

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

**SEMINAR**

**Dinamički problem raspoređivanja vozila – neke  
instance problema i metode rješavanja**

*Stjepan Zelić*

Voditelj: *Doc.dr.sc. Marko Đurasević*

Zagreb, svibanj 2020.

## Sadržaj

1. Uvod.....	3
2. VRP.....	5
2.1 CVRP – Capacitated Vehicle Routing Problem.....	5
2.1.1 Metode rješavanja .....	6
2.2 DVRP.....	7
2.2.1 DVRPTW.....	8
2.2.2 DPDVRP .....	8
2.3 Metode rješavanja.....	8
2.3.1 Multiagentski sustavi.....	9
2.3.2 Algoritam kolonije mrava.....	9
2.3.3 Genetičko programiranje.....	10
3. Zaključak.....	12
4. Sažetak .....	13
5. Literatura.....	14

## 1. Uvod

Zamislimo slijedeći scenarij: moramo doći na lokaciju na kojoj do sada nismo bili, pa ne znamo put na pamet. Vrijeme nam je važno, pa tamo želimo doći što prije ili želimo potrošiti čim manje goriva, pa nam je bitna duljina puta.

Ono što milijuni ljudi na zemlji svakodnevno rade u ovoj situaciji jest da pale navigacijski softver na svom telefonu, unesu odredište, i zabace se dok im računalo u milisekundama izračunava optimalnu putanju do odredišta i daje upute tijekom vožnje. Ovaj problem je prilično izravan, računalo koje nam računa rutu, bilo ono u oblaku ili na našem fizičkom uređaju, ima okolinu (ceste i raskrižja) reprezentirano kao grafove, te nekim algoritmom izračuna optimalnu rutu kretanja našeg vozila.

Što se događa ako toj ruti dodamo više odredišta? To je već malo kompleksniji problem. Tom problemu možemo dodati i više vozila, tako da svako vozilo mora doći na svoja odredišta.

Ali što ako smo, naprimjer, vlasnik lanca benzinskih crpki, svo gorivo nam se nalazi u centralnom skladištu, imamo  $n$  cisterni na raspolaganju, i trebamo koristeći tih  $n$  cisterni dostaviti gorivo do svih benzinskih crpki tako da potrošimo što manje vremena i goriva za cisterne? Taj problem više nije tako trivijalan kao onaj s kojim smo počeli, i zove se *Vehicle Routing Problem*, ili na hrvatskom Problem Usmjeravanja Vozila, kojeg ćemo dalje u tekstu oslovljavati s „VRP“.

VRP je formalno začet krajem pedesetih godina prošlog stoljeća, kada su George Dantzig, takozvani otac linearnog programiranja, i John Ramser postavili točno ovaj problem koji uključuje gorivo i benzinske crpke [1]. Taj problem su riješili bez upotrebe računala matematičkom metodom - linearnim optimiranjem. Čak su autori, kada su našli skoro optimalno rješenje s četiri rute za problem s dvanaest crpki, izjavili „Još nisu stvorene praktične aplikacije ove metode“. [2]

Brzim rastom svjetske ekonomije, samim time i logističkih potreba, razvojem računala i znanosti, ovaj problem se pokazao kao vrlo važan praktični i teorijski problem u znanosti. Interes za VRP kao kombinatorni optimizacijski problem ne proizlazi samo iz njegove praktične važnosti, već i iz značajne kompleksnosti – ovaj problem i njegove inačice su NP-teški problemi, što znači da ih se praktički ne može egzaktno riješiti u polinomijalnom vremenu, osim za male instance problema, odnosno za mali broj lokacija i vozila.

Svakodnevno se milijuni dostavnih vozila, pogona u tvornicama, logističkih tvrtki, i mnogih drugih entiteta oslanjaju na računala da im riješe instance VRP-a, ali kao što ćemo vidjeti kasnije u ovom radu, ne rješavaju se baš VRP problemi, nego njegove instance. Dobar primjer jest flota vozila UPS (engl. *United Parcel Service*), koja se sastoji od oko 123000 vozila, te dnevno dostave 3.2 milijuna pošiljki. Otprilike se po vozilu dostavi 26 pošiljki dnevno, a o broju ukupnih mogućnosti dostave nema smisla ni govoriti, kad znamo da već jedno vozilo ima preko  $4 \times 10^{26}$  mogućnosti za dostavu svojih 26 pošiljki. [3]

No, moramo se vratiti na naš osnovni problem kako bismo mogli prijeći u praktične dimenzije.

VRP ima  $n$  vozila s jednakim kapacitetom,  $m$  odredišta, ima unaprijed zadane zahtjeve koji se ne mijenjaju, i vozila odrade svoju rutu jednom i vrate se natrag u centralno skladište.

Što ako se klijenti ili odredišta tijekom dostave predomisle ili se nešto promijeni? Što ako im je potrebno dostaviti robu samo u određenom vremenskom okviru? Što ako imamo vozila s različitim kapacitetima, ako se vozila mogu vratiti u skladište i nastaviti s dostavom? U praksi, kao u primjeru UPS-a koji smo spomenuli ranije, ovakve se situacije zaista pojavljuju, pa ćemo vidjeti kako se VRP-u dodatno definiraju ograničenja kako bi se mogao primijeniti na širok spektar raznih scenarija.

U svojoj osnovnoj varijanti VRP rješava situaciju u kojoj unaprijed znamo sve informacije – broj dostupnih automobila, broj klijenata, lokacije klijenata, narudžbe i količine, ali takav ne pokriva ranije navedene stvarne situacije. Ako u opis problema dodamo mogućnost promjene bilo kojeg od ovih parametara tijekom izvršavanja algoritma, odnosno tijekom obilaska ruta, dobili smo novi problem.

Tako definiran VRP naziva se DVRP, ili *Dynamic Vehicle Routing Problem*, koji mnogo slobodnije može simulirati stvarne situacije.

Danas pretraga riječi „*vehicle routing problem*“ na Google Scholar platformi daje oko 891000 rezultata, što govori o popularnosti i važnosti ovog problema.

Naravno, u ovom radu nećemo opisati sve postojeće inačice problema, kao ni sve postojeće metode za rješavanje, jer ih ima previše, ali opisat ćemo neke odabrane i poznatije metode kao i zanimljivije inačice.

U prvom poglavlju obradit ćemo CVRP, inačicu VRP-a s ograničenim kapacitetom vozila. Formalno ćemo definirati VRP pomoću teorije grafova, a u njegovom potpoglavlju ćemo proći kroz neke metode rješavanja CVRP-a također se osvrćući na prošlost.

U drugom poglavlju precizno ćemo pokazati koje su razlike između statičkog i dinamičkog VRP-a uvodeći dodatne elemente, nakon čega ćemo istražiti dvije inačice dinamičkog VRP-a – inačicu s vremenskim prozorima i s prikupljanjem i dostavom.

U trećem poglavlju susrest ćemo se s nekim modernim metodama rješavanja problema proučenih u drugom poglavlju – dobit ćemo uvid u ideje, algoritme i rezultate različitih metoda kao i osvrt na usporedbu metoda baziranih na različitim paradigmama. Metode koje su obrađene su multiagentski sustavi, algoritam kolonije mrava i genetičko programiranje.

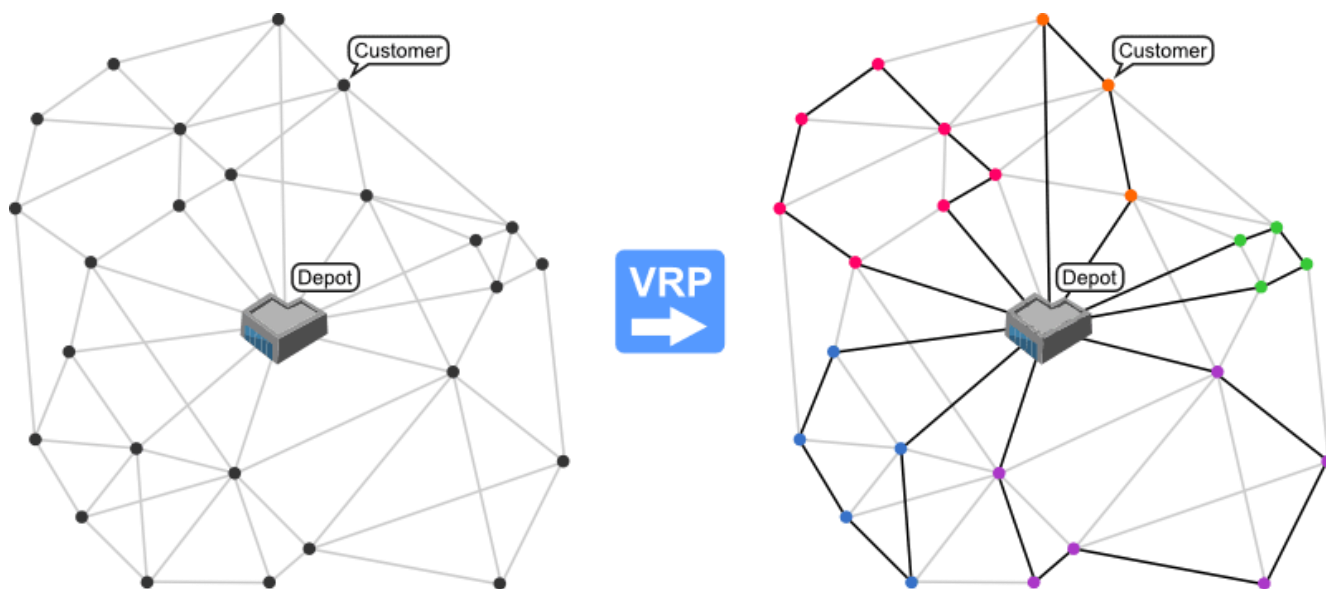
## 2. VRP

### 2.1 CVRP – *Capacitated Vehicle Routing Problem*

Osnovna varijanta VRP-a se zove Kapacitivni Problem Usmjerivanja Vozila, gdje vozila imaju ograničen kapacitet za robu koju dostavljaju. Formalnija definicija CVRP-a glasi ovako:

- $G(V, E)$  je neusmjereni graf
- $V = \{v_0, v_1, \dots, v_m\}$  - skup vrhova grafa, gdje se skladište (centralno skladište) nalazi u  $v_0$ , a svi ostali vrhovi (njih  $m$ ) osim  $v_0$  su klijenti ili odredišta.
- $n$  je broj vozila, gdje je jedna ruta  $R_i$  dodijeljena jednom vozilu, a  $R_i$  ruta, to jest skup lukova i vrhova koji čine dio rješenja.

Imamo jedno centralno skladište ili skladište kao čvor na grafu,  $n$  vozila, svako s jednakim kapacitetom,  $m$  odredišta kao čvorove na grafu, svako odredište  $i$  ima zahtjev za ukupno  $d_i$  robe (Slika 2.1).



Slika 2.1 : Vizualizacija problema i njegovog rješenja (dohvaćeno s <http://neo.lcc.uma.es/vrp/vehicle-routing-problem/> 10.5.2020. )

Jedino ograničenje u ovoj inačici VRP-a jest ograničenje svih vozila – suma traženih dostava ( $d_i$ ) na ruti jednog vozila  $j$  ne smije premašiti kapacitet tog vozila  $Q$ . S obzirom da su nam svi kapaciteti jednaki, ne moramo indeksirati  $Q$  (formula 1)

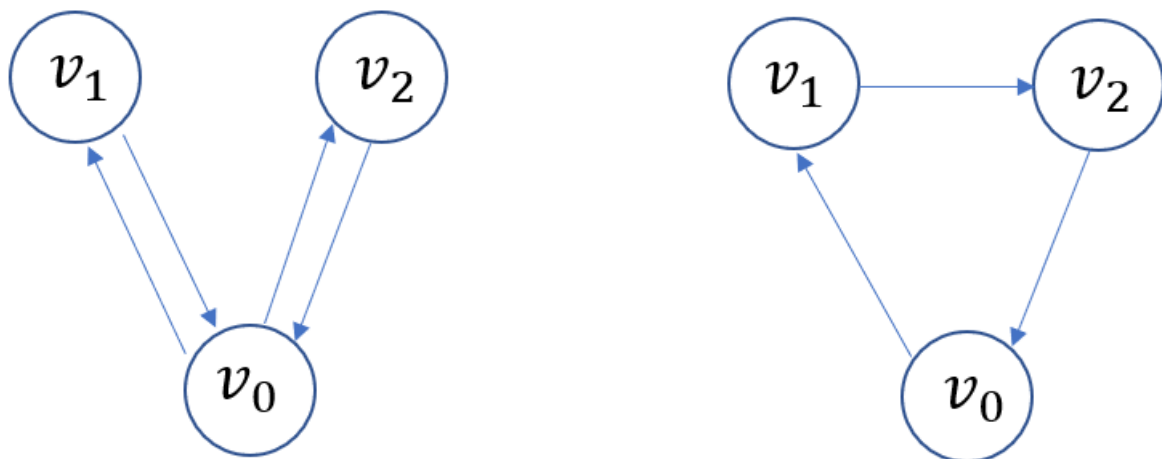
$$\sum_{i=1}^{m_j} d_i \leq Q \quad (1)$$

gdje  $m_j$  označava j-tu lokaciju za jedno vozilo,  $d_i$  označava količinu robe koja se treba dostaviti kupcu  $i$ . [4]

### 2.1.1 Metode rješavanja

Za CVRP su kroz povijest razvijana egzaktne metode rješavanja koje daju optimalan rezultat pretragom cijelog prostora stanja, ali i heurističke metode jer se egzaktne metode radi kombinatorne eksplozije ne mogu nositi s većim instancama problema. Na primjer, Clarke i Wright su nedugo nakon objave originalnog rada predložili poboljšanje algoritma heuristikom uštede (engl. *Savings algorithm*), čime su poprilično poboljšali vrijeme rješavanja i dali dobra rješenja za veće instance problema.

Heuristika je funkcionirala tako da je spajala različite rute na način da se dobije najveća ušteda u ukupnoj udaljenosti. Umjesto da trošimo vrijeme na povratak u skladište i odlazak prema novom klijentu, možemo direktno produžiti prema novome klijentu bez vraćanja u skladište. Primjer spajanja ruta nalazi se na slici Slika 2.2



Slika 2.2 : Clarke i Wright heuristika na 2 vrha [5]

Branch and Bound metoda rješavanja je egzaktna metoda rješavanja CVRP-a koja grananjem i ograđivanjem smanjuje prostor pretraživanja rješenja. Grananjem se problem smanji na potprobleme tako što grupiramo susjedne vrhove, pa ograničimo vrijednosti varijabli dijelimo prostor pretraživanja na manje potprostore. Na taj način možemo procjenom donjih i gornjih granica izbaciti potprostore koji ne zadovoljavaju unaprijed definirane granice. [6]

S obzirom da je ovo egzaktna metoda, njezina kompleksnost ovisi o broju vrhova u grafu što opet dovodi do kombinatorne eksplozije pa već se za veći broj vrhova moramo oslanjati na druge metode koje koriste heuristike i metaheuristike.

Metaheurističke metode, poput evolucijskih i genetskih algoritama, mogu optimirati sam problem, postojeće heuristike koje optimiraju problem, ili stvarati nove heuristike, a često su inspirirane prirodnim metodama, no o njima će detaljnije biti riječi u poglavlju dinamičkog problema raspoređivanja vozila.

## 2.2 DVRP

U statičkom problemu raspoređivanja vozila, koji smo obradili u prethodnom poglavlju, pretpostavka je da *a priori* imamo sve potrebne informacije o zahtjevima klijenata, te da se te informacije neće mijenjati, kao ni broj naših vozila, njihovi kapaciteti, i ostali parametri problema. U kontrastu, u obitelji dinamičkih problema, parametri problema se mogu, i općenito hoće, promijeniti tijekom izvršavanja algoritma i izračuna ruta vozila. Izvršavanje algoritma i izračunavanja ruta nisu odvojeni događaji, već se paralelno izmjenjuju u dinamičkom okruženju.

Formalna definicija DVRP-a proizlazi iz VRP-a, koju smo definirali u prethodnom poglavlju. U najvećem broju slučajeva cilj je minimizirati veličinu flote vozila i ukupnu udaljenost prijeđenu nakon što su sva vozila obavila dostave svim klijentima, ali u dinamičkoj simulaciji možemo optimirati i druge rezultate, poput broja odbijenih zahtjeva.

Najčešće se dinamičnost uzrokuje tako što dozvoljavamo novim zahtjevima da dolaze dok traje izvršavanje već postojećih zahtjeva. Ako za novi zahtjev možemo generirati rješenje koje će zadovoljiti naša ograničenja, osvježavamo trenutno rješenje problema novim rutama koje će uključiti novi zahtjev, a ako ne, odbijamo taj zahtjev i nastavljamo izvršavanje dotadašnjim rješenjem.

Druga varijanta dinamičnosti je stohastičnost klijenata. Umjesto pojave novih klijenata i zahtjeva, postojeći klijenti mogu odlučiti promijeniti svoja svojstva, poput promjene količine robe za isporuku, promjene koordinata, promjene vremenskog prozora, i slično. Dinamičnost se može postići kombinacijom ove dvije varijante, što je također nerijetko i slučaj u stvarnome svijetu.

Kao i u statičnom VRP-u, možemo uzimati u obzir homogene ili heterogene skupine vozila.

Varijanta s homogenom skupinom vozila je standardna varijanta VRP-a, gdje svako vozilo ima jednaki kapacitet, a u heterogenoj varijanti svako vozilo ima različiti kapacitet, pa u tom slučaju za svaku rutu mora vrijediti da suma pojedinačnih količina roba za svakog klijenta ne smije iznositi više od kapaciteta vozila određenog za tu rutu.

Korisnost dinamičke promjene parametara prepoznaje se u primjeni na razne situacije u praksi: naprimjer, možemo simulirati stanje u prometu u realnom vremenu tako da povećavamo težine bridova tamo gdje je promet zakrčeniji, možemo simulirati kvar nekog vozila, i slične pojave koje se događaju dinamički i bez mogućnosti da ih predvidimo.

### 2.2.1 DVRPTW

*Dynamic Vehicle Routing Problem with Time Windows* je inačica DVRP-a u kojoj svaki klijent za svoju dostavu ima odgovarajući vremenski prozor definiran kao  $[e_i, l_i]$ , gdje je  $e_i$  „earliest arrival time”, ili najranije vrijeme dolaska, a  $l_i$  latest arrival time, ili najkasnije vrijeme dolaska. Dostava klijentu  $i$  ne može se izvršiti prije  $e_i$  niti nakon  $l_i$ , a dostava također može trajati neko vrijeme  $s_i$  (engl. service time) koje je potrebno za istovar robe na odredištu. Također, skladište ima svoj vremenski prozor unutar kojeg sve dostave moraju započeti i završiti, tj. prvo vozilo mora krenuti nakon  $e_0$ , a zadnje dostavno vozilo se mora vratiti prije  $l_i$  kako bi se rješenje smatralo ispravnim.

### 2.2.2 DPDVRP

*Dynamic Vehicle Routing with Delivery and Pickup Problem* je klasa DVRP-a u kojoj se uvodi mogućnost povratka dobara kod klijenta tako da ta roba stane u vozilo, dakle uvodimo i utovar povrh istovara. Ovaj dodatak originalnom problemu otežava stvari - sada moramo paziti da ne popunimo vozila, ali i da nam vozila ne budu skoro prazna cijelim putem radi utovara pred kraj rute. Također, kao u prethodnim dinamičkim inačicama, tijekom izvršavanja algoritma mogu se promijeniti svi i stvoriti novi parametri, pa se tako može promijeniti količina robe za utovar, kao i pojaviti novi takvi zahtjevi.

Generalno se razmatra verzija u kojoj se roba prevozi od klijenta nazad u skladište, tako da ne vršimo dostavu dobara među klijentima, iako postoji i ta varijanta. Moguća simplifikacija problema je odvajanje na dvije faze u kojoj se tijekom prve faze vrše samo dostave iz depoa prema klijentima, a tijekom druge faze vrši samo skupljanje robe od klijenata i dostava u skladište.

## 2.3 Metode rješavanja

U ovom poglavlju proučit ćemo neke od metoda rješavanja inačica DVRP-a. Opisat ćemo metode neovisno o problemu, pa ćemo zatim za svaku metodu navesti na koju se inačicu fokusirala, kao i rezultate i zaključke radova koji su koristili te metode. Metode koje ćemo proučiti su MAS – Multi-Agent System, Algoritam kolonije mrava, i genetičko programiranje.



### 2.3.1 Multiagentski sustavi

U multiagentskom pristupu DVRP-u Barbuca [7] koristi slijedeću arhitekturu sustava:

Proces simulacije i rješavanja DVRP-a odrađuju agenti koji imaju iduće uloge:

- GlobalManager (GM) – inicijalizira sve agente
- RequestGenerator (RG) – generira nove zahtjeve ili ih čita iz datoteke
- Vehicle (Vh) – vozila koja poslužuju zahtjeve klijenata
- RequestManager (RM) – vodi brigu o zahtjevima i alokira ih vozilima
- Optimizing (OPT) – rješava parcijalne statičke probleme koji uključuju već postojeće zahtjeve.

U ovom pristupu DVRP se rješava tako da se dan u kojem se dostave moraju obaviti prvo podijeli na diskretan skup jednakih vremenskih perioda. Nakon svakog vremenskog perioda svi trenutno poznati zahtjevi se dodaju u skup zahtjeva. Ovakvim pristupom se na kraju svakog perioda svi zahtjevi poslože u jednu statičku instancu VRP-a koju potom rješava OPT agent.

Za optimizaciju koristi se VNS heuristika (Slika 2.3), ili Variable Neighborhood Search, algoritam koji dijeli čvorove na susjedstva i istražuje ih u dubinu.

---

#### Algorithm 2. VNS\_DVRP(s)

---

**Input:**  $s$  - current solution

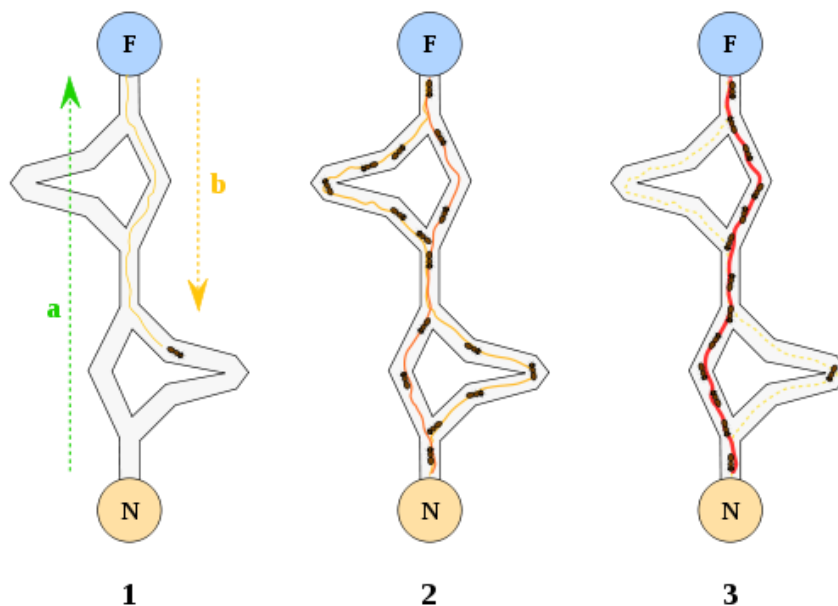
```
1:  $k \leftarrow 1$ 
2:  $k_{max} \leftarrow 3$ 
3: while ( $k \leq k_{max}$ ) do
4:   Select the group of neighborhoods  $N_k$ 
5:   Explore  $N_k$  using a predefined family of still deeper neighborhoods:  $N_{k1}$ , and
   next  $N_{k2}$  to find the most improving neighbor  $s'$  of  $s$  in  $N_k$ 
6:   if ( $f(s') < f(s)$ ) then
7:      $s \leftarrow s'$ 
8:      $k \leftarrow 1$ 
9:   else
10:     $k \leftarrow k + 1$ 
11:   end if
12: end while
```

**Output:**  $s$  - improved solution

Slika 2.3 : Pseudokod za VNS heuristiku u rješavanju DVRP [7]

### 2.3.2 Algoritam kolonije mrava

ACO (engl. *Ant Colony Optimization*) je poznati optimizacijski algoritam inspiriran prirodom. Bazira se na kretanju stvarnih mrava prilikom pronalaska rute do hrane. Pokazano je da od svih postojećih ruta, mravi uvijek pronađu onu najkraću. Kako bi to postigli, mravi putem ostavljaju trag feromona, kemijske tvari koju koriste za komunikaciju. Najkraći put ima najveću koncentraciju feromona pa to dovodi do sve više mrava koji koriste ovu rutu (Slika 2.4).



Slika 2.4 Algoritam kolonije mrava. Preuzeto s <https://notyetsecure.com/ant-colony-optimization.html> 10.5.2020.

Kako funkcionira algoritam kolonije mrava? U početnom stanju mravi će nasumično ići u raznim smjerovima u potrazi za hranom. Pretpostavimo, bez smanjenja općenitosti, da postoje samo dva smjera: kraći i dulji, i da oba završavaju hranom. Mrav koji krene kraćim putem će prije doći do hrane i vratiti se natrag, čime će ojačati feromone na kraćem putu prije nego to stigne učiniti drugi mrav.

Za mrave koji su još u mravinjaku veće su šanse da će krenuti kraćim putem nego duljim radi jačeg feromonskog traga. No, kako se više mravi nastavlja kretati kraćim putem nego duljim, i treba im manje vremena, tako se kraći put ojačava feromonskim tragom dok ne postane jedini put kojeg mravi biraju.

U radu [8] Euch, Yassine, i Chabchoub istražuju algoritam kolonije mrava s 2-opt heuristikom lokalne pretrage za rješavanje DPDVRP-a.

Za trenutno najbolje rješenje, 2-Opt algoritam traži nova rješenja zamjenama parova susjednih lokacija na istoj ruti, te tako potencijalno nalazi bolja rješenja.

Prema testiranjima obavljenima u [8], ova metoda se pokazala kao superiorna svim dotadašnjim algoritmima u rješavanju dinamičkog problema raspoređivanja vozila s dostavama i pokupljanjima.

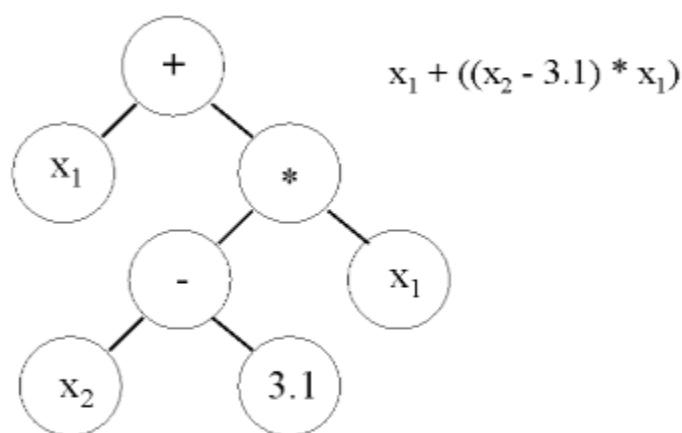
### 2.3.3 Genetičko programiranje

Genetičko programiranje, u tekstu označavana kao GP, koristi se kao primarna metoda rješavanja mnogih izazovnih kombinatornih problema radi uspjeha u rješavanju raznovrsnih problema, koji proizlazi iz drugačijeg pristupa problemima od većine evolucijskih algoritama. Do ideje genetičkog programiranja došao je John Koza tako što je promatrao računalne programe iz perspektive prirodne selekcije: programi su bili organizmi koji su se kroz generacije mijenjali i pri tome stvarali lošije i bolje verzije

sebe: lošije verzije ne „prežive” odabir, a bolje prežive, te se proces nastavlja kroz generacije kako bi nastali skoro optimalni programi za zadane probleme.

Prvo su se optimirale funkcije, onda se došlo do toga da se algoritmima genetičkog programiranja optimiraju heuristike koje se koriste za optimiranje zadanih problema. Ta klasa algoritama za optimizaciju nazivaju se hiperheuristike. Hiperheuristike su se pokazale vrlo uspješnima u rješavanju teških problema, a autori Jacobsen-Grocott, Mei, Chen i Zhang [9] su se u svom radu bavili primjenom genetičkog programiranja na DVRPTW klasu.

Reprezentacija jedinice u GP-u je binarno stablo kojemu su čvorovi operatori, tj. funkcije koje mogu primiti argumente ili terminali, tj. završni znakovi.



Slika 2.5 : Primjer reprezentacije funkcije kao jedinice u GP-u

Prvo se generira skup jedinica koje izgledaju kao Slika 2.5 : Primjer reprezentacije funkcije kao jedinice u GP-u, pa se sve takve jedinice evaluiraju na skupu za učenje te se vrši selekcija jedinica. Selekcija je jedan od parametara hiperheuristike, kao i mogući operatori ili terminali, a možemo koristiti neke od već postojećih selekcija u GP-u, kao i stvoriti svoju selekciju.

Ovakva stabla nakon selekcije se u postupku GP-a križaju i mutiraju, time stvarajući nove jedinice sve dok se ne postigne uvjet zaustavljanja, bio on broj generacija, ili granica na funkciju dobrote. Eksperimentalni rezultati ovog rada pokazali su kako GP hiperheuristike imaju značajno bolje performanse od ručno dizajniranih heuristika, što se dodatno pokazalo povećavanjem stupnja dinamičnosti [9].

### **3. Zaključak**

VRP i DVRP su se pokazale kao vrlo bitne klase problema u današnjem svijetu, a izgleda da postaju sve relevantnije rastom performansi računala, pronalaskom novih metoda rješavanja, te sve većom logističkom potrebom za raspoređivanjem vozila.

S obzirom da je VRP kao jedan od najbitnijih kombinatornih problema danas i među najpopularnijima, mnoge inačice VRP-a i DVRP-a su široko proučavane i unaprijeđivane kroz godine, no u tom području ima još mnogo mjesta za poboljšanja i napredak.

Radovi referencirani u ovom seminarskom radu pokazali su kako se određene metode na nekim inačicama problema ranije nisu isprobavale, te kako hiperheuristike i metaheuristike kontinuirano daju odlične rezultate u ovakvoj vrsti problema. Broj inačica je velik, kao i potencijal za uvođenje dodatnih ograničenja kako bi se reflektirali stvarni problemi, pa se može očekivati nastavak brzog rasta ovog područja.

## 4. Sažetak

U ovom radu proučavali smo Vehicle Routing Problem, a glavni fokus su nam bile inačice Dynamic Vehicle Routing problema. VRP je problem u kojem moramo rasporediti flotu vozila po različitim rutama na kojima se nalaze odredišta, na način da minimiziramo vrijeme ili ukupnu prijeđenu udaljenost. Važnost ovog problema je rasla napretkom računarskih mogućnosti i razvojem tehnologije, ogromnom rastu logističkih potreba, i sve širem spektru stvarnih problema na koje se mogu primijeniti metode rješavanja. Formalno smo definirali VRP i DVRP te smo dali pregled DVRP-a s vremenskim prozorima i DVRP-a s dostavama i preuzimanjima.

Analizirali smo neke metode koje se koriste pri rješavanju takvih problema, i njihovu učinkovitost na ovim problemima u usporedbi s drugim metodama. Dali smo kratki osvrt na prošlost i začetke VRP-a, kao i mišljenje o budućnosti i rastu ove grane računarske znanosti.

Metode koje smo pokrili su Branch and Bound pri rješavanju VRP-a, Algoritam Kolonije Mrava pri rješavanju DPDVRP-a, Multiagentske sustave pri rješavanju DVRP-a te genetičko programiranje pri rješavanju DVRPTW-a.

## 5. Literatura

- [1] G. Danzig i J. Ramser, »The truck dispatching problem,« 1959, *Management Science*, pp. 6:80-91.
- [2] B. Golden, S. Raghavan i E. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, 2008.
- [3] UPS, »UPS,« United Parcel Service, [Mrežno]. Available: <https://www.ups.com/hr/en/about/facts/worldwide.page>. [Pokušaj pristupa 10 5 2020].
- [4] L. Kraljević, *Uporaba simboličke regresije za rješavanja problema usmjeravanja vozila*, Zagreb: FER, 2019.
- [5] G. Clarke i J. Wright, »Scheduling of Vehicles from a Central Depot to a Number of Delivery Points,« *Operational Research*, 1964, pp. 568-581.
- [6] M. L. Fisher, »Optimal solution of vehicle routing problems using minimum k-trees.,« u *Operations Research*, 1994, pp. 42:626-642.
- [7] D. Barbuča, »VNS-Based Multi-agent Approach,« u *Computational Collective Intelligence*, Hendaye, Springer, 2019, pp. 556-562.
- [8] J. Euchi, A. Yassine i C. Habib, »The dynamic vehicle routing problem: Solution with Hybrid metaheuristic approach,« u *Swarm and Evolutionary Computation*, 2014.
- [9] J. Jacobsen-Grocott, Y. Mei, G. Chen i M. Zhang, »Evolving Heuristics for Dynamic Vehicle Routing,« 2017.
- [10] P. Toth i D. Vigo, *The Vehicle Routing Problem*, Philadelphia: SIAM, 2002.
- [11] T. Caric i H. Gold, *Vehicle Routing Problem*, In-Teh, 2008.
- [12] P. Toth i D. Vigo, *Vehicle Routing Problems, Methods, and Applications Second Edition*, SIAM, 2014.