje se nalaze u čvorovima. Koristeći

Hakimijevu teoremu, *RaVell i Swain* su 1970. godine prvi formulisali problem p-medijane (**p-median problem**), [RaV70]. *Kariv i Hakimi* su pokazali u [Kar79] da problem p-medijane na mrežama pripada grupi NP-teških problema.

Problemi p-medijane su često analizirani u rešavanju problema i na makro i na mikro nivou. Primer na makro nivou je donošenje odluke gde locirati skladište koje prima robu iz više fabrika sa poznatim lokacijama ili koje bi trebalo da distribuira robu mreži maloprodajnih objekata. Na mikro nivou, tipična primena problema je pronalaženje optimalne lokacije za novu mašinu u okviru pogona fabrike.

4.2.1 Formulacija problema

Neka je:

G=(N,A) – transportna mreža

N- skup čvorova mreže

a_i - potražnja u čvoru j

d_{ij} - rastojanje između čvora i i čvora j

p – ukupan broj objekata koje treba locirati

$$x_{ij} = \begin{cases} 1, \text{ukoliko i dobije uslugu u objektu lociranom u čvoru j} \\ 0, \end{cases}$$

Problem p medijana se defineše na sledeći način:

$$\min F= \sum_{i=1}^{n} \sum_{j=1}^{n} a_i d_{ij} x_{ij} , \qquad (1)$$

Pri ograničenjima:

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i=1,...n \qquad (2)$$

$$\sum_{j=1}^{n} x_{jj} = p \qquad (3)$$

$$x_{jj} \geq x_{ij}, \quad i,j=1,2,...n, \; i \neq j \qquad (4)$$

$$x_{ij} \in \{0,1\}, \; i,j=1,2,...n \qquad (5)$$

Funkcija cilja (1) minimizira sumu rastojanja između objekata i korisnika. Ograničenje (2) se odnosi na činjenicu da je svaki klijent (svaki čvor) uslužen od strane samo jednog objekta. Ograničenjem (3) se ukazuje da na mreži treba da postoji ukupno p objekata. Svaki klijent lociran u nekom od objekata dobija uslugu iz tog objekta. Ovo je iskazano kroz ograničenje (4).

4.2.2 Algoritam za određivanje jedne medijane mreže (p=1)

Jednostavan algoritam kojim se generiše skup dopustivih rešenja i određuje lokacija jedne medijane u slučaju neorijentisane mreže predložio je Hakimi, [Hak65]. Algoritam se sastoji iz sledećih algoritamskih koraka:

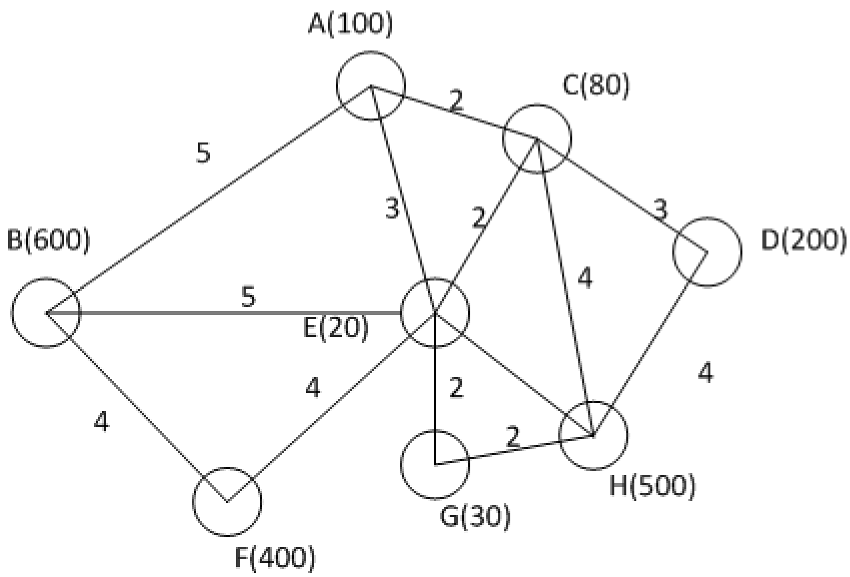
KORAK 1: Izračunati dužine najkraćih puteva d_{ij} između svih parova čvorova (i,j) mreže G i prikazati ih u matrici najkraćih rastojanja D. Čvorovi *i* predstavljaju moguće lokacije za medijanu, a čvorovi *j* predstavljaju lokacije klijenata koji zahtevaju uslugu.

KORAK 2: Pomnožiti j-tu kolonu matrice najkraćih puteva sa brojem zahteva za uslugom a_j iz čvora j. Element $a_j d_{ij}$ matrice D= $[a_j d_{ij}]$ predstavlja "rastojanje" koje prelaze korisnici iz čvora *j* koji dobijaju uslugu u čvoru *i*.

KORAK 3: Izvršiti sumiranje duž svake vrste *i* matrice D. Izraz $\sum_{i=1}^{n} a_j d_{ij}$ predstavlja ukupno "rastojanje" koje prelaze korisnici u slučaju kada je objekat lociran u čvoru *i*.

KORAK 4: Čvor čijoj vrsti odgovara najmanje ukupno "rastojanje" koje prelaze korisnici predstavlja lokaciju za medijanu.

Primer: Čvorovi transportne mreže su označeni respektivno sa A, B, C, ..., H (Slika 17.). U čvorovima se nalaze korisnici usluga. Dnevni zahtevi za uslugom dati su u zagradama pored čvorova. Takođe su označene i dužine svih grana u mreži. Treba rešiti sledeći problem: gde locirati objekat u kome se pruža određena usluga, tako da ukupno rastojanje koje prelaze korisnici usluga do objekta bude minimalno.



Slika 17.

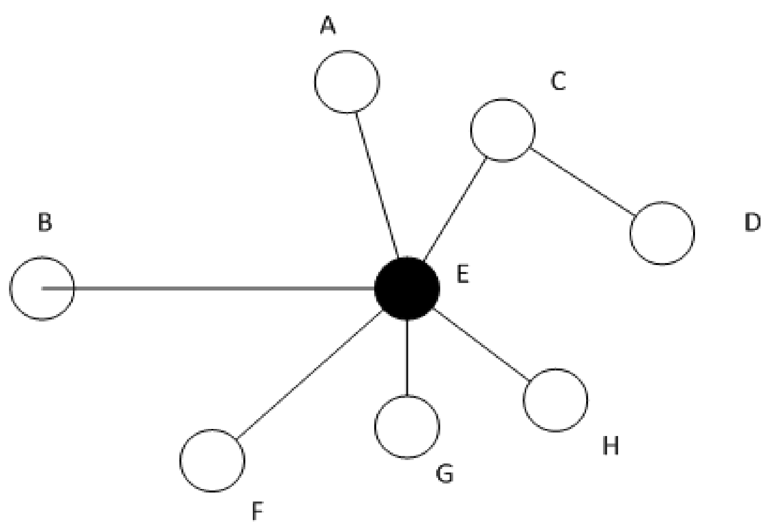
Na osnovu Hakimijeve teoreme [Hak64] možemo zaključiti da postoji 8 mesta kandidata za lociranje objekta. To su čvorovi A, B, C, D, E, F, G, H. Formirajmo matricu najkraćih rastojanja:

$$\begin{bmatrix} 0 & 5 & 2 & 5 & 3 & 7 & 5 & 6 \\ 5 & 0 & 7 & 10 & 5 & 4 & 7 & 8 \\ 2 & 7 & 0 & 3 & 2 & 6 & 4 & 4 \\ 5 & 10 & 3 & 0 & 5 & 9 & 6 & 4 \\ 3 & 5 & 2 & 5 & 0 & 4 & 2 & 3 \\ 7 & 4 & 6 & 9 & 4 & 0 & 6 & 7 \\ 5 & 7 & 4 & 6 & 2 & 6 & 0 & 2 \\ 6 & 8 & 4 & 4 & 3 & 7 & 2 & 0 \end{bmatrix}$$

U sledećem koraku izračunajmo izraze $a_j d_{ij}$, tako što ćemo svaku kolonu matrice najkraćih rastojanja pomnožiti sa brojem zahteva za uslugom u čvoru j.

Sumiranjem po vrstama matrice $[a_j d_{ij}]$ dobijaju se ukupna rastojanja koja bi prešli korisnici usluga, ukoliko bi objekat bio smešten u pojedinim čvorovima duž čijih vrsta se vrši sumiranje.Objekat treba locirati u onom čvoru duž čije vrste je dobijen najmanji zbir po izvršenom sumiranju. U datom primeru, traženi čvor je E (Slika18).

Čvor	Broj pređenih kilometara	Čvor	Broj pređenih kilometara
A	10170	E	7620
B	8970	F	9140
C	9760	G	9660
D	12620	H	9440



Slika 18.

Goldman je 1971. godine razvio algoritam za rešavanje problema jedne medijane mreže (p=1) [Gol71], koji umesto mrežnog dijagrama koristili linijski, kod koga se sve svodi na jednu putanju između samih objekata, što znatno pojednostavljuje problem čije optimalno rešenje u tom slučaju ne zavisi od razdaljine između objekata.

4.2.3 Metode za rešavanje problema p-medijane

Za rešavanje problema p-medijane u poslednjih 30 godina razvijeno je mnoštvo algoritama. Neki od njih su: algoritam za generisanje skupa dopustivih rešenja, algoritam zasnovan na teoriji grafova, heuristički algoritmi, algoritmi zasnovani na matematičkom programiranju. Metode za rešavanje problema p-medijane prvi su prezentovali Christofides i Beasley [Chr82], Hanjoul i Peeters [Han85], Beasley [Bea93] i Klose[Klo93].

Egzaktne metode koje se najčešće koriste za rešavanje problema p medijane su metoda grananja i ograničavanja [Jar72], metoda grananja i odsecanja ([Ces03], [Ces05]). Metoda Laganževe relaksacije opisana je u [Ave03].

Među tradicionalnim heurističkim metodama primenjenim na problem p-medijane je heuristika koju je predložio *Maranzana* u radu [Mar64]. Od ostalih heuristka najčešće se koriste pohlepni (Greedy) algoritam, metoda zamene čvorova (Vertex substitution) koje su opisane u [Das95].

Pohlepni algoritam u prvom koraku rešava problem 1-medijane. Nakon toga, čvor koji je identifikovan kao rešenje problema se uvrštava u skup rešenja, a zatim u svakoj sledećoj iteraciji na isti način u rešenja uključuje po još jedan čvor. Uvrštavani čvorovi se biraju tako što njihovo uključivanje u rešenja najviše doprinosi smanjenju ukupne sume rastojanja koja prelaze korisnici usluga. Na taj način se minimiziraju putni troškovi, vreme putovanja…

Metoda zamene čvorova primenjena na problem p-medijane kreće od postojećeg rasporeda p objekata, a zatim se sistematski razmatra relokacija za pojedinačni čvor. Ukoliko se zaključi da bi se na taj način dobila bolja vrednost ciljne funkcije, vrši se izmena, ukoliko to nije slučaj, razmatra se vrednost ciljne funkcije za relokaciju sledećeg čvora. Smatra se da je metoda završena kada se zaključi da se ponovnom iteracijom ne može doći do boljeg rešenja.

Rosing i ReVelle su izložili metodu pod nazivom *metoda heurističke koncentracije* opisanu u [Ros97]. Ova metoda prvo koristi heuristiku, poput metode zamene čvorova, kako bi se došlo do rešenja iz kojih je dobijen manji podskup originalnog skupa područja. Nakon toga se ovakav skup uključuje u formulaciju problema celobrojnog programiranja kako bi se izabrala najbolja alternativa. Tako dobijeno rešenje neće biti lošije od najboljeg rešenja koje smo dobili pri generisanju skupa alternativa. Ove dve faze u procesu su se pokazale kao jako dobre u pronalaženju optimalnog rešenja problema medijane.

Najpoznatije metaheuristike koje se koriste za rešavanje problema p-medijane su metoda promenljivih okolina ([Han97], [Cra03], [Cra04]), tabu ptretraživanje ([Glo90], [Rol96]), neuronske mreže ([Mer02], [Mer03]). Genetski algoritmi su takođe primenjeni za rešavanje problema p-medijane u radovima [Boz02], [Alp03], [Cor04],

Teitz i Bart [Tei68] su predložili, a *Larson i Odoni* (1981) poboljšali varijantu heurističkog algoritma za rešavanje problema p-medijane mreže. Algoritam počinje pronalaženjem lokacije jedne medijane, a nova medijana se nalazi u svakom dodatnom koraku. Algoritam se završava kada se pronađu svih p medijana.

Neka je M skup čvorova u kojima su medijane privremeno locirane i neka je m ukupan broj čvorova skupa M, koji uzima vrednosti od 1 do p.

KORAK 1: Neka je m=1. Nađimo lokaciju jedne medijane. Neka je ta medijana locirana u čvoru *i*, što znači $M = \{i\}$

KORAK 2: Sledeća medijana se nalazi u čvorovima iz skupa N\M, tako da se postigne najveće poboljšanje funkcije cilja. Zatim m uvećavamo za 1: m=m+1.

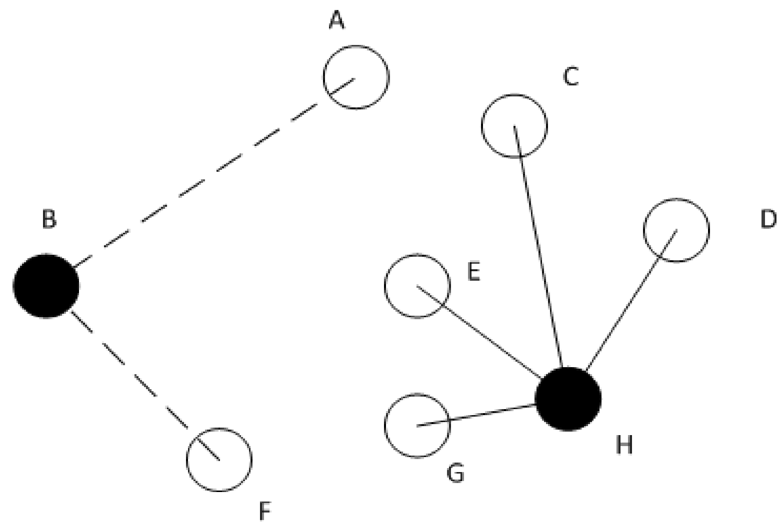
KORAK 3: Poboljšavamo vrednost funkcije cilja sistematičnim zamenjivnjem jednog čvora skupa M jednim od čvorova iz skupa N\M. Kada nema više poboljšavanja, može se napraviti funkcija cilja, i prelazi se na sledeći korak.

KORAK 4: Ako je m=p algoritam se završava, ako nije vraćamo se na KORAK 2.

Primer: Koristeći opisan algoritam nađimo 2 medijane u transportnoj mreži opisanoj na slici 17.

U prvom koraku određujemo da je optimalna lokacija za jednu medijanu čvor E, dakle M={E}.U sledećem koraku moramo uporediti vrednosti funkcije cilja u slučajevima kada su medijane locirane u sledećim parovima čvorova: (A,E), (B,E), (C,E), (D,E), (E,F), (E,G) i (E,H).

Za par čvorova (B,E) funkcija cilja ima najmanju vrednost (4620). Sada je skup M={B,E}. Upoređujemo ovo rešenje sa vrednostima funkcije cilja u sledećim parovima čvorova (A,B), (B,C), (B,D), (B,F), (B,G) i (B,H). Kako vrednost funkcije cilja za par (B,H) iznosi 3340, ovo je sada novo rešenje. To znači da je sada M={B,H}. Upoređujemo ovo rešenje sa vrednostima funkcije cilja u sledećim parovima čvorova (A,H), (C,H), (D,H), (E,H), (F,H) i (G,H). Nakon ovog upoređivanja nema poboljšanja u funkciji cilja. Rešenje (B,H) poredimo sa rešenjima (A,B), (B,C), (B,D), (B,E), (B,F) i (B,G). Nakon ovog poređenja nema poboljšanja funkcije cilja. Pošto je m=2=p, završavamo algortam sa rešenjem (B,H) (Slika 19.)



Slika 19.

4.3 Problem p-centra

Problem p-centra (***The p center problem***), koji je prvi uveo *Hakimi* [Hak64], pronalazi p objekata i dodeljuje ih korisnicima tako da se minimizira maksimalno rastojanje između korisnika i objekta koji mu je dodeljen. Suprotno problemu p-medijane, koji minimizira sumu transportnih troškova, problem p-centra na mreži traži skup objekata koji minimiziraju maksimalno rastojanje između čvora korisnika i najbližeg objekta. Postoji nekoliko mogućih varijanti osnovnog modela. Ukoliko se zahteva da traženi skup objekata bude smešten u čvorovima mreže, onda se problem naziva problem p-centra na čvorovima **(V*ertex center problem),*** dok su problemi koji omogućavaju objektima da budu bilo gde na mreži poznati kao problemi apsolutnog centra (***Absolute center problems).*** *Kariv* i *Hakimi* su pokazali da je problem p-centra na mreži NP težak, [Kar79a].

4.3.1 Formulacija problema p-centra na čvorovima

Formulacija problema p-centra na čvorovima opisana je u [Das95]:

Neka su:

d_{ij} –rastojanje od čvora - korisnika *i* do potecijalnog čvora - snabdevača *j*

p – broj objekata koje treba locirati

$$x_j = \begin{cases} 1, & \text{ako lociramo na mestu potražnje j} \\ 0, & \text{ako ne} \end{cases}$$

y_{ij}= deo čvora korisnika *i* koji je opslužen objektom u čvoru *j*

W= maksimalno rastojanje između čvora korisnika i najbližeg objekta

Min W (1)

pri ograničenjima:

$$\sum_j y_{ij} = 1 \qquad \forall i \qquad (2)$$

$$\sum_j x_j = p \qquad (3)$$

$$y_{ij} \le x_j \qquad \forall i,j \qquad (4)$$

$$W \ge \sum_j d_{ij} y_{ij} \qquad \forall i \qquad (5)$$

$$x_j = \{0,1\} \qquad \forall j \qquad (6)$$

$$y_{ij} \ge 0 \qquad \forall i,j \qquad (7)$$

Funkcija cilja (1) minimizuje maksimalno rastojanje između čvora korisnika i njemu najbližeg objekta u čvoru. Ograničenje (2) se odnosi na činjenicu da svi korisnici u čvoru *i* moraju biti opsluženi objektom u nekom čvoru *j* za sve korisnike *i*. Ograničenjem (3) se pokazuje da na mreži treba da bude ukupno p objekata. Korisnici u čvoru *i* ne mogu biti opsluženi objektom u čvoru *j* ukoliko objekat nije lociran u čvoru *j,* što je iskazano kroz ograničenje (4). Ograničenje

(5) govori da maksimalno rastojanje izmedju čvora korisnika i najblizeg objekta u čvoru W mora biti veće od rastojanja između bilo kog čvora korisnika *i* i objekta *j* kojim je opslužen. Uslovi (6) i (7) su ograničenja celobrojnosti i ne negativnosti.

Karakteristični primeri lokacijskih problema tipa min-max su optimalna razmeštanja raznih hitnih službi. Pretpostavimo da je za jednu službu koja pruža hitne usluge stanovništvu nekoliko naselja potrebno odabrati lokaciju i da zbog nekih razloga ta služba treba da se nalazi u naselju. Kao kriterijum optimalnosti moguće je koristiti:

- Najkraće rastojanje od posmatranog čvora – potencijalne lokacije do najudaljenijeg čvora u mreži.
- Najkraće rastojanje od najudaljenijeg čvora do posmatranog čvora – potencijalne lokacije.
- Najkraće rastojanje od posmatranog čvora – potencijalne lokacije do najudaljenijeg čvora u mreži i nazad.

Svako rastojanje može se otežati brojem stanovnika naselja ili verovatnoćom da će baš to naselje zatražiti uslugu. Zadatak je naći lokaciju hitne službe tako da težinsko rastojanje bude što je moguće manje.

Ako se kao kriterijum optimalnosti koristi rastojanje od čvora-potencijalne lokacije do najudaljenijeg čvora, tada se za svaki čvor i∈N određuje vrednost:

$$s_0(i) = \max_{j\in N}\left[V_j d(i, j)\right] ,$$

gde je $s_0(i)$ maksimalna vrednost u vrsti i (*i*=1, . . ., n) matrice najkraćih rastojanja D(G), koja nastaje množenjem kolone *j* (*j*= 1,. . . , n) matrice D(G) sa težinskim koeficijentom V_j.

Čvor i_0^* za koji važi:

$$s_0(i_0^*) = \min_{i\in N}\left[s_0(i)\right]$$

se naziva “spoljašnji centar” grafa G , a vrednost $s_0(i_0^*)$ “spoljašnji radijus” grafa G. Jedan graf može imati više spoljašnjih centara.

Ako se kao kriterijum optimalnosti koristi rastojanje od najudaljenijeg čvora do posmatranog čvora – potencijalne lokacije tada se za svaki čvor i∈N određuje vrednost:

$$s_t(i) = \max_{j\in N}\left[V_j d(j,i)\right] ,$$

gde je $s_t(i)$ maksimalna vrednost u koloni (*j*= 1,. . . , n) matrice D(G) koja nastaje množenjem vrste *i* (*i*=1,. . . , n) matrice D(G) sa težinskim koeficijentom V_j.

Čvor i_t^* za koji važi:

$$s_t(i_t^*) = \min_{i\in N}\left[s_t(i)\right]$$

se naziva se “unutrašnji centar” grafa G, a vrednost $s_t(i_t^*)$ “unutrašnji radijus” grafa G. Jedan graf može imati više unutrašnjih centara.

Kada je kriterijum optimalnosti ukupno rastojanje od čvora-potencijalne lokacije do najudaljenijeg čvora i nazad, tada se za svaki čvor i∈N određuje vrednost:

$$s_{ot}(i) = \max_{j\in N}\left[V_j(d(i,j)+d(j,i))\right],$$

gde je $s_{ot}(i)$ maksimalna vrednost u vrsti *i* (*i*=1,. . . , n) matrice $D^3(G)$ koja nastaje množenjem kolone *j* matrice rastojanja $D(G)=[d(i,j)+d(j,i)]_{n\times n}$ sa težinskim koeficijentom V_j.

U simetričnim mrežama se spoljašnji, unutrašnji i spoljašnje-unutrašnji centri grafa poklapaju.

Algoritam za određivanje spoljašnjeg centra grafa je sledeći:

1. Naći rastojanja između svaka dva čvora u mreži i formirati matricu rastojanja D=(d(i, j)), čiji su elementi rastojanja između čvorova *i* i *j*, i,*j*=1,. . . , n.

Svaku kolonu *j* matrice D pomnožiti težinom čvora V_j, tj. formirati matricu težinskih rastojanja D'

2. Naći maksimalni element (najveću vrednost) $s_o(i)$ za svaki red matrice D', tj. odrediti:

$s_0(i) = \max_{j\in N}\left[V_j d(i,j)\right]$, *i*=1,. . . , n

4. Od svih $s_o(i)$ određenih u prethodnom koraku naći najmanje, $s_0(i_0^*) = \min_{i\in N}\left[s_0(i)\right]$

4.3.2 Problem apsolutnog spoljašnjeg centra

Posmatrajmo problem lokacije službe za hitne intervencije kada ne postoji ograničenje da ona mora biti u naselju. Ponovo se razmatra problem lokacije na mreži tipa *min-max,* ali je novi objekat moguće postaviti u čvor ili na granu. Za modeliranje ovog problema koristi se pojam tačke koja pripada grafu, odnosno grani.

Kažemo da tačka x prpada grani (*i*, *j*) ukoliko je ispunjen uslov c_{ij}= c(*i, x*) + c(*x, j*), gde je ε=c(*i, x*) duzina dela (*i, x*).

Problem tipa *min-max* se formuliše kao zadatak nalaženja tačke grafa za koju važi da je rastojanje od nje do najdaljeg čvora (ili od najdaljeg čvora do nje) minimalno. Ovakvi zadaci se nazivaju problemi određivanja *apsolutnih centara* težinskog grafa.

Razmotrimo problem apsolutnog spoljašnjeg centa:
Neka je d(*x, i*) rastojanje od neke tačke x koja pripada grafu G do proizvoljnog čvora i∈N. Zadatak određivanja apsolutnog spoljašnjeg centra se definiše na sledeći način:

Odrediti x_0^* na grafu G tako da je: $s_o(x_o^*) = \min\left[\min_{i\in N}\{V_i d(x,i)\}\right]$, gde se prvi minimum odnosi na sve tačke koje pripadaju grafu G.

Ovaj zadatak rešava se primenom algoritma *Hakimija*, [Hak64] koji u osnovnom obliku ima dva koraka. U prvom je potrebno da se za svaku granu nađe tačka za koju važi da je otežano rastojanje od nje do najdaljeg čvora minimalno, tj. reše se problemi tipa *min-max* za svaku od grana. Ove tačke se nazivaju lokalni centri, a odgovarajuća rastojanja lokalnim radijusima grana grafa. U drugom koraku se od prethodno utvrđenih lokalnih centara izabere onaj koji daje najmanju vrednost radijusa.

Uočimo proizvoljnu granu *(m, l)* i na njoj prizvoljnu tačku x koja je definisana dužinom *ε=c(m, x)* dela grane *(m, x*). Izračunajmo najpre rastojanje od proizvoljne tačke *x* na grani do najdaljeg čvora grafa, da bismo izračunali lokalni radijus grane *(m, l)* koji ćemo označiti sa s_{ml}^*
Od tačke *x* do proizvoljnog čvora *i*∈*N* može se dospeti preko čvora *m* ili preko čvora *l*. Ukoliko se ide preko čvora *m* dužina od tačke *x* do čvora *i* iznosi:

$\tau_i(\varepsilon) = c(x, m) + d(m, i) = \varepsilon + d(m, i)$

Ukoliko se ide preko čvora *l*, dužina od tačke *x* do čvora *i* iznosi:

$\tau_i'(\varepsilon) = cml - c(x, m) + d(l, i) = c_{ml} - \xi + d(l, i).$

Rastojanje *d*(*x*, *i*) je dužina kraćeg od ova dva „puta“

$d(x, i) = \min \{ \tau i(\varepsilon) , \tau i'(\varepsilon) \}$

Težinsko rastojanje ima oblik:

$di'(\varepsilon) = Vi \min \{ \varepsilon + d(m, i), cml - \xi + d(l, i)\}.$

Najpre treba odrediti najdalji čvor za svako ε, $0 \leq \varepsilon \leq c_{ml}$, a potom se traži ono ε^* za koje je rastojanje do najdaljeg čvora minimalno. Dakle, za granu (m, l) rešava se sledeći problem:

$$s_{ml}^* = \min_{0 \leq \varepsilon \leq c_{ml}} \left[\max_{i \in N} d_i(\varepsilon) \right]$$

Pošto se za svaku granu odredi lokalni centar i lokalni radijus, izabere se onaj lokalni centar kome odgovara najmanji radijus.

4.3.3 Metode za rešavanje problema p-centra

Različiti autori su za optimalno rešavanje problema p-centa koristili još jedan pomoćni problem Set Covering Problem, SCP. Cilj SCP-a je nalaženje minimalnog broja objekata i njihovih lokacija, tako da svaki čvor-snabdevač mora da usluži objekat u okviru određenog maksimalnog vremena odgovora ili rastojanja, koje možemo nazvati prečnikom. Rešenje ovog problema lako se može naći rešavanjem njegove linearne progamske relaksacije sa povremenim primenama metode grananja i ograničavanja. Kada dođemo do velikih problema, veličina relaksirane verzije SCP-a može se redukovati smanjivanjem uzastopnih kolona i redova. Šira objašnjenja takvih redukcionih problema opisao je *Daskin* u [Das95]. Ova ideja služi da bi se našao najmanji prečnik, tako da optimalno rešenje razmatranog pomoćnog problema dovede do rešenja problema p-centra.

Minieka je u [Min70] predložio osnovni algoritam koji se oslanja na Set Covering problem. Ideja se sastojala u tome da se odabere optimalna daljina pokrivanja za radijus i da se proveri da li su svi zahtevi korisnika pokriveni u okviru radijusa korišćenjem samo p objekata. Na osnovu *Miniekine* ideje, *Daskin* je 1995. godine razvio algoritam za optimalno rešavanje ovog problema korišćenjem metode preseka koja sistematski smanjuje prazninu između gornjih i donjih ograničenja optimalnog rešenja i samim tim i optimalnu daljinju pokrivanja.

Daskin [Das00]*, Elloumi, Labbe, i Pochet* [Ell01] su predložili dva egzaktna algoritma za problem p-centa na čvorovima. Oba algoritma rešavaju uzastopne podprobleme i oslanjaju se pre na sprovođenje učestale pretrage daljina pokrića, nego na skupove lokacija objekata. *Daskin* je formulisao potproblem maksimalnog skupa pokrivanja i rešio ga pomoću Lagranževe relaksacije, dok je *Elloumi* koristio *Minekine* potprobleme i rešio ga pomoću pohlepnog heurističkog algoritma i IP formulacije potproblema.

Ilhan i Pinar su predložili u [Ilh01] inetresantnu egzaktnu metodu za rešavanje problema p-centra na čvorovima koja se sastoji iz dve faze. Prva faza, koju su nazvali LP faza, izračunava donju granicu optmalnog rešenja problema rešavajući niz problema izvodljivosti koji se zasnivaju na LP formulaciji. Druga faza, nazvana IP faza, koristi takođe probleme izvodljivosti da proveri da li je ili nije moguće uslužiti sve korisnike samo pomoću p objekata u okviru datog prečnika.

Sa metodološke strane *Martinich* je u [Mar88] predložio heuristiku za rešavanje problema p–centra na čvorovima. Ovaj algoritam može se posamtrati kao algoritam „pohlepnog oduzimanja“ u smislu da heuristika počinje sa pronalaženjem objekata na svim lokacijama kandidata i nastavlja da zatvara postrojenja na inteligentan način.

4.4 Lokacijski problemi pokrivanja

Lokacijski problemi pokrivanja često se koriste pri određivanju lokacija za hitne službe kao što su policijske i vatrogasne stanice, hitne pomoći itd. Clj je da se pokrije što veći broj stanovništva, a da pri tome troškovi opsluživanja budu što niži. U praksi su troškovi opsluživanja klijenta, uglavnom direktno proporcionalni rastojanju između klijenta i objekata. Smatra se da je određeno područje ''pokriveno'' uz pomoć datog objekta, ukoliko se nalazi na rastojanju manjem od nekog unapred definisanog ''kritičnog'' rastojanja. Primarni cilj je odabrati lokacije za uslužne objekte tako da što više potencijalnih klijenata bude pokriveno. Jedan od najvažnijih problema pokrivanja je problem maksimalnog pokrivanja korisnika (**The maximal covering location problem, MCLP**). U praksi se često sreću problemi lokacije nepoželjnih objekata čija funkcija cilja treba da pokrije što manje klijenata koji se nazivaju problemi minimalnog pokrivanja korisnika (**Minimum covering location problem with distance constrains, MCLPDC**). Lociranjem fiksiranog broja objekata cilj je u što većoj meri smanjiti broj pokrivenih klijenata (smatra se da je klijent pokriven ukoliko se nalazi na rastojanju manjem od definisanog za bar jednu od uspostavljenih lokacija). Problem pokrivanja lokacijama iz skupa (**Location set covering problem, LSCP**) pronalazi minimalan broj objekata iz ponuđenog skupa, koji pokrivaju sve zahteve klijenata. LSCP ima zadatak da pronađe objekte koji zadovoljavaju sve zahteve klijenata, a da pri tome cena uspostavljanja tih objekata, koja je dodeljena svakoj lokaciji, bude minimalna. U realnim situacijama često se događa da ovakvi objekti nisu poželjni u okolini klijenata koje treba da pokriju, tako da se u literaturi sreću i modeli kod kojih je, kao dodatni uslov, definisano minimalno rastojanje od najbližeg klijenta. Pokazano je da prošireni modeli LSCP-a imaju istu strukturu kao i originalni LSCP problem.

4.4.1 Lokacijski problem maksimalnog pokrivanja

Lokacijski problem maksimalnog pokrivanja (***The maximal covering location problem*, MCLP**) prvi su formulisali 1974. godine *Church i ReVelle* u radu [Chu74]. MCLP podrazumeva nalaženje jednog ili više čvorova na mreži (objekata snabdevača), takvih da se iz njih za neko unapred zadato rastojanje (ili vreme putovanja, troškove) pokriva najveći broj (najveća težina, tražnja) korisničkih čvorova. Detaljan opis ovog problema dat je u [Gal96].

Prema prostoru u kome se vrši lokacija, problem MCLP može biti: *diskretan (*skup potencijalnih lokacija snabdevača je unapred zadat), *kontinualan (*za prostor lokacija može se uzeti Euklidski prostor R^2 ili skup presečnih tačaka krugova poluprečnika R koji su opisani oko korisnika) i *mrežni (*skup potencijalnih lokacija je skup čvorova i lukova neke mreže), [Dre10]. *Megiddo* je pokazao u [Meg83] da je mrežni MCLP NP-težak problem.

MCLP se veoma često koristi za rešavanje problema postavljanja željenih objekata na ponuđene lokacije. U nekim situacijama ovi objekti su nepoželjni, kao na primer kada su u pitanju deponije smeća, nuklearni reaktori ili zatvori. Ovakvi objekti obično imaju nepovoljan efekat na stanovništvo koje se nalazi u okolini. Naravno, postoje objekti koji su stanovništvu veoma poželjni u okolini poput domova zdravlja, škola, fakulteta, vatrogasnih stanica, itd. Iz prethodno navedenih razloga, ovaj problem se može definisati kao problem maksimizacije ili minimizacije funkcije cilja.

Ograničeno vreme putovanja (ili ograničena dužina putovanja) od objekta snabdevača do korisničkog čvora će se razlikovati za različite veličine posmatranih teritorija:

- Za urbane lokacijske probleme, ograničeno vreme putovanja (dužina putovanja) predviđeno za uslugu je, na primer, red veličine nekoliko desetina minuta do jednog časa (nekoliko desetina kilometara).
- Za lokacijske probleme na "makro" teritoriji, red veličine vremena putovanja (dužine putovanja) predviđenog za uslugu je nekoliko sati (nekoliko stotina kilometara). Na primer, ako pretpostavimo da je radno vreme objekta opsluživanja 8 časova i da radnik koji opslužuje korisničke čvorove mora da obavi uslugu i vrati se nazad u toku radnog vremena, tada limitirano vreme putovanja predviđeno za uslugu iznosi 4 časa. Ako raspolažemo podatkom o prosečnoj brzini putovanja na putevima posmatrane teritorije, tada možemo odrediti maksimum dužine putovanja.

Maksimalno vreme putovanja (dužinu putovanja) možemo posmatrati kao radijus "kvazi" kružnice čiji oblik diktira konfiguracija mreže, a čiji je centar smešten u svaku od potencijalnih lokacija-čvorova. Opisujući kružnicu određenog radijusa oko svakog čvora-potencijalne lokacije, delimo čvorove-korisničke punktove na one koji se nalaze unutar kružnice i koji mogu biti usluženi u okviru zadatog vremena putovanja (dužine putovanja) i one koji to nisu i neće biti usluženi. Maksimalno vreme putovanja (dužina putovanja) od objekta snabdevača do korisničkih čvorova biće obeleženo sa R.

U opštem slučaju, sve čvorove mreže posmatramo kao potencijalne lokacije za objekat snabdevača i kao korisničke čvorove, istovremeno. Za proizvoljan čvor i∈N težinskog grafa G=(N,L), neka je P(i) podskup onih čvorova *j* iz N koje je moguće dostići iz čvora *i* iz N, uljučujući i čvor *i*, pod uslovom da rastojanje između čvorova *i* i *j* ne prevazilazi vrednost R.

$$P(i) = \{ j \mid d(i,j) \le R, i, j \in N \}$$

Za svaki čvor i iz N odredimo vrednosti B(i) koje predstavljaju ukupan broj težinskih čvorova čija je udaljenost manja ili jednaka od zadate vrednosti R u odnosu na posmatrani čvor

$B(i) = \sum_{j \in N} V_j x_{ij}$, gde je $x_{ij} = 1$ za svako $j \in P$ i $x_{ij} = 0$ za svako *j* ne pripada P

Vrednosti B(i) predstavljaju ukupan broj težinskih čvorova čija je udaljenost manja ili jednaka, od zadate vrednosti R u odnosu na posmatrani čvor $i \in N$.

Čvor $i^* \in N$ za koji važi:

$B(i^*) = \max_{i \in N}[B(i)]$

predstavlja najbolju lokaciju u zadatom smislu, odnosno čvor iz koga se pokriva najveći broj težinskih čvorova na mreži.

4.4.1.1 Metode za rešavanje MCLP-a

Algoritam koji će biti izložen sastoji se iz četiri koraka i jednostavan je čak i za mreže sa većim brojem čvorova. Obim računanja raste sa brojem čvorova po istom zakonu, po kom raste obim računanja u traženju najkraćih puteva između svaka dva čvora. Ovaj rast sledi polinomsku funkciju.

KORAK 1: Matrici najkraćih vremena putovanja(rastojanja) $D=[d(i,j)]_{n\times n}$ pridružiti matricu $X=[x_{ij}]_{n\times n}$ tako da važi:

$x_{ij}=1$ akko $d(i,j)\leq R$

$x_{ij}=0$ akko $d(i,j)>R$

KORAK 2: Svaku kolonu matrice $X=[x_{ij}]_{n\times n}$ pomnožiti težinskim koeficijentom čvora V_j, $j\in N$, tj. formirati matricu težinskih čvorova $X^*=[\,V_j\,x_{ij}]_{nxn}$.

KORAK 3: Naći sumu elemenata za svaku vrstu matrice $X^*=[\,V_j\,x_{ij}]_{nxn}$, tj. odrediti

$B(i)=\sum_{j\in N} V_j x_{ij}$, $i\in N$

KORAK 4: Od svih B(i) određenih u prethodnom koraku naći i^* za koje je B(i) najveće, tj.

$B(i^*)=\max[B(i)]$, $i\in N$

Izloženi algoritam za određivanje p najboljih lokacija, u smislu maksimalnog pokrivanja grafa, može se efikasno primenjivati za slučajeve malih i srednje velikih mreža u smislu broja čvorova– do red veličine 100 čvorova, naročito kada se ima u vidu praktična primena, gde je p najčešće mali ceo broj (na primer, p=2 ili p=3). Za mreže većih dimenzija pribegava se korišćenju raznih heuristika.

Church i ReVelle su predložili u [Chu74] dve heuristike za rešavanje MCLP-a koje su nazvali proždrljivo dodavanje (***The greedy adding***) i proždrljiv algoritam i heuristika supstitucije (***The***

greedy adding with substitution heuristic). Za rešavanje ovog problema velikih dimenzija postoji nekoliko heuristika od kojih tabu pretraživanje daje najbolje rezultate [Bea90].

4.4.2 Lokacijski problem anti-pokrivanja

Lokacijski problem anti-pokrivanja (***Anti-covering location problem, ACLP***) spada u klasu važnih problema prostorne optimizacije. ACLP su prvi definisali *Moon i Chaudhry* 1984. godine, [Moo84]. Ovaj problem podrazumeva da se u skupu potencijalnih lokacija kojima su pridruženi težinski koeficijenti (pokazatelji njihove važnosti), pronađe podskup takav da sve lokacije u njemu zadovolje uslov da su na međusobnoj udaljenosti ne manjoj od neke unapred zadate i da je suma težinskih koeficijenata pridruženih tim lokacijama maksimalna.

U teoriji grafova postoje problemi koji su vrlo bliski ACLP-u, koji su drugačije grafovski struktuirani, ali posmatraju isti ili komplementaran problem. To su: problem maksimalnog nezavisnog skupa (***Maximum Independent Set Problem***), problem pakovanja čvorova/tačaka (***Node/Vertex Packing Problem***) i problem maksimalne klike (***Maximum Clique Problem***), [Bom99]. Kako problemi maksimalne (otežane) klike i maksimalnog (otežanog) nezavisnog skupa spadaju u grupu prvih problema za koje je pokazano da su NP-potpuni [Gar79], sledi da je i lokacijski problem anti-pokrivanja NP-potpun problem.

Kod ovih problema funkcija cilja maksimizira broj lokacija koje zadovoljavaju uslov da su sve na međusobnoj udaljenosti ne manjoj od neke unapred zadate. Ovi problemi su prepoznatljivi kao LUF problemi, mada su prisutni i u slučajevima lociranja uslužnih objekata kod kojih koncept pokrivanja insistira na minimumu dozvoljene udaljenosti između tih objekata.

U problemima lociranja uslužnih objekata koji su u lancu velikih sistema, poput benzinskih pumpi, fast-food restorana, itd., poslovna politika u izboru lokacija može biti takva da nalaže minimum dozvoljene udaljenosti između bilo koje dve susedne lokacije, kao način da se pokrije što veća teritorija, odnosno što veći broj korisnika. U tom slučaju, zadatak bi bio maksimizirati profit ili pokrivenost korisnika izborom lokacija koje zadovoljavaju ograničenje u pogledu dozvoljene međusobne udaljenosti. Bitno je naglasiti da je ACLP jedna od mogućnosti za formulaciju problema kod kojih je poželjno rasporediti lokacije sa ciljem da se postigne veća pokrivenost regiona koji je predmet interesovanja. U tom smislu, modeliranje problema lociranja poslovnica telekomunikacionih operatora koje pružaju usluge korisnicima kroz direktan kontakt može u jednoj od vizija biti realizovano kao ACLP model.

ACLP je prisutan i u vojnoj primeni. Na primer, problem lociranja maksimalnog broja borbenih zona (jedinica), tako da sve međusobno budu na odgovarajućoj udaljenosti (ne manjoj od dozvoljene). Minimum propisane udaljenosti obezbeđuje da uništenje jedne od zona ima minimalan uticaj na ostale.

Veoma značajnu primenu ACLP ima u izboru lokacija za skladišta nepoželjnih i opasnih materija, poput eksploziva, zapaljivih materija, komprimovanih gasova, itd. U ovakvim problemima cilj je uskladištiti maksimalnu količinu nepoželjnih materija, tako da skladišta međusobno budu na udaljenosti većoj od ili jednakoj nekoj zadatoj koja se može nazvati bezbedno rastojanje, zbog mogućih eksplozija i neželjenih efekata koje one mogu izazvati. U slučaju lociranja raznih emitera mirisa, toplote, radioaktivnosti, buke, aero zagađenja, itd., minimum propisane udaljenosti između emitera obezbeđuje poštovanje propisa u vezi sa maksimalnom dozvoljenom koncentracijom konkretnog zagađenja. ACLP prepoznatljiv je i kao problem lociranja predajnika u radio-difuznim mrežama, gde predajnici koji rade na istim frekvencijama ne mogu biti na rastojanju manjem od propisanog.

4.4.2.1 Formulacija ACLP-a

Posmatra se potpuno povezana mreža, predstavljena grafom G=(N,A), gde je N={1, ..., i, j, ..., n} skup čvorova potencijalnih lokacija, a A skup grana *(i,j).* Svakoj grani $(i, j) \in A$ pridružen je nenegativni skalar c_{ij} koji predstavlja minimalnu udaljenost (prostornu ili vremensku) između svakog para čvorova $i \in N$, $j \in N$. Svakom čvoru $i \in N$ pridružen je težinski koeficijent v_i, koji predstavlja maksimalnu korisnost lokacije. Neka je sa R označena vrednost, unapred zadate, minimalne dozvoljene udaljenosti između čvorova. Neka je *Π(i)* podskup onih čvorova $j \in N$ koji su od čvora $i \in N$ udaljeni manje od unapred zadate dozvoljene udaljenosti R, (c_{ij}<R), ne uključujući čvor $i \in N$, odnosno:

$$\Pi(i) = \left\{ j \mid c_{ij} \le R, i \ne j \in N \right\} \qquad (1)$$

Prvu postavku ACLP-a, u formi problema binarnog celobrojnog programiranja, definisali su *Moon i Chaudhry* 1984. godine, [Moo84].

Max $Z = \sum_{i \in N} v_i x_i$ (2)

$Mx_i + \sum_{j \in \Pi(i)} x_j \leq M$, za svako $i \in N$ (3)

$x_i \in \{0,1\}$, za svako $i \in N$ (4)

gde je M dovoljno veliki pozitvan broj.

Zadatak je naći vektor rešenja x = (x_1, ..., x_i, ..., x_n), pri čemu je x_i =1, ako je lokacija koja odgovara čvoru $i \in N$ izabrana i x_i = 0 u suprotnom (4), koji čini da ukupna težina (korist) izabranih lokacija bude maksimalna (2). Ograničenje (3) se može nazvati ograničenje susedstva. Ako čvor $i \in N$ nije izabrana lokacija (x_i = 0), to nema uticaja na vrednosti x_j, a ako jeste (x_i = 1), svako x_j za koje važi $j \in \Pi(i)$ mora da uzme vrednost 0, jer se čvorovi $j \in \Pi(i)$ nalaze na udaljenosti manjoj od dozvoljene u odnosu na čvor $i \in N$.

Još dve formulacije ACLP-a kao problema binarnog celobrojnog programiranja mogu se naći u [Mur97]. Njima se postiže da koeficijent M u ograničenju (3) postane “strožiji“, pri čemu određivanje novih koeficijenata M zahteva izvestan računski napor. Formulacija ACLP-a kao problema nelinearnog, preciznije kvadratnog 0-1 programiranja, može se naći u [Dim05] i [Dim06].

4.4.2.2 Metode za rešavanje ACLP-a

ACLP je NP- težak problem (obim računanja eksponencijalno raste sa porastom broja čvorova), pa se za njegovo rešavanje primenjuju razne heuristike. U rešavanju ACLP-a testirani su sledeći heuristički pristupi: Lagranževa relaksacija [Mur97], genetski algoritmi [Cha86] i greedy algoritmi ([Dim05], [Dim06], [Cha86]).

U ovom radu biće predstavljene dve greedy (proždrljive) heuristike: α heuristika i β heuristika za rešavanje ACLP-a u algoritamskoj formi.

Osnovna ideja je da se za svaki čvor ustanovi koliko čvorova on pokriva (na udaljenosti su manjoj od dozvoljene), odnosno ne pokriva (na udaljenosti su većoj od dozvoljene). Na osnovu ova dva pokazatelja ustanovljeni su kriterijumi α i β, na osnovu kojih heuristike nose imena.

Neka je $\Omega^X \subseteq N$ skup svih čvorova "uključenih" u rešenje, $\Omega^X_t \subseteq \Omega^X$ skup čvorova "uključenih" u rešenje u t-toj iteraciji, $\Psi \subseteq N$ skup svih čvorova "isključenih" iz rešenja, $\Psi_t \subseteq N$ skup čvorova "isključenih" iz rešenja u t-toj iteraciji i $A_t \subseteq N$ skup "aktivnih" čvorova u t-toj iteraciji, odnosno čvorova koji su preostali da u narednoj iteraciji budu vrednovani. Neka je π(i) podskup svih onih čvorova $j \in A_t$ koji su od čvora $i \in A_t$ udaljeni manje od dozvoljene udaljenosti ($c_{ij}<R$), uključujući i čvor $i \in At$. Dakle:

$$\pi(i)=\left\{ j \mid c_{ij} \le R,\, j \in A_t \right\}$$

Vrednovanje svakog čvora $i \in A_t$ vrši se na osnovu sledećih kriterijuma:

$\alpha(i)= v_i - \sum_{j \in \{\pi(i) \setminus i\}} v_j$, koji predstavlja "korist" koja bi se ostvarila izborom čvora $i \in At$, odnosno njegovim "uključenjem" u rešenje, a čija se vrednost izražava kroz razliku težinskog koeficijenta tog čvora i sume težinskih koeficijenata svih onih čvorova koje bi ovaj izbor "isključio" iz rešenja, odnosno koji su na rastojanju manjem od dozvoljenog u odnosu na njega.

$\beta(i)= v_i + \sum_{j \in \{A_t \setminus \pi(i)\}} v_j$, predstavlja takođe "korist" koja bi se ostvarila izborom čvora $i \in At$, odnosno njegovim "uključenjem" u rešenje, ali čija se vrednost izražava kroz sumu težinskog koeficijenta tog čvora i težinskih koeficijenata svih onih čvorova koje ovaj izbor ne bi "isključio" iz rešenja, odnosno koji su na rastojanju većem od dozvoljenog u odnosu na čvor $i \in A_t$.

Algoritam za rešavanje ACLP-a je za obe heuristike dat u objedinjenoj formi, sa naznakom kada se nešto odnosi na konkretnu heuristiku, u pojedinim koracima algoritma.

KORAK1: inicijalizacija t=1

KORAK2: inicijalizacija $\Omega^X_t=\emptyset$, $\Psi_t=\emptyset$, $A_t=N$

KORAK3: za svaki čvor $i \in A_t$ odrediti skup π(i), tako da važi $\pi(i) = \{ j \mid c_{ij} \le R, j \in A_t \}$

KORAK4: za svaki čvor $i \in A_t$ izračunati vrednost kriterijuma $\alpha(i)$ za heuritiku α,

tako da važi $\alpha(i)= v_i - \sum_{j \in \{\pi(i) \setminus i\}} v_j$, odnosno izračunati $\beta(i)$ za heuristiku β, tako da važi

$\beta(i)= v_i + \sum_{j \in \{A_t \setminus \pi(i)\}} v_j$,

KORAK5: Odredti čvor i^* za koji važi $\alpha(i^*)=\max[\alpha(i)]$, $i \in At$, za heuristiku α, analogno za heuristiku β

KORAK6: postaviti t=t+1

KORAK7: Ažurirati skup Ω^X_t: $\Omega^X_t = \Omega^X_{t-1} \cup \{i^*\}$

KORAK8: Ažurirati skup Ψ_t: $\Psi_t = \Psi_{t-1} \cup \{ j \mid j \in \pi(i^*) \setminus i^* \}$

KORAK9: Ažurirati skup A_t: $A_t = A_{t-1} \setminus \{ \Psi_t \cup i^* \}$

KORAK10: Ako je $A_t \neq \emptyset$, vratiti se na KORAK 3, u suprotnom preći na KORAK 11

KORAK11: Usvojiti $\Omega^X = \Omega^X_t$, $\Psi = \Psi_t$

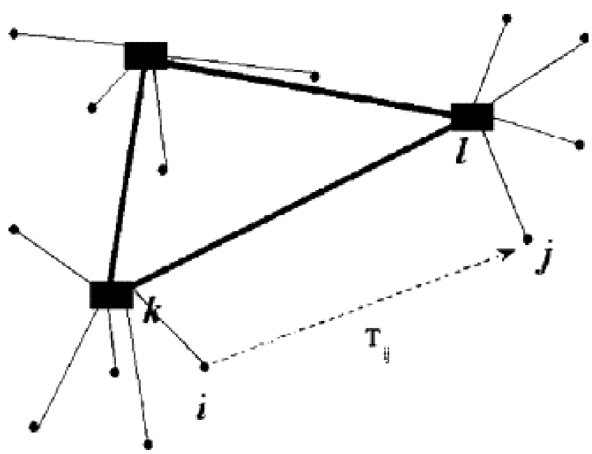
KORAK12: Izračunati vrednost funkcije cilja $Z = \sum_{i \in \Omega^x} v_i$

KORAK13: kraj algoritma

4.5 Hab lokacijski problemi

Problemi lociranja habova (**Hub location problems**), prisutni su u mrežama za hitnu isporuku pošiljaka, telekomunikacionim mrežama, kod avio i drumskog saobraćaja, u situacijama gde treba uspostaviti saobraćaj od početnog čvorova(snabdevača) do krajnjeg čvora(korisnika). Transport između ova dva čvora se ne uspostavlja direktno, već je usmeren preko skupa čvorova koji su označeni kao habovi. Opšti tip hab lokacijskog problema uključuje lociranje habova, gde se kao kriterijum optimalnosti koji se minimizira pojavljuje suma rastojanja (vremena putovanja) od svakog izvorišnog čvora preko haba do svakog odredišnog čvora, i definisanje ruta za transportne tokove između izvorišnih i odredišnih čvorova. Habovi predstavljaju centre konsolidacije i kolekcije protoka u mreži između dve lokacije. Korišćenjem habova kao tačaka preusmeravanja protoka i povećavanjem transporta između habova, kapacitet mreže se može iskoristiti dosta efikasnije, a troškovi transporta smanjiti. Pregled hab lokacijskih problema i njihova klasifikacija dati su u [Cam96], [Cam02].

Tipičan hab lokacijski problem mreže sadrži: lokacijski deo, u kome određeni čvorovi moraju biti izabrani za habove, i alokacijski deo, u kome je zahtev da određeni odlazno-dolazni parovi budu usmereni kroz habove (Slika 20.).



Slika 20. Mrežni hab lokacijski problem

U literaturi nalazimo razne varijante hab lokacijskih problema sa različitim ograničenjima i funkcijama cilja. Prema alokaciji koncepta, tj. u odnosu na to kako su ne-hab čvorovi alocirani prema habovima, hab lokacijski problem se dele na:

• šema jednostruke alokacije (single allocaton scheme), kod koje je svaki snabdevač/korisnik pridružen tačno jednom habu, tako da se sav transport od/do tog čvora obavlja isključivo preko određenog haba. Saobraćaj koji dolazi iz jednog ne-hab čvora ide ka samo jednom habu i saobraćaj koji dolazi ka odredišnom čvoru potiče iz samo jednog haba kome je pridružen.

• šema višestruke alokacije (multiple allocaton scheme), koja dopušta svakom ne-hab čvoru da komunicira sa jednim ili više habova čime se ostvaruje veća fleksibilnost modela, jer se saobraćaj može realizovati putem sa najnižom cenom transporta.

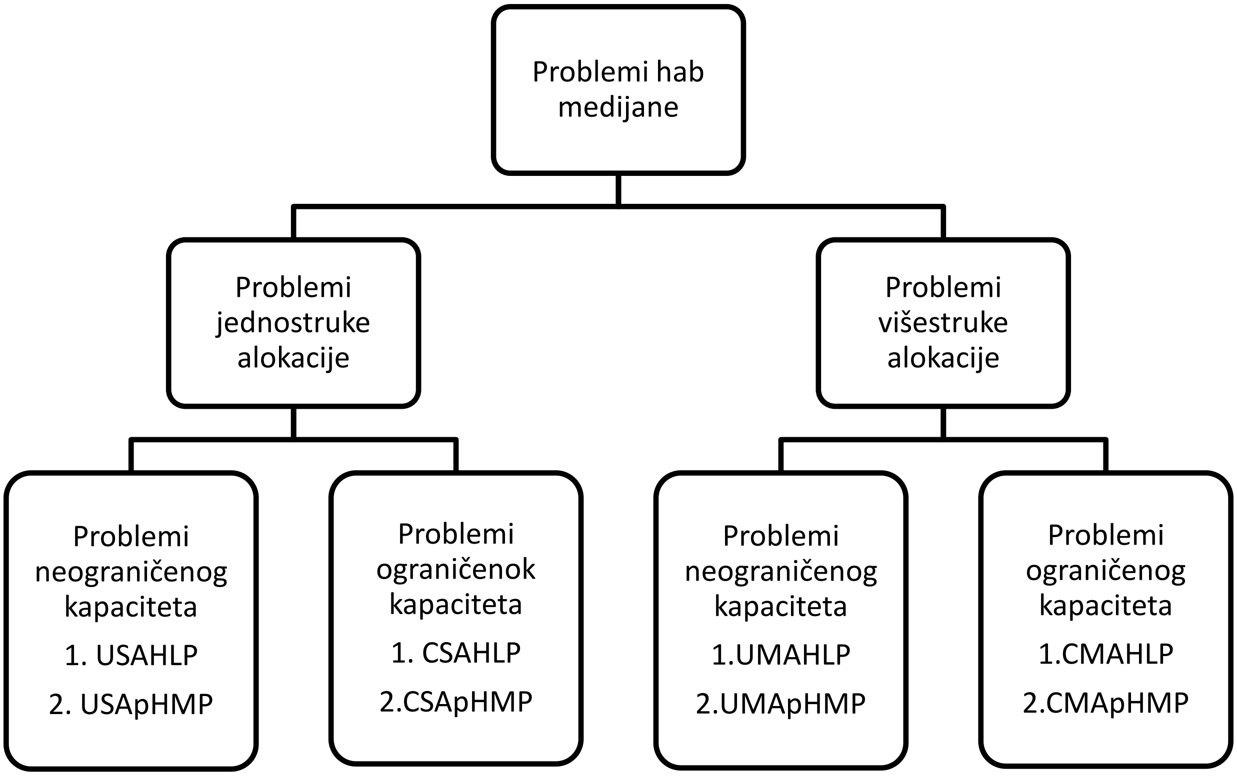
Prema kapacitetu , tj. ograničenjima količine protoka koji se može obraditi u svakom habu, hab lokacijski problemi se mogu klasifikovati:

- Hab lokacijski problem neograničenog kapaciteta (Uncapacitated hub location problems)
- Hab lokacijski problemi ograničenog kapaciteta (Capacitated hub location problems). Ograničenja kapaciteta u mreži mogu biti: ograničena količina robe koja dolazi ili prolazi kroz hab čvor, ograničen protok između habova kao i između habova pridruženih korisnika/snabdevača, ograničeni kapaciteti ne-hab čvorova i sl.

Hab lokacijski problemi ograničenih kapaciteta imaju veliki praktični značaj, ali su u odnosu na probleme neograničenih kapaciteta, zbog dodatnih ograničenja, teži za rešavanje.

4.5.1 Problemi hab medijane

Problemi hab medijane minimizuju sumu transportnih troškova za svaki odlazno-dolazni par preko odgovarajućeg skupa habova. Ukoliko je unapred zadat broj habova (p) koje treba locirati, govorimo o problemima p-hab medijane. U slučaju da broj habova nije unapred definisan, potrebno je platiti fiksne troškove za uspostavljanje habova. Problemi p-hab medijane minimizuju ukupne troškove transporta (vreme, rastojanje,...) da bi se opslužilo n zadatih čvorova, protok, saobraćaj između odlazno-dolaznih parova i broj habova koje treba locirati (p). Šema problema hab medijene prikazana je na Slici 21.



Slika 21. Problemi hab medijane

4.5.1.1 USAHLP

Hab lokacijski problem neograničenih kapaciteta sa jednostrukim alokacijama pozicije habova (**Uncapacitated single allocation hub location problem, USAHLP**) je problem neograničenog kapaciteta gde je potrebno locirati habove i njima pridružiti ne-hab čvorove tako da se minimizuje suma transportnih troškova u mreži i fiksnih troškova lociranja habova. Broj habova nije unapred definisan, ne-hab čvorovi mogu biti dodeljeni tačno jednom habu i postoje fiksni troškovi za uspostavljanje svakog haba. *O'Kelly* je pokazao u radu [O'Ke87] da je USAHLP NP-težak problem.

Neka su:

Z_{ik}=1 ako je ne-hab čvor *i* pridružen uspostavljenom habu *k*, inače 0,

Z_{kk}=1 ako je hab lociran u čvoru *k*, inače 0,

W_{ij}- količina robe koju treba transportovati od *i* do *j*,

D_{ij} – rastojanje između čvorova *i* i *j*,

f_k- fiksni trošak uspostavljanja haba *k*.

Formulacija USAHLP:

$$min \sum_{i,j,k,l \in I} W_{ij} (\chi D_{ik} Z_{ik} + \alpha D_{kl} Z_{ik} Z_{jl} + \delta D_{jl} Z_{jl}) + \sum_j Z_{jj} f_j \quad (1)$$

uz ograničenja:

$$\sum_k Z_{ik} = 1, \text{ za } \forall i \quad (2)$$

$$Z_{kk} - Z_{ik} \geq 0, \text{ za } \forall\, i, k \quad (3)$$

$$Z_{ik} \in \{0, 1\}, \text{ za } \forall\, i, k. \quad (4)$$

Funkcija cilja minimizuje sumu troškova kolekcije, transfera, distribucije i troškova uspostavljanja habova (1). Ograničenjem (2) se postiže jednostruka alokacija (ne-hab čvor je pridružen tačno jednom habu), a (3) da je čvor *i* prdružen čvoru *k* samo ako je *k* izabran za hab. Binarna prezentacija promenljive Z_{ik} je postignuta uslovom (4).

Za rešavanje USAHLP u radu [Abd98] navode se tabu pretraživanje i hibridni genetski algoritam, dok je u radu [Abd98a] opisano rešavanje ovog problema pomoću genetskog algoritma i metode grananja i ograničavanja (Branch-and-bound). Metoda grananja i odsecanja (branch-and-cut) za rešavanje USAHLP je navedena u [Lab03].

4.5.1.2 CSAHLP

Hab lokacijski problem ograničenih kapaciteta sa jednostrukim alokacijama pozicije habova **(Capacitated single allocation hub location problem**, **CSAHLP)** je problem čiji je cilj minimizovati ukupne troškove transporta u mreži i fiksne troškove uspostavljanja habova. Ovaj problem uveden je radi poboljšanja efikasnosti rada poštanskih sistema. Neka je zadato *n* poštanskih okruga (prikazani čvorovima) i neka je poznata prosečna prognoza količine pošte između svih parova okruga. Pošta, koju treba transportovati između svih odlazno-dolaznih parova mora da se usmeri kroz jedan, ili najviše dva, konsolidaciona centra (haba). Osim što je omogućen ekonomičniji prevoz (kroz konsolidaciju poštanskog saobraćaja), habovi predstavljaju i centre distribucije i sortiranja.

Neka su:

D_{ij} - rastojanje između čvorova *i*, *j*,

W_{ij} - količina protoka od snabdevača *i* do korisnika *j*,

Γ_k - kapacitet haba *k*,

f_k - fiksni troškovi uspostavljanja haba *k*,

$O_i = \sum_{j \in I} W_{ij}$, količina protoka koji polazi iz čvora *i*,

$D_j = \sum_{i \in I} W_{ij}$, količina protoka koji dolazi u čvor *j*,

Z_{ik}=1 ako je čvor *i* pridružen habu *k*, inače 0,

Y_{kl}^{i} = količina saobraćaja koja polazi od snabdevača *i*, sakuplja se u habu *k* i distribuira preko haba *l*.

Uz gore navedenu notaciju matematička formulacija CSAHLP je:

$$min \sum_{i=1}^{n} \sum_{k=1}^{n} D_{ik} Z_{ik} (\chi O_i + \delta D_i) + \sum_{i=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \alpha D_{kl} Y_{kl}^{i} + \sum_{k=1}^{n} f_k Z_{kk} \tag{1}$$

uz ograničenja:

$$\sum_{k=1}^{n} Z_{ik} = 1, \quad \forall\ i \in I \tag{2}$$

$$Z_{ik} \le Z_{kk}, \quad \forall\ i, k \in I \tag{3}$$

$$\sum_{j=1}^{n} W_{ij} Z_{jk} + \sum_{l=1}^{n} Y_{kl}^{i} = \sum_{l=1}^{n} Y_{lk}^{i} + O_i Z_{ik}, \quad \forall\ i, k \in I \tag{4}$$

$$\sum_{i=1}^{n} O_i Z_{ik} \le \Gamma_k Z_{kk}, \quad \forall\ k \in I \tag{5}$$

$$Y_{kl}^{i} \ge 0, \quad \forall\ i, k, l \in I \tag{6}$$

$$Z_{ik} \in \{0, 1\}, \quad \forall\ i, k \in I \tag{7}$$

Funkcija cilja (1) minimizuje ukupne troškove transporta u mreži i fiksne troškove uspostavljanja habova. Ograničenja (2) i (3) obezbeđuju sprečavnje direktne komunikacije između ne-hab čvorova. Konzervacija protoka u mreži je ograničenje (4), a (5) ograničava količinu protoka koja ulazi u svaki hab (odnosi se na kapacitet haba). Ograničenja (6) i (7) ukazuju na ne negativnu i binarnu prezentaciju promenljivih.

4.5.1.3 UMAHLP

Problem neograničene višestruke alokacije pozicije habova (**Uncapacitated multiple allocation hub location problem - UMAHLP**) ne postavlja ograničenja na kapacitet habova, broj habova nije unapred zadat i svaki ne-hab čvor može biti pridružen većem broju habova. U zavisnosti od potreba samog modela određuje se neophodni broj habova čije lociranje dovodi do izvesnih fiksnih troškova. Cilj je naći optimalno lokacijsko-alokacijsko rešenje kojim se minimizuju ukupni troškovi (suma transportnih i fiksnih troškova). U radu [O'Ke87] je dokazano da su UMAHLP NP-teški. Ukoliko je broj habova unapred definisan, na primer *p*, u pitanju je problem UMApHMP.

Neka su:

y_k – uzima vrednost 1 ako je hab lociran u čvoru *k*, 0 u suprotnom,

x_{ijkm} – deo toka od W_{ij} iz čvora *i* koji je prikupljen u habu *k*, distribuisan prema habu *m* i prosleđen u čvor *j*,

f_k – fiksni troškovi uspostavljanja haba *k* (za UMApHMP $\sum_{k \in J} f_k$ je obično 0 jer uglavnom nema fiksnih troškova lociranja habova),

D_{ij} – rastojanje između čvorova *i* i *j* (rastojanje na bazi troškova, koje je simetrično i zadovoljava nejednakost trougla),

W_{ij} – transportni troškovi po jedinici saobraćaja od *i* do *j*, $W_{ij} = \sum_{k \in J} \sum_{m \in J} x_{ijkm}$.

Parametri χ i δ predstavljaju jedinične troškove prikupljanja i raspodele, a 1-α predstavlja popust u ceni za transport između habova.

Koristeći gornju notaciju, UMAHLP se matematički može zapisati kao [Sta09]:

$$min \sum_{i,j,k,m} W_{ij} \left(\chi D_{ik} + \alpha D_{km} + \delta D_{mj} \right) x_{ijkm} + \sum_k f_k y_k \qquad (1)$$

uz uslove:

$$\sum_{k,m} x_{ijkm} = 1 \quad \forall i, j \tag{2}$$

$$\sum_{m} x_{ijkm} + \sum_{m,m\neq k} x_{ijkm} \leq y_k \quad \forall i, j, k \tag{3}$$

$$y_k \in \{0, 1\}, \forall k \tag{4}$$

$$x_{ijkm} \geq 0, \forall i, j, k, m \tag{5}$$

Funkcija cilja (1) minimizuje sumu troškova transporta $W_{ij}(\chi D_{ik}+\alpha D_{km}+\delta D_{mj})$ putanje $i \to k \to m \to j$ sa habovima u čvorovima *k* i *m* (snabdevač-hab, hab-hab i hab-korisnik, pomnoženi redom sa faktorima χ, α i δ) i fiksnih troškova uspostavljanja habova. Uslov (2) obezbeđuje da se celokupan saobraćaj šalje između svih parova habova, a (3) omogućava da se saobraćaj obavlja samo kroz otvoren hab. Ograničenje (4) omogućava binarnu prezentaciju y_k, a (5) ne-negativnu reprezentaciju promenljive x_{ijkm}. Uzimajući u obzir da ne postoji ograničenje kapaciteta habova, promenljive x_{ijkm} će u optimalnom rešenju biti binarne uzimajući vrednost 1 samo onda kada predstavljaju najjeftiniju put od *i* do *j*.

U radu [Cán04] je korišćena je metoda grananja i ograničavanja (Branch-and-bound) za rešavanje UMAHLP. U radu [Chen07] za nalaženje optimalnog broja habova UMAHLP korišćena je efektivna heuristika (effective heuristic) bazirana na metodi simuliranog kaljenja (simulated annealing) i tabu listi.

4.5.1.6 UMApHMP

Problem p-hab medijane neograničenih kapaciteta sa višestrukim alokacijama **(Uncapacitated multiple allocation p hub median problem-UMApHMP)** je problem ograničenog kapaciteta gde je cilj naći optimalno lokacijsko-alokacijsko rešenje kojim se minimizuju ukupni troškovi (suma transportnih i fiksnih troškova). Ovaj problem pripada grupi NP teških problema, [O'Ke87]. Jedna od formulacija problema p-hab medijane neograničenih kapaciteta sa višestrukim alokacijama jeste mešovita celobrojna linearna formulacija, [Bol04].

Neka je *I={1,...,n}* skup različitih čvorova mreže, pri čemu svaki čvor označava lokaciju korisnika/snabdevača ili potencijalnu lokaciju haba. Za svaki par čvorova (*i*, *j*), W_{ij} je količina robe koju treba transponovati od *i*-tog snabdevača do *j*-tog korisnika, D_{ij} rastojanje između njih. Neka je broj habova koje treba locirati fiksiran na p i neka su promenljive H_j, Z_{ik}, Y_{kl}^{i}, X_{lj}^{i} definisane na sledeći način:

- H_j uzima vrednost 1 ako je hab lociran na *j*-tom čvoru, inače 0

- Z_{ik} predstavlja količinu robe koja polazi od *i*-tog čvora a sakuplja se u habu *k*
- Y_{kl}^{i} je količina robe koja polazi od *i*-tog čvora, sakuplja se u habu *k* i distribuira preko haba *l*
- X_{lj}^{i} predstavlja količinu robe koja kreće od čvora *i*, čije je odredište čvor *j*, a transportuje se preko haba *l*.

Protoк od čvora-snabdevača do čvora-korisnika sastoji se od tri komponente: transfera od snabdevača do prvog haba, transporta između habova i distribucije od poslednjeg haba do korisnika, sa podrazumevanom šemom višestruke alokacije. Parametri χ i δ redom označavaju troškove (cenu) kolekcije i distribuciju robe po jedinici količine, dok 1-α predstavlja koeficijent uštede za transpot između habova.

Matematička formulacija UMApHMP glasi [Sta04]:

$$\min \sum_{i} \left[\chi \sum_{k} D_{ik} Z_{ik} + \alpha \sum_{k} \sum_{l} D_{kl} Y_{kl}^{i} + \delta \sum_{i} \sum_{j} D_{lj} X_{lj}^{i} \right] \qquad (1)$$

uz uslove:

$$\sum_{j} H_j = p \qquad (2)$$

$$\sum_{k} Z_{ik} = \sum_{j} W_{ij} , \qquad \text{za } \forall\, i \qquad (3)$$

$$\sum_{l} X_{lj}^{i} = W_{ij} , \qquad \text{za } \forall\, i, j \qquad (4)$$

$$\sum_{l} Y_{kl}^{i} + \sum_{j} X_{kj}^{i} - \sum_{l} Y_{lk}^{i} - Z_{ik} = 0 , \qquad \text{za } \forall\, i, k \qquad (5)$$

$$Z_{ik} \le \sum_{j} W_{ij} H_k , \qquad \text{za } \forall\, i, k \qquad (6)$$

$$\sum_{i} X_{lj}^{i} \le \sum_{i} W_{ij} H_l , \qquad \text{za } \forall\, i, j \qquad (7)$$

$$X_{lj}^{i}, Y_{lk}^{i}, Z_{ik} \ge 0 , \; H_k \in \{0,1\}, \qquad \text{za } \forall\, i, j, k, l \qquad (8)$$

Funkcija cilja (1) minimizuje sumu transportnih troškova snabdevač-hab, hab-hab i hab-korisnik pomnoženih sa koeficijentima χ, α i δ respektivno. Uslovom (2) se fiksira broj uspostavljenih habova na *p*, a uslovi (3)-(5) predstavljaju za svaki čvor *i* jednačine divergencije protoka u mreži. Ograničenja (6) i (7) ne dopuštaju direktnu komunikaciju između ne-hab čvorova, a (8) označava ne-negativnu i/ili binarnu reprezentaciju promenljivih H_j, Z_{ik}, Y_{kl}^i i X_{lj}^i.

Egzaktna metoda za rešavanje UMApHMP predložena je u [Soh98]. Metod najkraćeg puta (shortest path method) za prebrojavanje svih mogućih lokacija habova u UMApHMP korišćen je u [Ern98]. Ovaj algoritam na instancama problema sa relativno malim brojem habova koje treba uspostaviti daje tačna rešenja u razumnom vremenu računanja, dok na instancama većih dimenzija, metod najkraćeg puta se može uspešno kombinovati sa branch-and-bound algoritmom. Do sada je ovim algoritmom optmalno rešen problem do najviše 100 čvorova i on predstavlja najefikasniju egzaktnu metodu za rešavanje UMApHMP.

U literaturi postoje brojne heuristike za rešavanje UMApHMP. Lagranževa relaksacija je najefikasnija heuristika za rešavanje problema *p*–hab medijane [Pir98], a Lagranževu relaksaciju zajedno sa algoritmom grananja i ograničavanja za rešavanje UMApHMP koristio je *Klincewich* [Kli96]. Pokazuje se da je tabu pretraživanje efikasna metoda za rešavanje UMApHMP. Rešavanje ovog problema pomoću tabu pretraživanja predstavljen je u radu [Sko94]. Posle izbora skupa od p habova, u prvoj fazi korisnici/snabdevači se pridružuju njima najbližim habovima, čime se dobija početno rešenje za tabu search algoritam. TS ispituje različite mogućnosti pridruživanja svakog ne-hab čvora nekoj od hab-lokacija iz prethodno izabranog skupa. Kasniji rad Skorin-Kapova [Sko96] dokazuje da predloženi TS algoritam nalazi optimalno rešenje za sve testirane instance ovog problema.

Neke manje instance ovog problema rešavane su pomoću neuronskih mreža, [Dom03]. U radu [Per04] prikazana je evolutivna metaheuristika pod nazivom metoda ponovnog povezivanja puteva (Path relinking method-PR), koja koristi prostore susedstva kao osnovu za rekombinaciju rešenja.

4.5.2 Problemi hab centra

Problemi hab centra lociraju (*p*) habova unutar mreže tako da se minimizuje maksimalna udaljenost/vreme/troškovi transporta između parova korisnik-snabdevač. Problemi hab centra imaju veliku primenu kod brzih ili vremenski ograničenih sistema isporuka,kod lociranja službe za pružanje pomoći kao što je lokacija vatrogasnih stanica ili službe za hitnu pomoć. U ovakvim sistemima dato maksimalno vreme predstavlja najbolje vreme koje se može garantovati svim klijentima. Problemi p-hab centra spadaju u NP-teške probleme, što su pokazali *Kariv* i *Hakimi*, [Kar79].

4.5.2.1 HCSAP

Problem hab centra jednostruke alokacije (**Hub center single allocation problem, HCSAP**), alocira svaki ne-hab čvor tačno jednom habu tako da se minimizuju maksimalni troškovi transporta između svih odlazno-dolaznih parova. U [Ern04a] je dokazano da je HCSAP NP-težak problem.

Neka je $I=\{1,..,n\}$ skup čvorova, J skup habova, $J \subseteq I$, D_{ij} rastojanje od i-tog do j-tog čvora (najkraće rastojanje od i do j), troškovi hab luka $[k, m]$ se definišu kao αD_{km} (α zadati faktor uštede transporta među habovima), troškovi luka koji spaja ne-hab čvor sa habom jednaki su dužini datog luka. Rastojanje od polaznog čvora i do odredišta j, kroz habove k i m, se računa po formuli $D_{ik}+\alpha D_{km}+D_{mj}$. U slučaju da je $i=j$, odnosno saobraćaj ide od čvora pa natrag, rastojanje je zadato sa $2D_{im}+D_{kk}$. Promenljiva Z_{ik} je jednaka 1 ako je čvor i pridružen habu k, inače 0.

Sada možemo uvesti matematičku formulaciju HCSAP, [Ern04a]:

$$\min \quad \max_{i,j\in I,k,m\in J} (D_{ik}+\alpha D_{km}+D_{mj})Z_{ik} Z_{jm} \qquad (1)$$

uz uslove:

$$\sum_{k\in J} Z_{ik} = 1, \text{ za } \forall\, i \in I \qquad (2)$$

$$Z_{ik}\in\{0, 1\}, \ \forall\, i \in I, \forall\, k \in J \qquad (3)$$

Funkcija cilja (1) minimizuje maksimalne transportne troškove između svih odlazno-dolaznih parova. Uslovi (2) i (3) garantuju da će čvor biti pridružen tačno jednom habu.

U [Mey09] opisan je dvofazni algoritam za rešavanje problema p-hab centra jednostruke alokacije. U prvoj fazi algoritam koristi metodu grananja i ograničavanja pomoću koje računa skup potencijalnih optimalnih hab kombinacija. Da bi se dobila dobra gornja granica algoritma grananja i ograničavanja razvijena je heuristika za probleme p-hab centra jednostruke alokacije na osnovu mravljih kolonija. Nakon toga sledi faza raspodele, koristeći formulaciju smanjene veličine koja daje optimalno rešenje. Prva heuristika za rešavanje problema p–hab centra jednostruke alokacije je heuristika jednostruke relokacije (**Single-relocation heuristics**), [Pam01]. Ova heuristika se bazira na jednostavnim zamenama habova sa ne-hab čvorovima, primenjujući tabu pretraživanje da bi se izbegao lokalni minimum.

4.5.2.2 UMApHCP

Problem p-hab centra neograničenih kapaciteta sa višestrukim lokacijama **(Uncapacitated multiple allocation p-hub center problem, UMApHCP)** uspostavlja p habova na nekim od n potencijalnih lokacija tako da se minimizuje maksimalno rastojanje, odnosno troškovi ili vreme transporta između bilo koja dva čvora u mreži preko jednog ili više habova.

Formulacija UMApHCP koristi nove promenljive. Neka je Y_{ijkm} binarna promenljiva koja uzima vrednost 1 ako se saobraćaj od *i* -tog do *j* -tog čvora usmerava kroz habove *k* i *m* . Binarna promenljiva Z_k jednaka je 1 ako i samo ako je čvor *k* izabran za hab. Promenljive D_{ij}, W_{ij}, Γ_k, α, *I* i *J* imaju isto značenje kao u prethodnim poglavljima.

Sada možemo uvesti matematičku formulaciju UMApHCP [Ern09].

$$Min\ Z \qquad (1)$$

uz uslove:

$$\sum_{k \in I} \sum_{m \in I} Y_{ijkm} = 1, \quad \forall\, i, j \in I \qquad (2)$$

$$\sum_{k \in I} Y_{ijkm} \leq Z_m, \quad \forall\, i, j, m \in I \qquad (3)$$

$$\sum_{m \in I} Y_{ijkm} \leq Z_k, \quad \forall\, i, j, k \in I \qquad (4)$$

$$Z \geq \sum_{k \in I} \sum_{m \in I} Y_{ijkm} (D_{ik} + \alpha D_{km} + D_{mj}), \quad \forall\, i, j \in I \qquad (5)$$

$$Z_k,\ Y_{ijkm} \in \{0, 1\}, \quad \forall\, i, j, k, m \in I \qquad (6)$$

$$\sum_{k=1}^{n} Z_k = p \qquad (7)$$

Funkcija cilja, *Z*, minimizuje maksimalne troškove između svih polaznih i krajnjih odredišta, (1). Uslovom (2) je obezbeđena jedinstvena putanja od početnog do krajnjeg odredišta. Uslovi (3) i (4) obezbeđuju da ako je čvoru pridružen neki drugi čvor, onda on mora biti selektovan za hab, dok je uslovom (5) definisana najniža donja granica za promenljivu *Z* funkcije cilja. Uslov (6) omogućava binarnu reprezentaciju promenljivih. Konačno, uslovom (7) je definisano tačno *p* habova. Problemi UMApHCP su NP-teški, ali alokacijski deo problema je polinomijalno rešiv za $O(pn^2)$ vremena, rešavanjem niza problema najkraćeg puta [Cam07, Ern09].

U radu [Ern04] date su dve ceobrojne formulacije problema i prikazane dve heurističke metode EH1 i EH2 za rešavanje varijacija ovog problema sa šemom jednostruke alokacije i sa šemom višestruke alokacije. Ove heuristike, od kojih je bolja EH2, relativno brzo dovode do rešenja, ali u nekim slučajevima rešenja su nezadovoljavajuća, čak i za probleme malih dimenzija. Egzaktni brunch-and-bound algoritam (EBnB) opisan u [Ern04] zasnovan je na metodi najkraćih puteva i metodi grananja i ograničavanja. Ovaj metod brzo dodvodi do rešenja, ali je kod nekih instanci odstupanje dobijenog rešenja od optimalnog i preko 10%. U radu [Ern09] date su dve nove IP formulacije i opisana je BnB metoda za rešavanje UMApHCP-a.

Genetski algoritam za rešavanje UMApHCP, predložen u [Sta04] koristi binarnu reprezentaciju jedinki, odnosno genetski kod duzine n. Svaki gen tog koda uzima vrednost 1 ako je na toj poziciji uspostavljen hab, u suprotnom 0. Niz (Z_k) se dobija iz genetskog koda, a kako korisnici/snabdevači mogu biti pridruženi isključivo uspostavljenim habovima, vrednost promenljive Y_{ijkm} se dobija tokom računanja funkcije cilja koja je identički jednaka funkciji prilagođenosti. Za konačan skup habova koristi se *Floyd-Warshall-ov* algoritam najkraćeg puta. Za svaki par čvorova u mreži, nakon nalaženja najkraćih puteva, lako se računa funkcija cilja jednostavnim sumiranjem najkraćih rastojanja snabdevač-hab, hab-hab i hab-korisnik pomnoženim odgovarajućim parametrima α, χ i δ.

4.5.3 Problemi hab pokrivanja

Problemi hab pokrivanja **(Hab covering problems)** imaju veliku primenu u određivanju lokacija za hitne službe, vatrogasnih i stanica hitne pomoći, za postizanje pokrivenosti samo na određenim udaljenostima od snabdevača, kao što su repetitori mobilnih telefona itd. Jedna od najvećih primena hab lokacijskih problema pokrivanja je u oblasti transporta. Za razliku od većine lokacijskih problema koji su usmereni na saobraćaj uz minimalne troškove zanemarujući vreme isporuke, problemi hab pokivanja uzimaju u obzir vreme putovanja između odlazno-dolaznih parova.

Smatra se da je određeno područje ''pokriveno'' uz pomoć datog objekta, ukoliko se nalazi na rastojanju manjem od nekog unapred definisanog ''kritičnog'' rastojanja. Problemi hab pokrivanja se dodatno razlikuju od standardnih problema pokrivanja zbog interakcije između habova koje stvaraju dodatne komplikacije, jer su odlazno-dolazni parovi pokriveni parovima habova i naplata troškova se vrši po svakom habu. Ovi problemi pripadaju klasi NP-teških problema [Kar03].

Radijus pokrivenosti je maksimalno vreme putovanja kojim se utvrđuje pokrivenost klijenta. Objekti pokrivanja mogu biti potpuno ili individualno pokriveni, odnosno nepokriveni. Kod potpune pokrivenosti svaki klijent je u radijusu pokrivenosti snabdevača, dok kod individualne pokrivenosti, pokrivenost klijenta zavisi od blizine njemu najbližeg snabdevača.

Najpoznatiji predstavnici problema hab pokrivanja su: modeli postepenog pokrivanja (***The gradual cover models),*** model kooperativnog pokrivanja (***The cooperative cover model***), model promenljivog radijusa (***Variable radius model***), lokacijski problemi promenljivog pokrivanja (***The variable radius location problem***).

Calic i saradnici su u radu [Cal09] predložili heuristiku za rešavanje problema hab pokrivanja. Predložena heuristika za rešavanje problema hab pokrivanja sa nekompletnom mrežom habova koja je bazirana na tabu pretraživanju, pokazala se veoma efikasnom. Heuristika se sastoji od faza izgradnja i poboljšanja. Algoritam polazi od skupa potencijalnih lokacija habova, zatim alocira ne-hab čvorove ka habovima i formira hab mrežu. Primetimo da algoritam ne startuje sa kompletnim, već sa delimičnim rešenjem, tj. skupom habova, koje ne garantuje izvodljivu alokacijsku raspodelu. Tokom faze izgradnje rešenja korišćene su tri različite konstrukcijske metode bazirane na alokacijskoj strategiji.U svakoj od ovih alokacijskih strategija, moguća rešenja se traže u okviru komletne hab mreže. Kad je pronađeno moguće rešenje, u fazi poboljšanja, hab lukovi koji ne vode do neizvodljivosti se uklanjaju iz hab mreže da bi se dobilo što bolje dopustivo rešenje [Rus11].

4.6 Kvadratni problem pridruživanja

Kvadratni problem pridruživanja (***Quadratic Assignment Problem,*****QAP***)*, prvi su formulisali *Koopmans i Beckmann* , [Koo57] 1957. godine, prilikom rešavanja problema određivanja lokacija za skup objekata uz minimizaciju ukupnih troškova toka materijala među objektima. Ulazni podaci problema su tri matrice: matrica toka između objekata $A = (a_{ij})$, gde je a_{ij} tok materijala između objekata i i j, matrica udaljenosti $B = (b_{kl})$, gdje je b_{kl} udaljenost između lokacija k i l i matrica $C = (c_{ik})$ koja predstavlja trošak postavljanja i-tog objekta na k-tu lokaciju. Potrebno je naći takav raspored objekata na lokacijama kako bi ukupan trošak bio minimalan.

Kvadratni problem pridruživanja spada u klasu NP-teških problema, što su dokazali *Sahni i Gonzales* [Sah76], zbog čega je u praksi optimalno rešavanje kvadratnog problema pridruživanja nemoguće za veličine problema veće od 20. Takođe je pokazano da QAP spada među najteže optimizacijske problem, jer se čak i nalaženje približnog rešenja smatra teškim problemom.

Problem pridruživanja elemenata jednog skupa elementima drugog skupa se pojavljuje u mnogim praktičnim problemima alokacije resursa. Ukoliko su pridruženi troškovi konstante za svako moguće sparivanje, tada se radi o klasičnom problemu pridruživanja. S druge strane, ukoliko je struktura troškova složenija, tako da trošak određenog pridruživanja zavisi i od drugih sparivanja, dolazimo do teških kombinatornih problema, poput kvadratnog problema pridruživanja.

Kvadratni problem pridruživanja se primenjuje kao model kod problema raspoređivanja, u elektronici, paralelnog i distribuiranog računanja, sportu, hemiji, proizvodnji računara. Više o primeni QAP-a može se naći u [Ste61], [Els77].

4.6.1 Formulacija QAP-a

Problem se može modelirati pomoću dve matrice dimenzija n×n:

A = (a_{ij}), gde je a_{ij} kretanje od objekta i do objekta j

B = (b_{kl}), gde je b_{kl} udaljenost između lokacija k i l

Kvadratni problem pridruživanja sa matricama koeficijenata *A* i *B*, koji ćemo označavati sa QAP(A,B), se tada formuliše kao :

$$\min_{\pi \in S_n}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} a_{\pi(i)\pi(j)} b_{ij}\right)$$

gde je S_n skup svih permutacija celih brojeva od 1 do n. Svaki broj $a_{\pi(i)\pi(j)} b_{ij}$ je cena dobijena dodelom objekta π_i lokaciji *i* i objekta π_j lokaciji j. Ova formulacija QAP-a se još zove i ***Koopmans-Beckmannov QAP***, prema imenima istraživača koji su ga prvi formulirali [Koo57]. Ukoliko su matrice *A* i *B* simetrične, tada QAP(A,B) zovemo *simetrični QAP*, inače se radi o asimetričnom QAP-u.

Opštiju definiciju QAP-a postavio je *Lawler* u [Law63]:

$$\min_{\pi \in S_n}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} d_{\pi(i) i \pi(j) j}\right)$$

Gde je $D = d_{kilj}$ 4-dimenzionalno polje realnih bojeva, *i, j, k, l = 1, 2, ..., n.*

4.6.2 Metode za rešavanje QAP-a

Egzaktne metode koje se koriste za rešavanja QAP-a su dinamičko programiranje, metode presecanja ravni i metode grananja i ograničavanja. Od ove tri metode, metoda grananja i ograničavanja se pokazuje kao najefikasnija.

Za rešavanje kvadratnog problema pridruživanja, metodom grananja i ograničavanja postojeći algoritmi se mogu podeliti u tri grupe:
- metode jednostrukog pridruživanja (***Single assignment algorithms***) koje je razvio *Gilmore* [Gli62].
- metode pridruživanja u paru (***Pair assignment algorithms***) koje su opisali *Gavett i Plyter* [Gav66], ali numerička izračunavanja su pokazala da ovaj pristup ne daje dobre rezultate.
- metode relativnog pozicioniranja (***Relative positioning algorithm***) koje su predložili *Mirchandani i Obata* u radu [Mir79].

Sve tri metode polaze od prazne permutacije, koja se u toku izvršavanja algoritma proširuju do potpune permutacije. Numerički eksperimenti su pokazali da među navedena tri tipa algoritma grananja i pretraživanja, metoda jednostrukog pridruživanja daje najbolje rezultate, dok se metoda relativnog pozicioniranja eventualno može koristiti kod problema sa retko popunjenim matricama.

Kako egzaktni algoritmi za rešavanje kvadratnog problema pridruživanja teško mogu rešiti probleme dimenzija većih od 20, jedina mogućnost je da se problem pokuša rešiti heurističkim algoritmima. Međutim, kod primene heurističkih metoda, nemoguće je dokazati optimalnost rešenja, pa se kvalitet rešenja mora uporediti sa rešenjima koja su dobijena nekim drugim metodama. Najelementarniji pristup je metoda najbržeg spusta (***Steepest descent method***), [Fed01], koja iz zadate početne tačke u prostoru rešenja kreće u smeru najbržeg smanjivanja funkcije cilja (ukoliko se radi o problemu minimizacije) dok ne dođe do minimuma funkcije. Ova metoda nalazi rešenje koje predstavlja lokalni minimum, pa je izrazito neprikladna ukoliko imamo višedimenzionalni konfiguracijski prostor.

Najpoznatiji algoritmi koji imaju ugrađen u sebi mehanizam kojim izlaze iz lokalnog minimum i nastavljaju pretraživanje su tabu pretraživanje (**Tabu search, TS**), simulirano kaljenje (**Simulated annealing, SA),** genetski algoritmi (**Genetic algorithms, GA**). Međutim, čak i ukoliko se spreči konvergencija algoritma u nekom od lokalnih minimum, i dalje postoji mogućnost da će se pojaviti granično kolo (*limit cycle*) koje predstavlja zatvorenu putanju u konfiguracijskom prostoru stanja kroz koje algoritam stalno prolazi. Tabu-search metode su eksplicitno definisane kako bi se sprečilo pojavljivanje takvih zatvorenih putanja, jer će lista zabranjenih poteza sprečiti ponavljanje određene sekvence pomaka u prostoru stanja.

Postoji i treća mogućnost, a to je pojava *haotičnog atraktora* (*chaotic attractor*), gde je pretraživanje ograničeno na određeni deo konfiguracijskog prostora, bez pojavljivanja fiksnih tačaka ili graničnih kola. Iako algoritam ne konvergira u određenoj tački (kao kod pojavljivanja fiksne tačke) niti stalno prolazi kroz određeni skup stanja (kao kod graničnog kola), ipak je pretraživanje ograničeno na određeni deo skupa rešenja, pa se globalni optimum neće naći, osim ukoliko se ne nalazi unutar tog skupa. Prepoznavanje pojavljivanja haotičnog atraktora, i nalaženje načina za izbegavanje, je od velike važnosti kod implementacije heurističkih algoritama.

Tabu pretraživanje korišćenjem liste zabranjenih poteza sprečava konvergenciju algoritma u lokalnom minimumu. Rešavanje QAP-a pomoću tabu pretraživanja predložila je *Skorin-Kapov* u radu [Sko90]. Verovatno je najefikasniji tabu search algoritam za rešavanje QAP-a ***robust tabu search*** (RoTS) koji je opisao *Taillard* u [Tai91]. On je u svojoj implementaciji primenio tehniku kojom se za svaku jedinku i svaku lokaciju zapisuje poslednja iteracija u kojoj je data jedinka bila na toj lokaciji. Potez se proglašava tabu potezom ukoliko se njime obe jedinke dodeljuje na lokacije koje su zauzimale tokom poslednjih *s* iteracija. Ovakav tip tabu poteza je vrlo sličan onom primenjenom kod Skorin-Kapov, ali je pogodniji kod tabu liste koja je promenjive veličine, što je odlika *Taillardove* implementacije.

Kod definisanja liste zabranjenih poteza primenjuju se dva pristupa. Prvi se sastoji u tome da se zabranjuju potezi zamene dve jedinke koje su bile zamenjene tokom poslednjih s iteracija (s je kontrolni parametar koji definiše veličinu tabu liste). Formalno se tabu lista sastoji od parova (i,j) jedinki koje ne mogu biti zamenjene, a vrlo se jednostavno implementira pomoću simetrične matrice gde element $A[i][j]$ sadrži u sebi broj iteracije u kojoj će ponovno biti dopuštena zamena ta dva elementa. Takav pristup je efikasan i vrlo brz, jer da bi se ustanovio status poteza troši se samo jedno poređenje sa elementom dvodimenzionalnog polja. Drugi se sastoji u tome da se za svaku jedinku i svaku lokaciju zapisuje poslednja iteracija kada je ta jedinka bila na toj lokaciji. Potez se proglašava tabu potezom ukoliko se njime obe jedinke dodjeljuje na lokacije koje su zauzimale unutar poslednjih *s* iteracija. Ovakav tip tabu poteza je pogodniji kod tabu liste koja je promenjive veličine.

Da bi se izbegle zavisnosti tabu search algoritama od spoljnih faktora (veličina tabu liste, odabir aspiracijske funkcije), *Battiti i Tecchiolli* u radu [Bat94] su predložili modifikaciju tabu search algoritma, nazvanu ***Reactive tabu search***, kod koga se automatski obavljaju modifikacije parametara u potrazi za što boljim rešenjem. Kod implementacije ovog algoritma, pored standardnih elemenata tabu-search procedure, definiše se automatizovan način prilagođavanja veličine liste zabranjenih problema u skladu sa trenutnim stanjem pretraživanja. Pored toga, definiše se i način diversifikacije pretraživanja kada standardni mehanizam tabu-search procedure zakaže. Reaktivni tabu-search algoritam se u praksi pokazao superiornijim, iako su zahtevi za memorijske resurese znatno veći budući da ima ugrađen mehanizam dugotrajne memorije koja se iskorišćava za praćenje napredovanja algoritma, uočavanje graničnih kola i haotičnih atraktora.

Simulirano kaljenje su na kvadratni problem pridruživanja primenili *Burkard i Rendl* [Bur84], koji su u svom radu implementirali elementarni raspored kaljenja koji se sastojao u smanjivanju temperature za određeni multiplikativni faktor, s time da se temepratura menjala nakon određenog, preddefinisanog broja obavljenih zamena. Za implementaciju algoritma simuliranog kaljenja, potrebno je definisati raspored kaljenja, odnosno skup $S = \{t1, t2, ..., tr\}$ gde je ($t1 > t2 > ... > tr$) i t_i predstavlja temperaturu u *i*-toj iteraciji kaljenja. *Wilhelm i Ward* u [Wil87] su koristili raspored kaljenja koji se prvi put pojavljuje kod *Kirkpatricka i Gelatta*, [Kir83], gde se predlaže da se raspored kaljenja definiše sledećim nizom temperatura: $t_i = (10)(0.9)^{i-1}$, za i =1, 2, ..., r. *Connoly* je u svom radu [Con90] uočio da potpuno slučajno biranje sledeće konfiguracije za ispitivanje može biti neefikasno iz dva razloga: jedan je što se na nižim temperaturama može propustiti ispitivanje nekih zanimljivih poteza, a drugi što se napori za pomeranje iz lokalnog optimuma mogu oslabiti zbog preranog odbacivanja uzlaznog poteza (poteza koji pogoršava funkciju cilja). *Thonemann i Bolte* su predložili poboljšan algoritam simuliranog kaljenja u [Tho94].

Genetski algoritmi primenom operatora ukrštanja i mutacije nad reprezentacijama rešenja pretražuju prostor rešenja, čime su se pokazali kao vrlo efikasan način za rešavanje QAP-a. Obećavajući heuristički algoritmi za rešavanje QAP-a trenutno su genetski algoritmi, tabu prtraživanje i GRASP metoda (Greedy randomized adaptive search), [Li94], dok se kao najefikasniji način rešavanja QAP-a pokazala primena hibridnih genetskih algoritama, gde je unutar genetskog algoritma ugrađen algoritam lokalnog pretraživanje koji poboljšava jedinke u populaciji.

Yongzhong je u [Yon07] opisao model genetskog algoritma koji je zasnovan na novoj strategiji zamena. Strategija zamena odnosi se na strategiju selekcije, koja definiše kako selektovati sledeću generaciju potomaka. Posebno je važna za problem sa velikim brojem ograničenja jer vodi algoritam pretraživanja kroz dozvoljeno područje i tako utiče na performanse algoritma. Najčešća upotrebljavana strategija zamena je *steady-state*. U svakoj generaciji jedinke su selektovane kako bi se nad njima izvršili genetski operatori. Svaki novi potomak upoređuje se sa najgorim članom u trenutnoj generaciji, i njime će biti zamenjen. Cilj nove strategije je poboljšanje sposobnosti globalnog pretraživanja algoritma. Uključuje dve različite strategije: zamena najlošijeg (*replace-worst*) i zamena roditelja (*replac-parent*).

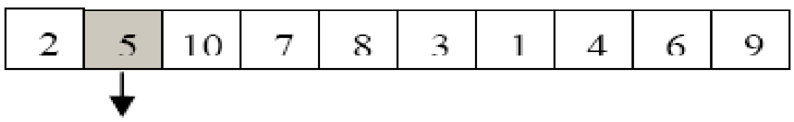
- strategija zamene najlošijeg (*replace-worst*) primjenjuje se jednom svakih T_c generacija. Prilagođenost potomka se upoređuje sa najgorom jedinkom u populaciji. Ako je prilagođenost potomka bolja nego prilagođenost najlošije jedinke tada zamenjujemo potomka i najlošiju jedinku. Prilagođavanjem učestalosti primene ove strategije, konvergencija procesa pretraživanja i raznolikost populacije može biti dobro uravnotežena.
- strategija zamene roditelja (*replace-parent)* primjenjuje se jednom u svakoj generaciji, svaki potomak upoređuje se sa svojim roditeljom (jedinkom pre mutacije). Ako je prilagođenost potomka bolja nego prilagođenost njegovog roditelja tada potomak zamenjuje roditelja.

Model genetskog algoritma zasnovanog na novoj strategiji zamena je sledeći [Yon07]:

```
Begin
     Konstruiši početnu populaciju;
      za svaku generaciju
            slučajno odaberi jedinke iz populacije;
            primeni genetske operatore na generisane potomke;
            if nije u i_Tc generaciji
                  uporedi svakog potomka sa njegovim roditeljom i
                  obriši lošijeg;
            inače
                  uporedi svakog potomka sa najlošijim članom
                  populacije i obriši lošijeg;
      ponavljaj dok nije zadovoljen jedan od uslova
End
```

U prikazanom modelu T_c se naziva period strategije zameni najlošijeg (*replace-worst).*

U genetskom algoritmu za QAP iskorišćen je prikaz permutacije.



Gen predstavlja da je objekt 5 postavljen na lokaciju 2

Slika 22. Izgled hromozoma

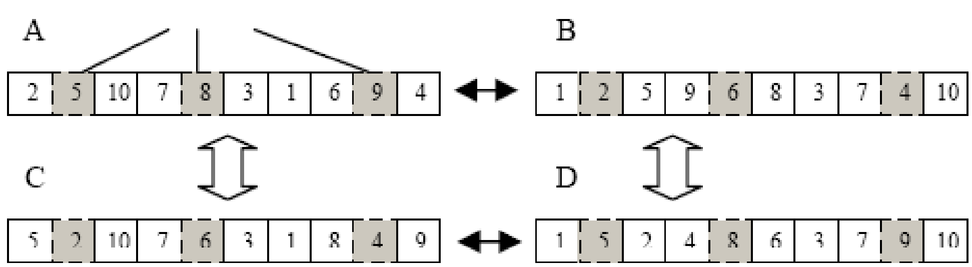
Vrednost svakog gena predstavlja objekat koji je dodjeljen određenoj lokaciji. U hromozomu prikazanom na Slici 22. postoji 10 objekata postavljenih na 10 lokacija. Npr. drugi gen u hromozomu znači da je objekt 5 postavljen na lokaciju 2.

Cilj genetskog algoritma je minimizacija ukupne cene C. Funkcija cilja za hromozom u genetskom algoritmu je definisana na sledeći način:

$$f_i = \frac{1}{C_i}$$

gde je f_i prilagođenost jedinke *i,* a C_i je stvarna vrednost.

Korišćen je specijalan tip uniformnog ukrštanja sa konstantnom verovatnoćom izmene gena. Procenat izmene gena je postavljen na malu vrednost tako što će svaki od dva potomka biti puno sličniji jednom od svojih roditelja. Nakon ukrštanja, jedinke A i B stvaraju C i D. Definišemo A kao direktnog roditelja od C pošto su međusobno slični, i definišemo B kao direktnog roditelja od D.



Slika 23.Operator ukršatanja

U svakoj generaciji svaki hromozom se sparuje sa drugim hromozom nasumično. Operator ukrštanja će biti primenjen na sve parove hromozoma (Slika 23.). U ovoj GA implementaciji nije korišćen operator mutacije.

Zaključak

Poslednih godina javlja se veliko interesovanje za teoriju lokacije, a razlozi su mnogobrojni. Donošenje odluka o lociranju nekog objekta je vrlo prisutna pojava na makro i na mikro ekonomskom nivou, u inženjerskoj praksi, u javnim službama, vojsci, itd.

U ovom radu opisani su sledeći lokacijski problemi na mrežama: prost lokacijski problem, problem p medijane, problem p centra, lokacijski problem maksimalnog pokrivanja, lokacijski problem anti-pokrivanja, osnovni hab lokacijski problemi i kvadratni problem pridruživanja. Za svaki od rešavanih problema date su matematička formulacija i kratak opis problema, kao i egzaktne i metaheurističke metode za rešavanje datih problema.

Lokacijski problemi su obično originalno definisani u skladu sa konkretnom postavkom problema. Ograničenja i kriterijumi optimalnosti tj. funkcija cilja, menjaju se u zavisnosti od konkretne problematike i moraju se identifikovati za svaki zadatak posebno.

Pokazuje se da izbor lokacija, a naročito izbor optimalnih lokacija, predstavlja izazovan teorijski problem. Reč je o vrlo složenim, multidisciplinarnim problemima gde su i postavke problema izvanredno složeni zadaci, a njihovo rešavanje traži poznavanje više matematičkih teorija i vrlo intenzivnu primenu računara.

Literatura

[Abd98] **Abdinnour-Helm S.,** ''A hybrid heuristic for the uncapacitated hub location problem'', European Journal of Operational Research, **106**, 489-499, (1998).

[Abd98a] **Abdinnour-Helm S., Venkataramanan M.,** "Solution approaches to hub location problems", Annals of Operations Research 78, 31-50, (1998).

[AbH98] **Abdinnour-Helm S.**, "A hybrid heuristic for the uncapacitated hub location problems", *European Journal of Operational Research*, Vol. 106, pp.489-499 (1998).

[AbH99] **Abdinnour-Helm S.**, "Network design in supply chain management", *Internatonal Journal of Agile Mangement Systems,* Vol. 1, No. 2., pp.99-106 (1999).

[Alp03] **Alp O., Erkut E., Drezner Z.** "An efficient genetic algorithm for the p-median problem" *Annals of Operations Research, 122:21–42*, (2003).

[Ape77] **Appel, Kenneth and Haken, Wolfgang**. "The Solution of the Four-Color-Map Problem," Scientific American, Volume 237 (1977): 108-121.

[Aqe90] **Aqeev A.A., Beresnev V.S.**, "Polynomially Solvable Cases of the Simple Plant Location Problem", In: *Proceeding of the First Integer Programming and Combinatorial Optimization Conference*, Kannan R. and Pulleyblank W.R. (eds), University of Waterloo Press, ON, Canada, pp. 1-6 (1990)

[Ave03] **Avella P., Sassano A., Vasil'ev I.** "A heuristic for large-scale p-median instances" *Electronic Notes in Discrete Mathematics, 13:14–17,* (2003).

[Bä00a] **Bäck T., Fogel D. B., Michalewicz Z**.: "Basic Algorithms and Operators", in: Evolutionary Computation *1*, Institute of Physics Publishing, Bristol-Philadelphia, (2000).

[Bä00b] **Bäck T., Fogel D. B., Michalewicz Z.:** "Advanced Algorithms and Operators", In: Evolutionary Computation 2, Institute of Physics Publishing, Bristol-Philadelphia, (2000).
http://msmga.ms.ic.ac.uk/jeb/orlib/info.html

[Bal65] **Balinski, M.L.,** “ Integer programming: Methods, uses, computation”, Management Science 12, 253–313, (1965).

[Bat94] **Battiti R., Tecchiolli G.,** “The Reactive Tabu Search”, *ORSA Journal on Computing* **6**, (1994).

[Bea90] **Beasley J.E.,** OR-library-distributing test problems by electronic mail, *Journal of the Operational Research Society,* 41: 1069-72, (1990). http://people.brunel.ac.uk/mastjjb/jeb/orlib/pmedinfo.html

[Bea93] **Beasley, J.E.** “Lagrangean heuristics for location problems”, *European Journal of Operational Research*, vol. 65, pp. 383-399, (1993).

[Bel10] **Belenguer J.M., Benavent E., Prins C., Prodhon C., Calvo R.W.,** “A Branch-and-Cut method for the Capacitated Location-Routing Problem”, Spain (2010).

[Ber03] **Berman O., Drezner Z., Weslowsky G.O**., “The expropriation location problem”,Journal of the Operational Research Society,(2003).

[BeJ93] **Beasley J.E.**, "Lagrangean heuristic for location problems", *European Journal of Operational Research*, Vol. 65, No. 3, pp. 383-399 (1993).

[Blm58] **Bellman R.E.**, "On a Routing Problem", Quarterly Applied Mathematics, Vol. 16, pp. 87-90 (1958).

[Bol04] **Boland N., Krishnamoorthy M., Ernst A.T., Ebery J.**, "Preprocessing and Cutting for Multiple Allocation Hub Location Problems", *European Journal of Operational Research*, Vol. 155, pp.638-653 (2004).

[Bom99] **Bomze I.M., Budinich M., Pardalos P.M., Pelillo M.,** “The maximum clique problem”, *Handbook of CombinatorialOptimization, Kluwer Academic Publisher, Dordrecht, The Netherlands,* pp. 1-74, (1999).

[Boz02] **Bozkaya, B., Zhang J., Erkut E. “** An efficient genetic algorithm for the p-median problem, In Facility location: Applications and theory, Drezner, Z.& Hamacher, H. (Ed.), pp.179-205, Springer Verlag Berlin Heidelbeg, ISBN 3-540-21345-7, New York (2002).

[Bra89] **Brandeau M., Chiu S**. “An overview of representative problems in location research”, *Management Science*, **35**,(1989) 645-674.

[Bur84] **Burkard R.E. i Rendl F.,"** A thermodynamically motivated simulation procedure for combinatoril optimization problems", *European Journal of Operational research* **17** (1984), 164-174.

[Cán04] **Cánovas L., García S., Marín A.,** "Solving the uncapacitated multiple allocation hub location problem by means of dual-ascent technique", *Working paper no 1*, Departamento de Estadısticae Investigacion Operativa, University of Murcia, Spain, http://www.optimization-online.org/DB FILE/2004/01/812.pdf, (2004).

[Cal09] **Calic H., Alumur S.A., Kara B.Y., Karasan O.E.,** "A tabu-search based heuristic for the hub covering problem over imcomplete hub network", *Computers & Operations Research 36, 3088-3096,* (2009).

[Cam96] **Campbell J.F.,** "Hub Location and the p-hub Median Problem", *Operations Research,* Vol. 44, No. 6, pp. 923-935 (1996)

[Cam02] **Campbell J.F.,Ernst A. and Krishnamoorthy M.,** "Hub Location Problems" in Hamacher H. and Drezner Z., eds.: Facility Location : Applications and Theory, Springer-Verlag, Berlin-Heidelberg, pp. 373-407 (2002).

[Ces03] **Ceselli A.** " Two exact algorithms for the capacitated p-median problem" *4OR: Quarterly Journal of the Belgian, French and Italian Operations Research Society,* 1:319–340, (2003).

[Ces05] **Ceselli A., Righini A.** "A branch-and-price algorithm for the capacitated p-median problem" Networks, 45(3):125–142, (2005).

[Cha86] **Chaudhry, S.S., McCormick, S.T., Moon, I.D.,"** Locating independent facilities with maximum weight: greedy heuristics", *Omega – International Journal of Management Science,* Vol. 14, No. 5, pp. 383-389, (1986).

[Chen07] **Chen J. F.,** "A hybrid heuristic for the uncapacitated single allocation hub location problem", *OMEGA The Internacional Journal of Menagement Science*, 35: 211-220. (2007).

[Chr82] **Christofides N., Beasley J.E.** "A tree search algorithm for the p-median problems", *European Journal of Operational Research*, vol. 10, pp. 196-204 (1982).

[Chu74] **Church, R. L. and ReVelle, C.** "The maximal covering location problem" *Papers of Regional Science Association*, 32(1974), 101-118.

[Con90] **Connolly D.T.,** "An improved annealing scheme for the QAP", *European journal of* *operational research* **46** (1990).

[Cook71] **Cook S.A.**, "The complexity of theorem-proving procedures", In: *Proceedings of Third Annual ACM Symposium on the Theory of Computing*, pp. 151-158 (1971).

[Cor04] **Correa E.S., Steiner M. T. A., Freitas A. A., Carnieri C. "** A genetic algorithm for solving a capacitated p-median problem", *Numerical Algorithms, 35:373–388,* (2004).

[Cor06] **Correa F. A., Lorena L. A., N. Senne E. L. F.**, "Lagrangean Relaxation with Clusters for the Uncapacitated Facility Location Problem", *Proceedings of the XIII Congreso Latino Iberoamericano de Investigacion Operativa - CLAIO, Montevideo, Urugvay,* (2006).

[Cve96] **Cvetković D., Čangalović M., Dugošija Đ.,Kovačević-Vujčić V.,Simić S., Vuleta J.,** "Kombinatorna optimizacija", *Društvo operacionih istraživača Jugoslavije-DOPIS Beograd,* (1996).

[Cra03] **Crainic T.G., Gendreau M., Hansen P., Mladenović N."** Parallel variable neighbourhood search for the p-median" *Les Cahiers du GERAD, G-2003-4,* (2003).

[Cra04] **Crainic T.G., Gendreau M., Hansen P., Mladenović N."** Cooperative parallel variable neighborhood search for the p-median", *Journal of Heuristics, 10(3):293–314,* (2004).

[Cre97] **Crescenci P., Kann V.,** "A compendium of NP optimization problems" (1997). http://www.nada.kth.se/theory/problemlist.html

[Čer85] **Černý, V.** , "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm", *Journal of Optimization Theory and Applications* **45**: 41–51.(1985).

[Dav06] **Davidović T.,** "Raspoređivanje zadataka na višeprocesorske sisteme primenom metaheuristika" , *Doktorska disertacija, Matematički fakultet, Beograd (2006).*

[Das95] **Daskin M.S.,** "Network and Discrete Location: models, algorithms, and applications" *John Wiley & Sons, Inc., New York,* (1995).

[Das00] **Daskin M.,** "A new approach to solve the vertex P-center problem to optimality: algorithm and computational results", *Communications of the Operations Research* Society of Japan, 45(9), 428-436*,* (2000).

[Dea85] **Dearing P.M.,** "Location problems", *Operations Research Letters, Vol. 4, pp.* 95-98 (1985).

[Dij59] **Dijkstra E.W.**, "A note on two problems in connexion with graphs", *Numerische Mathematik*, Vol. 1, pp. 269-271 (1959).

[Dim 05] **Dimitrijević B., Vidović, M.,** "Heuristics for One Class of Minimal Covering Problem in Case of Locating Undesirable Goods", *The 7th Balkan Conference on Operational Research, Constanta, Romania,* 25-28 May, (2005).

[Dim06] **Dimitrijević B., "** Prilog rešavanju jedne klase lokacijskih problema", *doktorska disertacija, Saobraćajni fakultet, Beograd,(2006*).

[Dim07] **Dimitrijević B., Vidović M.,** "Lokacijski problem nepokrivanja u telekomunikacionim mrežama", *Simpozijum o novim tehnologijama u poštanskom i telekomunikacionom saobraćaju, Saobraćajni fakultet , Beograd ,* (2007).

[Dre02] **Drezner Z., Klamroth K., Schobel A., Weslowsky G.O.,** The Weber problem, in H. Hamacher and Z. Drezner, eds.: Facility Location : Applications and Theory, Springer-Verlag, Berlin-Heidelberg, 1-25 (2002).

[Dre03] **Drezner Z., Erkut E.,** "An Efficient Genetic Algorithm for the p-median Problem", *Annals of Operations Research*, Vol. 122, pp.21-42 (2003).

[Dre10] **Drezner Z., Krass D., Berman O.,** "Generalized coverage: New developments in covering location models", *Computers & Operations Research 37,* 1675-1687, (2010).

[Dom03] **Dominguez E., Munoz J., Merida E**., "A recurrent neural network model for the p-hub problem", Lecture notes in computer science 2687, (2003).

[Dor96] **Dorigo M., Maniezzo V., Colorni A.**, "Ant System: Optimizing by a Colony of Cooperating Agents", *IEEE Transactions on Systems, Man and Cybernetics - Part B*, Vol. 26, No. 1, pp. 29-41 (1996).

[DSi96] **De Simone C., Mannino C.**, "Easy Instances of the Plant Location Problem", *Technical Report R-427*, Gennaio 1996, University of Roma, Italy (1996).

[DJo75] **De Jong K.E.,** "An analysis of the behavior of a class of genetic adaptive systems", PhD thesis, University of Michigan (1975).

[Ell01] **Elloumi S., Labbe M., Pochet Y.,** " New formulation and resolution method for the P-center problem" http://www.optimization- online.org/DB HTML/2001/10/394.html.

[Els77] **Elshafei A.N.**, "Hospital layout as a quadratic assignment problem2, *Operations Research Quaterly* **28**, (1977).

[Erl78] **Erlenkotter D., "**A dual-based procedure for uncapacitated facility location", *Operations Research*, Vol.26, pp 992-1009, (1978).

[Ern96] **Ernst A.T., Krishnamoorty M.,** "Efficient algorithms for the uncapacitated singl allocation p-hub median problem", CSIRO Division of Mathematics and Statistics, (1996).

[Ern98] **Ernst A.T., Krisnamoorthy M.**, "Exact and Heuristic Algorithms for the Uncapacitated Multiple Allocation p-hub Median Problem ", *European Journal of Operational Research,* Vol.104., pp.100-112 (1998).

[Ern04] **Ernst, A.T., Hamacher H. , Jiang H., Krishnamoorthy M., Woeginger G.**, "Uncapacitated Single and Multiple Allocation p-Hub Center Problems", *Operations Research*, (2004).

[Ern04a] **Ernst, A.T., Hamacher H. , Jiang H., Krishnamoorthy M., Woeginger G.,** "Heuristic Algorithms for the Uncapacitated Hub Center Single Allocation Problem", *European Journal of Operational Research* (2004.to appear)

[Fan90] **Fang L., Li T.**, "Design of competition based neural networks for combinatorial optimization", *International Journal on Neural System,* Vol. 3, pp. 221-235 (1990).

[Fed01] **Fedoryuk, M.V.** , "Method of steepest descent", in Hazewinkel, Michiel, *Encyclopedia of Mathematics*, Springer, (2001).

[Fil97] **Filipović V.,**"Određivanje performansi genetskih algoritama u teoriji i praksi", *Prolećna škola o programskim jezicima - Tekstovi predavanja, Novi Sad,* (1997).

[Fil98] **Filipović V.** "Predlog poboljšanja operatora turnirske selekcije kod genetskih algoritama", *Magistarski rad, Univerzitet u Beogradu, Matematički fakultet* (1998).

[Fil00] **Filipović V., Kratica J., Tošić D., Ljubić I.,** "Fine Grained Tournament Selection for the Simple Plant Location Problem", *Proceedings of the 5th Online World Conference on Soft Computing Methods in Industrial Applications - WSC5,* pp. 152-158, (2000).

[Fil06] **Filipović V.,** "Operatori selekcije i migracije i WEB servisi kod paralelnih evolutivnih algoritama", *Doktorska disertacija, Matematički fakultet, Beograd* (2006).

[Gal96] **Galvao R.D., ReVelle C.S.**, "A Lagrangean heuristic for the maximal covering location problem", European Journal of Operational Research, Vol. 88, pp. 114–123 (1996).

[Gar79] **Garey M.R.**, **Johnson D.S.**, "Computers and Intractibility: A Guide to the Theory of NP Completness", W.H. Freeman and Co. (1979).

[Gav66] **Gavett J.W., Plyter N.V.**, “The optimal assignment of facilities to locations by branch and bound”, *Operations Research* 14 (1966), 210-232.

[Glo86] **Glover F.**, "Future paths for integer programming and links to artificial intelligence", *Computers & Operations Research*, Vol. 5, pp. 533-549 (1986).

[Glo90] **Glover F.**, "Tabu search: A Tutorial", *Interfaces*, Vol 20, pp. 74-94 (1990).

[Glo03] **Glover F., Kochenberger G.A.**, "Handbook of Metaheuristics", Kluwer Academic Publishers, Boston-Dordrecht-London (2003).

[Geo74] **Geoffrion A., McBride R.**, "Lagrangean Relaxation for Integer Programming", *Mathematical Programming Study*, Vol. 2, pp. 82-114 (1974).

[Gib85] **Gibbons A.,** "Algorithmic Graph Theory", Cambridge University Press, Cambridge, (1985).

[Gli62] **Gilmore P.C.,** Optimal and suboptimal algorithms for the quadratic assignment problem, *J.SIAM* **10** (1962), 305-313.

[Gol71] **Goldman A. J.** “Optimal Center Location in Simple Networks”, *Transportation Science,*5,212-221, (1971).

[Gol89] **Goldberg D.E**., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley Publ. Comp., Reading, Mass., 412 pp (1989).

[Gom63] **Gomory R.E.** “ An algorithm for integer solutions to linear programs”, In R. L. Graves and P.Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269-302. McGraw-Hill, New York, (1963).

[Gro91] **Grotschel M., Holland O.** “Solution of large-scale travelling salesman problems”, *Mathematical Programming*, 51(2):141-202, (1991).

[Ham96] **Hamacher H.,Nickel S.,Schneider A.,** “Classification of Location problems”, Fachbereich Mathematik Universitat Kaserlautern,(1996).

[Han85] **Hanjoul, P., Peeters, D.,** “A comparison of two dual-based procedures for solving the p-median problem”, European Journal of Operational Research 20,(1985) 387–396.

[Han97] **Hansen P., Mladenović N.** “Variable neighborhood search for the p-median” *Location Science, 5(4):*207–226, (1997).

[Han99] **Hansen P., Mladenović N.**, "An Introduction to Variable Neighborhood Search", In: Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Voss S., Martello S., Osman I.H., Roucairol C. (eds.), Kluwer Academic Publishers, pp. 433-458 (1999).

[Han01] **Hansen P.,Mladenović N.**, "Variable neighborhood search: Principles and applications," European Journal of Operational Research 130, pp. 449-467, 2001.

[Han07] **Hansen P., Brimberg J., Urošević D., Mladenović N.,** "Primal-Dual Variable Neighborhood Search for the Simple Plant-Location Problem", INFORMS Journal on Computing 19, pp. 552-564 (2007).

[Hak64] **Hakimi S.L.,** "Optimum locations of switching centers and the absolute centers and medians of a graph " , *Operations Research 12,* (1964) 450–459.

[Hak65] **Hakimi S.L.,** " Optimum distribution of switching centers in a communication network and some related graph theoretic problems", *Operations Research 13,* (1965)462–475.

[Hll75] **Holland J.H.,** "Adaptation in Natural and Artificial Systems", *The University of Michigan Press, Ann Arbor (1975).*

[Hlm95] **Holmberg K.**, "Experiments with primal-dual decomposition and subgradient methods for the uncapacitated facility location problem", *Research Report LiTH-MAT/OPT-WP-1995-08*, Optimization Group, Department of Mathematics, Linkoping Institute of Technology, Sweden (1995).

[Ilh01] **Ilhan T., Pinar M. C.,** "An efficient exact algorithm for the vertex P-center Problem", (2001).

[Ili99] **Ilić V**. "Neuronske mreže", (2003). http://solair.EUnet.rs/~ilicv/neuro.html

[Jar72] **Jarvinen P., Rajala J.,Sinervo H.** "A branch-and-bound algorithm for seeking the p median." *Operations Research,* 20(1):173–178,(1972).

[Kar79] **Kariv, O., Hakimi, S.L.,** "An algorithmic approach to network location problems"Part II: The p-median. SIAM, (1979) Journal of Applied Mathematics 37, 539–560.

[Kar03] **Kara, B.Y., Tansel B.,** "The single assignment hub covering problem", *Journal of the Operation Research Society,* 54: 59-64, (2003).

[Kli96] **Klincewicz J.G.,** " A dual algorithm for the uncapacitated hub location problem", Location Science, 4 (3): 173-84, (1996).

[Klo93] **Klose, A.** “Das kombinatorische p-Median-Modell und Erweiterungen zur Bestimmung optimaler Standorte” *Dissertation Nr. 1464, Hochschule St. Gallen, Dufourstrasse 50, CH-9000 St. Gallen* (1993).

[Kra99] **Kratica J.,** "Improvement of Simple Genetic Algorithm for Solving the Uncapacitated Warehouse Location Problem", Advances in Soft Computing - Engineering Design and Manufacturing, R.Roy, T. Furuhashi and P.K. Chawdhry (Eds), Springer-Verlag London Limited, (1999), pp. 390-402.

[Kra00] **Kratica J.,** "Paralelizacija genetskih algoritama za rešavanje nekih NP- kompletnih problema", *Doktorska disertacija, Matematički fakultet, Beograd* (2000).

[Kra01] **Kratica J., Tošić D., Filipović V., Ljubić I.**, "Solving the Simple Plant Location Problem by Genetic Algorithm", RAIRO Operations Research, Vol. 73, No. 1, pp.127-142 (2001).

[Kra02] **Kratica J., Tošić D., Filipović V., Ljubić. I**., ”A genetic algorithm for the uncapacitated network design problem”, *Soft Computing in Industry – Recent Applications*, Engineering series, pp. 329-338. Springer (2002).

[Kra03] **Kratica J., Ljubić I., Tošić D.,** ”A genetic algorithm for the index selection problem”, *Springer Lecture Notes in Computer Science*, Vol. 2611, pp. 281-291 (2003).

[Krr83] **Krarup J.**, **Pruzan P.M.**, "The simple plant location problem: Survey and synthesis", *European Journal of Operational* Research, Vol. 12, pp. 36-81 (1983).

[Kir83] **Kirkpatrick S., Gellat C., Vecchi M.**, "Optimization by simulated annealing", *Science*, Vol. 220, pp. 671-680 (1983).

[Koo57] **Koopmans T.C.,Beckmann M.J.**, “Assignment problems and location of eceonomic Activities”, *Econometrica* **25**,(1957), 53-76.[Krp72] **Karp R.**, "Reducibility among combinatorial problems", In: *Complexity of Computer Computations*, Miller R.E and Thatcher J.W. (eds.), Plenum Press, New York, NY, pp. 85-104 (1972).

[Kov08] **Kovačević, J.**, "Hybrid Genetic Algorithm For Solving The Low-Autocorrelation Binary Sequence Problem", Yugoslav Journal of Operations Research (2008).

[Lab03] **Labbe M., Yaman H., Gordiny E**., “A Branch and Cut Algorithm for Hub Location Problems with Single Assignment”(2003).

[Lan60] **Land A.H.,Doig A.G.,** "An automatic method of solving discrete programming problems", *Econometrica* **28** (3): pp. 497-520 (1960).

[Law63] **Lawler E.L.,** “The quadratic assignment problem”, *Management Science* **9**, (1963).

[Li94] **Li Y., Pardalos P. M. , Resende, M. G. C**.,”A greedy randomized adaptive search procedure for the quadratic assignment problem”, DIMACS Series in Discrete Mathematics and Theoretical Computer Science,(1994), pp.237-261.

[Lov88] **Love R.F., Morris J.G., Wesolowshy G.O.,** Facilities Location: Models & Methods, North-Holland, New York (1988).

[Lju04] **Ljubić I.**, "Exact and Memetic Algorithms for Two Network Design Problems", *PhD thesis*, Institute of Computer Graphics, Vienna University of Technology (2004).

[Luč02] **Lučić P., Teodorović D.,** “Transportation Modeling: An Artificial Life Approach”, *Proceedings of the 14th IEEE “International Conference on Tools with Artificial Intelligence”*, pp. 216-223, November 4-6, 2002 Washington D.C.

[Mac03] **MacKay, J. C. D**., “Information theory, inference and learning algorithms”, Cambridge University Press, (2003).

[Mar64] **Maranzana, F.F**., “ On the Location of Supply Points to Minimize Transport Costs”,*Operations Research Quart.,* vol 15, (1964).

[Mar08] **Marić M**., “Rešavanje nekih NP-teških hijerarhijsko-lokacijskih problema primenom genetskih algoritama” , *Doktorska disertacija, Matematički fakultet, Beograd* (2008).

[Mar88] **Martinich J.S.** “A Vertex Closing Approach to the p-Center Problem”, *Naval Research Logistics, 35, 185-201,* (1988).

[Meg83] **Megiddo, N., Zemel, E., and Hakimi, S. L.** “The maximum coverage location problem”, *SIAM Journal on Algebraic and Discrete Methods*, 4 (1983).

[Mer02] **Merino E.D., Perez J.M. “** An efficient neural network algorithm for the p-median problem” *In IBERAMIA 2002: Proceedings of the 8th Ibero-American Conference on AI, volume 2527 of Lecture Notes in Computer Science, pages 460–469,* (2002).

[Mer03] **Merino E.D., Perez J.M., Aragones J.J**., ” Neural network algorithms for the p-median problem”, In ESANN 2003: Proceedings of 11th European Symposium on Artificial Neural Networks, Belgium, April 2003, pages 385–391, 2003.

[Mey09] **Meyer T., Ernst A.T., Krishnamoorthy M.,** “A2-phase algorithm for solving the single allocation p-hub center problem“, Computers & Operations Research, 36: 3143-3151, (2009).

[Mla95] **Mladenović N.**, "A variable neighborhood algorithm - a new metaheuristics for combinatorial optimization", Abstracts of papers presented at Optimization days,Montreal (1995).

[Mla97] **Mladenović N., Hansen P.** "Variable neighborhood search", *Computers Operations Research,* Vol. 24, pp. 1097-1100, (1997).

[Mla03] **Mladenović N.,** “Kontinualni lokacijski problem”, Matematički Institute, Srpska akademija nauka i umetnosti, Beograd (2003).

[Mih03] **Mihelic J., Robic B.**, "Genetic algorithm for the k-center location problem", *XIV EWGLA*, Corfu, Greece (2003).

[Mis05] **Misevicius A.**, "A tabu search algorithm for the quadratic assignment problem," Computational Optimization and Applications, vol. 30, no. 1, pp. 95-111, (2005).

[Mir79] **Mirchandani P.B.,Obata T.**, “Locational decisions with interactions between facilities: the quadratic assignment problem a review”, Working Paper Ps-79-1, Rensselaer Polytechnic Institute, Troq, New York, (1979).

[Mic96] **Michalewicz Z.,** "Genetic Algorithms + Data Structures = Evolution Programs", Third Edition, Springer Verlag, Berlin Heideleberg (1996).

[Min70] **Minieka E.**, “The m-center problem”, SIAM Review, 12, 138-139.(1970).

[Mit96] **Mitchell M.,** "An Introduction to Genetic Algorithms", MIT Press (1996).

[Mit99] **Mitchell John E.,** “Branch-and-Cut Algorithms for Combinatorial Optimization Problems” Mathematical Sciences Rensselaer Polytechnic Institute Troy, NY, USA (1999).

[Mor78] **Morris, J.G**., “ On the extent to which certain fixed-charge depot location problems can be solved by LP”Journal of the Operational Research Society 29, 71–76 (1978).

[Moo84] **Moon D., Chaudhry S.,** “An analysis of network location problems with distancecontrains, Management Science, Vol. 30, No.3, pp. 290-307, 1984.

[Mur97] **Murray A.T., Church R.L.,** “Solving the anti-covering location problem using lagrangian relaxation”, Computers and Operations Research , Vol. 24, No.2, pp. 127-140, 1997.

[O'Ke87] **O'Kelly M.,** ''A quadratic integer program for the location of interacting hub facilities“, European Journal of Operational Research 32, 393-404, (1987).

[Pad91] **Padberg M., Rinaldi G. "**A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems", *SIAM Review*, 33(1):60-100, (1991).

[Pam01] **Pamuk F.S., Sepil C.,** "A solution to the hub center problem via a single-relocation algorithm with tabu search", *IIE Transactions* 33 (5), 399-411, (2001).

[Per04] **Perez Perz M., Almeida Rodriguez F., Moreno Vega J.M.,** "On the use of the path relinking for the p-hub median problem, *Lecture notes in computer science* 3004, (2004).

[RaV70] **ReVelle, C.S., Swain, R.W.,** "Central facilities location", *Geographical Analysis* 2, 30–42 (1970).

[Rol96] **Rolland E., Schilling D.A., Current J.R. "**An efficient tabu search procedure for the p-median problem", *European Journal of Operational Research, 96(2):329–342,* (1996).

[Ros97] **Rosing K.E., ReVelle C.S.** "Heuristic concentration: Two stage solution construction" , *European Journal of Operational Research, 97(1):75–86,* (1997).

[Ryu92] **Ryu C., Guignard M.**, "An Exact Algorithm for the Simple Plant Location Problem with an Aggregate Capacity Constraint", *TIMS/ORSA Meeting*, Orlando, FL, 92-04-09 (1992).

[Rus11] **Rusov J.,** "Pregled hab lokacijskih problema i metoda za njihovo rešavanje", *Master rad, Matematičk fakultet, Beograd* (2011).

[Sah76] **Sahni S.,Gonzalez T.,"** P-complete approximation problems", *Journal of the ACM* **23**, (1976), 555-565.

[Sim89] **Simao H.P., Thizy J.M.**, "A dual simplex algorithm for the canonical representation of the uncapacitated facility location problem", *Operations Research Letters*, Vol. 8, No. 5, pp. 279-286 (1989).

[Soh98] **Sohn J., Park S.**, "Efficient Solution Procedure and Reduced Size Formulations for p-hub Location Problems", *European Journal of Operational Research*, Vol. 108, pp.118-126 (1998).

[Sko90] **Skorin-Kapov J.** " Tabu search applied to the quadratic assignment problem" *ORSA. Journal on Computing*, 2, 33–45, (1990).

[Sko94] **Skorin-Kapov D.**, **Skorin-Kapov J.** "On tabu search for the location of interacting hub facilities", European Journal of Operational Research 73,(1994).

[Sta04] **Stanimirović Z.,** “Rešavanje nekih diskretnih lokacijskih problema primenom genetskih algoritama“, *Magistarska teza, Matematički fakultet, Beograd* (2004).

[Sta07] **Stanimirović Z**., “Genetski algoritmi za rešavanje nekih NP-teških hab lokacijskih problema“, *Doktorska disertacija, Matematički fakultet, Beograd* (2007).

[Sta09] **Stanimirović Z., Kratica J., Filipović V., Tošić D.,** “Evolutivni pristup za rešavanje hab lokacijskih problema“, *Univerzitet u Beogradu, Matematički fakultet,* (2009).

[Ste61] **Steinberg L.,** “The backboard wiring problem : a placement algorithm”, *SIAM Review* **3**,(1961).

[Sun06] **Sun M.,** "Solving the uncapacitated facility location problem using tabu search", Computers and Operations Research, Vol. 33, No. 9, pp. 2563-2589 (2006).

[Tai90] **Taillard. É.D** , “Some efficient heuristic methods for the flow shop sequencing problem”,European Journal of Operational Research 47, 65-74(1990)

[Tai1991] **Taillard É. D. “**Robust taboo search for the quadratic assignment problem” *Parallel Computing*, 17, 443–455, (1991).

[Tca88] **Tcha D.W., Ro H.B, Yoo C.B.** “Dual based Add Heuristic for Uncapacitated Facility Location”, *Journal of Operational Research Society, Vol.39, No. 9,* (1998).

[Tei68] **Teitz M.B., Bart P.,** "Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph," *Operations Res.,* Vol. 16 (1968), pp. 955-961.

[Toš04] **Tošić D., Mladenović N., Kratica J., Filpović V.,** "Genetski algoritmi",Matematički institut SANU, Beograd (2004).

[Van] **Vanjak Z.,** “Metode rješavanja kvadratičnog problema pridruživanja”, Magistarski rad, Fakultet elektrotehnike i računarstva, Zagreb

[Vuj96] **Vujošević M., Stanojević M., Mladenović N.,** "Metode optimizacije: mrežni, lokacijski i višekriterijumski modeli", Društvo operacionih istraživača Jugoslavije, Beograd (1996).

[Web09] **Weber A.,** „Über den Standort der Industrien“, Tübingen (1909), English translation by Friedrrich C.J., Theory of the Location of Industries, University of Chicago Press (1909).

[Wil87] **Wilhelm M.R., Ward T.L.,** “Solving quadratic assignment problems by simulated Annealing”, *IEEE Transactions* **19** (1987), no. 1, 107-119.

[Yig03] **Yigit V., Turkbey O.**, "An approach to the facility location problems with hill-climbing and simulated annealing", Journal of The Faculty of Engineering and Architecture of Gazi University, Vol. 18, No. 4,pp. 45-56 (2003).

[Yon07] **Yongzhong W., Ping J.** "Solving the Quadratic Assignment Problems by a Genetic Algorithm with a New Replacement Strategy", *World Academy of Science, Engineering and Technology*, (2007).

[Yur94] **Yuret D.,** "From Genetic Algorithms to Efficient Optimization", MSc Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science (1994).
http://www.dai.ed.ac.uk/groups/evalg/Local_Copies_of_Papers/Yuret.MSc_Thesis.From_Genetic_Algorithms_to_Efficient_Optimization.ps.gz

Sadržaj