**math.e****Hrvatski matematički elektronički časopis**

Genetski algoritmi i biomorfi

genetski algoritam grafovi heuristički algoritmi optimizacijaNela Bosner (nela.bosner@math.hr) i Tomislav Droždjek

Genetski algoritmi su jedna vrsta evolucijskih algoritama. Evolucijski algoritmi, kao što i samo ime govori, posebna su vrsta algoritama inspirirana procesom evolucije. Glavna ideja evolucijskih algoritama je, koristeći metodu pokušaja i pogrešaka, simulirati proces evolucije te ga primijeniti na rješavanje raznih optimizacijskih problema. Promotrimo sada detaljnije kako je pojam evolucije povezan s evolucijskim algoritmima. U teoriji evolucije, neku okolinu nastanjuje populacija jedinki kojima je "cilj" preživjeti i razmnožavati se. Podobnost (eng. fitness) tih jedinki govori nam koliko je pojedina jedinka uspješna u ispunjavanju tih ciljeva, odnosno, ona reprezentira šansu jedinke da preživi dovoljno dugo kako bi se razmnožavala. U kontekstu rješavanja problema, jedinke izjednačavamo s kandidatima za rješenje. Kvaliteta tih potencijalnih rješenja nam govori koliko dobro ona aproksimiraju rješenje problema. Nju možemo iskoristiti kako bismo odlučili s kolikom će vjerojatnošću određeni kandidat za rješenje sudjelovati u konstrukciji sljedećih kandidata (intuitivno, što kandidat za rješenje bolje aproksimira rješenje ta bi vjerojatnost trebala biti veća).

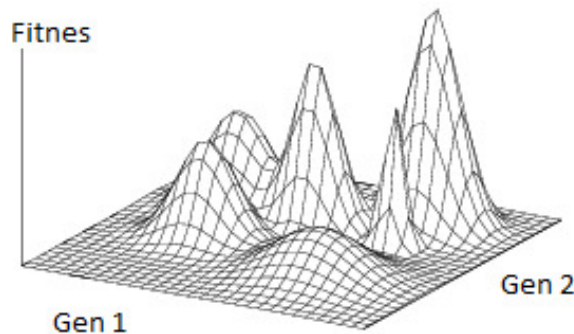
1 Inspiracija u biologiji

Darwinova teorija evolucije daje nam objašnjenje bioraznolikosti i mehanizama kojima se postiže bioraznolikost. U makroskopskom pogledu na evoluciju, glavnu ulogu igra proces prirodne selekcije. U okolini u kojoj može živjeti samo određen broj jedinki, očito je da je potreban neki oblik selekcije, ako se želi izbjeći eksponencijalan rast populacije. Prirodna selekcija favorizira one jedinke koje su najbolje prilagođene uvjetima okoline, tj. koje najbolje mogu iskoristiti dostupne resurse.

Okako opisana prirodna selekcija je jedna od dvije osnovne ideje teorije evolucije. Druga osnovna ideja rezultat je fenotipskih varijacija unutar populacije. Fenotip jedinke su karakteristike jedinke (fizičke ili biheviorističke) koje imaju direktan utjecaj na interakciju jedinke s okolinom. Odnosno, to su karakteristike koje utječu na njenu podobnost, a preko toga i na vjerojatnost njenog preživljavanja. Svaka jedinka jedinstvena je kombinacija fenotipskih karakteristika. Ako okolina te fenotipske karakteristike ocijeni povoljno, onda se one zadržavaju u populaciji kroz potomstvo te jedinke. S druge strane, negativno ocijenjene karakteristike gube se, jer negativno ocijenjene jedinke češće umiru bez potomstva. Darwin je uočio da se male nasumične varijacije, mutacije, u fenotipu događaju tijekom reprodukcije iz generacije u generaciju. Kao rezultat tih varijacija, pojavljuju se i ocijenjuju nove kombinacije svojstava. Najbolje među njima prežive i razmnožavaju se, i time omogućuju evoluciju.

Ukratko, populacija se sastoji od određenog broja jedinki. Vjerojatnost da se te jedinke razmnožavaju direktno ovisi o tome koliko su one dobro prilagođene okolini. Razmnožavanjem uspješnijih jedinki, uz povremene mutacije, pojavljuju se nove jedinke. Time se, uz dovoljno vremena, mijenja cijela populacija, dakle ona evoluirala.

Grafički se taj proces može prikazati kao na slici 1. z-os grafa prikazuje podobnost (fitness), dakle veći z pridružujemo boljoj podobnosti i obrnuto. x i y os pridružujemo nekim karakteristikama jedinke. Očito, na x–y ravnini sada se nalaze sve moguće kombinacije karakteristika, dok se na z-osi može očitati podobnost jedinke s tim karakteristikama. Neku populaciju sad možemo zamisliti kao skup točaka u prostoru, gdje svaka točka predstavlja jednu jedinku. Evoluciju tada možemo zamisliti kao proces postupnog pomaka populacije na veću visinu. Valja spomenuti da je, zbog konačnog broja jedinki te zbog nasumičnosti u cijelom procesu, moguć gubitak dobro prilagođenih jedinki iz populacije. To nam, za razliku od procesa optimizacije, omogućuje i "pomicanje nizbrdo" te nam ništa ne jamči da će se populacija vratiti istim putem. Iz toga slijedi da je moguće pobjeći iz lokalnih optimuma te postići globalni optimum.



Slika 1: Primjer ovisnosti vrijednosti podobnosti o kombinaciji 2 gena.

2 Što je evolucijski algoritam?

Kao što smo već spomenuli u prethodnom poglavlju, ideja na kojoj se baziraju evolucijski algoritmi je da će se opća podobnost populacije podići kao rezultat prirodne selekcije. Polazište evolucijskih algoritama je reprezentacija jedinki. Ona definira vezu između prostora u kojem se nalazi problem i prostora u kojem će evolucijski algoritam tražiti rješenje. Kandidate za rješenje unutar prostora u kojem se nalazi problem zovemo *fenotipi*, dok njihove reprezentante u algoritmu zovemo *genotipi*. Za danu funkciju podobnosti koju trebamo maksimizirati, nasumično kreiramo skup kandidata za rješenje te na njih primijenimo funkciju. Na temelju rezultata, možemo odabrati bolje kandidate kao roditelje sljedeće generacije i na njih primijeniti rekombinaciju i/ili mutaciju.

.

Rekombinacija je binarni operator varijacije (eng. recombination, crossover). Kao što i samo ime govori, ovaj operator spaja genetske informacije dva roditelja u jedan ili više potomaka. Operator rekombinacije je stohastički; izbor dijelova koji će svaki roditelj prenijeti na potomke te način spajanja tih dijelova je nasumičan.

.

Mutacija je unarni operator varijacije. Ulazni podatak operatora mutacije je jedan genotip, a izlazni je modificirani genotip. Promjena između ta dva genotipa je obično mala. Mutacija je uvijek stohastički operator; genotip potomka je rezultat niza nasumičnih promjena.

Primjenom tih operatora dobijemo novi skup kandidata za rješenje za koje opet izračunamo njihovu podobnost. Nakon toga, novi zajedno sa starim kandidatima ulaze u izbor za kandidate koji će se prenijeti u sljedeću generaciju. Taj izbor najčešće se zasniva na temelju podobnosti, ali se može zasnivati i na temelju starost kandidata.

Opisani proces ponavljamo sve dok ne pronađemo dovoljno dobrog kandidata za rješenje.

Dakle, gore navedeni proces zasniva se na dvije stvari:

.

Operatori mutacije i rekombinacije stvaraju potrebnu raznolikost i time omogućavaju varijacije u populaciji

.

Funkcija podobnosti omogućuje nam selekciju rješenja.

Kombinacija tih dviju pokretačkih sila vodi do porasta podobnosti kandidata kroz generacije.

Primijetimo da je evolucijski proces stohastički. Iako je vjerojatnost odabira jedinki s višom podobnosti veća od vjerojatnosti odabira jedinki s manjom podobnosti, čak i te manje kvalitetne jedinke imaju pozitivnu vjerojatnost preživljavanja, kao i vjerojatnost da će postati roditelj. Naime, i kod odabira roditelja kod rekombinacije, i kod odabira jedinke za mutaciju, taj odabir se vrši nasumično. Algoritam 1 prikazuje pseudokod evolucijskog algoritma.

Algorithm 1: Evolucijski algoritam

```

1 INICIJALIZIRAJ populaciju sa nasumičnim kandidatima;
2 IZRAČUNAJ podobnost za svakog kandidata;
3 while nije zadovoljen UVJET ZAUSTAVLJANJA do
4   Odaberi roditelje;
5   Izvrši REKOMBINACIJU roditelja;
6   MUTIRAJ potomstvo;
7   EVALUIRAJ nove kandidate;
8   ODABERI jedinke koje će sačinjavati sljedeću generaciju;
9 end

```

Algoritam 1: Evolucijski algoritam

3 Genetski algoritmi

Ne postoji jedan način za kreiranje genetskog algoritma; on nastaje kombinacijom različitih raspoloživih operatora tako da bude prikladan za rješavanje nekog konkretnog problema. Ideju genetskog algoritma je inicijalno začeo Holland 1973. godine kao sredstvo proučavanja adaptivnog ponašanja. Oni se u glavnom smatraju optimizacijskim metodama. Klasični genetski algoritam ima binarnu reprezentaciju, selekciju proporcionalnu podobnosti, nisku vjerojatnost mutacije, naglasak na genetski inspiriranoj rekombinaciji, i generacijski izbor kandidata koji će se prenijeti u sljedeću generaciju. Kao što smo već prije spomenuli, prvi korak u konstrukciji bilo kojeg evolucijskog algoritma je određivanje reprezentacije kandidata za rješenje. To uključuje definiciju genotipa te preslikavanja s genotipa u fenotip. Kod genetskih algoritama najčešće se koriste

.

binarna reprezentacija: genotip prikazan kao string binarnih znamenki,

.

reprezentacija prirodnim brojevima: genotip prikazan kao string prirodnih ili cijelih brojeva, ili nekog njihovog podskupa,

.

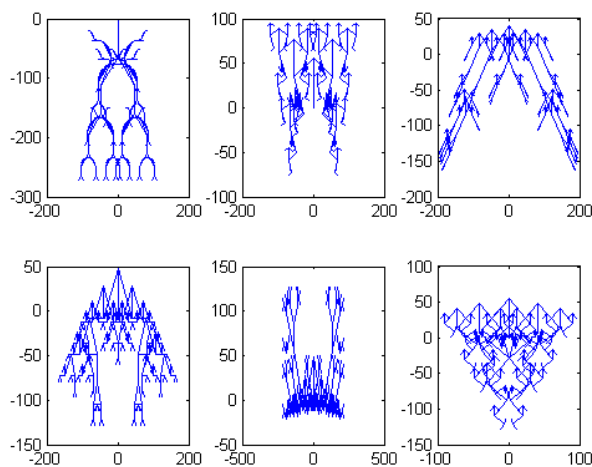
reprezentacija realnim brojevima: genotip prikazan kao string realnih brojeva,

.

reprezentacija permutacijama: genotip je prikazan kao permutacija skupa prirodnih brojeva.

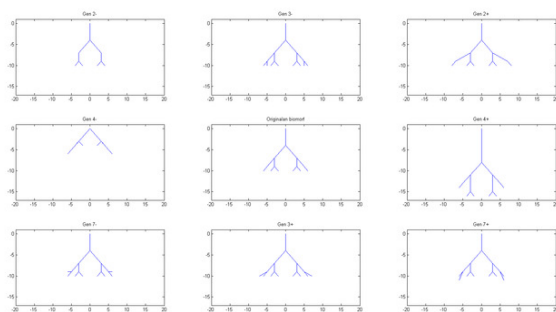
4 Eksperimenti s biomorfima

Rad genetskih algoritama ilustrirat ćemo primjerima biomorfa, koji su inspirirani pravim biološkim oblicima i njihovom evolucijom. *Biomorf* je u širem smislu riječi bilo koji objekt koji nalikuje na živo biće. Njih je u računarstvo prvi uveo Richard Dawkins u knjizi "The blind watchmaker", kako bi pokazao da se gomilanjem malih promjena na genima kroz dovoljno generacija mogu razviti bitno drugačiji organizmi od početnog. U našim eksperimentima biomorfi su dvodimenzionalna "bića" nalik grančicama, koja nastaju uzastopnim grananjem linija na način koji je određen njihovim genima. Slika 3 prikazuje nekoliko tipičnih primjera biomorfa. Eksperimente slične Dawkinsovima pokušat ćemo ponoviti i u ovom članku, kako bismo demonstrirali efikasnost genetskih algoritama, uz bitnu razliku da ćemo podobnost organizama vrednovati nekom funkcijom podobnosti, dok je Dawkins u originalnom radu sam birao organizme koji će biti roditelji sljedeće generacije na temelju njihovog izgleda. Konkretno, pokrenut ćemo neki genetski algoritam na različitim problemima s različitim parametrima te usporediti rezultate.



Slika 3: Neki tipični biomorfi.

Pozabavimo se detaljnije strukturom biomorfa. Biomorfi koje ćemo koristiti u našim pokusima imat će 9 gena. Prvih 8 reprezentirat ćemo cijelim brojevima u rasponu od -8 do 7, dok ćemo deveti gen reprezentirati prirodnim brojevima od 1 do 8. Taj, deveti gen reprezentira broj "nivoa" od kojih će se sastojati određeni biomorf, dok prvih 8 utječe na duljinu i nagib njegovih grana. Konkretno, predznak prvih 8 gena govorit će nam u kojem smjeru vodi određena grana, dok će njihov iznos utjecati na duljinu. Slika 4 prikazuje jedan, relativno jednostavan biomorf u sredini (genotipa $[-1,-1,-1,-1,-1,-1,-1,-1,4]$), dok su oko njega neki biomorfi nastali mutacijom samo jednog od njegovih gena za +1 ili -1. Svaka grana biomorfa se, kad dođe do kraja, razdvaja na točno dvije nove grane. Duljina grane, osim što ovisi o genima, linearno se smanjuje s brojem grananja od korijena do te grane. Ukoliko se neka od tih grana ne vidi na crtežu, njezina duljina je 0, no još uvijek se iz nje mogu granati nove grane.

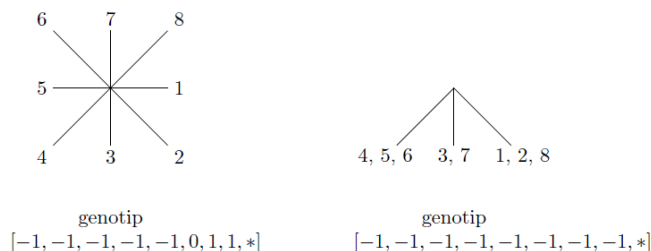


Slika 4: Prikaz biomorfa i njemu bliskih biomorfa.

Objasnimo sada malo pobliže proces grananja biomorfa. U trenutku kada grana dođe do svojeg kraja, geni određuju duljinu i kut grananja grana nastalih iz nje. Primijetimo da je prva grana biomorfa usmjerena ravno prema dolje, dakle raste "u smjeru juga". Grane nastale iz nje tada rastu u smjeru "jugoistoka" i "jugozapada". Zapravo, ako zamislimo ružu vjetrova s 8 krakova, grananje iz neke grane uvijek će ići "u smjeru" 2 susjedna kraka. Navodnike smo ovdje stavili jer ti smjerovi nisu pravilno raspoređeni kao na ruži vjetrova, bitno je samo da ih ima 8. Kad bi za svaki od tih smjerova jedan gen na neki način određivao pomak u smjeru x, a jedan u smjeru y, očito bi trebali 16 različitih gena. No, primijetimo da su svi biomorfi simetrični s obzirom na y. To nam omogućuje redukciju broja gena na 8. U tablici 1, za svaki smjer ruže vjetrova dan je indeks gena koji na njega utječe u smjeru x i y osi. Smjer 3 je ravno dolje, 7 je ravno gore (zbog toga jer je dx u tim smjerovima jednak 0), dok su ostali smjerovi raspoređeni u smjeru kazaljke na satu.

$dx(1) = -dx(5)$	$dy(1) = dy(5)$
$dx(2) = -dx(4)$	$dy(2) = dy(4)$
$dx(3) = 0$	$dy(3) = gen(4)$
$dx(4) = gen(1)$	$dy(4) = gen(5)$
$dx(5) = gen(2)$	$dy(5) = gen(6)$
$dx(6) = gen(3)$	$dy(6) = gen(7)$
$dx(7) = 0$	$dy(7) = gen(8)$
$dx(8) = -dx(6)$	$dy(8) = dy(6)$

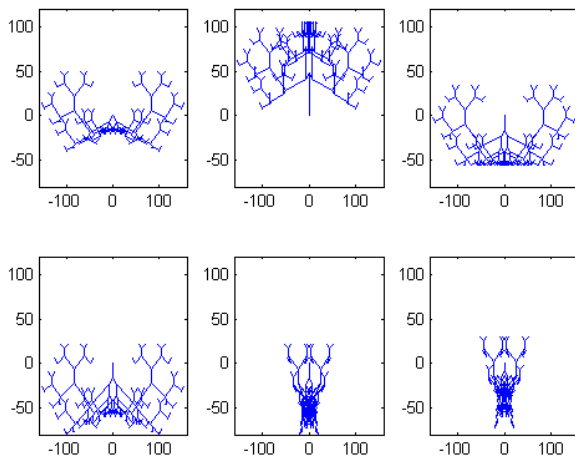
Tablica 1: Veza gena i rasta biomorfa.



Slika 5: Ruže vjetrova za dva genotipa.

Iako na prvi pogled i nije jasno koliko veliku varijabilnost fenotipa biomorfa možemo dobiti na ovaj način, na slici 6 dano je 6 biomorfa koji su dobiveni jedan od drugog mutacijom samo jednog gena, dakle drugi biomorf dobiven je mutacijom jednog gena iz prvog itd. (Mutacija je ovdje implementirana kao nasumično biranje bilo kojeg

$n \in [-8,7]$). Razlike u fenotipu su drastične, iako je promijenjen samo jedan gen.



Slika 6: Ilustracija promjena na biomorfu nastalih mutacijom jednog gena.

4.1 Primjeri

Naše primjere modelirat ćemo genetskim algoritmom sa sljedećim elementima.

.

Populacija se sastoji od 500 jedinki, nasumično inicjaliziranih.

.

Mehanizam odabira roditelja je turnirska selekcija (deterministički odabir, biramo λ roditelja). Ovakav mehanizam odabira pogodan je za situacije kad možemo uspoređivati samo koliko je neka strategija dobra u usporedbi s nekom drugom. Turnirska selekcija ne zahtijeva znanje o cijeloj populaciji, dovoljno je imati relaciju koja može usporediti bilo koje 2 jedinke. Pseudokod turnirske selekcije koja odabire λ roditelja dan je sa algoritmom 2. Vjerojatnost odabira jedinke kao pobjednika turnira ovisi o rangu jedinke u populaciji, i o veličini turnira k .

Algorithm 2: Pseudokod za turnirsku selekciju

```

1 broj trenutnih članova = 1;
2 while broj trenutnih članova ≤ λ do
3   odaberi nasumičnih k jedinki;
4   odaberi najboljeg među njima s obzirom na njihovu podobnost,
   označi ga s i;
5   roditelj[broj trenutnih članova] = i;
6   broj trenutnih članova = broj trenutnih članova+1;
7 end

```

Algoritam 2: Pseudokod za turnirsku selekciju

.

Mutacija je implementirana kao biranje nasumičnog broja iz $[-8,7]$ za jedan gen u jedinki, pri čemu do mutacije dolazi s vjerojatnošću p .

.

Rekombinacija je implementirana kao križanje u jednoj točki, a broj nivoa uzimamo nasumično od jednog roditelja. U ovom tipu operatora rekombinacije odabiremo nasumičan broj iz $[0, l-1]$, gdje je l duljina stringa genoma, te nakon toga "presječemo" oba roditelja u toj točki i zamijenimo njihove zadnje dijelove. Time dobijemo dvoje nove djece.



Slika 8: Primjer rekombinacije u jednoj točki.

.

Odabir preživjelih se obavlja izbacivanjem λ najmanje podobnih jedinki iz populacije.

.

Uvjet zaustavljanja je zadovoljen kad je postigunt maksimalan broj generacija ili nema nikakvog poboljšanja u podobnosti populacije u 20 generacija.

Genetski algoritam pokrenuti ćemo 100 puta sa svakom kombinacijom parametara te ćemo za svaki od njih, u svakoj generaciji pamtiti najbolju i prosječnu podobnost, kao i genotip najbolje jedinke. Parametri koje smo koristili dani su u tablici 2.

	Set1	Set2	Set3	Set4
Parametar mutacije (p)	0.05	0.05	0.05	0.25
Broj djece u svakoj generaciji (λ)	100	60	100	100
Veličina turnira (k)	2	2	6	2

Tablica 2: Parametri genetskog algoritma

Zamislimo da želimo konstruirati takav biomorf kojem podobnost raste sa svakim slobodnim krajem grane unutar nekog prostora, no za konstrukciju njegovih grana potrebni su resursi, dakle što je suma duljina grana dulja, to je veći negativni efekt na podobnost jedinke (ovdje smo cijenu definirali kao duljinu grane). Možemo zamisliti da se radi o korijenu drva koje raste na tlu u kojem su nejednoliko raspoređene hranjive tvari. Točnije, količina hranjivih tvari linearno raste porastom dubine, do $y=-200$ nakon čega počne linearno padati. Primijenili smo gore opisani genetski algoritam, koristeći sva 4 seta parametara. Maksimalan broj iteracija stavili smo na 200.

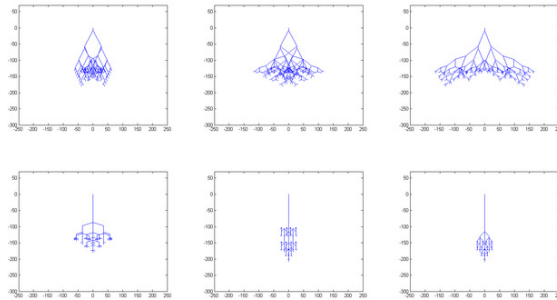
Tablica 3 prikazuje broj pokretanja kod kojih rješenje nije pronađeno te broj dobivenih rješenja koja su različita od najboljeg dobivenog rješenja, kao i odstupanja od najboljeg rješenja.

	Set1	Set2	Set3	Set4
Nije nađeno rješenje	0	0	0	0
Suboptimalna rješenja	29	49	45	46
Prosječno iteracija do nalaženja rješenja	85.23	108.09	56.90	95.08
Najveće odstupanje od najboljeg rješenja	17.89%	34.5%	4.56%	26.73%
Prosječno odstupanje od najboljeg rješenja	0.7%	2.21%	1.00%	1.34%

Tablica 3: Podaci o izvedbi algoritma

Možemo primijetiti da je najbrže nalaženje rješenja kod parametara iz seta 3 zbog najvećeg selekcijskog pritiska, kao i najsporije uz korištenje parametara iz seta 2 zbog najmanjeg broja djece u svakoj generaciji. Zanimljivo je uočiti da odabir prevelikog parametra mutacije negativno utječe na kvalitetu rješenja. No, crtanjem najbolje jedinke ispostavilo se da je algoritam kao optimalno rješenje pronašao okomitu crtu prema dolje (točnije, više crta koje sve putuju ravno prema dolje; svaki pomak po osi x je nepoželjan jer ne nosi nikakvu nagradu, a negativno utječe na podobnost). Slika 9 prikazuje nekoliko biomorfa koji su u jednom pokretanju algoritma bili najbolje rješenje u svojoj generaciji. Promatramo li ih po redu (od gore lijevo prema dolje desno), možemo jasno vidjeti kako se biomorf stanjuje i približava

ravnoj okomitoj liniji. Ponovimo sada pokus tako da dodatno nagradimo jedinke koje će imati veći raspon "korijena".



Slika 9: Neki biomorfi dobiveni u jednom pokretanju algoritma.

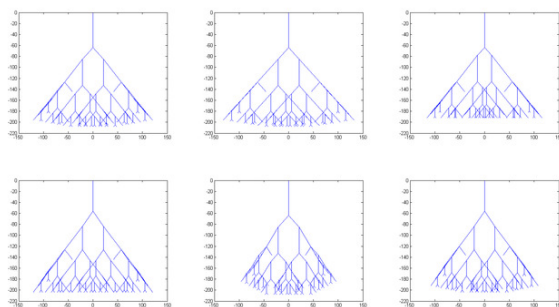
Koristit ćemo istu funkciju podobnosti kao i u prethodnom primjeru, no dodat ćemo još jedan pribrojnik koji je proporcionalan najvećoj udaljenosti 2 slobodna kraja grana. Time se nadamo dobiti biomorfe koji se neće grupirati oko samo jedne linije, a ujedno ćemo dodatno zakomplicirati funkciju podobnosti te promotriti kako se naš algoritam nosi s tim. Opet smo algoritam pokrenuli 100 puta sa svakim setom podataka, s maksimalno 200 iteracija. Tablica 4 prikazuje broj pokretanja kod kojih rješenje nije uspješno pronađeno, broj dobivenih rješenja koja su različita od najboljeg dobivenog rješenja te odstupanja od najboljeg rješenja.

	Set1	Set2	Set3	Set4
Nije nađeno rješenje	0	0	0	0
Suboptimalna rješenja	36	56	58	36
Prosječno iteracija do nalaženja rješenja	90.26	108.81	65.64	89.94
Najveće odstupanje od najboljeg rješenja	2.9%	20.66%	4.98%	3.22%
Prosječno odstupanje od najboljeg rješenja	0.44%	2.04%	0.80%	0.48%

Tablica 4: Podaci o izvedbi algoritma

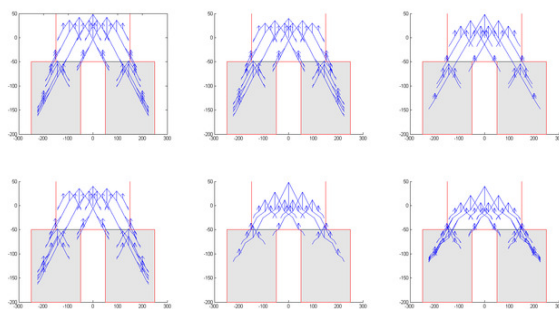
Možemo primijetiti da se prosječno odstupanje od najboljeg rješenja smanjilo, iako se broj suboptimalnih rješenja povećao. Posebno vrijedi istaknuti porast u kvaliteti performansi seta parametara s većim parametrom mutacije. To možemo pripisati strukturi prostora rješenja. U prošlom primjeru velik parametar mutacije onemogućavao je brzo penjanje prema optimalnom rješenju što je usporavalo algoritam i često nas izbacivalo s puta prema tom optimumu; ovdje se to pokazalo korisnim zbog kompleksnije strukture prostora (većeg broja bliskih po podobnosti lokalnih optimuma) te omogućilo skakanje između njih. Međusobni odnos rezultata dobivenih različitim parametrima ostao je vrlo sličan kao i u prethodnom primjeru. Možemo primijetiti da su se performanse algoritma s parametrima iz seta 4 skoro izjednačile s onima iz prvog seta, iako su u jednostavnijem primjeru bile dosta lošije. Set 2 i dalje je najlošiji, dok set 3 daje najbrže nalaženje rješenja, no i nešto lošiju točnost rezultata.

Također, zanimljivo je pogledati koliko su različiti biomorfi koje je algoritam vratio kao rješenja. Na slici 10 prikazan je biomorf koji je optimalno rješenje problema (gore lijevo), kao i 5 najboljih rješenja koja nisu optimalna. Ostala rješenja poredana su po podobnosti (dolje desno je najlošije rješenje).



Slika 10: Fenotip biomorfa koje je algoritam vratio kao rješenje problema.

Kao zadnji primjer sa simetričnim biomorfima promotrimo prostor u kojem su biomorfi kažnjeni ukoliko im grane imaju vrhove u određenim dijelovima ravnine, a nagrađeni ukoliko imaju vrhove u nekim drugim dijelovima ravnine. To znači da postoji velika vjerojatnost da algoritam zapne u lokalnom optimumu (situacija u kojoj nijedan biomorf nema nijednu granu ni u području koje se nagrađuje, ni u području koje se kažnjava). Neka rješenja (gore lijevo je optimalno rješenje), kao i dijelovi za koje biomorfi dobivaju kaznu, odnosno nagradu prikazani su na slici 11.



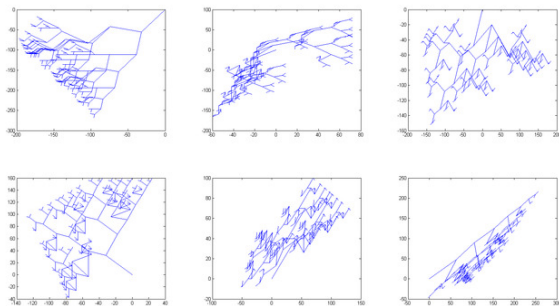
Slika 11: Fenotip biomorfa koje je algoritam vratio kao rješenje problema. Crvena crta je granica koju biomorfi ne mogu prijeći bez kazne. Sivo osjenčano područje je područje u kojem dobivaju nagradu.

U tablici 5 dani su podaci o izvršavanju algoritma. Vidimo velik pad u kvaliteti rješenja, tj. izrazito veliko najveće i prosječno odstupanje od najboljeg rješenja. Ipak, algoritam je sa sva 4 seta podataka pronašao identično najbolje rješenje. Pregledom rješenja s jako velikim odstupanjem od najboljeg rješenja ispostavilo se da to zaista jesu biomorfi koji nisu uspjeli doći do područja s nagradom (ostali su u neosjenčanom području oko ishodišta, ali i unutar crvene linije). Ovo je dobar primjer problema u kojem bi dobra inicijalizacija mogla bitno poboljšati algoritam. Naime, ukoliko na početku u populaciju ubacimo par jedinki koje imaju neke grane u sivom području, a da ne prelaze crvenu liniju, možemo smanjiti vjerojatnost zapinjanja u lokalnom optimumu. Primijetimo pozitivan učinak koji ima povećanje broja jedinki u turniru, kao i negativan utjecaj koji ima smanjenje broja djece. Također, povećanje parametra mutacije bitno je pogoršalo prosječnu kvalitetu rješenja.

	Set1	Set2	Set3	Set4
Nije nađeno rješenje	0	0	0	0
Suboptimalna rješenja	87	94	88	98
Prosječno iteracija do nalaženja rješenja	91.95	68.27	64.49	65.23
Najveće odstupanje od najboljeg rješenja	89.98%	79.97%	32.33%	71.24%
Prosječno odstupanje od najboljeg rješenja	13.59%	37.21%	7.35%	36.16%

Tablica 5: Podaci o izvedbi algoritma

I za kraj, spomenimo da biomorfi mogu biti i asimetrični. Tada, umjesto 8 gena koji određuju smjer i kut među granama, koriste se 16 gena, dok sedamnaesti gen određuje broj nivoa u biomorfu. Neki primjeri asimetričnih biomorfa nalaze se na slici 12.



Slika 12: Neki asimetrični biomorfi.

5 Zaključak

Kao zaključak ovih eksperimenata možemo istaknuti da izbor optimalnih parametara uvelike ovisi o tipu problema koji želimo riješiti. Dok kod najjednostavnijih problema izbor najboljih parametara i nije bio od presudne važnosti jer su se svi setovi parametara dobro nosili s problemom, kod kompleksnijih problema počele su se primijećivati neke razlike. Od bitnijih stvari možemo istaknuti negativan efekt koji relativno velik parametar mutacije ima kod traženja rješenja na prostoru s jednim globalnim optimumom mnogo boljim od lokalnih optimuma, dok kod prostora s više lokalnih optimuma bliskih po vrijednosti globalnom optimumu povećanje parametra mutacije može imati pozitivan efekt. Također, na svim primjerima potvrdilo se da povećanje broja jedinki u turniru dovodi do mnogo bržeg pronalaska rješenja, no povećava i šansu zapinjanja u lokalnom optimumu. Ipak, najnegativniji je učinak na izvedbu algoritma u svim primjerima imalo preveliko smanjivanje broja djece. Na kraju napomenimo da su ovi primjeri ilustrirali samo jedan mali dio raznolikosti biomorfa, koji su reprezentirani sa samo 9 gena.

Bibliografija

[1]

R. Dawkins, *The blind watchmaker*, Norton & Company, Inc, New York, NY, 1986.

[2]

A.E. Eiben i J.E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.

