

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Inteligencija roja čestica

Viran Ribić

Voditelj: Marin Golub

Zagreb, svibanj 2015.

Sadržaj

1.	Uvod.....	1
2.	Seminarski rad	2
2.1.	Uvod u rojeve estica.....	2
2.1.1	Umjetna inteligencija i problem automatskog planiranja.....	2
2.1.2	Evolucijski algoritmi	3
2.2.	Rojevi estica	4
2.2.1	Potreba za novim algoritmima	4
2.2.2	Inspiracija	4
2.2.3	Definicija.....	4
2.2.4	Prednosti i nedostaci	5
2.2.5	Op i algoritam SI.....	6
2.3.	Pregled algoritma	7
2.3.1	Algoritam kolonije mrava	7
2.3.2	Algoritam p eli.....	8
2.3.3	Algoritam prepoznavanja imunolozkog sustava	12
2.3.4	Optimizacija rojem estica.....	14
2.4.	Primjene u praksi.....	17
3.	Zaklju ak	19
4.	Literatura	200
5.	Sa0etak	211

1. Uvod

Od ranih dana informacijske ere do danas brzina i količina podataka koja nastaje postigla je enormne razmjere. Da bi se ti podaci uspjezno koristili, potrebni su sustavi koji mogu brzo i efikasno obavljati i upravljati tim informacijama. Iako su procesori danas iznimno napredni, kako bi se postigla najveća iskoristivost sustava, algoritmi za obradu tih podataka moraju biti vježto i precizno osmisljeni. Poznati postupci pretraživanja ovih velikih nakupina podataka nisu vise dovoljno dobri i inovatori su suočeni s novim izazovom. Jedno od rješenja dolazi iz svijeta biologije gdje se razmatra zajedničko ponazanje životinjskih kolonija kao inspiracija za modeliranje suvremenih algoritama. Znajući doprinos na ovom području dali su Marco Dorigo, Russell C. Eberhart, James Kennedy te Gerardo Beni i Jing Wang koji su uveli sam pojam rođestva.

Inteligencija rođestva (eng. Swarm Intelligence) naziv je za ponazanje većeg broja subjekata koje na razini grupe međusobnom interakcijom postižu rezultate koji ne bi bili ostvarivi pojedinačnim jedinkama te grupi. Zajedničke karakteristike svih takvih sustava su izmjena prikupljenih informacija koja na razini grupe regulira kako će se pojedinačni agenti ponazati.

Ovaj rad predstavlja osnovne koncepte umjetne inteligencije i problem automatskog planiranja i zakazivanje kao i uvid u evolucijske algoritme s ciljem pojasnjavanja ključnih pojmova i boljeg razumijevanja koncepta rođestva. Iznijeti će se motivacija za koristenjem rođestva, odnosno predstaviti probleme za koje su algoritmi rođestva pogodni. Slijedi pregled ove koritenih algoritama u kojem će se razmatrati ideja iza pojedinog pristupa i opisati način rada tog algoritma. Na kraju se daje pogled praktične primjene rođestva na konkretnim slučajevima sa ciljem boljeg razumijevanja izloženih koncepta.

Uz svaki stručni pojam bit će naveden i njegov engleski naziv, radi lakzeg pronalaženja dodatnih informacija u stranoj literaturi, dok će, gdje je moguće, tekst biti proziren smjernicama na povezanu literaturu za one koji se vise zainteresiraju za ovu ili neku od podtema.

2. Seminarski rad

2.1. Uvod u rojeve čestice

2.1.1. Umjetna inteligencija i problem automatiziranog planiranja

Umjetna inteligencija (eng. artificial intelligence) grana je razlike znanosti koja se bavi proučavanjem strojeva i programa koje karakterizira intelligentno ponazanje. Takvo ponazanje izvode intelligentni agenti (eng. intelligent agents) koji percipiraju okruženje u kojem se nalaze, uzimaju i u svojim izražajima i do tada prikupljene podatke s ciljem maksimaliziranja mogućnosti za uspjehom. Intelligentni agenti su različiti modeli stvarnih objekata pa se u skladu s tim svojstvom nazivaju i apstraktni intelligentni agenti (eng. abstract intelligent agents), a ako se naglasak stavlja na njihovu neovisnost, koristi se i izraz autonomni intelligentni agent (eng. autonomous intelligent agents). Kako bi se opisalo intelligentno ponazanje, problem razlikujemo na iduće podskupine: rasu, ivanje, znanje, automatizirano planiranje i zakazivanje, učenje, obrada prirodnog jezika, percepcija te mogućnost micanja i manipulacije objektima. Sustav koji uspješno izvodi svaku od navedenih vještina te, što je još važnije, povezuje i nadovezuje rezultate tih operacija zadovoljava osnovne kriterije intelligentnog ponazanja. U fokus smo dalje staviti područje automatiziranog planiranja i zakazivanja (eng. automated planning and scheduling), dalje u ovom radu nazvano samo planiranje. Planiranje je aspekt umjetne inteligencije u kojem intelligentni agent analizira sve opcije koje vode do cilja u trenutnom okruženju, te biraju najbolji od ponuđenih putova. Rojevi čestici koriste vize agenata pri toj analizi (eng. multi-agent planning) i odluke se donose na razini zajednice što sam proces planiranja i nezavise kompliksnijim od klasičnog planiranja izoliranih agenata u drugim modelima. Za sve zainteresirane, knjiga *Artificial Intelligence: A Modern Approach* je fundamentalna knjiga autora Stuarta Russella i Petera Norviga koja vrlo detaljno obrađuje ovu temu.

2.1.2. Evolucijski algoritmi

Evolucijski algoritmi (eng. evolutionary algorithm) podgrana je umjetne inteligencije i podvrsta evolucijskog računanja. Kao i rođeni evolucijski algoritmi inspirirani su prirodnim procesima i kao takvi dijele mnoge sličnosti. Oba pristupa rješavanju problema sastoje se od idućih koraka:

- i) nezavisno pretraživanje prostora problema
- ii) evaluacija prikupljenih podataka i određivanje najboljeg rezultata
- iii) pomak svih agenata u domeni pretraživanja prema najboljem rješenju

U evolucijskim algoritmima agenti su modelirani genima nad kojima se izvrzavaju operacije reproducije+, smutacije+, srekombinacije+ i sprirodne selekcije+. Takvi se algoritmi opisuju kao metaheuristički ili stohastički. Heurističke su tehnike one koje koriste aproksimacije rješenja kada klasične metode ne mogu dati nikakvo rješenje ili pak za ubrzavanje procesa koji bi bili prespori za izrađivanje. Metaheuristika proizvodi sam koncept heuristike kontroliranjem osnovnih heurističkih algoritama uz pomoć strojnog učenja. Takve su nam metode, za sada, jedina opcija pri izrađivanju NP-potpunih problema. Stohastički model definiramo kao onaj koji prepoznaće slučajnu prirodu ulaznih komponenti odnosno pokus iji ishod nije unaprijed određen.

Evolucijski su algoritmi predstavljeni radi lakzeg snalaženja i tatalja me u kategorizacijom algoritama umjetne inteligencije i neće se detaljnije razmatrati u okviru ovog rada, no razumijevanje jednog od ova dva koncepta olakzava usvajanje drugog. Za one koji se zainteresiraju za tu temu posebnu pažnju usmjerujem na rad: "Genetski algoritam" (Golub Marin, 2004.), u kojem je na jasan i razumljiv način taj reprezentativni princip.

2.2. Rojevi čestica

2.2.1. Potreba za novim algoritmima

ZnatiOelja i razumijevanje svijeta oko sebe primarna je ambicija ljudi od kad su prvi put poeli percipirati svijet koji ih okruuje. Ta je potraga obiljeđena mnogim borbama s preprekama i problemima koje je ljudski intelekt uvijek iznova uspjevao nadi i, time zire i granice poznatog i unaprijeju uj uivot zajednice u kojoj Oivi. Klju za uspjezno rješavanje problema bio je sistematičan pristup problemu, njegova analiza i razlambu, ali povrh svega dobro definiran postupak dolaska do cilja. Takav niz jednostavnih operacija naziva se algoritam koji karakterizira uvijek točan rezultat nad zadatkom koji izvrzava. I dok su neki problemi brzo izrađeni, od posebnog su nam interesa problemi koji se klasificiraju kao NP-C potpuni problemi (eng. nondeterministic polynomial time complete problems). Problem trgovina kog putnika, problem zaustavljanja (eng. halting problem), nauka o kriptografiji, samo su neka od područja u kojima se susrećemo s NP-C problemima. I dok veliki broj znanstvenika traga za odgovorom postoji li ekvivalencija između polinomialnih i NP problema ($P=NP?$), u praksi se često možemo zadovoljiti aproksimacijama rješenja ako bi to znalo da rezultate možemo dobiti u prihvatljivom vremenu. Koristeći se metaheuristikom i probabilističkim tehnikama, možemo smanjiti domenu pretraživanja rezultata, no tada smo suočeni s modeliranjem specifičnog zadatka što postaje pothvat sam za sebe.

2.2.2. Inspiracija

U prethodnom smo poglavljiju spomenuli s kojim se ograničenjima pri izradi unavanju susrećemo u različitim stohastičkim modelima rješavanja, mora odrediti poseban pristup određenoj skupini problema.

Između 70-tih i 80-tih godina Gerardo Beni prvi proučava ponazanja rojevačestica razmatrajući njihovu primjenu u robotici gdje velik broj jednostavnih entiteta surađuju pri realiziranju zajedničkog cilja. U prirodi takvo ponazanje vidimo u mravljinim kolonijama prilikom potrage za hranom, na primjeru na koji jata riba izbjegavaju predatore ili obrani organizma od toksičnih supstanci protiv kojih djeluje imunološki sustav. Takve su pojave poznate i proučavane već dugo vremena, a njihova efikasnost verificirana je samim ishodom njihovog promatranja.

2.2.3. Definicija

Inteligencija rojevačestica (eng. Swarm Intelligence, u daljem tekstu označava se sa SI) bavi se raznim sustavima inspiriranim skolektivnom inteligencijom. Takvo je ponazanje svojstveno velikim skupinama jedinki, inteligentnim agentima, koje djeluju na nekom području te međusobno komuniciraju. Ta komunikacija može biti direktna, kao što je slučaj kod problema koje rezultate svoje pretrage dijele s ostatkom kolonije, ili indirektna, kroz okoliz, što se može pojaviti u algoritmu.

slu0e u potrazi za hranom, kasnije obra eno detaljnije u optimizaciji kolonije mrava. Kako je ve naglazeno, inteligentni agenti reprezentiraju jednostavne organizme ije ponazanje nije tezko implementirati na ra unalu. No zbog relativno smalog%broja razli itih modela nasuprot svim problemima koji se mogu razmatrati, izazov pri primjeni ovih algoritama le0i u ispravnom odabiru prirodnog modela i prilago avanju nazeg problema zahtjevima algoritma. Otkad su se prvi put pojavili, inteligencija roj estica uspjezno je nazla primjenu u mnogo optimizacijskih i istra0iva kih problema, kao zto su kombinatorna optimizacija, optimizacije funkcija, pronala0enje optimalnih ruta, zakazivanje, struktorna optimizacija te analiza slika i podataka. Razmjere koje je ova tehnologija postigla vide se i u injenici da danas pronalazi i primjenu u strojnom u enju (eng. machine learning), bioinformatici i medicinskoj informatici (eng. bioinformatics and medical informatics), dinami kim sustavima (eng. dynamical systems and operations) pa ak i u financijama i poslovanju.

2.2.4. Prednosti i nedostaci

S ciljem postizanja boljeg razumijevanja SI algoritama joz u spomenuti generalne prednosti i nedostatke, odnosno ograni enja koja su postavljena. Mogu nost skaliranja sustava je prva ve a prednost, pri emu je osiguran mehanizam kontroliranja svih agenata neovisno o tome koriste li se male grupe od dvadesetak ili ve e grupe od vize tisu a subjekata. Adaptivnost je idu a klju na zna ajka gdje se zbog samoorganiziraju ih i autokonfiguriraju ih svojstava mogu dinami ki prilago avati promjenjivoj okolini, sve za vrijeme izvo enja. Kolektivna robusnost cijelog sustava omogu uje vrlo visoku sposobnost tolerancije grezke To proizlazi iz samog modela gdje individua sama za sebe nije klju na za daljnje izvo enje. Najlakze je povu i paralelu s ponazanjem u prirodi: jedna ptica koja se odvoji od jata ne e naruziti formaciju cijelog jata ukoliko je ono dovoljno veliko. Time se zaobilazi problem jedinstvene to ke neuspjeha (eng. single point of failure), dio sustava u kojem pojava grezke zna i potpuni kolaps itavog sustava. S druge strane, ono zto SI algoritmima zadaje najvize potezko a pri izvo enju su postavljanje parametara sustava, zada e koje iziskuju brzo rjezavanje i opasnost od stagnacije. Podezavanje parametara generalno je izazovno za sve stohasti ke optimizacijske metode pa se za naj ez e naztimavaju metodom pokuzaja i pogreznice ili se adaptiraju tijekom izvo enja. Kako algoritam nije predefiniran za konkretan problem, ve je samo na in rjezavanja, postoji svojevrsno prilago avanje algoritma na problem, ispitivanje domene rjezenja i prilago avanje u skladu s tim. Vjerojatnost ispravnog rjezenja raste s porastom izmjenjenih generacija agenata, ali i brojem samih agenata. Iz tih zahtjeva nu0no slijedi duga obrada podataka koja uklju uje evaluacije okru0enja svake jedinke unutar svake generacije, kroz vize uzastopnih generacija. No sve to vrijeme ra unanja mo0e biti uzalud utrozeno ako do e do prijevremene konvergencije algoritma k lokalnom optimumu, koji mo0e, ali i ne mora, biti tra0eni globalni optimum. Posebnim mehanizmima nasumi nih odabira mo0e se unijeti raznovrsnost u populaciju zto mo0e rezultirati uspjeznim oporavkom sustava iz takvih situacija, koje su naj ez e uzrok nedovoljno dobrim po etnim parametrima.

2.2.5. Općeniti algoritam SI

```
1    stvoriAgenteIRasporediNaDomeniProblema();
2    DOK_JE (  $\neg$  uvjetZaustavljanja() ) :
3        ocjeniTrenutnePozicije();
4        agregirajPodatke();
5        evaluirajPodatke();
6        prilagodiNoviRasporedNaDoemniProblema();
7    KRAJ;
```

Pseudokod 1: Općeniti algoritam SI

Pseudokod 1 prikazuje općeniti pristup pisanja algoritama pokretanih inteligencijom robočestica. Osnovni elementi su ocjena svog položaja u sustavu, agregacija informacija za svakog od lanova, obrada prikupljenih podataka te korak napredovanja u iduću iteraciju.

2.3. Pregled algoritma

U ovom poglavlju dan je opis najzastupljenijih algoritama i njihovih pseudokodova.

2.3.1. Algoritam kolonije mrava

Opis:

Algoritam kolonije mrava (eng. *Ant colony system*, u dalnjem tekstu oznaka ACS) oponaša kretanje mrava kroz okolinu u potrazi za resursom od primarnog interesa, u našem slučaju optimalnim rješenjem. Interakcija među agentima ostvarena je feromonima koje mravi otpuštaju u okoliš kad se žele sporazumjeti s lokalnim susjedima (eng. *Stigmergy*). Bitna karakteristika feromona je propadanje s protokom vremena što je ključno za gubljenje suboptimalnih putova, ali može i predstavljati veliki problem ukoliko je vrijeme raspada kao ulazni parametar algoritma premaleno. Feromoni se počinju odlagati tek nakon što je rješenje pronađeno, i tada počinje povratak ostatku kolonije. Svi su mravi jednako jako privučeni feromonima i kako broj putovanja od izvora prema rješenju raste, tako se utvrđuje i najkraći put kao posljedica sve većeg intenziteta odloženog feromona na njemu. ASO prvenstveno je dizajniran za: problem trgovačkog putnika(eng *TSP*), problem naprtnjače (eng. *knapsack problem*), problem bojanja grafa (eng. *graph coloring problems*) itd.

Implementacija:

```
1   Rnajbolji .vrijednost= generirajHeurističkoRješenje(problem);
2   Rnajbolji .evaluacija= evaluiraj(Rnajbolji);
3   feromonpočetni = 1/(domenaProblema.veličina* Rnajbolji .evaluacija);
4   feromon = inicjalizirajFeromon(feromonpočetni);
5   DOK_JE (  $\neg$  uvjetZaustavljanja() ) :
6       ZA i=1 DO m :
7           agent[i] .vrijednost =odrediRješenje(feromon,problem, $\beta$ ,q0);
8           agent[i] .evaluacija = evaluiraj(agent[i] .vrijednost);
9           AKO_JE agent[i] .evaluacija<= Rnajbolji .evaluacija ONDA
10          Rnajbolji .evaluacija= agent[i] .evaluacija;
11          Rnajbolji .vrijednost= agent[i] .vrijednost;
12          KRAJ;
13          lokalnoAžurirajISmanjiFeromon(feromon, agent[i], σ);
14      KRAJ;
15      globalnoAžurirajISmanjiFeromon(feromon, Rnajbolji, ρ);
16      KRAJ ;
17      VRATI Rnajbolji ;
```

Pseudokod 2: Algoritam kolonije mrava

ACS se oslanja na jednostavnije heurističke algoritame (npr. greedy algoritma) pri inicijalnoj procjeni vrijednosti. Zatim se generira odgovarajuća struktura koja bijektivno preslikava vrijednost feromona s vrijednošću problema. Slijedi odsječak optimizacije/pretrage koji se izvršava sve dok uvjet zaustavljanja nije ispunjen. Uvjet zaustavljanja može biti zabilježena konvergencija prema nekoj vrijednosti, vrijeme izvršavanja, broj iteracija bez poboljšanja, broj iteracija općenito itd. Rješenje određuje vjerojatnost pojedine komponente $R_{i,j}$ da bude izabrana koristeći (i).

$$R_{i,j} \leftarrow \frac{\tau_{i,j}^\alpha * \eta_{i,j}^\beta}{\sum_{k=1}^c \tau_{i,k}^\alpha * \eta_{i,k}^\beta} \quad (i)$$

Oznake u jednadžbi su iduće:

$R_{i,j}$: rješenje (vjerojatnost) odabira komponete
c	: broj iskoristivih komponenti
τ	: feromon pojedine komponente
η	: snaga kontribucije promatrane komponente cijelokupnoj ocjeni
α	: koeficijent povijesti
β	: heuristički koeficijent

Varijabla q_0 , označena kao ulazni argument funkcije odredi Rješenje, naziva se faktor pohlepe (eng. greediness factor) i koristi se pri odlučivanju hoće li se vjerojatnost izračunati pomoću izraza (i) ili greedy algoritma.

$$\tau_{i,j} \leftarrow (1 - \sigma) * \tau_{i,j} + \sigma * \tau_{i,j}^0 \quad (ii)$$

Funkcija lokalnoAžurirajISmanjiFeromon smanjuje feromon komponente $\tau_{i,j}$ prema izrazu (ii). Nerijetko je slučaj da se problemi koji se vežu za ACS modeliraju grafovima, pa jednu dionicu na kojoj mrav odlaže feromon možemo promatrati kao brid između dva vrha. σ je lokalni faktor feromona a $\tau_{i,j}^0$ označava početnu vrijednost feromona (rezultat izvođenja inicijalizirajFeromon).

$$\tau_{i,j} \leftarrow (1 - \rho) * \tau_{i,j} + \rho * \Delta \tau_{i,j} \quad (iii)$$

Funkcija globalnoAžurirajISmanjiFeromon slična je lokalnoj inačici. Komponentu σ sada zamjenjuje faktor raspadanja ρ , a $\Delta \tau_{i,j}$ je faktor maksimalizacije rješenja za najbolje pronađeno rješenje ukoliko je komponenta i,j označena kao globalno najbolja, u suprotnom $\Delta \tau_{i,j} = 0$. Ta je funkcija opisana izrazom (iii).

Slični algoritmi:

Kolonije s elitizmom (eng. Elitist ant system) – algoritam ACS proširuje se funkcionalnošću otpuštanja dodatnog feromona od najboljeg rješenja kako bi se spriječilo slučajno gubljenje istog.

Max-Min kolonija mrava (eng. Max-Min ant system, MMAS) – ova inačica uvodi maksimalnu i minimalnu količinu feromona, od najboljih prema lošijim putanjama. Algoritam započinje maksimalnim iznosom feromona među relacijama podataka i prepusta se stagnaciji dok mravi obavljaju svoj posao. Više informacija o ovom pristupu može se pronaći u izvornom radu „MAX-MIN Ant System“ (Thomas Stützle, 2000).

Rangirana kolonija mrava (eng. Rank-based ant system) – nešto općenitiji pristup, feromon se dodjeljuje za cijelu dužinu nekog puta, rangira se po udaljenosti izvora od odredišta te se svi bridovi takvog puta ocjenjuju jednakom količinom feromona.

Rekurzivna optimizacija kolonijom mrava(eng. Recursive Ant Colony Optimisation) – principom podijeli pa vladaj, domena problema dijeli se na poddomene koje se zasebno procjenjuju, a najbolji se kandidati šalju u idući korak rekurzije.

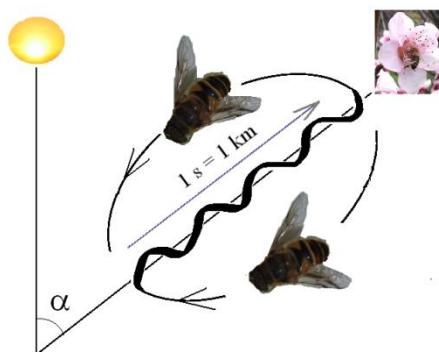
2.3.2. Algoritam pčela

Opis:

Algoritam pčela (eng. Bees algorithm, u dalnjem tekstu oznaka BS) oponaša ponašanje pčela pri potrazi za cvjetnim nektarom. Poznato je kako izviđači mogu prijeći čak i do 10 km u potrazi za adekvatnim zemljишtem. Nakon što je pronašla povoljnog kandidata, izvidnica se vraća u košnicu gdje prenosi prikupljenu informaciju izvodeći poseban ples (eng. waggle dance) prikazan na Slici 1. Iz tog gibanja ostatak pčela može razabrati smjer, udaljenost od košnice i ocjenu same lokacije. Izviđač koji je pronašao tu lokaciju predvodi sad grupu pčela natrag do cvijeća gdje se započinje s prikupljanjem resursa. Manji broj izviđača ostaje aktivan u svojoj potrazi za slučaj da postoji bolje rješenje od onog na kojem je trenutni fokus. Odvajanje rezultata koji su dali bolji odaziv na funkciju evaluacije od ostalih naziva se elitizam i njime se služimo radi očuvanja trenutnog optimuma rješenja. Kako svi znaju procjenu određenog zemljишta, jednom pronađeno bolje odredište lako zamjenjuje trenutno optimalno. BA uspješno je primjenjen u redovima linearnih antena (eng. linear antenna arrays), optimizaciji neuralnih mreža namijenjenih klasifikaciji (eng. optimization of neural networks for classification) te optimizaciji metoda formiranja klastera podataka (eng. optimization of clustering methods).

Implementacija:

BA započinje generiranjem intelligentnih agenata, virtualnih pčela, i njihovim oslobođanjem na domenu problema. Slijedi prikupljanje informacija o dodjeljenim lokacijama, uspoređivanje istih te raspoređivanje raspoloživih pčela na jednu od dviju uloga: analizu susjedstva elitnih lokacija iz prvog kruga pretrage ili daljnje traganje za moguće neotkrivenim boljim izvorima. Tako raspoređen posao prenosi se kroz iduću generaciju u naredni korak iteracije. BA je nešto jednostavniji od ACS algoritma jer se oslanja na kontinuiranost domene problema, odnosno na neprekinutost kao odnos između elemenata unutar domene. Za domenu koja je tako definirana možemo lako zamisliti postojanje gradijenta koji će rasti u smjeru optimuma tog susjedstva. Opasnost da će rješenje konvergirati samo ka lokalnom optimumu izbjegnuto je dalnjom potragom ostalih izviđača ukoliko uvjet zaustavljanja daje mogućnost oporavka od suboptimalnog rješenja.



Slika 1. Pčelin ples u službi izmjene informacija

Slični algoritmi:

Algoritam krijesnica(eng. Firefly algorithm) – algoritam oponaša interakciju među krijesnicama koristeći bioiluminaciju kao način privlačenja. Privlačenje među jedinkama funkcija je samo jačine svjetlosti, koja mora biti opadajućeg intenziteta s povećanjem udaljenosti između dva entiteta. Više o toj temi možete naći u radu „Nature-Inspired Metaheuristic Algorithms“ (Xin-She Yang, 2010).

Algoritam šišmiša (eng. Bat algorithm) – inspiriran snalaženjem šišmiša u prostoru pomoću bio-radara, ultrazvuka, alternirajući njegovu frekvenciju i glasnoću. Članak „A New Metaheuristic Bat-Inspired Algorithm“ iz rada „Nature Inspired Cooperative Strategies for Optimization“ (Xin-She Yang, 2011) sadrži detaljnije informacije o toj temi.

Algoritam sivog vuka (eng. Grey wolf optimizer) – ovaj model optimizacije oponaša strategiju lova sivih vukova. Čopor ima hijerarhijsku strukturu i sastoji se od alfa, beta, delta i omega jedinica koje tragaju za plijenom, okružuju plijen i napadaju plijen.

```

1 populacija = inicjalizirajPopulaciju(pčelebroj,problemveličina);
2 DOK_JE (  $\neg$  uvjetZaustavljanja() ) :
3     evaluirajPopulaciju(populacija);
4     pčelenajbolji = računajNajboljeRješenje(populacija);
5     idućaGeneracija=0;
6     zemljišteveličina =(zemljištePočetniveličina * zemljišteSmanjenjefaktor);
7     lokacijenajbolja =odrediNajboljuLokaciju(populacija, lokacijebroj );
8     ZA_SVAKOG lokacija[i] ∈ lokacijenajbolja :
9         aktivnePčele = 0;
10        AKO i<elitneLokacijebroj ONDA
11            aktivnePčelebroj= elitnePčelebroj;
12        INAČE
13            aktivnePčelebroj= ostalePčelebroj;
14        KRAJ;
15        Susjedstvo=0;
16        ZA j DO aktivnePčelebroj :
17            susjedstvo= stvoriSusjedstvo(lokacija[i],zemljišteveličina);
18        KRAJ;
19        idućaGeneracija=odrediNajboljeRješenje(susjedstvo);
20    KRAJ;
21    preostalePčelebroj = (pčelebroj - lokacijebroj);
22    ZA j DO preostalePčelebroj:
23        idućaGeneracija=stvoriNasumičnuPčelu();
24    KRAJ;
25    populacija=idućaGeneracija;
26 KRAJ;
27 VRATI pčelenajbolji ;

```

Pseudokod 3: Algoritam pčela

2.3.3. Algoritmi prepoznavanja imunološkog sustava

Opis:

Algoritmi prepoznavanja imunološkog sustava (eng. Artificial immune recognition system, u dalnjem tekstu oznaka AIRS) adaptivni su sustavi inspirirani imunitetnim funkcijama složenih organizama. Napad antigena na organizam rezultira reakcijom limfocita koji se vežu na strana tijela. Slijedi samoumnožavanje imunološke stanice u više tisuća kopija uz diferencijaciju samog tipa. Rezultat umnožavanja su plazma i memorijске stanice. Plazma stanice žive kraće i razvijaju velike količine molekula kao odgovor na napad s ciljem nadjačavanja napadača. Memorijске stanice imaju duži vijek života i njihova je uloga prepoznavanje i brži odgovor organizma na mogući budući napad. Cijeli proces karakterizira bitna pojava nazvana somatska hipermutacija koja označava izloženost reproducirajuće stanice manjim greškama pri umnožavanju. Posljedica toga je promjenjena sposobnost prepoznavanja limfocita i antitijela stvorenih od plazma stanica. AIRS algoritmi koriste se kao nadzorni algoritam za klasifikaciju domena problema.

Implementacija:

Inicijalizacija nakupine memorije ili memorijskog bazena (eng. memory pool) prvi je korak koji AIRS algoritam mora poduzeti. Stanice se potom stimuliraju ulaznim uzorcima uz izdvajanje predstavnika, no ako se procijeni da niti jedna ne zadovoljava minimalna očekivanja, stvaraju se nove stanice za specifični uzorak. Najbolja stanica biva klonirana te u tom procesu biva podvrgnuta manjim mutacijama, a njezini se potomci natječu za povratkom u memorijski bazen ne bi li prenijeli najbolja svojstva u sljedeću iteraciju. Općenito algoritam za svaki ulazni uzorak ulaznog skupa uzorka generira pripadajuće rješenje koje se pohranjuje u memoriju stanica i tako ostaje očuvano za usporedbu s narednim uzorcima.

Slični algoritmi:

*Optimizacija kolonijom bakterija (eng. Bacterial Colony Optimisation) – kroz kemotaksiju (sposobnost pokretnih stanica da prepoznaju odgovarajući kemijski gradijent), komunikaciju, eliminaciju, reprodukciju i migraciju algoritam simulira životni ciklus bakterije *Escherichia coli*. Eliminacija, reprodukcija i migracija izvršavaju se tek kad je zadovoljen određen kriterij, a sama se komunikacija bazira na individualnu i grupnu razmjernu informaciju.*

```

1 stanicememorija = inicjalizirajNakupinuMemorije(ulazniUzorci);
2 ZА_SVAKI ulazniUzorak[i] ∈ ulazniUzorci :
3     stimuliraj(stanicememorija,ulazniUzorci);
4     stanicanajbolja = najvećaStimulacija(ulazniUzorak[i], stanicememorija);
5     AKO stanica ≠ ulazniUrorka[i] ONDA
6         stanicememorija= stvariNovuMemorijskuStanicu(ulazniUzorak[i]);
7     INAЧЕ
8         klonovibroj = stanicanajbolja * klonkoef * mutacijakoef ;
9         staniceklonovi = stanicanajbolji;
10    ZА i DO klonovibroj :
11        staniceklonovi = klonirajIMutiraj(stanicanajbolji);
12    KRAJ;
13    DOK_JE prosječnaStimulacija(staniceklonovi) <=stimulacijagranična:
14        ZА_SVAKI stanica[i] ∈ stanice :
15            staniceklonovi=klonirajIMutiraj(stanice[i]);
16        KRAJ;
17        stimuliraj(staniceklonovi, ulazniUzorci);
18        reducirajNakupinuResursa(staniceklonovi,resursimax);
19    KRAJ;
20    stanicas = najvećaStimulacija(ulazniUzorak[i], staniceklonovi);
21    AKO stanicas > stanicanajbolja ONDA
22        stanicamemorija = stanicas;
23        AKO afinitet(stanicas,stanicanajbolja) <=afinitetgranični ONDA
24            uništiStanicu(stanicanajbolja,stanicamemorija);
25        KRAJ;
26        KRAJ;
27    KRAJ;
28    KRAJ;
29    VRATI stanicememorija;

```

Pseudokod 3: Algoritam imunološkog sustava

2.3.4. Optimizacija rojem čestica

Opis:

Optimizacija rojem čestica (eng. Particle swarm optimisation, u dalnjem tekstu oznaka PSO) najopćenitiji je od svih SI algoritama i čini bazu za mnoge varijacije. Formalno, zadaća algoritma je jednoznačno definiranje globalnog optimuma u multidimenzionalnom prostoru. Osnovna ideja algoritma je simulacija jata koje navigira kroz okruženje prilagođavajući svoje ponašanje vodećim jedinkama. Inteligentni agenti koji sačinjavaju jato definirani su svojim pozicijama u prostoru, domeni problema. Svakim korakom iteracije algoritma pridružuje se nova brzina pojedine čestice, zavisna o kretanju cijelog jata, i novoodređena pozicija kao rezultat napredovanja. Zbog svojih dobrih svojstava PSO se može koristiti u optimizacijskim problemima koji se dinamički mijenjaju s vremenom ili sadrže šumove i nejasne informacije među podacima.

Implementacija:

PSO započinje raspršivanjem čestica među nasumično odabranim točkama ulaznog problema. Tu svaka čestica ocjenjuje svoju lokaciju i glasa za inicijalni optimum. Rezultat metode novaBrzina vrijednost je dobivena prema formuli (iv).

$$v_i(n+1) = v_i(n) + (A * \text{rand}() * (pos_i^M - p_i(n))) + (B * \text{rand}() * (pos_G - p_i(n))) \quad (\text{iv})$$

Oznake u jednadžbi su iduće:

$v_i(n+1)$: brzina čestice i u aktualnom ($n+1$) koraku iteracije
$v_i(n)$: brzina čestice i u prethodnom (n) koraku iteracije
A	: težinski koeficijent za osobnu najbolju poziciju
B	: težinski koeficijent za globalnu najbolju poziciju
pos_i^M	: najbolja pozicija koju je čestica i posjetila
$p_i(n)$: trenutna pozicija čestice i u (n) koraku iteracije
pos_G	: najbolja pozicija koju je posjetio član jata

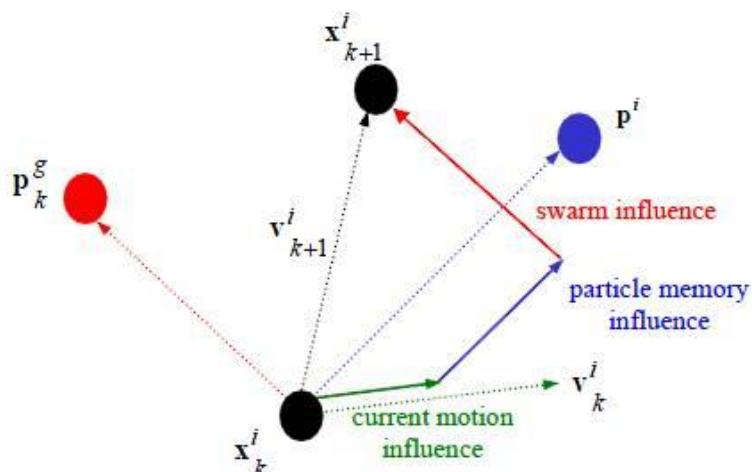
Pozicija se ažurira jednostavnim pridodavanjem vektora brzine na aktualnu poziciju. Konstante A i B najčešće se uzimaju kao brojevi u intervalu od 0 do 4 uključeno. Preporuka je da se i maksimalna brzina koju čestica može postići ograniči na fiksnu brzinu kako bi se postiglo preciznije manevriranje i mogućnost da se optimalno rješenje „preleti“ više puta bez da je identificirano. Primjer takvog izračuna prikazuje Slika 2.

```

1 populacija=0;
2 P_globalni=0;
3 ZA i=1 DO populacijaveličina :
4     Pbrzina=nasumičnaBrzina();
5     Ppozicija=nasumičnaPozicija(populacijaveličina);
6     Ppartikularni=Ppozicija;
7     AKO procjena(Ppartikularni)<=procjena(Pglobalni) ONDA
8         Pglobalni=Ppartikularni;
9     KRAJ;
10    KRAJ;
11    DOK_JE (  $\neg$  uvjetZaustavljanja() ) :
12        ZA_SVAKI P  $\in$  populacija :
13            Pbrzina=novaBrzina(Pbrzina, Pglobalni, Ppartikularni);
14            Ppozicija=novaPozicija(Ppozicija, Pbrzina);
15            AKO procjena(Ppozicija)<=procjena(Ppartikularni) ONDA
16                Ppartikularni=Ppozicija;
17                AKO procjena(Ppartikularni)<=procjena(Pglobalni) ONDA
18                    Pglobalni=Ppartikularni;
19                KRAJ;
20            KRAJ;
21        KRAJ;
22    KRAJ;
23    VRATI Pglobalni ;

```

Pseudokod 4: Optimizacija rojem čestica



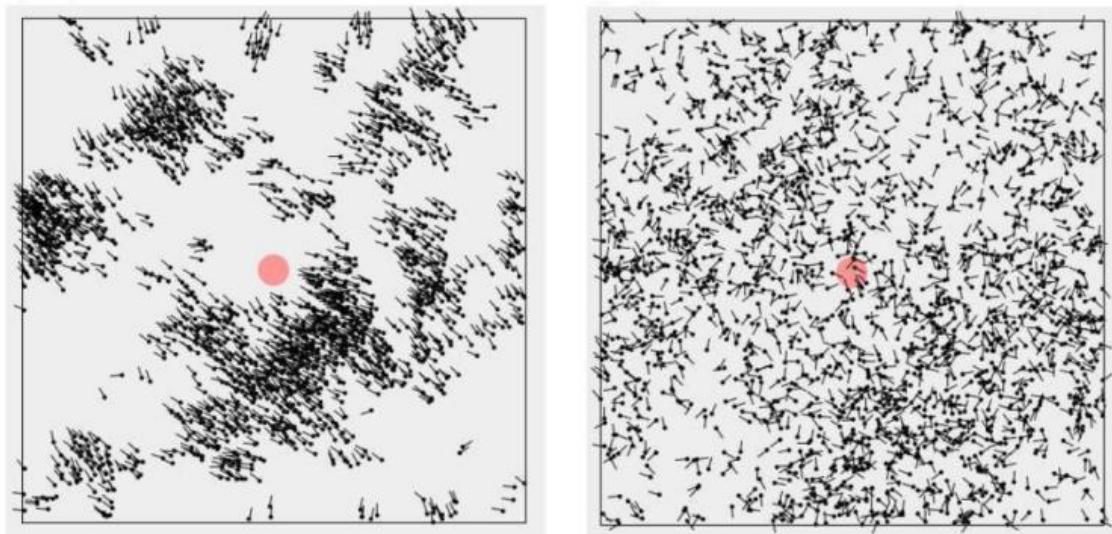
Slika 2. Izra unavanje nove pozicije u PSO algoritmu

Slični algoritmi:

Samoupravljuće čestice (eng. *Self-propelled particles*, SSP) – specijalni slučaj PSO, SSP karakterizira konstantna brzina čestica koje srednju vrijednost ukupnog smjera kretanja prilagođavaju promjenama u doprinosu pojedinih agenata. Tamás Vicsek u svom radu "Novel type of phase transition in a system of self-driven particles" objašnjava ovaj model kolektivnog ponašanja. Slika 3. prikazuje utjecaj ulaznih parametra na ponašanje algoritma. Lijevo je zabilježeno usmjereni gibanje zbog pravilnog podešavanja parametara algoritma dok je desno kaotično gibanje pojedinih agenata zbog neispravnog postavljanja istih.

Mnogostruka optimizacija rojem čestica (eng. *Multi-swarm optimization*, MSO) – općenito, postupak optimizacije može se podijeliti u dva segmenta: pretraga i iskorištavanje prikupljenih informacija. Predrugotrajna pretraga rezultira klasičnim postupkom uspoređivanja elemenata, rani fokus na obradu prikupljenih procjena i algoritam stagnira u uskom području pretrage. Klasičan pristup PSO mora balansirati te zadatke ne bi li uspješno donosio odluke. Za razliku od njega, MSO dijeli te zadatke u posebne faze koje se izmjenjuju: primjena pod-roja za pretragu i dodatne metode diverzifikacije za tempiranje i okidanje novog pod-roja.

Pretraga stohastičkom difuzijom (eng. *Stochastic diffusion search*) – primjenjuje se kad se cilj obrade može razložiti i grupirati u konačan broj manjih podcjelina. Takve se podfunkcije, čje su odluke da/ne prirode, dodjeljuju agentima koji postavljaju hipotezu. Hipoteze se testiraju, i uspoređuju u interakciji jedan naprema jedan. Iskustvo boljeg se usvaja kao ispravno i pretraga se nastavlja. Za bolje razumijevanje ovog algoritma preporučuje se analiza „The restaurant game“ primjera povezanog s pojmom „Stochastic diffusion search“.

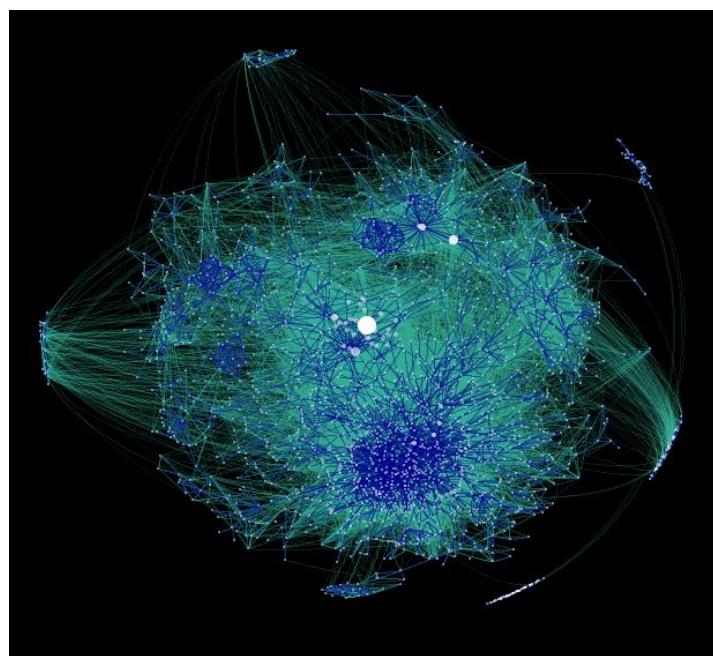


Slika 3. Samoupravljuće čestice

2.4. Primjene u praksi

Unato injenici da algoritmi inteligencije roja estica izra0avaju svoja rjezenja s nesigurnoz u koju uvode stohasti ke tehnike ra unanja, kona ni je rezultat dovoljno dobra aproksimacija, primjenjiva u in0enjerskoj struci. U zadnjih 20 godina u estalost primjena SI razrijezile su sve sumnje u njihovu u inkovitost. Organizacije kao zto su ESA (akronim, eng. European Space Agency) i NASA (ackronim, eng. National Aeronautics and Space Administration) oslanjaju se na efikasnost tih algoritama prilikom planetarnih mapiranja, upravljanja satelitima i analizi elekromagnetskih valova (eng. interferometry) uz sve ve i interes za novim primjenama.

Dobra svojstva SI algoritama potakla su njihovu primjenu pri analizi specifi nih uzoraka u velikim bazama podataka. Ti su procesi, poznati kao data mining, najidealniji za primjenu predstavljenih algoritama gdje se mo0e vidjeti njihova prava mo pri rjezavanju ne samo kompleksnih problema, ve i onih s enormnim brojem ulaznih podataka. Slika 4. prikazuje mapiranje i kompleksnu strukturu baze nad kojima se zahtijeva izvrzavanje data mininga.



Slika 4. Grafi ki prikaz slo0ene baze podataka

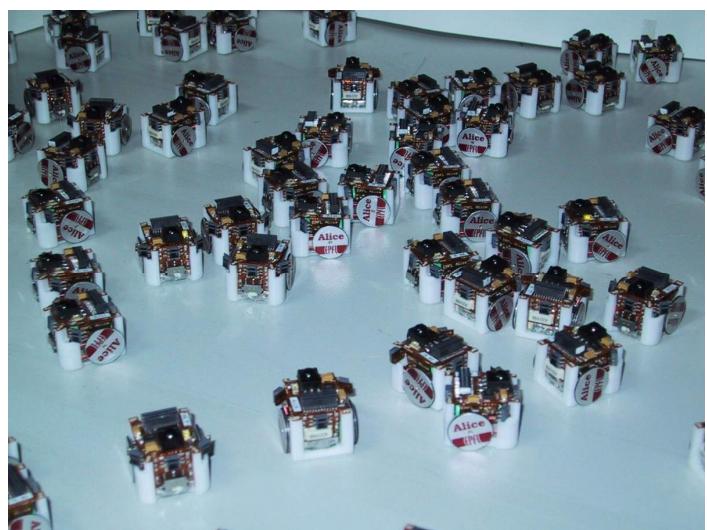
Vojna industrija koristi ih pri upravljanju bespilotnih vozila, avio kompanije pri rasporedu prtljage, u medicini lije nici se koriste SDS (eng. Stochastic-Difusion Search) algoritmom za prepoznavanje tumora sa slika dobivenih magnetskom

rezonancom. U telekomunikacijama ACS algoritmi preferirano su rjezenje prilikom ostvarivanja veza putem komunikacijskih mreža.

Filmska industrija oslanja se na tehnologiju izvedenu iz SI, prikladno nazvana **sMassive+**, u scenama velikih bitki i kontroliranja masa ljudi generiranih pomoću unarne grafike. Spomenimo kako su SI sustavi dali doprinos i u radu i istraživanju na hardware aspektima modernih sustava. Oblik antene prikazan na Slici 5. rezultat je optimizacije evolucijskih algoritama. Na Slici 6. prikazani su fizici ekvivalentni inteligentnim agentima, jednostavnii roboti ograničenih mogunosti, koji zajedno radom mogu formirati kompleksne strukture.



Slika 5. Oblik antene - rezultat SI optimizacije



Slika 6. Rojevi robova

3. Zaključak

Inteligencija roja estica revolucionaran je pristup rjezavanju problema koji svojim jednostavnim, a opet mo nim sposobnostima ra unanja privla e sve ve i interes stru njaka. Iako je tek metaheuristi ka tehnika, potreba za rjezavanjem zahtjevnih problema potakla je njihovo istra0ivanje i ulaganje kako bi danas postala valjan i zto je najbitnije pouzdan pristup. Sve ve i broj radova i novih istra0ivanja proziruje podru je njihove primjene, otkrivaju i nove koncepte i razvijaju i nove ideje, ali i poboljzavaju i ve poznate algoritme ine i ih efikasnijima ili jednostavnijim za koriztenje. Od ra unarne znanosti do telekomunikacija, istra0iva i programeri slu0e se predstavljenim algoritmima pri analizi mre0a, razvoju i upravljanjem sustava, obradi i pretra0ivanju podataka te dizajniranju strojeva i njihovim radom. Iako su neka ograni enja, kao dugo vrijeme obrade i izmjene generacija, problem koji ne e biti lako uklonjen, kako ini samu bit modela koji imitiramo, SI je daleko najbolji izbor za dobivanje rjezenja na ina e nerjezive ra unarske probleme.

4. Literatura

1. Kennedy J., Eberhart R.C., Shi Y., Swarm intelligence, Morgan Kaufmann Publishers, San Francisco, 2001
2. Golub, M. Genetski algoritmi: Prvi dio, 2004
3. Brownlee, J. Clever Algorithms: Nature-Inspired Programming Recipes, 2011.
4. Hinckley M.G., Rash J.L., Truszkowski W.F., Autonomous and Autonomic Swarms, NASA Goddard Space Flight Center, 2005

5. Sažetak

Inteligencija roja estica opisni je naziv za skup algoritama koje karakterizira zajedni ko ponazanje grupe jednostavnih subjekata, koji me usobnom interakcijom uspijevaju nadi i komplikirane probleme. Rad daje uvod u inteligenciju roja estica uz kratak opis umjetne inteligencije kao grane ra unarne znanosti iz koje se u aspektu automatskog planiranja i zakazivanja nalaze evolucijski algoritmi i sami rojevi estica. Iznose se izazovi u ra unarstvu, stvara motivacija za njihovo rjezavanje i predla0e sam koncept rojeva estica. Uz formalnu definiciju i generalne zna ajke, analiziraju se prednosti i nedostaci SI algoritama koji su zajedni ki svim pristupima. Fokus ovog rada je na samim algoritmima za koje se objaznjava njihova inspiracija iz realnog svijeta izla0u i njihov ekvivalentni virtualni model. Od mnogobrojnih algoritama koji su do danas poznati odabrani su algoritam kolonije mrava, algoritam p elja, algoritam prepoznavanja imunoložkog sustava te optimizacija rojem estica. Uz sam pseudokod navode se i alternativne ina ice ili pak prozirenja navedenih algoritama kako bi se potaknula znati0elja itatelja za ovim neisrpnim podru jem. Rad zavrjava pregledom nekih od primjena obra enih algoritama po institucijama i zadacima u kojima je SI pronazao stalnu primjenu.