SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Primjena optimizacije kolonijom mrava na rješavanje problema usmjeravanja vozila

Karlo Knežević

Voditelj: doc.dr.sc. Domagoj Jakobović

Zagreb, svibanj, 2010.

Sadržaj

1. Uvod

Jedna od glavnih odlika računala je da u kratkom vremenu egzaktno riješi probleme koji uključuju složene računske operacije. Ipak, uz svu tehnologiju i napredak postignut u računarskoj znanosti, postoje problemi koji današnjim metodama nisu rješivi u stvarnom vremenu. Posebnu skupinu problema čine NP[3]-teški i NP-potpuni problemi. U NP-teške i NP-potpune probleme spadaju i mnogi kombinatorni problemi, poput problema usmjeravanja vozila. Poznati algoritmi za rješavanje NP-teških problema su u najboljem slučaju eksponencijalne složenosti, tj. vrijeme izvođenja je eksponencijalnog rasta. Za ilustraciju, to znači da se pojedini problemi ne mogu poznatim algoritmima riješiti milijardama godina. Pitanje je kako pristupiti takvim problemima. Jedan od učinkovitih načina rješavanja takvih problema jest korištenje heurističkih[2] algoritama.

Heuristički algoritmi mogu se podijeliti na heuristike specifičnih problema i metaheuristike. Metaheuristički[3] algoritmi obuhvaćaju širok skup algoritama namijenjenih optimizacijskim problemima. Metaheuristički algoritmi dijele se po različitim osnovama, a jedna od podjela su prirodom inspirirani algoritmi, među kojima se nalazi algoritam optimizacije kolonijom mrava.

Problem usmjeravanja vozila se u posljednje vrijeme počeo sve više razmatrati zbog sve većih zahtjeva gospodarstva za uštedama u transportu. Problem usmjeravanja vozila spada u NP-teške probleme. Racionalizacijom prijevoznih procesa moguće je ostvariti znatne uštede. Cilj racionalizacije je poboljšati plan prijevoza odnosno pronaći skup ciklusa koji će smanjiti ukupne troškove prijevoznog procesa, pri čemu se troškovi reflektiraju u ukupnoj duljini puta, broju upotrijebljenih vozila ili vremenu potrebnom da se cijeli posao obavi.

U seminarskom radu bit će objašnjeno kako primjena optimizacije kolonijom mrava na problem usmjeravanja vozila može u zadovoljivom vremenu dati rješenje u skupu zadovoljivih rješenja.

---

1. NP - klasa složenosti NP je skup problema odluke koji mogu biti riješeni nedeterminističkim Turingovim strojem u polinomnom vremenu

2. grč. ευρίσκω – pronaći, otkriti

3. grč. μετά – iza, preko; u kontekstu metaheuristike: viša razina

2. Optimizacija kolonijom mrava

Mravi su izuzetno jednostavna bića – usporedimo li ih, primjerice, s čovjekom. No i tako jednostavna bića zahvaljujući socijalnim interakcijama postižu zadivljujuće rezultate. Sigurno ste puno puta u životu naišli na mravlju autocestu – niz mrava koji u koloni idu jedan za drugim prenoseći hranu do mravinjaka. O ponašanju ovih stvorenja snimljen je čitav niz dokumentarnih filmova: o njihovoj suradnji, o organizaciji mravinjaka, o nevjerojatnoj kompleksnosti i veličini njihovih kolonija. No kako jedno jednostavno biće poput mrava može ispoljavati takvo ponašanje?

Čitav spektar različitih znanstvenih disciplina iskazao je interes za mrave. A znanstvenicima s područja računalne znanosti mravi su posebno zanimljivi iz vrlo praktičnog razloga – mravi rješavaju optimizacijske probleme. Naime, uočeno je da će mravi uvijek pronaći najkraći put između izvora hrane i njihove kolonije, što će im omogućiti da hranu dopremaju maksimalno brzo. Da bi misterij bio još veći, spomenimo da mravlje vrste ili uopće nemaju razvijen vidni sustav, ili je on ekstremno loš.

Optimizaciju kolonijom mrava[4] prvi je uveo Marco Dorigo 1992. godine. Kolonija mrava predstavlja multi-agentski sustav gdje je ponašanje pojedinog, umjetnog, mrava nadahnuto ponašanjem mrava u prirodi. Primjene obuhvaćaju klasične NP – teške probleme i sežu do usmjeravanja u računalnim i telekomunikacijskim mrežama.

2.1 Mravi u prirodi

Kako bi istražili ponašanje mrava, Deneubourg i suradnici [1] napravili su niz eksperimenata koristeći dvokraki most postavljen između mravinjaka i hrane. Mravi prilikom kretanja ne koriste osjet vida, već je njihovo kretanje određeno socijalnim interakcijama s drugim mravima. Na putu od mravinjaka do hrane, te na putu od hrane do mravinjaka, svaki mrav ostavlja kemijski trag – feromone.

Mravi imaju razvijen osjet feromona, te prilikom odlučivanja kojim putem krenuti tu odluku donose obzirom na jakost feromonskog traga koji osjećaju. Tipično, mrav će se kretati smjerom jačeg feromonskog traga.

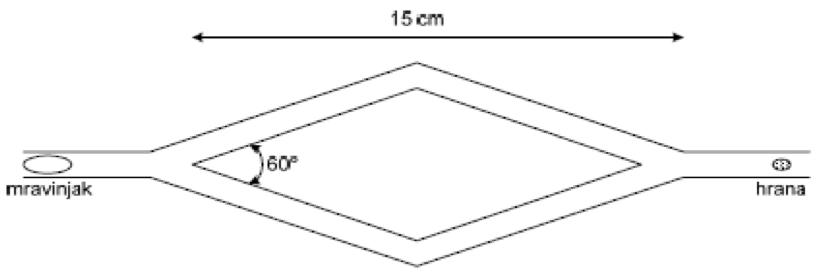
Deneubourg i suradnici prilikom eksperimenata varirali su duljine krakova mosta. U prvom eksperimentu oba su kraka jednako duga. Mravi su na početku u podjednakom broju krenuli preko oba kraka. Nakon nekog vremena, međutim,

---

4. engl. ant colony optimization

dominantni dio mrava kretao se je samo jednim krakom, slučajno odabranim. Objašnjenje je sljedeće. Na početku, mravi slučajno odabiru jedan ili drugi krak, i krećući se njima ostavljaju feromonski trag. U nekom trenutku dogodi se da nekoliko mrava više krene jednim od krakova, uslijed slučajnosti, i tu nastaje veća koncentracija feromona. Privučeni ovom većom količinom feromona, još više mrava kreće tim krakom što dodatno povećava količinu feromona. S druge strane, kako drugim krakom kreće manje mrava, količina feromona koja se osvježava je manja, a feromoni s vremenom i isparavaju.

Ovo u konačnici dovodi do situacije da se stvara jaki feromonski trag na jednom kraku, i taj feromonski trag privlači najveći broj mrava. Eksperiment je ponovljen više puta, i uočeno je da u prosjeku u 50% slučajeva mravi biraju jedan krak, a u 50% slučajeva biraju drugi krak.



Slika 2.1: *eksperiment dvokrakog mosta*

Sljedeći eksperiment napravljen je s dvokrakim mostom kod kojeg je jedan krak dvostruko dulji od drugoga. U svim pokušajima eksperimenta pokazalo se da najveći broj mrava nakon nekog vremena bira kraći krak.

Ovakvo ponašanje na makroskopskoj razini – pronalazak najkraće staze između hrane i mravinjaka rezultat je interakcija na mikroskopskoj razini – interakcije između pojedinih mrava koji zapravo nisu svjesni šire slike.

Konačno, kako bi se provjerila dinamika odnosno sposobnost prilagodbe mrava na promjene, napravljen je treći eksperiment.

Kod ovog eksperimenta mravinjak i izvor hrane najprije su spojeni jednokrakim mostom, čiji je krak dugačak. Mravi su krenuli tim krakom do hrane i natrag. Nakon 30 minuta kada se je situacija stabilizirala, mostu je dodan drugi, dvostruko kraći krak. Međutim, eksperiment je pokazao da se je najveći dio mrava i dalje nastavio kretati duljim krakom, zahvaljujući stvorenom jakom feromonskom tragu.

2.2 Pronalazak minimalnog puta u grafu

Analogno prirodnim mravima, zadatak je umjetnih mrava pronaći minimalni put između čvorova grafa. Neka je G = (V, E) povezani graf, gdje je V skup svih čvorova, a E skup svih lukova. Broj čvorova je V = n . Rješenje problema predstavlja put na grafu koji povezuje izvorišni čvor *f* s odredišnim *d*, čija duljina je određena brojem lukova na putu.

Sa svakim je lukom (i,j) grafa G povezana varijabla τ_{ij} koja predstavlja umjetni trag feromona. Tragove feromona čitaju i pišu mravi. U svakom čvoru grafa dolazi do stohastičke odluke koji je čvor sljedeći. $k-ti$ mrav lociran u čvoru i koristi trag feromona τ_{ij} za izračun vjerojatnosti u koji čvor $j \in N_i$ treba otići. N_i je skup susjeda čvora i.

Neka je zadan problem takav da je u grafu G=(V,E) potrebno pronaći najkraći put koji povezuje sve gradove, a da je početak i završetak puta u istom čvoru. Čvorove nazovemo gradovima. Broj gradova koje treba obići je n, broj mrava u sustavu je m, a $b_i(t)$ je broj mrava u gradu i u trenutku t. U početku se programski inicijaliziraju feromonski tragovi na vrijednost τ_{ij}=Q, između onih gradova između kojih postoje u grafu G bridovi $e \in E$.

Mrav se odlučuje u koji će sljedeći grad j ići po funkciji koja kao parametre ima udaljenost gradova i trag feromona na njihovoj poveznici. Kako bi se izbjeglo obilaženje istih gradova više puta, svaki mrav ima svoju *tabu listu* u koju stavlja svaki grad koji je obišao. Kada tabu lista mrava obuhvati svih n gradova, mrav postavlja feromone na svaki put kojeg je koristio za obilazak gradova te resetira svoju tabu listu. Broj obilazaka koje će svaki mrav napraviti ograničen je zadanom vrijednosti, koju zovemo broj ciklusa.

Globalno gledajući, svaki mrav, *k*, se u trenutku *t* odlučuje gdje će biti u trenutku $t+1$, te se pri idućoj iteraciji koraka pomiče tamo. Na taj način svi će mravi nakon n iteracija proći kroz sve gradove, u trenutku $t+n$, i postaviti trag. Pri tome se učitaju novi podaci za feromone na način da se stara vrijednost pomnoži faktorom isparavanja ρ, te mu se nadoda trag svakog mrava koji je tuda prolazio. Faktor isparavanja govori koliki dio feromonskog traga ne isparava. U jednanžbi 2.1 napisan je izraz za obnavljanje feromonskog traga na putu (i,j).

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \tag{2.1}$$

U jednadžbi 2.2 napisan je izraz za dodanu vrijednost feromonskog traga.

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{2.2}$$

Količina feromona koju mrav, k, doda na neki put je 0 ako nije prošao tim putem. Ukoliko je mrav prošao tim putem, ostavlja neki određeni iznos traga ili obrnuto proporcionalan s duljinom puta koju je prošao.

$$\Delta\tau_{ij}^{k} = \frac{Q}{L_k} \tag{2.3}$$

U jednadžbi 2.3, Q je zadana konstanta, a L_k ukupni put kojeg je k-ti mrav prošao za svoj obilazak svih gradova.

Pri odluci kamo da ide, osim traga feromona, mrav koristi i *vidljivost*. Vidljivost je parametar obrnut od udaljenosti. Što je grad udaljeniji to je vidljivost manja. Jedadžba 2.4 opisuje vidljivost.

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{2.4}$$

Ukupni račun za odluku mrava prikazan je jednadžbom 2.5:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum [\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ik}]^{\beta}} \tag{2.5}$$

Ova jednadžba vrijedi u slučaju da mrav ima kamo otići, odnosno njegova tabu lista ne obuhvaća sve postojeće gradove. η_{ij} predstavlja vidljivost između gradova i i j. α i β su parametri koji upravljaju odnos važnosti traga u odnosu na vidljivost. Što je α veći to će mrav svoje odluke više bazirati na tragu feromona. Što je β veći, analogno tragu, mrav će pri donošenju odluke veću važnost pridijeliti vidljivosti gradova. Rezultat $p_{ij}^{k}(t)$ predstavlja vjerojatnost da će mrav k krenuti iz grada i u grad j trenutku t. Vrijednost p poprima vrijednosti iz intervala $(0,1)$.

2.3 Svojstva mrava i kolonije

U prethodnom odlomku, matematički je postavljen model prema kojem mravi u grafu s n gradova pronalaze najkraći put. Parametri jednadžbi opisuju svojstva mrava, ali i svojstva kolonije.

Svojstva kolonije kao cjeline:

1) dobra rješenja mogu proizaći samo kao rezultat kolektivne interakcije;
2) svaki mrav koristi samo privatne informacije i lokalne informacije čvora kojeg posjećuje;
3) mravi komuniciraju samo indirektno;
4) mravi se sami po sebi ne adaptiraju, naprotiv, mijenjaju način prezentacije problema i percepcije drugih mrava.

Svojstva mrava kao jedinke:

1) mrav traži najisplativije rješenje;
2) svaki mrav ima memoriju M_k za pohranu informacija o putu i može ju koristiti:
    a. za generiranje smislenog rješenja,
    b. za evaluaciju pronadenog rješenja,
    c. za povratak po istom putu;
3) mrav k može se pomaknuti u bilo koji čvor j u susjedstvu N_i^k;
4) mravu k može se dodijeliti početno stanje i jedan ili više terminirajućih uvjeta;
5) mrav stvara rješenje inkrementalno i konstrukcija završava kada je zadovoljen jedan od uvjeta zaustavljanja;
6) odabir sljedećeg čvora temelji se na vjerojatnosti;
7) vjerojatnost pojedine odluke mrava temelji se na:
    a. vrijednostima pohranjenim u strukturi čvora $A_{ij} = [a_{ij}]$,
    b. privatnoj memoriji mrava,
    c. ograničenjima problema;
8) pomicanjem iz čvora i u čvor j mrav ažurira trag feromona;
9) kada je izgradio rješenje, mrav se može vratiti istim putem i ažurirati

trag feromona na povratku;

10) kada je izgradio rješenje i vratio se na izvorište, mrav umire i oslobađa resurse.

2.4 Algoritamsko ostvarenje osnovnih funkcija kolonije mrava

Pri pokretanju algoritma najprije se učitavaju koordinate ili međusobne udaljenosti gradova iz datoteke i učitavaju se u listu gradova, te se postavljaju tragovi feromona na ceste koje ih povezuju. Zatim se stvaraju i postavljaju svi mravi. Pri postavljanju mrava stvaraju se tabu liste, mravi se postavljaju u početni grad na način da svaki mrav ima svoj početni grad ili može biti i više mrava ili nijedan mrav u nekom gradu, te se taj početni grad stavi kao prvi član tabu liste.

Zatim počinju iteracije kretanja mrava. Nakon n iteracija svi mravi imaju pune tabu liste te se postavlja trag feromona svakog mrava. Mravi se nanovo postavljaju u neki početni grad i tabu liste im se prazne, odnosno ostane samo novi početni grad. Takav ciklus od n iteracija se ponavlja toliko puta dok korisnik ne prekine rad programa ili program dostigne maksimum dozvoljenih ciklusa.

Inicijalizacija programa prikazan je slikom 2.1:

```
brojac_ciklusa = 0;
        Najbolja_Ruta = [maximalna vrijednost];
        postaviti n gradova (učitati iz datoteke ili radne memorije);
        postaviti Trag[i,j] na početnu vrijednost c za svaki i i j od 1 do n;
        stvori m mrava;
```

Slika 2.1: *Inicijalizacija programa*

Odrada unaprijed definiranog broja ciklusa na slici 2.2:

```
dok (brojac_ciklusa < MAX_broj_ciklusa){
        brojac_ciklusa++;
        Odradi_Ciklus();
        Provjeri_Najbolju_Rutu();
        Ispari_Trag();
        Nadodaj_Trag();
        Resetiraj_Mrave();
    }
```

Slika 2.2: *Odrada unaprijed definiranog broja ciklusa*

Funkcija Odradi_Ciklus():

```
za svaki i od 0 do n{
        za svakog mrava k od 1 do m{
                k.Sljedeći_Korak();
        }
}
```

Slika 2.3: *funkcija Odradi_Ciklus()*

Funkcija Provjeri_Najbolju_Rutu():

```
za svakog mrava k od 1 do m{
        ako je (Najbolja_Ruta > k.Duljina_Rute){
                Najbolja_Ruta = k.Duljina_Rute;
        }
}
```

Slika 2.4: *funkcija Provjeri_Najbolju_Rutu()*

Funkcija *Ispari_Trag* množi sve elemente matrice *Trag* s faktorom isparavanja, a funkcija *Nadodaj_Trag* prolazi kroz sve mrave, te za svaki njihov korak dodaje određen iznos traga u matricu Trag na odgovarajuće mjesto. Funkcija *Resetiraj_Mrave* briše Tabu liste svih mrava i vraća ih u svoje početne gradove.

3. Problem usmjeravanja vozila

Određivanje najpovoljnijeg puta, kojeg koristi grupa vozila prilikom posluživanja skupa korisnika, predstavlja problem usmjeravanja vozila. Prvo spominjanje ovog prometnog problema zabilježeno je 1959. godine [2]. Povod proučavanja ovog kombinatoričko-optimizacijskog problema bilo je unapređenje organizacije dostave goriva benzinskim crpkama. Tada su Dantzig i Ramser [2] prvi put predložili matematičku interpretaciju problema i algoritamski pristup rješavanju. Clark i Wright su 1964. g. predložili poboljšanje Dantzig-Ramserovog pristupa kroz heuristička rješenja. Na tragu ovih pristupa razvijeno je više od stotinu modela i algoritama za dobivanje optimalnih i aproksimativnih rješenja.

S današnjim stanjem tehnologije i s trenutno najdjelotvornijim egzaktnim algoritmima moguće je riješiti problem usmjeravanja vozila samo u slučajevima gdje postoji manje od 50 korisnika, i to na način da se do optimuma kod najvećih problema može doći samo za pojedine slučajeve. Realni problemi koje je potrebno rješavati imaju znatno više korisnika, pa se pomoću egzaktnog pristupa ne mogu riješiti u prihvatljivom vremenu. Zato se za rješavanje koristi heuristički i metaheuristički pristup, poput algoritma kolonije mrava, koji dovodi do približno optimalnog rješenja u realnom vremenu.

Da bi se definirao problem usmjeravanja vozila za distribuciju ili prikupljanje dobara, potrebno je dati osnovna ograničenja problema. U zadanom vremenu, skup vozila poslužuje skup korisnika. Vozila kreću iz skladišta i pri transportu koriste mrežu prometnica. Rješenje problema predstavlja skup ruta. Svaka ruta ima polaznu i završnu točku u skladištu vozila koje koristi rutu. Svi zahtjevi korisnika moraju biti ispunjeni, a sva nametnuta ograničenja poštovana. Cilj je da ukupni transportni trošak bude minimalan.

Moguće je uvoditi različita ograničenja i ciljeve koji mogu izravno utjecati na konstrukciju ruta prilikom optimizacijskog procesa. Mreža prometnica koja se koristi za transport dobara najčešće se opisuje grafom čiji lukovi predstavljaju prometnice, a točke raskrižja, skladišta ili korisnike. Lukovi mogu biti jednosmjerni ili dvosmjerni, a uz svaki luk postoji cijena koja najčešće korespondira sa udaljenošću i vremenom putovanja koje je ovisno o tipu vozila.

Informacije potrebne za dobar opis korisnika pri rješavanju problema usmjeravanja vozila su:

- polazna točka koja predstavlja skladište;
- količina dobara, moguće je da postoji više tipova dobara, koje je potrebno prikupiti ili dostaviti;
- vremenski period (vremenski prozor) u kojem je potrebno poslužiti korisnika;
- vrijeme potrebno da se obavi dostava ili prikupljanje kod korisnika;
- podskup dostupnih vozila koja se mogu koristiti kod pojedinog korisnika obzirom na mogući pristup za iskrcaj i ukrcaj.

3.1 Inačice problema usmjeravanja vozila

Vozači moraju poštivati zahtjeve kao što su radno vrijeme, broj obveznih stanki, maksimalno trajanje vožnje itd. Takvi zahtjevi prenose se na značajke vozila. Rute moraju zadovoljavati operativne zahtjeve koje nameće priroda transporta, kvaliteta usluge, te karakteristike vozila i korisnika.

Tipični operativni zahtjevi su:

- pri vožnji na svakoj ruti vozilo ne smije imati više tereta u odnosu na nazivni kapacitet koji je dopušten;
- korisnik može zahtijevati dostavu, prikupljanje ili i jedno i drugo;
- korisnik se poslužuje unutar vremenskog perioda koji zahtijeva korisnik, a nije u suprotnosti s radnim vremenom vozača.

Korisnik odabire u odnosu na što će se raditi optimizacija, a taj čin određuje samo inačicu problema usmjeravanja vozila[5]. Inačice problema usmjeravanja vozila odnose se na:

1.) ograničenje kapaciteta, CVRP
(Capacitated vehicle routing problem)
2.) višebrojnost skladišta, MDVRP
(Multi-depot vehicle routing problem)
3.) vremenska ograničenja, WRPTV ili PVRP
(Vehicle routing problem with time windows ili Period vehicle routing problem)
4.) dostavu i povratno prikupljanje, VRPB
(Vehicle routing problem with Backhauls)
5.) prikupljanje i dostavu, VRPPD
(Vehicle routing problem with pick-up and delivery)

Najčešće funkcije cilja za problem usmjeravanja vozila su:

- smanjenje broja vozila i vozača potrebnih da se posluže svi korisnici;
- smanjenje ukupnih troškova prijevoza koji direktno ovise o prevaljenom putu vozila i fiksnim troškovima vozila i vozača;
- uravnoteženje ruta s obzirom na vrijeme putovanja na ruti i ukrcanih dobara;
- smanjenje kazni zbog nemogućnosti posluživanja korisnika;
- kombinacija pojedinih ciljeva.

U narednom poglavlju opisat će se rješavanje problema usmjeravanja vozila s ograničenjem kapaciteta vozila[6] pomoću algoritma kolonije mrava.

---

[5] engl. vehicle routing problem (VRP)

[6] engl. capacitated vehicle routing problem (CVRP)

3.2 Definicija problema CVRP

Osnovni model problema usmjeravanja vozila je VRP s ograničenjima kapaciteta, CVRP. U ovom problemu svi su korisnici i njihovi zahtjevi unaprijed poznati, vozila su identična, a zajednička polazna točka im je u centralnom skladištu. Jedino ograničenje koje postoji jest kapacitet pojedinog vozila. Funkcija cilja izražava zahtjev za minimiziranje ukupnog troška.

CVRP se može izraziti korištenjem notacije teorije grafova na slijedeći način: neka je $G=(V,E)$ potpuni graf, $V=(0,...,n)$ skup vrhova, E skup lukova. Vrhovi *i=1, ..., n* predstavljaju korisnike. Vrh s indeksom 0 predstavlja skladište. Nenegativna cijena c_{ij}, a to može biti i neki drugi parametar poput vremena ili udaljenosti, povezana je s lukom $(i,j)\in E$ i predstavlja trošak prijevoza između korisnika *i* i *j*. Uobičajeno je ne koristiti lukove *(i,i)* i pretpostavlja se da je cijena $c_{ij}=\infty$ za sve $i\in V$.

Svaki korisnik *i (i=1,...,n)* povezan je s poznatim nenegativnim zahtjevom d_i, dok skladište ima fiktivni zahtjev $d_0=0$. Za podskup $S\rightarrow V$ postoji $d(S)=\sum_{i\rightarrow V} d_i$ koji predstavlja ukupni zahtjev skupa. U skladištu se nalazi skup od *K* identičnih vozila, svako sa kapacitetom *C*. Da bi rješenje problema bilo izvedivo potrebno je da vrijedi $d_i\leq C$ za svaki *i (i=1,...,n)*. Svako vozilo može izvesti najviše jednu rutu. Pretpostavlja se da K nije manji od K_{min}, gdje je K_{min} minimalni broj vozila[6] potrebnih da se posluže svi korisnici, iako je nemoguće uporabom algoritma kolonije mrava dobiti manji broj vozila od K_{min}.

Rješavanje CVRP-a predstavlja određivanje K ruta, s tim da je svaka ruta povezana samo s jednim vozilom, gdje ukupan trošak rute, ili duljina, treba biti minimalan. Ukupan trošak dobiva se kao suma cijena na svakom luku koji pripada ruti. Rješenje treba zadovoljavati ove uvjete:

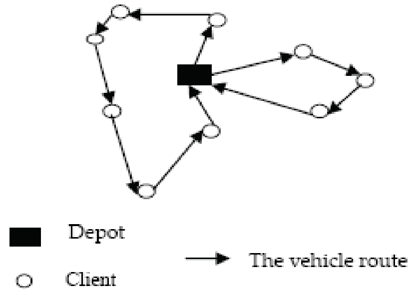
- svaka ruta treba započeti i završiti u skladištu;
- svaki korisnik sudjeluje u samo jednoj ruti;
- suma zahtjeva korisnika koji su posluženi u jednoj ruti ne smije biti veća od kapaciteta vozila.

[6] vrijednost K_{min} moguće je odrediti rješavajući BPP problem (engl. Bin Packing Problem)

3.3 Rješenje problema usmjeravanja vozila uz pomoć kolonije mrava

Model problema usmjeravanja vozila može biti predstavljen kao težinski graf $G = (V, E)$ gdje V{v_0, v_1, v_2, ..., v_n} predstavlja skup vrhova, a $E((v_i, v_j) : i \neq j)$ predstavlja skup bridova. Vrh v_0 predstavlja centralno skladište, a drugi vrhovi predstavljaju klijente. Svakom bridu (v_i, v_j) pridružena je nenegativna vrijednost d_{ij}. Prije navođenja jednadžbi, opisane su još neke dodatne varijable koje su potrebne za ispravnu formulaciju.

- $G = (V, E)$
- V={v_0, v_1,..., v_n}, v_0 je centralno skladište, a ostalo su klijenti
- q_i je zahtijev klijenta v_i, $i \in V$
- d_{ij} predstavlja udaljenost ili cijenu između v_i i v_j
- K={k_1, k_2,..., k_n} predstavlja skup vozila
- C je kapacitet svakog vozila $k_i \in K$, homogeni skup vozila



Slika 3.1: *graf CVRP-a*

Da bismo pronašli poredak posjeta klijentima, definiraju se varijable odluke:

$x_{i,j}^k$ = { 1, ako vozilo *k* posjećuje klijenta *j* odmah nakon klijenta *i*;

0, inače }

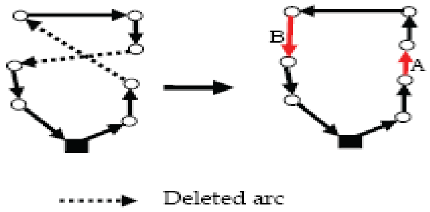
y_i^k = { 1, ako vozilo *k* poslužuje klijenta *i*;

0, inače }

Cilj algoritma usmjeravanja vozila može se skraćeno prikazati sljedećom funkcijom:

$$Min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} \cdot x_{ij}^k .$$

U svrhu poboljšanja učinkovitosti algoritma, može se primijeniti i 2-optimizacija. Princip 2-optimizacije, prikazan slikom 3.2, jest izbrisati dva brida na ruti i zamijeniti ih s neka druga dva brida u svrhu optimizacije duljine ili cijene rute.



Slika 3.2: *princip 2-optimizacije*

Algoritam rješavanja problema usmjeravanja vozila temelji se na hibridnom algoritmu optimizacije kolonijom mrava i lokalnog pretraživanja, 2 -optimizacija. U algoritam se uvodi heuristička optimizacija koja omogućuje bolju pretragu i računa je li bolje ostaviti dva klijenta u istoj ruti ili ih odvojiti. 2-optimizacija se primjenjuje na svakoj definiranoj ruti mrava trenutak prije osvježavanja feromona.

Prvenstveno je cilj, prilikom pretrage mrava, utvrditi isplati li se klijente v_i i v_j staviti u istu ili u različite rute. Klijenti v_i i v_j bit će smješteni u istu rutu ukoliko je faktor g_{ij} veći od iznosa koji odabere programer, a ovisi o vrijednostima *g* i *f*. U jednadžbi 3.1 opisan je izraz za faktor g_{ij}.

$$\gamma_{ij} = d_{i0} + d_{0j} - g \cdot d_{ij} + f \cdot |d_{i0} - d_{0j}| \tag{3.1}$$

Vrijednosti parametara g i f određene su po kriteriju koji odabire programer.

Algoritam kolonije mrava uzima u obzir intenzitet feromona i udaljenost najbližeg susjeda. Pristup koloniji mrava koji uključuje dodatnu uštedu zahtijeva i 2-optimizaciju. Prilikom inicijalizacije, *n* mrava postavimo u *n* gradova tako da je u svakome gradu jedan mrav. Svakom mravu je cilj posjetiti klijente, pritom pazeći na ograničenja i ponovno se vratiti u centralno skladište. Svaki mrav u jednom ciklusu posjećuje jednog klijenta samo jednom, ali skladište može posjetiti više puta. U trenutku kada mrav dosegne kapacitivno ograničenje, vraća se u skladište. Tada je zaokružena potpuna ruta vozila. Kad se mrav vrati u skladište, započinje ciklus ispočetka posjećujući klijente koje prethodno nije posjetio.

Tijekom procesa izgradnje ruta, mrav osvježava količine feromona na izabranoj ruti primjenom lokalnog pravila osvježavanja. Nakon što su svi mravi završili s izgradnjom ruta, primjenom globalnog pravila osvježavanja, obnavlja se feromonski trag.

Ukoliko se mrav nalazi u gradu v_i, za odabir prijelaza u jedan grad u skupu gradova V={v_i, v_{i+1},...}, koristi jednadžbu 3.2 i 3.3 kojom izračunava vjerojatnost odlaska u taj grad. Mrav na kraju odlazi u onaj grad za koji je vjerojatnost odlaska najveća.

$$j = (\arg\max_{u \in Fk(i)} ((\tau_{iu})^{\alpha} \cdot (\eta_{iu})^{\beta} \cdot (\gamma_{iu})^{\lambda}) \tag{3.2}$$

Jednadžba 3.3 opisuje vjerojatnost odlaska iz grada *i* u grad *j*.

$$p_{ij} = \frac{(\tau_{ij})^{\alpha} \cdot (\eta_{ij})^{\beta} \cdot (\gamma_{ij})^{\lambda}}{\sum_{u \in Fk(i)} (\tau_{iu})^{\alpha} \cdot (\eta_{iu})^{\beta} \cdot (\gamma_{iu})^{\lambda}} \ , \ u \in F_k(i)$$

$p_{ij} = 0$, inače. (3.3)

- F_k je popis gradova koje mrav *k* pozicioniran u gradu i nije posjetio
- t_{ij} je količina feromona na bridu e(i,j)
- h_{ij} je vidljivost između gradova *i* i *j*
- a, b, l su parametri koji određuju važnost pretraživanja na temelju feromonskog traga, ne temelji vidljivosti i važnost uštede

Prilikom odlaska iz grada *i* u grad *j*, mrav na temelju jednadžbe 3.4 osvježava feromonski trag na temelju lokalnog pravila osvježavanja.

$$\tau_{ij}^{novi} = \rho \cdot \tau_{ij}^{stari} + \rho \cdot \tau_0 \tag{3.4}$$

Nakon što mravi završe ciklus, globalno se osvježava feromonski trag na temelju 3.5.

$$\tau_{ij}^{novi} = \rho \cdot \tau_{ij}^{stari} + \Delta \cdot \tau_{ij} \tag{3.5}$$

- $\Delta \tau_{ij} = \frac{1}{L}$, *L* je udaljenost koju je prošao mrav *k*, a koji je prošao bridom e(i,j), a čiji L je najmanji od svih preostalih mrava koji su prošli tim bridom

Na ovako gore opisan način odluke odlaska u sljedeći grad, pritom pazeći na vrijednost kapaciteta vozila te pazeći na pravila lokalnog i globalnog osvježavanja feromonskog traga, u grafu G(V,E) pronalazi se *K* optimalnih ruta, koje poslužuju *K* vozila pritom prolazeći najoptimalnijim rutama u smislu cijene ili udaljenosti.

Pseudokod za optimizaciju kolonijom mrava na problem usmjeravanja vozila prikazan je na slici 3.1.

```
Inicijaliziraj program;
   Za I^max iteracija{
         Za sve mrave iz K generiraj novu rutu koristeći formulu 3.3;
         Poboljšaj sve rute koristeći 2-optimizaciju;
         Osvježi feromonske tragove koristeći formulu 3.5;
   }
```

Slika 3.1: *pseudokod za ACO VRP*

4. Usporedba optimizacije kolonijom mrava i drugih algoritama za rješavanje problema usmjeravanja vozila

Optimizacija kolonijom mrava pokazuje se kao jedan od najboljih i najbržih metaheurističkih algoritama. Ukoliko se ne bi koristila optimizacija kolonijom mrava na problem usmjeravanja vozila, već rješenje koje se generira na temelju mogućih permutacija ili mogućih petlji, izračun bi trajao tisućama, milijunima ili milijardama godina.

Broj mogućih rješenja jest približno n!, a optimizacijom kolonijom mrava problem je praktički sveden na polonomijalni problem. Naglašavam praktički jer je u uvodu objašnjena klasa problema u koju ulazi problem usmjeravanja vozila.

[6] Nakon opisa algoritma usmjeravanja vozila pomoću kolonije mrava, pokazani su dobiveni rezultati. Eksperimenti su rađeni na laptopu s procesorom Pentium 4, frekvencije 2.4 GHz, operacijskim sustavom Windows XP i radnom memorijom od 512 MB. Algoritam je pisan u programskom jeziku Java. Eksperiment se sastojao od više inačica problema koji su prethodno riješeni nekim drugim algoritmom, npr. genetskim algoritmom. Cilj je bio pokazati koliko je rješenje problema usmjeravanja vozila s ograničenjem kapaciteta dobro ili loše u odnosu na dosad najbolji dobiven rezultat.

Početni parametri inicijalizirani su na sljedeće vrijednosti:

$$\alpha = 1 \;, \beta = 2 \,, \lambda = 1 \,, \rho = 0.9 \,, f = g = 2 \,, \tau_0 = 10^{-6} \,.$$

Opis ovih parametara opisan je u poglavljima 2. i 3.

Ime inačice problema sastoji se od 3 dijela [12]:

1. skup u koji inačica pripada
2. broj klijenata
3. najmanji broj vozila kojim je problem rješiv

Tablica 4.1.: inačice problema za testiranje

Inačica	n	Q	Inačica	n	Q	Inačica	n	Q
B-n68-k9	68	100	C1	50	160	P-n76-k5	76	280
B-n78-k10	78	100	C2	75	140	E-n76-k10	76	140
B-n66-k9	66	100	C3	100	200	E-n76-k15	76	100
B-n50-k8	50	100	C4	150	200	A-n60-k9	60	100
B-n57-k9	57	100	C5	199	200	A-n63-k10	63	100
B-n63-k10	63	100	C11	120	200	A-n69-k9	69	100
B-n67-k10	67	100	C12	100	200	A-n80-k10	80	100
P-n51-k10	51	80	A-n63-k9	63	100	-	-	-

U tablici 4.1 [6] *n* predstavlja broj klijenata, a *Q* kapacitet vozila.

U tablici 4.2 [6] prikazani su rezultati eksperimenta. Prva kolona opisuje inačicu problema, druga kolona opisuje dosad najbolje rezultate, udaljenost i broj vozila, dobivene nekim drugim algoritmom, treća kolona opisuje rezultate dobivene eksperimentom, a četvrta kolona opisuje devijaciju između udaljenosti dobivene eksperimentom i već poznate najbolje udaljenosti.

Tablica 4.2: *CVRP rezultati eksperimenta*

inačica	dosad najbolji rezultat		dobiveni rezultat		devijacija
	udaljenost	broj vozila	udaljenost	broj vozila	
B-n68-k9	1304	9	**1300.91**	9	**-0,23%**
B-n78-k10	1266	10	**1238.28**	10	**-2,19%**
B-n66-k9	1374	9	**1371.67**	9	**-0,17%**
B-n50-k8	1313	8	1317.34	8	0,30%
B-n57-k9	1598	9	**1598.0**	9	**0%**
B-n63-k10	1537	10	1540.71	10	0,19%
B-n67-k10	1033	10	1035.46	10	0,19%
A-n60-k9	1408	9	**1357.70**	9	**-3,57%**
A-n63-k9	1634	9	1636.94	9	0,18%
A-n63-k10	1315	10	1322.93	10	0,60%
A-n69-k9	1168	9	1177.36	9	0,80%
A-n80-k10	1764	10	1787.05	10	1,30%
P-n51-k10	745	10	**743.26**	10	**-0,23%**
P-n76-k5	631	5	**631.0**	5	**0%**
E-n76-k10	832	10	836.37	10	0,52%
E-n76-k15	1032	15	**1030**	15	**-0,19 %**

Iz tablice 4.2 vidi se da je prosječna devijacija 0.022%, što upućuje da je algoritam kolonijom mrava našao približno identične rezultate kao i prijašnji algoritmi. Štoviše, masno su podebljani rezultati koji upućuju da je algoritam kolonijom mrava pronašao bolje rješenje od dosad najboljeg. Ovakav rezultat dobiven je nakon 500 iteracija, što znači da bi nakon većeg broja iteracija mogli očekivati čak i bolja rješenja. Broj vozila u oba slučaja ostao je isti, a to nam potvrđuje da algoritam ne povećava broj minimalnih resursa, već samo možemo očekivati, u ovisnosti o broju iteracija, promjenu udaljenosti.

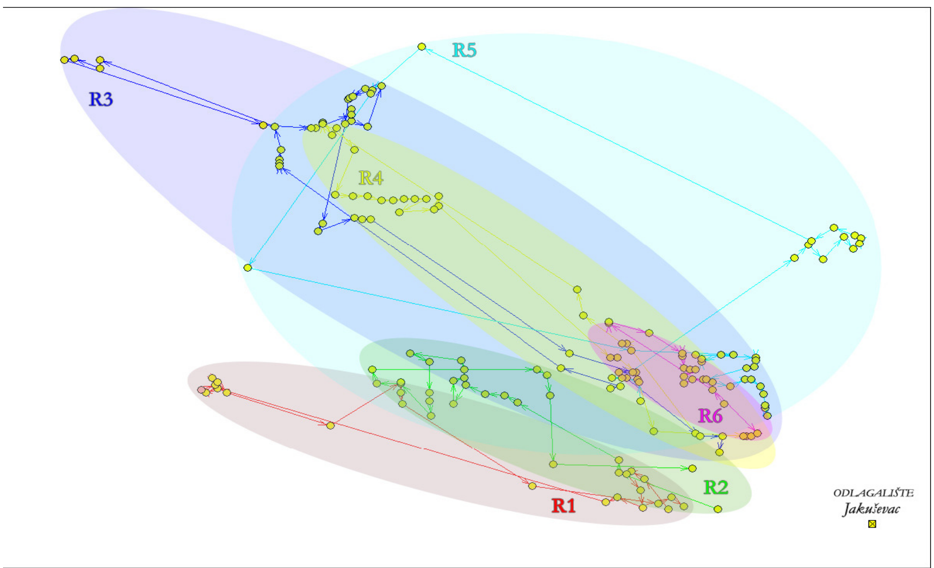
Osim što algoritam kolonije mrava pokazuje odlična svojstva u pronalasku najkraćih ruta, važan faktor predstavlja i vrijeme izvođenja. U tablici 4.3 [6] pokazana su vremena izvođenja algoritma kolonije mrava bez 2-optimizacije i s 2-optimizacijom, te devijacija udaljenosti od najboljeg rezultata.

Tablica 4.3: *vremena izvođenja algoritama*

inačica	dosad najbolji rezultat	ACO bez 2-opt			ACO s 2-opt		
		cijena	vrijeme	devijacija	cijena	vrijeme	devijacija
C1	524.61	524.61	6	0%	524.61	**2**	0%
C2	835.26	844.31	78	1.08%	838.86	**20**	0.43%
C3	826.14	832.32	288	0.75%	834.77	**35**	1.04%
C4	1028.42	1061.55	4048	3.22%	1035.23	**80**	0.66%
C5	1291.45	1343.46	5256	4.03%	1304.25	**200**	0.99%
C11	1042.11	1065.21	552	2.22%	1046.81	**32**	0.45%
C12	819.56	819.56	300	0%	819.56	**30**	0%

Iz tablice 4.3 vidi se da je algoritam kolonije mrava nadopunjen s 2-optimizacijom brži i točniji.

S problemom usmjeravanja vozila susrećemo se svakodnevno. Jedan takav problem može se projecirati na zagrebačku Gradsku čistoću. Svakodnevno kamioni određenog kapaciteta kreću s odlagališta "Jakuševac" pokupiti otpad i nakon što napune vozilo, vraćaju se na odlagalište. Cilj Uprave gradske čistoće je obavljati posao po najoptimalnijem ekonomskom modelu-potrošiti što manje novca. Problem se može prikazati problemom usmjeravanja vozila s ograničenjem kapaciteta, a može se efikasno i učinkovito riješiti algoritmom kolonije mrava.



Slika 4.1: *graf kretanja vozila Gradske čistoće u Zagrebu*

5. Zaključak

Za rješavanje problema distribucije ili prikupljanja potrebno je opise tehnoloških procesa približiti opisu matematičkog problema. Sam problem distribucije i prikupljanja dobara razmatra se kao prošireni problem usmjeravanja vozila s kapacitivnim ograničenjima. Taj se matematički kombinatorno optimizacijski problem, osim za slučaj manjeg broja vozila, ne može općenito i egzaktno riješiti u prihvatljivom vremenu. Do praktičnih rješenja dolazi se aproksimativnim metodama koje su najčešće heurističke prirode ili metaheurističke prirode od kojih je jedna optimizacija kolonijom mrava.

Optimizacija kolonijom mrava rješava vrlo efikasno i učinkovito problem usmjeravanja vozila ograničenog kapaciteta što se može iskoristiti u svrhu pozitivnog ekonomskog pomaka kod tvrtki koje se svakodnevno susreću s takvim problemom.

Smatram da su metaheuristički algoritmi odlično sredstvo u borbi protiv velikih složenosti i dugotrajnog računanja. Rezultati ukazuju na opravdanost upotrebe algoritma kolonije mrava za planiranje ruta u cilju predlaganja novih ruta koje obavljaju isti posao uz vidne uštede.

6. Literatura

[1] Čupić Marko, Algoritam kolonije mrava , 2009., *Prirodom inspirirani optimizacijski algoritmi*,http://www.fer.hr/_download/repository/Cupic2009-PrirodomInspiriraniOptimizacijskiAlgoritmi.pdf, 15.travnja 2010.

[2] Vehicle routing problem, 3. veljače 2007., *Vehicle routing problem*,

http://en.wikipedia.org/wiki/Vehicle_routing_problem, 15. travnja 2010.

[3] Carić, T., Unapređenje organizacije transporta primjenom heurističkih metoda, doktorska disertacija, Fakultet prometnih znanosti, 2004.

[4] Stokić, I., Rješavanje problema usmjeravanja vozila genetskim algoritmom, završni rad br. 958, Fakultet elektrotehnike i računarstva, 2009.

[5] Bullnheimer, B., An improved ant system algorithm for the vehicle routing problem, Sveučilište u Beču, Institut ekonomskih znanosti, 1997.

[6] Gold, H. i Carić,T. , Vehicle routing problem, ISBN 978-953-7619-09-1, 2008., Sveučilište u Rijeci, 25.ožujka 2010.

[7] Montemanni, R., Ant colony system for a dinamic vehicle routing problem, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, 2008., 25. ožujka 2010.

[8] Maniezzo, V., Ant colony optimization, 2007., http://www.google.hr/#hl=hr&q=ant+colony+optimization+Maniezzio&aq=f&aqi=&aql=&oq=&gs_rfai=&fp=a29b5b2763d62a6a, 15. travnja 2010.

[9] Ant colony optimization, 2009., http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html, 12. travnja 2010.

[10] Genetic and ant colony optimization algorithms, http://www.codeproject.com/KB/recipes/GeneticandAntAlgorithms.aspx, 12. travnja 2010.

[11] Ant colony optimization, 2007., http://www.scholarpedia.org/article/Ant_colony_optimization, 25. travnja 2010.

[12] CVRP Instances, http://osiris.tuwien.ac.at/~wgarn/VehicleRouting/neo/Problem%20Instances/CVRPinstances.html, 25. travnja 2010.

7. Sažetak

Opisan je metaheuristički algoritam optimizacije kolonijom mrava za rješavanje kapacitivnog problema usmjeravanja vozila. Objašnjena su prirodna i programska svojstva mrava i kolonije te algoritamsko ostvarenje osnovnih funkcija mrava. Prikazane su jednadžbe na kojima mrav temelji svoje odluke.

Algoritam kolonije mrava upotrebljen je u svrhu optimizacije rješavanja kapacitivnog problema usmjeravanja vozila. Objašnjen je i definiran problem usmjeravanja vozila i svojstva koja su nadodana algoritmu kolonije mrava u svrhu bolje optimizacije.

Usporedbom s drugim algoritmima vidljiva je kvaliteta rješenja dobivena kolonijom mrava te pokazano je rješenje jednog realnog problema kapacitivnog usmjeravanja vozila.