

# Optimizacija kolonijom pčela

Seminarski rad u okviru kursa  
Metodologija stručnog i naučnog rada  
Matematički fakultet

Maja Vukolić, Marija Marković, Lea Petković, Marina Pilipović  
vukolic.maja97@gmail.com, markovic.n.maja@gmail.com,  
lea.bela.97@gmail.com, beka.pilipovic@gmail.com

30. mart 2020

## Sažetak

U ovom tekstu je ukratko prikazana osnovna forma seminarskog rada.

## Sadržaj

<b>1 Uvod</b>	<b>2</b>
1.1 Inteligencija roja . . . . .	2
<b>2 Pčele i strategija</b>	<b>2</b>
2.1 Pčelinji ples . . . . .	3
<b>3 Istorija algoritma</b>	<b>4</b>
<b>4 Pčelinji algoritam</b>	<b>4</b>
4.1 BCO algoritam . . . . .	5
4.1.1 Lojalnost pčele . . . . .	6
4.1.2 Regrutacija . . . . .	7
<b>5 Primene algoritma</b>	<b>7</b>
5.1 Problem trgovackog putnika . . . . .	8
5.2 Problem deljenja vozila . . . . .	8
<b>6 Zaključak</b>	<b>9</b>
<b>Literatura</b>	<b>10</b>

# 1 Uvod

Priroda je poslužila kao nadahnuće za mnoštvo naučnih istraživanja. Pored onoga što je već otkriveno, dosta je ostalo neistraženo. Danas, dosta novih algoritama je inspirisano prirodnim procesima, tj. karakteristike koje su se dobro pokazale u biologiji, programeri teže da primene u rešavanju mnogih problema.

## 1.1 Inteligencija roja

Specijalnu klasu ovakvih algoritama predstavljaju oni koji počivaju na **inteligenciji roja** (engl. **Swarm intelligence, SI**). Jedan od vodećih eksperata, Eric Bonabeau, definisao je inteligenciju roja kao „*svaki pokušaj dizajniranja algoritama ili distribuiranih uređaja za rešavanje problema, inspirisanih kolektivnim ponašanjem društvenih kolonija insekata i drugih društava životinja.*“<sup>[4]</sup> Pojam *roj* odnosi se na bilo kakvu skupinu interaktivnih agenata/pojedinaca. Primer roja su pčele. Metafora se lako može proširiti na druge (slične) sisteme: kolonija mrava može se smatrati rojem čiji su agensi mravi; a slično važi za jato ptica. Neki algoritmi koji se zasnivaju na inteligenciji roja, odnosno - ponašanju pojedinačnih jedinki unutar određene grupe, su:

- Optimizacija rojevima čestica (engl. *Particle Swarm Optimization, PSO*),
- Optimizacija mravlјim kolonijama (engl. *Ant Colony Optimization, ACO*),
- Optimizacija kolonijom pčela (engl. *Bee Colony Optimization, BCO*).

Nabrojani algoritmi rade nad skupom jedinki, rojem. Elementi ovog skupa nazivaju se čestice. Čestice se na unapred definisan način kreću po prostoru pretrage. Njihovo kretanje se usmerava imajući u vidu njihovu trenutnu poziciju, njihovu trenutno najbolju poziciju, kao i najbolju poziciju čitavog roja. Na ovaj način stvara se model koji je prilagođen računaru, kao i potrebama određenih zadataka. Ovaj rad će dati kratak teorijski uvod o kretanju pčela u prirodi, algoritmu koji je iz toga proistekao, kao i primere njegove realne primene.

# 2 Pčele i strategija

Pčele se prema ulozi koje imaju dele na: matice, trutove i radilice. Kako autorima rada u algoritmu stvaranje pčela nije od značaja, akcenat se stavlja na treću vrstu – radilice. Postoje dva tipa radilica: izviđači (engl. scout bees) i sakupljači (engl. forager bees).

Izviđači konstantno pretražuju okruženje tražeći nove izvore nektara. Kreću se nasumično u okolini košnice, procenjujući vrednost resursa na koje najdu. Kada pronađu izvor hrane, vraćaju se u košnicu, odlaze u deo koji se naziva „podijum za igru“ i izvode pčelinji ples, prikazan na slici 2. Pomoću ovog plesa objašnjavaju drugim pčelama put do novog izvora hrane. Nakon toga, one se vraćaju na izvor, ali sada u pratnji drugih pčela. Broj pčela koji će im se pridružiti zavisi od kvaliteta poruke koju prenose. Zahvaljujući ovom procesu, kolonija je u stanju da se fokusira samo na najvrednije izvore hrane.

Prirodno se nameće da je broj sakupljača mnogo veći od broja izviđača. Sakupljači na osnovu izvedenog plesa odlučuju da li će poći do izvora hrane



Slika 1: Pčela „sakupljač” prilikom sakupljanja nektara.

ili ne. Zainteresovane pčele kreću na put. Takođe, pčela izviđač je, osim putanje, prenela i miris cveta na kom je bila, čime sakupljači znaju da se nalaze na pravom putu.

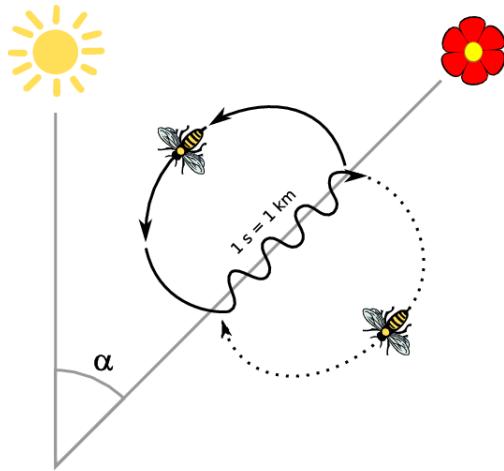
## 2.1 Pčelinji ples

Nemački zoolog Karl Von Frisch je 1973. otkrio da pčele među sobom komuniciraju plesom (engl. waggle dance). Na tzv. „plesnom podijumu” u košnici izviđači prenose poruku u dve faze:

- migoljenje
- kružni povratak, prvo u jednu, a zatim u drugu stranu.

Na ovaj način se kodiraju: smer, udaljenost i kvalitet izvora hrane, a pčele koje nikad nisu napuštale košnicu tačno znaju gde treba da idu.

Najsloženiji podatak je *smer*. Pčele koriste položaj Sunca, odnosno, ugao pod kojim izvodi ples označava ugao u odnosu na pravac Sunca pod kojim pčele treba da krenu ka izvoru - cvetu. Ovo je prikazano na slici 2:



Slika 2: Ples pčele koji vrši u košnici, kako bi ukazala drugim pčelama na smer i udaljenost izvora hrane.

*Udaljenost* izvora se određuje na osnovu krive koju pčela izviđač opisuje prilikom plesa. Ako kruži, izvor se nalazi u radijusu od 50m. Ako je kriva u obliku znaka beskonačno, onda je izvor dalji od 150m; dok je u preostalim slučajevima izvor na udaljenosti između 50m i 150m[7].

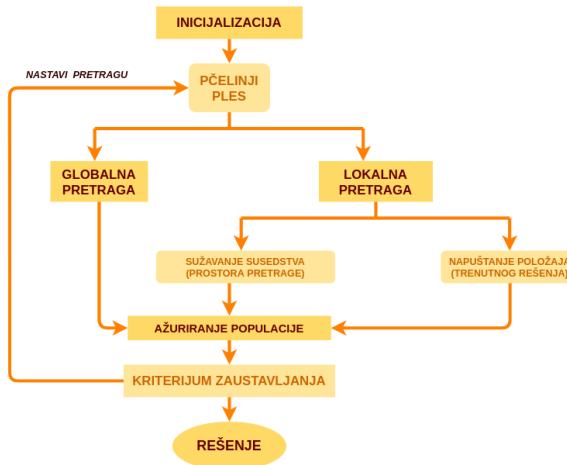
Brzina i vreme trajanja plesa je proporcionalno oceni resursa pčele izviđača. Drugi izviđači takođe mogu opisivati isti izvor nektara. *Kvalitet* je veći što je veći broj pčela koje oglašavaju za jedan isti izvor, a samim tim sa njima će poći veći broj sakupljača[6]. Dokle god se smatraju vrednim, izvori hrane će "reklamirati" pčele sakupljači kroz ples u košnici. Novi regruti mogu takođe učestvovati u plesu i na taj način pozivati što više novih pčela da se priključe sakupljanju.

### 3 Istorija algoritma

U periodu 1999-2003, srpski akademici - Panta Lučić i Dušan Teodorović, predstavili su osnovne koncepte optimizacije kolonijom pčela (engl. bee colony optimization, BCO) kroz sistem pčela (engl. bee system) dok su vršili istraživanja u *Virginia Tech*[2]. Optimizacija kolonijom pčela uvedena je 2005. godine kao metaheuristika za rešavanje problema uparivanja putovanja. Iste godine su Drias i Yahi modifikovali algoritam optimizacije rojem pčela (engl. bee swarm optimization) za problem MAX-W-SAT (engl. maximum weighted satisfiability problem). Takođe, D. T. Pham uvodi pčelinji algoritam (engl. bees algorithm) koji kombinuje pretraživanje susedstva i nasumično pretraživanje.[5] Dok računarski inženjer D. Karaboga uvodi algoritam veštačke kolonije pčela (engl. artificial bee colony, ABC) prvenstveno namenjen numeričkoj optimizaciji.

### 4 Pčelinji algoritam

Pčelinji algoritam je metod pretrage koji oponaša kolonije pčela prilikom potrage za hranom[1]. Profesor D. T. Pham, zajedno sa svojim saradnicima, je napravio i opisao algoritam 2005. godine.



Slika 3: Dijagram sa koracima opštег pčelinjeg algoritma.

U svojoj osnovnoj verziji izvršava pretraživanje susedstva u kombinaciji sa globalnim pretraživanjem, i može se upotrebiti i za kombinatornu i za kontinuiranu optimizaciju. Koristi populaciju ageansa, odnosno veštačkih pčela, kako bi uzorkovao prostor rešenja[6]. Pčele izviđači sumično traže regije visoke funkcije prilagođenosti (globalna pretraga). Najuspješniji izviđači regrutuju promenljiv broj pčela koje tragaju za hranom kako bi pretraživali u okolini najboljih rešenja (lokalna pretraga). Ciklusi globalne i lokalne pretrage ponavljaju se dok se ne otkrije prihvativljivo rešenje ili ne prođe određeni broj iteracija. Efikasnost ovog algoritma se pokazala u mnogim istraživanjima. Među najpoznatijim pčelinjim algoritmima nalazi se **BCO** (engl. Bee Colony Optimization). Osnovni koraci dati su dijagramom na slici 3.

## 4.1 BCO algoritam

Optimizacija rojem pčela je zasnovana na analogiji između prirodnog ponašanja pčela u potrazi za nektarom i ponašanja optimizacionih algoritama u potrazi za optimumom[6]. Ova metaheuristika je zasnovana na pretraživanju populacije.

Postoje 2 osnovna pristupa BCO algoritma:

- **Konstruktivni BCO** - zasnovan na koracima izgradnje, gde pčele grade rešenje korak po korak
- **BCOi** - zasnovan na poboljšanju celovitih rešenja kako bi se dobila što bolja konačna rešenja [6]

Potrebito je napraviti koloniju veštačkih pčela koje bi tražile dobra rešenja problema u prostoru pretrage. Kako bi kvalitet dobijenih rešenja bio bolji, pčele sarađuju i razmenjuju informacije. Na ovaj način pčele su skoncentrisane na rešenja koja više obećavaju, te postepeno generišu i poboljšavaju pronađena rešenja (sve dok nije postignut zadati kriterijum zaustavljanja). Opšti algoritam dat je u nastavku:

```

1 // B - broj pčela uključenih u pretragu
2 // N - broj letova od kosnice do cveta i nazad
3 // u jednoj BCO iteraciji
4 // Inicijalizacija: Dodeljuju se vrednosti promenljivama B i N,
5 // određuje se kriterijum zaustavljanja
6 do:
7   (1) Dodeli prazno rešenje svakoj pčeli
8   (2) for (k = 0; k < N; k++) :
9     // let unapred
10    (a) for (i = 0; i < B; i++) :
11      for (j = 0; j < f(N); j++) :
12        (i) Proceni sve mogućnosti;
13        (ii) Izaberi naredni korak koristeci pravila
14          ruleta;
15    // let unazad
16    (b) for (i = 0; i < B; i++) :
17      Proceni kvalitet (parcijalnog/celokupnog)
18      rešenja za trenutnu pčelu;
19    (c) for (i = 0; i < B; i++) :
20      Odrediti lojalnost na osnovu pravila ruleta
21      trenutne pčele i;
22    (d) for (i = 0; i < B; i++) :
23      Ako pčeli i kojoj nije dodeljeno rešenje,
24      pravilom ruleta izaberi rešenje;
25    (3) Uporedi sva rešenja i izaberi najbolje
26      Azuriraj x_best i f(x_best) dok kriterijum zaustavljanja
27      ne bude zadovoljen
28
29 while (nije zadovoljen kriterijum)
```

Listing 1: Pseudokod BCO algoritma [2]

Populacija se sastoji od  $B$  veštačkih pčela. Svaka od njih zadužena je za proizvodnju jednog rešenja. Različite putanje koje pčele prelaze predstavljaju različita rešenja. Algoritam ima dve faze - **let unapred** i **let unazad**. Navedene faze dešavaju se naizmenično[2].

Kod leta unapred, pčele pretražuju prostor pretrage. Postoji unapred definisani broj koraka koji izgrađuju i/ili poboljšavaju rešenje, koje onda postaje bolje rešenje. Zatim se započinje se s letom unazad.

Kod leta unazad, svaka pčela deli informaciju o kvalitetu svog parcijalnog/celokupnog rešenja. Do optimalnog rešenja dolazi se izračunavanjem funkcije cilja za svako parcijalno/celokupno rešenje, što je analogno plesu pčela u prirodi. Nakon što su rešenja evaluirana, svaka pčela odlučuje (s određenom verovatnoćom) da li će ostati pri svom rešenju ili ne. Pčele koje su pronašle bolja rešenja imaju veće šanse da ih zadrže i reklamiraju. Za razliku od pčela u prirodi, u algoritmu - druge pčele će razmatrati rešenja kao potencijalno dobra, onih pčela koje su ostale verne svojim rešenjima. Kad god neka pčela odbaci svoje rešenje, mora da izabere jedno od reklamiranih rešenja. Pčela donosi odluku sa određenom verovatnoćom - tako da bolja rešenja imaju veće šanse da budu izabrana za naredne korake. Na ovaj način, pri svakom letu unazad, sve pčele se dele u dve grupe. Prvu grupu čine pčele regruteri  $R$ , a drugu neopredeljene pčele  $B - R$ . Vrednosti broja regrutera i neopredeljenih pčela se menjaju od iteracije do iteracije (jednog leta unazad do drugog leta unazad).

U narednom letu unapred, prema konstruktivnom BCO-u, svaka pčela dodaje novu komponentu na pre generisano delimično rešenje. Kod BCO-i a pčele menjaju delove celokupnog rešenja kako bi se postigla rešenja što većeg kvaliteta.

Faze leta unapred i leta unazad se smenjuju  $N$  puta, dok svaka pčela ne izgradi sopstveno rešenje ili ne uradi izmenu rešenja.  $N$  je predstavlja parametar koji služi za definisanje učestalosti komunikacije između pčela. Kada se izvrši  $N$  iteracija/koraka, određuje se najbolje rešenje od svih mogućih  $B$  rešenja. To rešenje se posle koristi pri ažuriranju globalno najboljeg rešenja. Ovime je iteracija algoritma gotova i brišu se svih  $B$  rešenja, te započinje nova iteracija[6].

Sve dok kriterijum zaustavljanja nije ispunjen, vrše se iteracije. Mogući kriterijumi zaustavljanja mogu biti maksimalan/zadati broj iteracija ili maksimalan broj iteracija bez poboljšanja funkcije cilja. Najbolje pronađeno rešenje se daje kao konačno. To rešenje i globalno rešenje.

#### 4.1.1 Lojalnost pčele

Nakon završetka leta unapred, svaka pčela odlučuje da li će ostati verna prethodno pronađenom rešenju ili ne. Ova odluka zavisi od trenutnog kvaliteta rešenja pčele u odnosu na rešenja ostalih pčela. Verovatnoća da je pčela  $i$  lojalna svom prethodnom parcijalnom/celokupnom rešenju, na početku novog leta unapred, opisana je formulom:

$$p_i^{u+1} = e^{-\frac{O_{max} - O_i}{u}} \quad (*)$$

$O_i$  - normalizovana vrednost funkcije parcijalnog/celokupnog rešenja

$O_{max}$  - maksimalna vrednost svih normalizovanih

$u$  - brojač letova unapred (1, 2, ..., N)

U zavisnosti da li nam treba minimalna ili maksimalna funkcija, normalizacija se izvodi na sledeći način:

$$O_i = \frac{C_{max} - C_i}{C_{max} - C_{min}} \text{ za minimizaciju}$$

$O_i = \frac{C_i - C_{min}}{C_{max} - C_{min}}$  za maksimizaciju  
gde  $C_i$  označava ciljnu vrednost funkcije i-tog pčelinjeg rešenja.

Lojalnost pčele zavisi od formule (\*), kao i generatora slučajnih brojeva. Ako je slučajni broj manji od izračunate vrednosti, pčela ostaje pri svom rešenju. U suprotnom, neopredeljena je[2].

#### 4.1.2 Regrutacija

Za svaku neopredeljenu pčelu, treba odrediti kog regruta će slediti. Pritom, uzima u obzir kvalitet reklamiranih rešenja. Verovatnoća da će i-to parcijalno/celokupno rešenje izabrati neopredeljene pčele je:

$$p_i = \frac{Q_i}{\sum_{k=1}^R (Q_k)}, i = 1, 2, \dots R$$

Gde je  $Q_k$  normalizovana vrednost funkcije cilja k-tog reklamiranog rešenja.  $R$  je broj regrutera. Svaka neopredeljena pčela se pridružuje jednom od regrutera preko ruleta[6].

## 5 Primene algoritma

Ovaj algoritam našao je mnoge primene u inžinerstvu: optimizacija klaster sistema, manufaktura, upravljanje, bioinžinjeri, mnogi problemi optimizacije itd. U narednoj tabeli su nabrojani problemi koje su naučnici rešavali pomoću optimizacije kolonijom pčela.

Godina	Autori	Primena
2001, 2002, 2003	Lučić i Teodorović	Problem trgovačkog putnika
2003	Lučić i Teodorović	Problem sa usmeravanjem vozila u slučaju nesigurne potražnje
2005	Teodorović i Dell' Orco	Problem deljenja vozila
2006	Teodorović, Lučić, Marković, Dell' Orco	Problem trgovačkog putnika i problemi sa usmeravanjem mreža
2007	Marković, Teodorović i Aćimović-Raspopović	Usmeravanje i dodela talasne dužine svim optičkim mrežama
2007	Teodorović i Šelmić	Problem p-medijane
2008	Teodorović	Poređenje performansi BCO algoritma sa performansama drugih algoritama zasnovanih na inteligenciji roja
2009	Davidović, Teodorović i Šelmić	Statičko zakazivanje nezavisnih zadataka kod homogenih višeprocesorskih sistema

Tabela 1: Primene BCO algoritma od 2001. do 2009. godine

## 5.1 Problem trgovačkog putnika

U postavci ovog problema dato je  $n$  gradova i potrebno ih je sve obići, tako da ni u koji grad ne dođemo dva puta, a da dužina ukupnog puta bude što manja. Lučić i Teodorović su ovo pokušali da reše koristeći metod optimizacije kolonijom pčela. Oni su testirali optimizaciju na velikom broju dobro poznatih testova za testiranje performansi (*engl. test benches*) kao što su *Eil51*, *Berlin52*, *St70* itd[8]. Algoritam je izvršavan na IBM-ovom računaru sa PIII procesorom (533MHz). Rezultati testiranja su prikazani u sledećoj tabeli:

Test	Optimalna vrednost(O)	Najbolja postignuta vrednost pomoću BCO algoritma(B)	(B-O)/O (%)	CPU (s)
Eil51	428.87	428.87	0	29
Berlin52	7544.366	7544.366	0	0
St70	677.11	677.11	0	7
Pr76	108159	108159	0	2
Kroa100	21285.4	21285.4	0	10
Eil101	640.21	640.21	0	61
Tsp225	3859	3899.9	1.06%	11651
A280	2586.77	2608.33	0.83%	6270
Pcb442	50783.55	51366.04	1.15%	4384
Prl1002	259066.6	267340.7	3.19%	28101

Tabela 2: Vrednosti dobijene primenom algoritma [8]

Iz tabele može se videti da su najbolje vrednosti, dobijene BCO algoritmom, približne optimalnim vrednostima, kao i da je vreme rada procesora malo. Stoga, može se zaključiti da je BCO dao kvalitetne rezultate.

## 5.2 Problem deljenja vozila

Današnji saobraćaj je veoma zagušen zbog neprestane gradnje novih puteva i novih račvanja. To doprinosi većem broju kašnjenja, dužem putovanju, velikom broju zaustavljanja i povećanju zagadenosti vazduha u gradovima. Zato je bitno pametno iskoristiti postojeće puteve.

Deljenje vožnje (*engl. ride-sharing*) je jedna od korišćenijih TDM (*engl. Travel Demand Management*) tehnika. U okviru ovog koncepta dve ili više osoba dele isto vozilo kako bi stigli do svojih destinacija. Prevoznik mora da poseduje sledeće informacije kako bi dobro isplanirao put:

- kapacitet vozila (broj ljudi u vozilu)
- mesto odakle se kreće za svaki dan u nedelji
- krajnja destinacija za svaki dan u nedelji
- vreme željenog polaska i/ili dolaska za svaki dan u nedelji
- dani u nedelji, kada je osoba spremna da učestvuje u deljenju vozila

Teodorović i Dell' Orco želeli su da osmisle rutu i raspored vozila i putnika za celu nedelju, tako da se ukupna dužina putovanja za sve učesnike bude minimalna. Razvili su model koji se oslanja na BCO. Ovaj model testiran je na uzorku vozila koja su kretali iz malog grada u Italiji -

Trane, do mesta Bari. Učestvovalo je 97 putnika. Kapacitet svakog vozila bio je 4 putnika.

U ovom slučaju, algoritam je odredio da 96 (= 24\*4) od 97 putnika daju najbolju putanju. Ne postoji standardni metod u literaturi za određivanje najbolje metode za optimalno deljenje vožnje. Ovo je bio samo pokušaj da se razvije metodologija koja bi mogla da reši ovaj problem[3].

## 6 Zaključak

U ovom radu predstavljena je optimizacija kolonijom pčela, metaheuristička metoda inspirisana ponašanjem pčela i najmlađa tehnika koja koristi inteligenciju roja. Bavili smo se komunikacijom između pčela i međusobnoj podeli posla u košnici. Posebna pažnja posvećena je samom algoritmu, koji se uz određene modifikacije može prilagoditi za različite probleme optimizacije u upravljanju, inženjeringu ili kontroli. Takođe, ukratko je opisana primena na problem trgovackog putnika i problem deljenja vozila, sa kojim se svakodnevno susrećemo. BCO ima mogućnost da putem razmene informacija i regrutovanja intenzivnije vrši pretragu u onim regionima prostora rešenja koji više obećavaju. Ipak, nije široko korišćen za rešavanje stvarnih problema.

Zanimljivi aspekti budućeg istraživanja mogu biti homogenost pčela, različiti mehanizmi razmene informacija i različiti mehanizmi saradnje. Takođe, jedna od interesantnih praktičnih primena mogla bi da bude i primena pčelinjeg algoritma na naše elektronske uređaje. Naime, svima nam je dobro poznata situacija u kojoj se mobilni telefon, laptop ili neki drugi uređaj zaređao ili prestao sa radom usled velikog korišćenja. O rešavanju ovog problema, inspirisan prirodom (tačnije pčelama), govori Ali Abuassal u svom doktorskom radu. Rad možete pronaći na:  
[https://pure.york.ac.uk/portal/en/publications/artificial-bee-colonyinspired-runtime-task-management-for-manycore-systems\(83f1aad7-3e26-4609-b0df-03af36d0d330\).html](https://pure.york.ac.uk/portal/en/publications/artificial-bee-colonyinspired-runtime-task-management-for-manycore-systems(83f1aad7-3e26-4609-b0df-03af36d0d330).html).

## Literatura

- [1] Marco Castellani. The Bees Algorithm WebPage. on-line at: <http://beesalgorithbsite.altervista.org>.
- [2] Tatjana Davidović, Dušan Teodorović, and Milica Šelmić. Bee colony optimization part i: The algorithm overview. *Yugoslav Journal of Operations Research*, 25 (2015), Number 1, 33–56, 2014.
- [3] Mauro Dell' Orco Dušan Teodorović. Migrating traffic congestion: Solving the ride-matching problem by bee colony optimisation. *Taylor and Francis*, 18 January 2008, 2008.
- [4] G. Theraulaz E. Bonabeau, M. Dorigo. Swarm intelligence: From natural to artificial systems. *Oxford University Press*, 1999, 1999.
- [5] Dino Klemen. *Diplomski rad: Raspoznavanje uzoraka primjenom umjetne kolonije pčela*. Sveučilište u Zagrebu, Fakultet Elektrotehnike i računarstva, 2011.
- [6] Stejpan Lojen. *Diplomski rad: Primjena heurističkih metoda u mještajstvu*. Sveučilište u Zagrebu, Fakultet Strojarstva i brodogradnje, 2016.
- [7] David R. Tarpy. The honey bee dance language. *North Carolina Cooperative extension service*, 2016.
- [8] Dušan Teodorović. *Bee Colony Optimization (BCO)*. Univerzitet u Beogradu, Saobraćajni fakultet.