

UPORABNOST IN UČINKOVITOST KANONIČNEGA GENETSKEGA ALGORITMA

GAJA ŽUMER

Fakulteta za matematiko in fiziko
Univerza v Ljubljani

60J20, 68W20

Genetski algoritem je stohastična optimizacijska metoda za reševanje zahtevnejših oziroma slabše obvladljivih optimizacijskih problemov. V članku je najprej opisana njegova implementacija, sledeči primeri pa opozarjajo na pasti, ki se lahko pri tem pojavijo. Pri iskanju rezultata genetski algoritem preiskuje območja, za katera je bolj verjetno, da bodo vsebovala globalno optimalno rešitev. O tem govori izrek o shemah, ki nakazuje na mehanizem napredovanja algoritma, ne moremo pa ga uporabiti za analizo konvergence. V ta namen potrebujemo teorijo končnih homogenih markovskih verig. Članek vsebuje komentar na konvergenco kanoničnega genetskega algoritma in dveh njegovih različic.

APPLICABILITY AND EFFICIENCY OF THE CANONICAL GENETIC ALGORITHM

Genetic algorithm is a stochastic optimisation method for solving difficult optimisation problems. This article first discusses its implementation, followed by examples indicating the inconveniences which may appear when dealing with putting genetic algorithm into practice. When searching for the best solution, genetic algorithm inspects areas with the higher probability of containing a globally optimal solution. Schema theorem tries to explain the mechanics behind genetic algorithm, but it cannot be used for the analysis of its convergence properties. For this purpose, finite homogeneous Markov chains need to be applied. The article comments on convergence of canonical genetic algorithm and two of its versions.

1. Uvod

V začetku 60-ih let se je z revolucionarnimi idejami Alana Turinga močno povečalo zanimanje za računalniške simulacije bioloških sistemov. Razvili so se evlucijski algoritmi, katerih delovanje temelji na zakonih biološke evolucije. Pri reševanju optimizacijskih problemov namreč uporabljajo mehanizme, kot so selekcija, križanje in mutacija. Omembe vredna je tudi vpeljava naključnosti, zaradi česar jih uvrščamo med stohastične optimizacijske metode. Obstaja več vrst evlucijskih algoritmov. Mednje spadajo evlucijske strategije (ES), evlucijsko programiranje (EP), genetsko programiranje (GP) in genetski algoritmi (GA). Ti so doživeli razcvet v poznih 70-ih in zgodnjih 80-ih letih s profesorjem Johnom H. Hollandom z Univerze v Michiganu, ki je leta 1975 izdal knjigo *Adaptation in natural and artificial systems* [3]. Omembe vredna je tudi knjiga njegovega učenca, Davida E. Goldberga, z naslovom *Genetic algorithms in search, optimization and machine learning* [2].

Zaradi njihove splošnosti se lahko evlucijske algoritme uporabi pri reševanju najrazličnejših problemov. Nekateri izmed njih so problem trgovskega potnika, problem transporta, sestavljanje urnika, 0/1 nahrbtnik, krmiljenje mobilnih robotov, procesiranje slik in prepoznavanje vzorcev, podatkovno rudarjenje itd. Kljub širokemu spektru uporabe jih ni priporočljivo uporabiti pri reševanju problemov, za katere že poznamo hevristične algoritme reševanja. Praktični so predvsem takrat, ko se želimo hitro dokopati do rešitve, ki je blizu optimalni; bodisi nas natančna vrednost ne zanima bodisi želimo zgolj grobo oceno, ki jo bomo uporabili za nadaljnje računanje točnejše rešitve.

V naslednjem razdelku bo različica genetskega algoritma uporabljena pri iskanju globalnega maksimuma zvezne realne funkcije dveh spremenljivk h . Običajno bi ta problem reševali z gradientno metodo ali s pregledom vrednosti na robu območja in z iskanjem ničel parcialnih odvodov h_x in h_y . Pri tem bi lahko uporabili Newtonovo metodo, pri kateri potrebujemo približek za ničlo. V primeru, da funkcije ne poznamo najbolje, si lahko pri oceni približka pomagamo z genetskim algoritmom.

Za boljšo predstavo delovanja genetskih algoritmov si velja pogledati enostaven primer iz sveta biologije: evolucijo populacije zajcev. Recimo, da se na travniku pase skupina zajcev. Nekateri zajci so hitrejši od drugih in imajo zato večjo možnost preživetja, kar pomeni, da lahko lažje uidejo lisici. V očeh *naravne selekcije* imajo ti zajci večjo možnost preživetja. Naravna selekcija torej skrbi, da se geni manj uspešnih osebkov ne prenesejo v naslednjo generacijo. Kadar zajci niso preveč zaposleni z bežanjem pred lisicami ali s prehranjevanjem, se razmnožujejo. Pri tem se njihovi geni med seboj *križajo*, lahko pa pride tudi do naključnih *mutacij*. Te skrbijo za pestrost genskega zapisa. Na ta način se namreč lahko skoti zelo hiter zajec, ki prekaša vse ostale. Ali pa se na primer skoti zajec z izjemnim sluhom, kar je tudi dobrodošla sprememba v genetskem materialu. Nova generacija zajcev je v povprečju boljša od prejšnje, prav tako pa je boljša nova generacija lisic. V nasprotnem primeru bi bile manj uspešne pri lovu, kar bi jih pripeljalo do izumrtja. Populacija zajcev se s tem spet znajde na začetku in evolucija se nadaljuje.

Omenjen primer evolucije populacije zajcev se lahko strne v psevdokodo genetskega algoritma, ki je povzeta po [1].

Algoritem 1 Splošen genetski algoritem.

Vhod: krmilni parametri $N, p_m, p_c, \text{max_gen}$

Izhod: rešitev predstavljena v obliki kromosomov

```

1: function GA
2:    $t \leftarrow 0$ 
3:    $P(t) \leftarrow$  inicializiraj( $N$ )
4:    $F(t) \leftarrow$  ovrednoti  $P(t)$ 
5:   while ustavitveni pogoji niso izpolnjeni do
6:      $\bar{P}(t) \leftarrow$  selekcija  $P(t)$ 
7:      $\tilde{P}(t) \leftarrow$  križanje  $\bar{P}(t)$ 
8:      $P(t+1) \leftarrow$  mutacija  $\tilde{P}(t)$ 
9:      $F(t+1) \leftarrow$  ovrednoti  $P(t+1)$ 
10:     $t \leftarrow t+1$ 
11:  end while
12: end function

```

Iz algoritma 1 je razvidno, da vsak genetski algoritem sestavljajo naslednje komponente:

- krmilni parametri: velikost populacije N , verjetnost križanja p_c , verjetnost mutacije p_m , ustavitveni pogoji, itd.,
- elementi prostora možnih rešitev morajo biti zapisani v obliki kromosomov,
- uspešnostna funkcija, ki vsakemu osebk v populaciji pripiše njegovo uspešnost,
- genetski operatorji: selekcija, križanje, mutacija.

2. Komponente genetskega algoritma

V tem razdelku bodo opisani vsi sestavni deli genetskega algoritma, poudarek pa bo na komponentah kanoničnega genetskega algoritma, ki bo natančneje opisan na začetku razdelka 3. Pri tem so kot glavni vir uporabljene knjige [1], [5] in [6]. Za lažjo predstavbo bo vzporedno potekalo reševanje optimizacijskega problema

$$\max_{(x,y) \in I} h(x,y), \quad h : I \subseteq \mathbb{R}^2 \rightarrow \mathbb{R},$$

kjer je

$$h(x,y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{-x^2-y^2} - \frac{1}{3}e^{-(x+1)^2-y^2} \quad (1)$$

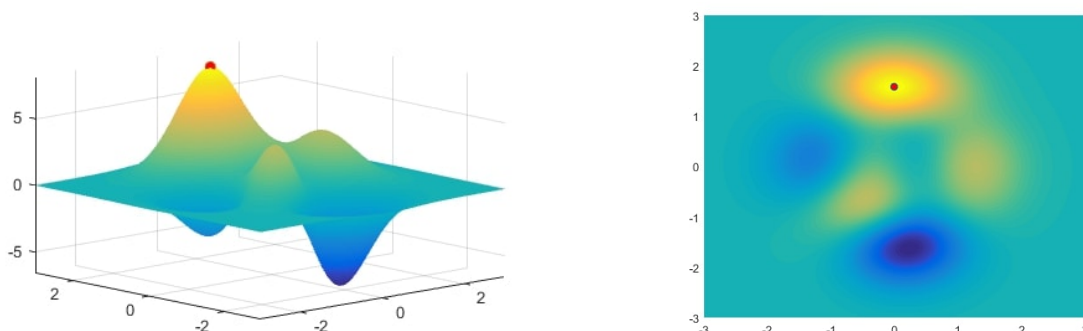
in

$$I = [-3, 3] \times [-3, 3].$$

Slika 1 prikazuje grafa funkcije h z označenim globalnim maksimumom

$$h(x,y) = h(-0,0093; 1,5814) = 8,1062,$$

ki je bil izračunan z uporabo Matlabove vgrajene funkcije `fminunc`. Rezultatu se bomo tekom razdelka poskušali čim bolj približati z uporabo kanoničnega genetskega algoritma.



Slika 1. Grafa funkcije h , podane v (1), z označenim globalnim maksimumom.

2.1 Predstavitveni problem

Z genetskimi algoritmi se lahko lotimo reševanja številnih problemov oblike

$$\max_{x \in A} f(x), \quad f : A \rightarrow B.$$

Množica A je lahko množica realnih števil, naravnih števil, množica mest ali pa na primer množica predmetov, ki jih želimo spraviti v nahrbtnik. Pomembno je, da so elementi množice A zapisani v obliki podobni kromosomom. Po navadi je to eden izmed težjih korakov pri snovanju genetskega algoritma. Problemu pravimo *predstavitveni problem*. Pri tem se moramo domisliti funkcije oziroma algoritma, ki sprejema elemente množice A in vrača elemente množice kromosomov K , s katerimi lahko genetski algoritem dejansko operira. K problemu trgovskega potnika je možno pristopiti na več načinov: pot lahko predstavimo vektorsko, s permutacijami ali pa z matriko. Pri problemih kot so 0/1 nahrbtnik, minimalno povezan graf, iskanje maksimuma realne funkcije pa uporabimo predstavitev z dvojiškimi nizi.

Tudi pri iskanju globalnega maksimuma funkcije h bo uporabljena predstavitev z dvojiškimi nizi. Lahko bi uporabili standardno dvojiško kodiranje. Izkaže pa se, da taka predstavitev ni najprimernejša, saj lahko že majhna razlika med elementoma v množici A pomeni veliko razliko v prostoru predstavitev K .

Primer 1. Naj bosta 7 in 8 elementa množice A . Pri uporabi navadnega dvojiškega zapisa dobimo naslednji predstavitvi:

$$(7)_2 = 0111,$$

$$(8)_2 = 1000.$$

Evklidska razdalja med številoma 7 in 8 je 1, medtem ko je Hammingova razdalja med njunima dvojiškima predstavitevima enaka 4. Ta je namreč definirana kot število mest, na katerih se dvojiška niza X in Y razlikujeta: $H(X, Y) := |\{i; x_i \neq y_i\}|$. Problemu se izognemo z uporabo Grayeve kode [5], pri kateri se zaporedna dvojiška niza razlikujeta zgolj v enem bitu. Številoma 7 in 8 z uporabo Grayeve kode pripadata naslednja dvojiška niza:

$$(7)_G = 0100,$$

$$(8)_G = 1100.$$

Predpostavimo, da imamo opravka z realno funkcijo f več spremenljivk, kar pomeni, da moramo z dvojiškim nizom predstaviti n -terice $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$. Najprej moramo določiti natančnost d dobljene rešitve. Vsako spremenljivko predstavimo z dvojiškim nizom dolžine ℓ_i , končni dvojiški niz pa bo zlepljenka dobljenih posameznih dvojiških nizov dolžine $\ell = \sum_{i=1}^n \ell_i$. Dolžina posameznega dvojiškega niza ℓ_i je odvisna od dolžine intervala, na katerem se spremenljivka nahaja, in od števila točnih decimalnih mest d , ki jih zahtevamo. Če želimo, da je spremenljivka x_i natančna na d decimalnih mest, moramo njen interval $[\alpha_i, \beta_i]$ razdeliti na $(\beta_i - \alpha_i) \cdot 10^d$ delov. Dolžino dvojiškega niza ℓ_i določimo na podlagi neenačbe

$$2^{\ell_i-1} \leq (\beta_i - \alpha_i) \cdot 10^d \leq 2^{\ell_i} - 1.$$

Enačba za pretvorbo dvojiškega niza $b_{\ell_i-1}b_{\ell_i-2}\dots b_0$ v realno vrednost spremenljivke x_i je

$$x_i = \alpha_i + \sum_{j=0}^{\ell_i-1} b_j 2^j \frac{\beta_i - \alpha_i}{2^{\ell_i} - 1}. \quad (2)$$

V našem primeru sta spremenljivki funkcije h omejeni in ležita na intervalu $[-3, 3]$, $(x, y) \in [-3, 3] \times [-3, 3]$. Če želimo oceno končne rešitve natančno na 4 decimalna mesta, moramo posamezen interval $[-3, 3]$ razdeliti na vsaj $(3 - (-3)) \cdot 10^4$ delov. Dolžina posameznega dvojiškega niza mora biti 16, saj velja

$$32768 = 2^{15} \leq 60000 \leq 2^{16} - 1 = 65535.$$

Ker je celotna rešitev oblike $(x, y) \in [-3, 3] \times [-3, 3]$, bo dolžina celotnega dvojiškega niza 32, pri čemer prvih 16 bitov pripada zapisu x , drugih 16 pa predstavlja dvojiški zapis za koordinato y . Očitno je, da dvojiški niz s samimi ničlami predstavlja koordinati $(-3, -3)$, dvojiški niz sestavljen iz samih enic pa predstavlja koordinati $(3, 3)$.

Naj bo $b_{31}b_{30}\dots b_0$ dvojiški niz, ki bi ga radi zapisali kot par realnih števil (x, y) , pri čemer velja $(x, y) \in [-3, 3] \times [-3, 3]$. Najprej je potrebno dvojiški niz razdeliti na dva enaka dela $B_1 = b_{31}b_{30}\dots b_{16}$ in $B_2 = b_{15}b_{14}\dots b_0$. Na dvojiških nizih B_1 in B_2 uporabimo formulo (2) in dobimo

$$x = -3 + \sum_{i=0}^{15} b_{i+16} 2^i \cdot \frac{6}{2^{16} - 1},$$

$$y = -3 + \sum_{i=0}^{15} b_i 2^i \cdot \frac{6}{2^{16} - 1}.$$

Primer 2. Za primer si pogledjmo niz

$$00111011000010101101011111001010,$$

ki ga želimo pretvoriti v par realnih števil (x, y) . Razdelimo ga na dva dela:

$$\begin{aligned}(b_{31}b_{30} \dots b_{16}) &= 0011101100001010, \\ (b_{15}b_{14} \dots b_0) &= 1101011111001010.\end{aligned}$$

Sledi

$$\begin{aligned}x &= -3 + (0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + \dots + 0 \cdot 2^{15}) \frac{6}{2^{16} - 1} = -1,6163 \\ y &= -3 + (0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + \dots + 1 \cdot 2^{15}) \frac{6}{2^{16} - 1} = 2,0576.\end{aligned}$$

2.2 Začetna populacija

Začetna množica je sestavljena iz N naključno generiranih možnih rešitev. Množici pravimo tudi populacija, njenim elementom pa osebki. Oblika njihovega zapisa naj bi spominjala na kromosom, saj bomo na njih izvajali podobne operacije, kot jih izvaja evolucija na osebkih v naravi.

2.3 Uspešnostna funkcija

Kot že samo ime pove, uspešnostna funkcija

$$\text{eval} : \{\text{osebki}\} \rightarrow (0, \infty)$$

meri uspešnost posameznega osebka. V našem primeru bi to lahko bila kar translirana funkcija $h(x, y)$, npr.

$$\text{eval}(x, y) = h(x, y) + 7.$$

Z uspešnostno funkcijo lahko natančno določimo uspešnost posameznega osebka, ki vpliva na verjetnost napredovanja v naslednjo generacijo. Formula za *verjetnost preživetja posameznega osebka* o_i ob času t se glasi

$$p_i = p_i(t) = \frac{\text{eval}(o_i)}{F(t)}, \quad (3)$$

kjer je

$$F(t) = \sum_{i=1}^N \text{eval}(o_i)$$

uspešnost celotne populacije ob času t . Iz formule je razvidno, zakaj mora uspešnostna funkcija slikati v prostor pozitivnih števil. V nasprotnem primeru bi bila lahko verjetnost preživetja posameznega osebka negativna.

Primer 3. Naj bosta o_1 in o_2 edina osebka v populaciji ob času t ,

$$\begin{aligned}o_1 &= 00111011000010100101011111001010, \\ o_2 &= 01010101010101110101011011110101.\end{aligned}$$

Na način, ki je opisan v razdelku 2.1, izračunamo njune realne vrednosti

$$\begin{aligned}o_1 &= (-1,6163; -0,9424), \\ o_2 &= (-0,9998; -0,9619).\end{aligned}$$

Uspešnosti osebkov sta

$$\begin{aligned}\text{eval}(o_1) &= \text{eval}((-1,6163; -0,9424)) = h(-1,6163; -0,9424) + 7 = 7,0064, \\ \text{eval}(o_2) &= \text{eval}((-0,9998; -0,9619)) = h(-0,9998; -0,9619) + 7 = 8,9091.\end{aligned}$$

Osebek o_2 je uspešnejši od osebka o_1 , zato bo njegova verjetnost preživetja večja:

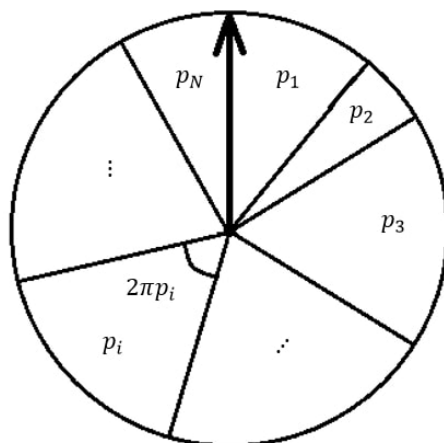
$$p_1 = \frac{\text{eval}(o_1)}{F(t)} = \frac{7,0064}{7,0064 + 8,9091} = 0,4402,$$

$$p_2 = \frac{\text{eval}(o_2)}{F(t)} = \frac{8,9091}{7,0064 + 8,9091} = 0,5598.$$

2.4 Genetski operatorji

2.4.1 Selekcija

Selekcija ne skrbi za pestrost populacije, temveč na podlagi uspešnosti izbira osebke, ki bodo sestavljali novo generacijo. Obstaja več vrst selekcije. Ena izmed njih je *proporcionalna selekcija*, pri kateri je verjetnost kopiranja osebka v novo generacijo sorazmerna z njegovo uspešnostjo. V angleščini se takšni selekciji reče *roulette wheel selection*, saj si lahko predstavljamo, da se osebki nahajajo na vrtečem se ruletnem kolesu (glej sliko 2). Razlika med kolesom pri ruleti in kolesom v našem primeru je v velikosti predalčkov. V našem primeru predalčki niso nujno enako veliki, pri čemer večji predalček pomeni večjo verjetnost preživetja p_i . Središčni kot i -tega osebka tako meri $2\pi p_i$. Ko zavrtimo kazalec kolesa, se bo ta ustavil pri nekem kotu $2\pi \cdot \text{rand}$, kjer je rand enakomerno porazdeljeno naključno število na intervalu $[0,1]$.



Slika 2. Ruletno kolo pri proporcionalni selekciji.

Ta postopek ne izključuje najslabših osebkov. Tudi ti imajo možnost preživetja, vendar z zelo majhno verjetnostjo. Prav tako ni nujno, da bodo izbrani le najboljši osebki. Kolo se mora zavrteti N -krat. Nekateri osebki so lahko izbrani večkrat, spet drugi pa niso izbrani nikoli. Pri proporcionalni selekciji lahko pride do prehitre konvergence k lokalnemu maksimumu. To se zgodi v primeru, ko imamo v populaciji neprimerno uspešnejši osebek od ostalih, ki pa ne predstavlja globalno optimalne rešitve problema. Temu se lahko izognemo z uporabo drugih vrst selekcije, kot sta na primer elitistična in turnirska selekcija.

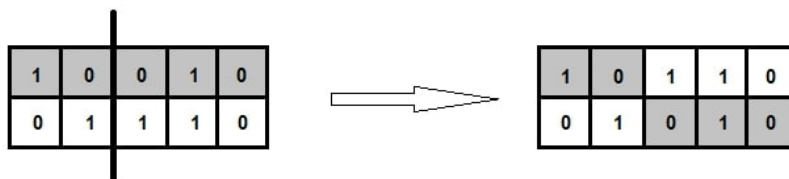
Elitistična selekcija pomeni, da ohranimo neka (po navadi enega) najboljših osebkov. Tem osebkom se ni potrebno boriti za obstoj in se samodejno kopirajo v naslednjo generacijo.

Pri *turnirski selekciji* najprej naključno iz populacije izberemo k osebkov, kjer k predstavlja velikost turnirja. Med k izbranimi osebki, se nato za uvrstitev v nadaljnjo generacijo izbere najuspešnejšega. Opisani postopek se ponovi N -krat.

V primeru iskanja globalnega maksimuma funkcije h sta uporabljeni tako proporcionalna kot tudi elitistična proporcionalna selekcija. Primerjava obeh selekcij se nahaja v razdelku 2.6.

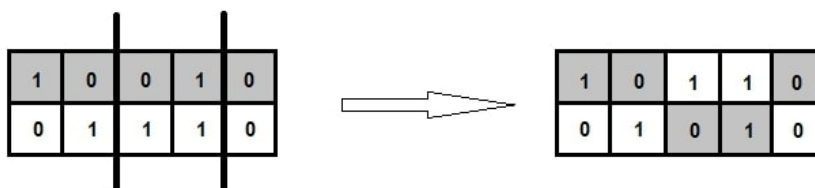
2.4.2 Križanje

S selekcijo, ki je opisana v 2.4.1, se izbere osebk, iz katerih se nato ustvari generacija potomcev. Pri križanju je najprej potrebno naključno izbrati par osebkov, ki so potencialni starši potomcev. Pripiše se jima enakomerno porazdeljeno naključno število rand iz intervala $[0,1]$. Osebk se križata, če velja $\text{rand} \leq p_c$, kjer je p_c vnaprej določena *verjetnost križanja* dveh osebkov. V nasprotnem primeru se brez posledic premakneta na naslednji korak - mutacijo. Tako kot selekcije je tudi križanja več vrst. V primeru računanja maksimuma funkcije h je uporabljeno enotočkovno križanje. To pomeni, da izbranemu paru, ki se bo križal, naključno določimo mesto križanja k ; $k \in \{1, 2, \dots, \ell_i - 1\}$. Na tem delu se oba starša razdelita in si med sabo izmenjata dela (glej sliko 3).



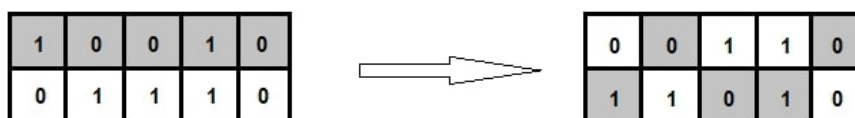
Slika 3. Enotočkovno križanje.

Dvotočkovno križanje je podobno enotočkovnemu, s to razliko, da se starša razdelita na tri dele. Pri tem si izmenjata srednji del in s tem tvorita dva nova potomca (glej sliko 4).



Slika 4. Dvotočkovno križanje.

Poznamo še uniformno križanje, pri katerem obravnavamo vsak bit posebej. Lahko si predstavljamo, da za vsak bit vržemo kovanec (pošten ali nepošten). Izid meta kovanca odloča o tem, kateremu izmed dveh potomcev se deduje posamezen bit prvega starša (glej sliko 5).



Slika 5. Uniformno križanje.

2.4.3 Mutacija

Operator mutacije deluje na vsakem bitu posebej, pri čemer spremeni vrednost posameznega bita z vnaprej določeno *verjetnostjo mutacije* p_m . Vsakemu bitu v osebku se določi enakomerno porazdeljeno naključno število rand_i iz intervala $[0,1]$. V primeru, da je $\text{rand}_i \leq p_m$, bit spremeni svojo vrednost iz 0 v 1 oziroma iz 1 v 0. V nasprotnem primeru bit ostane nespremenjen.

Primer 4. Naj bo $p_m = 0,05$ verjetnost mutacije in 001110 dvojiški niz, ki mu pripišemo vektor v naključnih števil rand_i med 0 in 1, $i = 1, 2, \dots, 6$.

$$v = [r_1, r_2, \dots, r_6] = [0,2769; \mathbf{0,0462}; 0,0971; 0,8235; 0,6948; \mathbf{0,0344}].$$

Vidimo, da je $r_i \leq p_m$ za $i \in \{2, 6\}$, zato bo mutacija delovala na drugem in šestem bitu. Nov dvojiški niz bo tako oblike **011111**.

2.5 Krmilni parametri

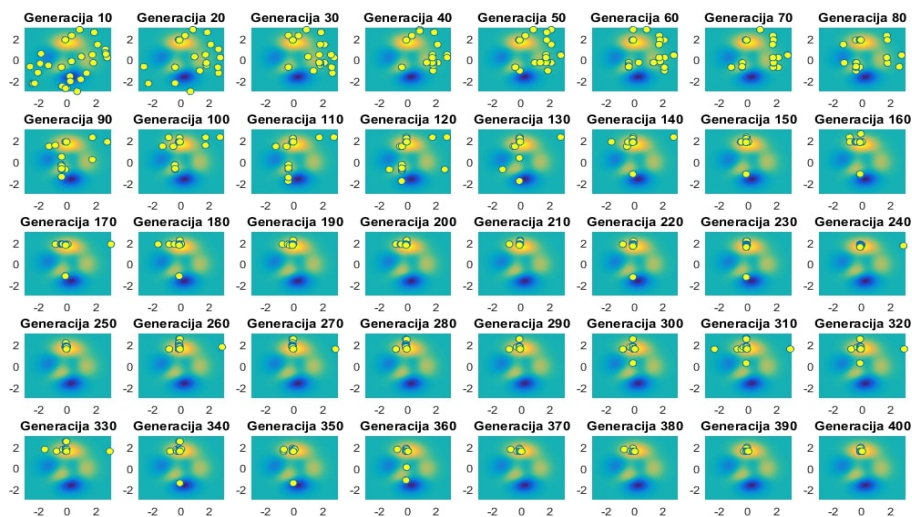
Krmilni parametri vplivajo na hitrost delovanja genetskega algoritma in na kakovost rešitve. Eden izmed njih je *velikost populacije* N . Z večanjem populacije algoritem preiskuje večji del prostora rešitev, zaradi česar se zmanjšuje verjetnost, da bo algoritem obtičal v lokalnem maksimumu. Po drugi strani pa večje število osebkov pomeni večjo računsko zahtevnost in posledično počasnejše delovanje algoritma. Velikost populacije naj se tekom izvajanja algoritma ne bi spreminjala, lahko pa tudi se, če se za to odločimo. Naslednja krmilna parametra se nanašata na križanje in mutacijo. Če bi bil vpliv mutacije prevelik, bi lahko uspešne osebkke, ki so bili ustvarjeni s križanjem, pokvarili z mutacijo. Premajhna verjetnost mutacije pa pomeni manjšo pestrost genskega zapisa v primerjavi z začetno populacijo. S tem bi bila končna rešitev preveč odvisna od izbire začetne populacije. Podobno velja za verjetnost križanja dveh osebkov p_c . Manjša verjetnost križanja pomeni zgolj prenos osebkov iz populacije ob času t v populacijo ob času $t + 1$, kar pa spet pomeni manjšo pestrost genskega zapisa. Poleg ravnokar naštetih parametrov moramo določiti tudi *ustavitvene pogoje*. Genetski algoritem lahko ustavimo po določenem številu preteklih generacij. Druga možnost je spremljanje uspešnosti najboljšega osebka v vsaki generaciji. Ko se uspešnost ob času $t + 1$ ne razlikuje za več kot konstanto ϵ od uspešnosti ob času t (gledano relativno), si ta podatek zapomnimo. Če se ta pojav ponovi večkrat zapored, izvajanje genetskega algoritma ustavimo. Obstajajo tudi drugi ustavitveni pogoji, ki se razlikujejo od problema do problema.

2.6 Povzetek rezultatov

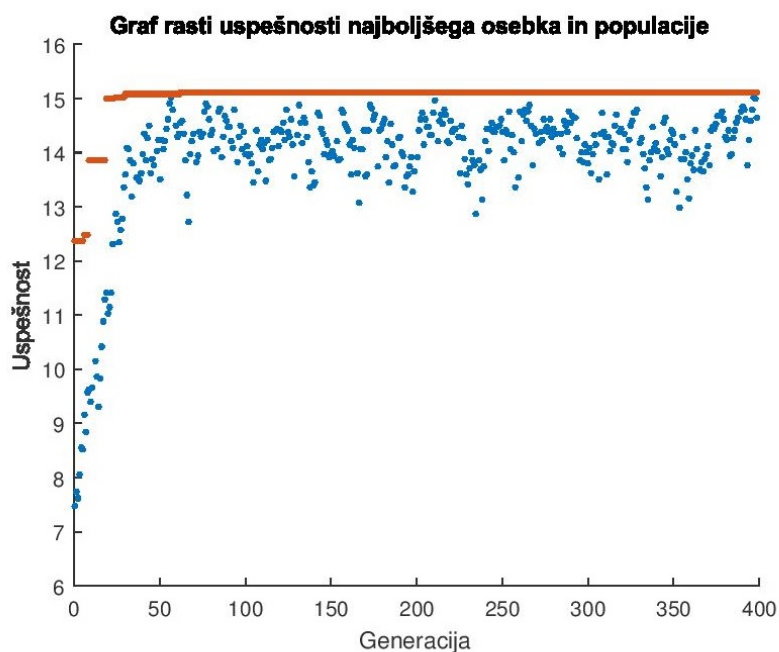
Prednost genetskega algoritma je pregledovanje obsežnega prostora možnih rešitev. Algoritem skuša v množici kandidatov za rešitev izvesti inteligentno iskanje. Istočasno je to tudi njegova slabost, saj lahko ob slabi izbiri krmilnih parametrov namesto globalne najde lokalno optimalno rešitev. Na sledečih primerih bodo predstavljene nekatere pasti in zanimiva opažanja povezana z delovanjem algoritma in točnostjo rešitve.

Primer 5 (Globalni maksimum funkcije h). Skozi razdelek 2 je bil predstavljen vpliv posameznih komponent na delovanje genetskega algoritma. Pri reševanju optimizacijskega problema (1) je bila uporabljena elitistična proporcionalna selekcija, enotočkovno križanje z verjetnostjo $p_c = 0,25$ in verjetnostjo mutacije $p_m = 0,007$. Število osebkov v populaciji je znašalo $N = 40$, algoritem pa je prenehal delovati po določenem številu generacij, $\text{max_gen} = 400$. Slika 6 prikazuje migracijo osebkov po grafu funkcije h in njihovo ustalitev pri globalnem maksimumu. Iz slike 7 se vidi razvoj najboljšega osebka (oranžno) in populacije (modro) skozi generacije.

Uporabnost in učinkovitost kanoničnega genetskega algoritma



Slika 6. Migracija osebkov pri računanju maksimuma funkcije h , podane v (1).



Slika 7. Rast uspešnosti najboljšega osebka in populacije skozi generacije.

Nekaj rešitev, ki jih je vrnil genetski algoritem, je zapisanih v tabeli 1.

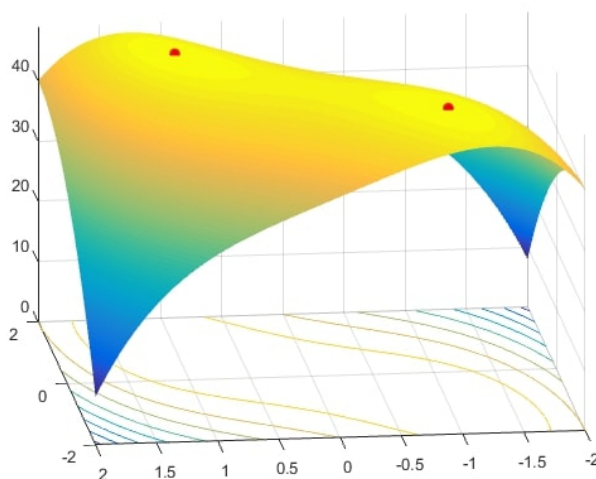
Tabela 1. Rezultati desetih zaporednih zagonov genetskega algoritma pri računanju maksimuma funkcije h .

x	y	$h(x,y)$
-0,0079	1,5829	8,1062
0,0015	1,5821	8,1053
-0,0094	1,5813	8,1062
0,0000	1,5828	8,1055
-0,0123	1,5814	8,1061
-0,0096	1,5821	8,1062
-0,0091	1,5828	8,1062
-0,0118	1,5821	8,1062
0,0000	1,5814	8,1055
-0,0093	1,5814	8,1062

Primer 6 (Funkcija z več globalnimi maksimumi). Pri računanju globalnega maksimuma funkcije z več kot enim globalnim maksimumom genetski algoritem najde samo enega naenkrat. Naj bo z funkcija, katere maksimum iščemo:

$$z(x,y) = -2x^2 + 4xy - y^4 + 48. \quad (4)$$

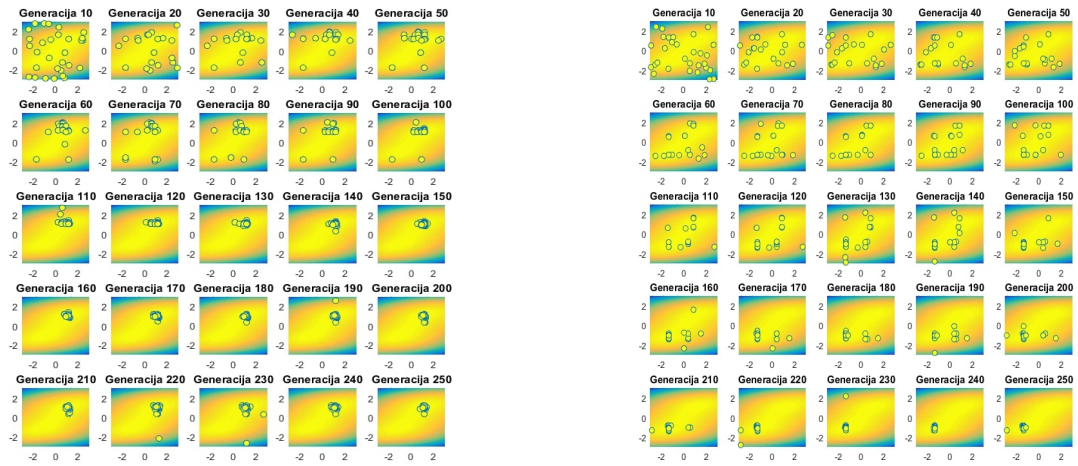
Funkcija je prikazana na sliki 8.



Slika 8. Graf funkcije z , podane v (4), z označenima maksimumoma.

Funkcija z ima dva globalna maksimuma, ki sta dosežena v točkah $(-1, -1)$ in $(1, 1)$, kjer zavzame vrednost $z(-1, -1) = z(1, 1) = 49$. Kot je razvidno iz slike 9, genetski algoritem najde enega ali drugega, vendar nikoli obeh hkrati.

Uporabnost in učinkovitost kanoničnega genetskega algoritma



(a) Maksimum odkrit v točki $(1, 1)$.

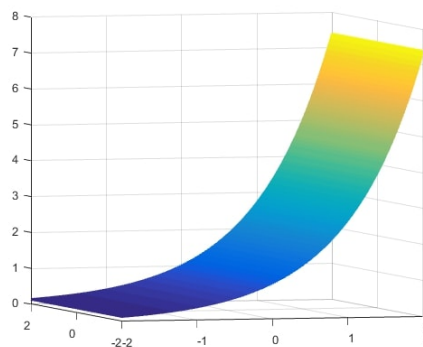
(b) Maksimum odkrit v točki $(-1, -1)$.

Slika 9. Iskanje maksimuma funkcije z , podane v (4), z dvema globalnima maksimumoma.

Podoben sklep velja za vse funkcije z več kot enim globalnim maksimumom. Kot skrajni primer pogledjmo iskanje maksimuma funkcije

$$g(x, y) = e^x, \quad (5)$$

pri čemer pregledujemo območje $[-2, 2] \times [-2, 2]$.



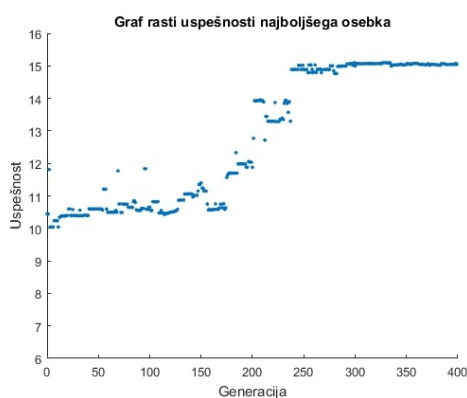
Slika 10. Graf funkcije g , podane v (5).

Iz slike 10 je očitno, da so globalni maksimumi doseženi na daljici med točkama $(2, -2)$ in $(2, 2)$. V tabeli 2 so prikazani globalni maksimumi dobljeni z 10-kratnim zagonom genetskega algoritma. Vsakokrat je algoritem našel novo vrednost koordinate y oziroma nov globalni maksimum.

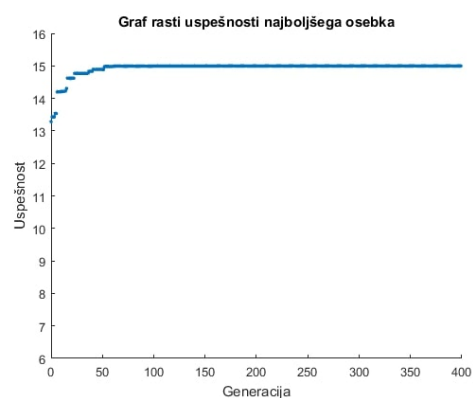
Tabela 2. Rešitve desetih zaporednih zagonov genetskega algoritma pri računanju maksimuma funkcije g .

x	y	$g(x,y)$
2,0000	0,3462	7,3891
2,0000	-1,3204	7,3891
2,0000	-1,5133	7,3891
2,0000	-1,7969	7,3891
2,0000	1,3485	7,3891
2,0000	-0,4038	7,3891
1,9999	1,5142	7,3882
2,0000	0,1545	7,3891
2,0000	0,7781	7,3891
1,9999	0,9711	7,3886

Primer 7 (Primerjava proporcionalne in elitistične proporcionalne selekcije). Poglejmo si spreminjanje uspešnosti najboljšega osebkov v odvisnosti od števila generacij. Na sliki 11(a) je uporabljena proporcionalna selekcija, na desni pa elitistična proporcionalna selekcija. Preostali krmilni parametri se med seboj ne razlikujejo.



(a) Uporabljen proporcionalna selekcija.



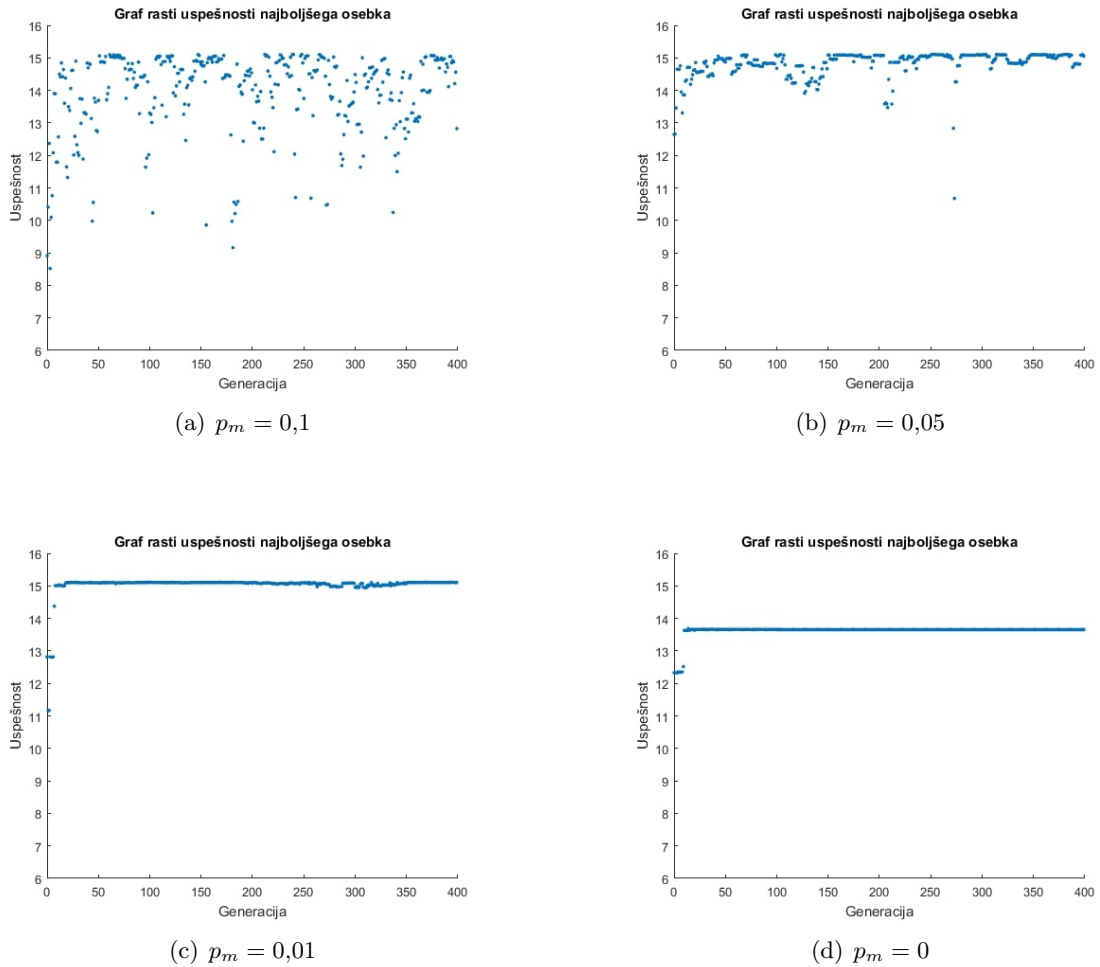
(b) Uporabljen elitistična proporcionalna selekcija.

Slika 11. Razlika v razvojih najuspešnejšega osebkov pri proporcionalni in elitistični proporcionalni selekciji.

Pri uporabi elitistične selekcije je opaziti stalnost in dejanski napredek osebkov, saj najuspešnejši osebek napreduje v naslednjo generacijo z verjetnostjo 1. Z uporabo zgolj proporcionalne selekcije pa se napredek lahko iz generacije v generacijo izgubi.

Primer 8 (Izbira krmilnega parametra p_m). Mutacija vnaša pestrost v genetski zapis. Ob generiranju začetne populacije (oziroma v katerikoli od sledečih generacij) se lahko zgreši določen bit informacije, ki je nujno potreben pri iskanju optimalne rešitve. Brez mutacije bi bilo nemogoče vnesti ta manjkajoči delček genskega zapisa v populacijo. Če pestrosti genskega zapisa ni, bodo vse naslednje generacije vsebovale približno enake osebkov kot so v začetni populaciji. To lahko vidimo na sliki 12 (d), kjer je $p_m = 0$. Osebkov se ne razlikujejo od tistih v začetni populaciji, zato tudi uspešnost najboljšega osebkov ostaja praktično nespremenjena in ne raste proti globalnemu maksimumu. Druga skrajnost je prevelika pestrost. To je jasno vidno na primeru, ko izberemo $p_m = 0,1$ (slika 12 (a)). Pri tem algoritmu sploh ne damo možnosti, da bi skonvergirala, saj neprestano naključno spreminjamo osebkov. Na sliki 12(c) je mutacije ravno prav, da ne moti razvoja algoritma in še vedno vsake toliko časa ustvari super osebek.

Uporabnost in učinkovitost kanoničnega genetskega algoritma

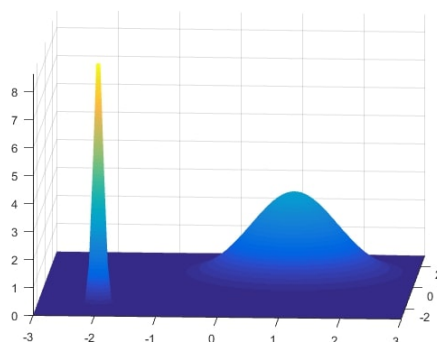


Slika 12. Razvoj najuspešnejšega osebkca pri različnih vrednostih parametra p_m .

Primer 9 (Najden lokalni maksimum). V tem primeru je obravnavana funkcija

$$s(x,y) = 10e^{-1000(x+2)^2-1000(y+2)^2} + 3e^{-(x-1)^2-(y-1)^2}, \quad (6)$$

pri čemer iščemo maksimum na območju $[-3,3] \times [-3,3]$. Njen graf je prikazan na sliki 13.

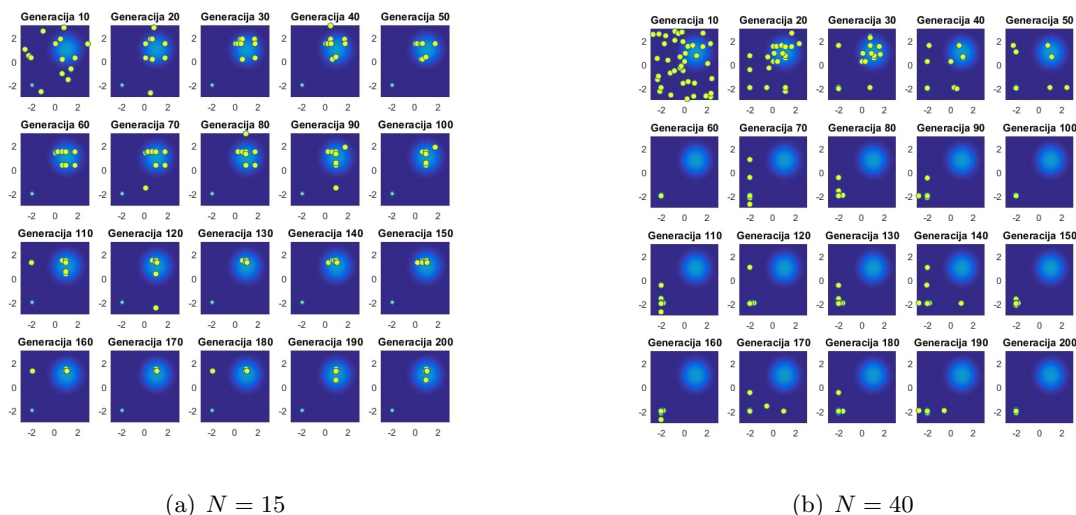


Slika 13. Graf funkcije s , podane v (6).

Iz slike 13 je razvidno, da vrednosti funkcije s hitro naraščajo proti globalnemu maksimumu v njegovi relativno majhni okolici. Okolica lokalnega maksimuma, kjer se vrednosti funkcije približujejo maksimalni vrednosti, je večja. Verjetnost, da se bodo uspešni osebki nahajali v bližini

lokalnega maksimuma, je potemtakem večja od verjetnosti, da kakšen izmed osebkov zadane okolico globalnega maksimuma. Zgodi se, da genetski algoritem za rešitev prepozna lokalni maksimum. Tej napaki se izognemo tako, da povečamo število osebkov v populaciji in s tem povečamo verjetnost, da se bo kateri izmed osebkov nahajal v neposredni bližini globalnega maksimuma.

Na sliki 14 je prikazan potek iskanja rešitve pri velikosti populacije $N = 15$ (levo) in $N = 40$ (desno). Pri izbiri večjega števila osebkov v populaciji, nam genetski algoritem vrne globalni maksimum, pri manjšem številu pa zgolj lokalni maksimum.



Slika 14. Vpliv velikosti populacije na kakovost rešitve.

3. Konvergenca kanoničnega genetskega algoritma

V tem razdelku bo opisano teoretično ozadje za delovanjem in konvergenco različice genetskega algoritma - kanoničnega genetskega algoritma. To je genetski algoritem, za katerega mora veljati:

- dvojiška predstavitev osebkov,
- osebki s končno in vnaprej znano dolžino,
- proporcionalna selekcija,
- enotočkovno križanje,
- operator mutacije deluje na vsak bit v populaciji z verjetnostjo p_m .

Primer uporabe kanoničnega genetskega algoritma je bil prikazan že v razdelku 2, kjer smo iskali maksimum zvezne realne funkcije dveh spremenljivk.

V delovanje algoritma nas bo popeljal izrek o shemah, ki ga je formuliral John Holland v letu 1970 [3], pri tem pa nas bo spremljal pojem pričakovane vrednosti.

Definicija 1. Naj bo X diskretna slučajna spremenljivka s porazdelitvijo

$$X \sim \begin{pmatrix} x_1 & x_2 & \dots \\ p_1 & p_2 & \dots \end{pmatrix},$$

kjer je $p_i = P(X = x_i)$ in $\sum_i p_i = 1$. Potem se *pričakovano vrednost* izračuna po formuli

$$E(X) = \sum_i x_i P(X = x_i) = \sum_i x_i \cdot p_i.$$

3.1 Izrek o shemah

Preden se posvetimo izpeljavi in razlagi izreka o shemah, se moramo seznaniti s shemami na splošno. Snov za ta razdelek je vzeta predvsem iz [1], [2] in [5]. Shema S je niz znakov 0, 1 in *, pri čemer je * univerzalni simbol in predstavlja 0 ali 1. Shema določa množico vseh možnih dvojiških nizov, ki ustrezajo njenemu zapisu.

Primer 10. Naj bo $S = **10$ shema. Potem je $\{0010, 0110, 1010, 1110\}$ množica vseh pripadajočih dvojiških nizov.

Iz primera 10 je očitno, da shema z m univerzalnimi simboli generira množico dvojiških nizov moči 2^m . Po drugi strani pa poljuben dvojiški niz dolžine ℓ pripada 2^ℓ različnim shemam.

Primer 11. Dvojiški niz 10 z dolžino $\ell = 2$ pripada shemam **, 1*, *0, 10.

V jeziku genetskih algoritmov sheme predstavljajo množice kandidatov za rešitev problema. Z $E(S,t)$ označimo pričakovano število nizov, ki pripadajo shemi S ob času t . V nadaljevanju si bomo pogledali, kako operatorji genetskih algoritmov vplivajo na preživetje sheme, torej v kakšnem razmerju sta $E(S,t)$ in $E(S,t+1)$ po selekciji, križanju in mutaciji. Pred tem moramo navesti še nekaj definicij in oznak, začenši s tremi osnovnimi lastnostmi shem.

Definicija 2. Red sheme S definira specifičnost sheme in je enak številu njenih fiksnih bitov. Označimo ga z $o(S)$.

Definicija 3. Dolžina sheme S je razlika med položajema prvega in zadnjega fiksnega bita ter predstavlja strnjenost informacij v shemi. Označimo jo z $\delta(S)$.

Primer 12. Naj bo $S = *1*01$ shema. Potem je njen red $o(S) = 3$ in njena dolžina $\delta(S) = 3$. V primeru sheme $H = 0****$ dolžina znaša $\delta(H) = 0$.

Definicija 4. Uspešnost sheme S ob času t označimo z $u(S,t)$ in je enaka povprečni vrednosti uspehov vseh pripadajočih elementov sheme, ki se nahajajo v populaciji ob času t , to je

$$u(S,t) = \frac{\sum_{i=1}^n \text{eval}(s_i)}{n}, \quad (7)$$

kjer so s_1, s_2, \dots, s_n osebki v populaciji, ki pripadajo shemi S .

3.1.1 Vpliv operatorjev GA na preživetje sheme

Preživetje sheme S je odvisno od števila osebkov populacije, ki se v danem trenutku t nahajajo v shemi S . Večje število osebkov v trenutni populaciji, ki pripada določeni shemi, pomeni večjo možnost preživetja te sheme v naslednjih generacijah. Vemo namreč, da se sestava populacije nenehno spreminja zaradi delovanja genetskih operatorjev. V nadaljevanju nas bo zanimalo, katere lastnosti sheme pozitivno vplivajo na njeno preživetje.

Vpliv selekcije Zapisali smo formulo (7) za povprečno uspešnost sheme. Na podoben način lahko zapišemo povprečno uspešnost populacije

$$\overline{F(t)} = \frac{F(t)}{N}.$$

S q označimo njuno razmerje

$$q = q(t) = \frac{u(S,t)}{\overline{F(t)}}$$

in ga poimenujmo *uspešnostni količnik*. Grobo rečeno ta predstavlja povprečen delež preživetja sheme v populaciji. Sledi enačba

$$E(S, t + 1) = \frac{E(S, t) \cdot u(S, t)}{F(t)} = E(S, t) \cdot q, \quad (8)$$

ki opisuje zvezo med pričakovanim številom osebkov v populaciji, ki pripadajo shemi S ob času t in $t+1$. Zvezi pravimo *reprodukcijska rast sheme*. Shemi, za katero velja $q > 1$, pravimo nadpovprečna, saj je v povprečju uspešnejša od povprečne uspešnosti populacije. Sledi, da nadpovprečna shema povečuje število pripadajočih dvojiških nizov v naslednjih generacijah, shemi z uspešnostjo manjšo od povprečne uspešnosti populacije pa se manjša število nizov v prihodnjih populacijah.

Sheme, ki so bolj uspešne od povprečja populacije, imajo večjo verjetnost nadaljevanja v naslednjo generacijo. Enačba (8) pokaže, da selekcija povečuje povprečno uspešnost populacije, slabost pa je, da ne prinaša novih potomcev. Za raznolikost potomcev poskrbita križanje in mutacija.

Vpliv križanja

Definicija 5. Shema S *preživi* enotočkovno križanje, če vsaj eden izmed staršev in vsaj eden izmed potomcev pripada shemi S .

Primer 13. Naj bo $S = *10**$ shema. Izbor staršev in mesta križanja odloča o preživetju sheme:

$$\begin{aligned} P_1 = 110|10 \in S & \rightarrow S_1 = 11011 \in S & \rightarrow \text{shema } S \text{ preživi} \\ P_2 = 101|11 \notin S & \rightarrow S_2 = 10110 \notin S & \\ \\ P_1 = 11|010 \in S & \rightarrow S_1 = 11111 \notin S & \rightarrow \text{shema } S \text{ ne preživi} \\ P_2 = 10|111 \notin S & \rightarrow S_2 = 10010 \notin S & \end{aligned}$$

Recimo, da so elementi sheme S dolžine m . Vseh možnih mest za križanje je tako $m - 1$. Na podlagi prejšjega primera je očitno, da shema ne bo preživela križanja na toliko mestih, kolikor znaša njena dolžina $\delta(S)$. Iz tega sledi formula za verjetnost uničenja sheme S ,

$$p_d(S) = \frac{\delta(S)}{m - 1},$$

kot tudi formula za verjetnost preživetja sheme S ,

$$p_s(S) = 1 - \frac{\delta(S)}{m - 1}.$$

Ti dve formuli veljata v primeru, da bomo izbrani niz zagotovo križali z drugim nizom, kar pa ne drži vedno. Vemo, da je izbira osebkov za križanje naključna in odvisna od verjetnosti p_c . Sledi, da je verjetnost preživetja sheme med operacijo križanja enaka

$$p_s(S) = 1 - p_c \cdot \frac{\delta(S)}{m - 1}. \quad (9)$$

Primer 14. Naj bo $S = *10**$. Križanje izvedemo med drugim in tretjim bitom. Kljub temu, da sta to fiksna bita sheme, je shema S preživela.

$$\begin{aligned} P_1 = 11|010 \in S & \rightarrow S_1 = 11001 \in S & \rightarrow \text{shema } S \text{ preživi} \\ P_2 = 00|001 \notin S & \rightarrow S_2 = 00010 \notin S & \end{aligned}$$

Kljub temu, da smo v primeru 14 križanje izvedli na mestu, kjer shema načeloma naj ne bi preživela, se je zgodilo ravno nasprotno. Shema lahko torej preživi, če drugi starš nadomesti del sheme, ki se je izgubil med križanjem.

Zaradi tega predstavlja izraz na desni strani neenačbe (9) zgolj spodnjo mejo za verjetnost preživetja sheme. Sledi neenakost

$$p_s(S) \geq 1 - p_c \cdot \frac{\delta(S)}{m-1}.$$

Združena vpliva selekcije in križanja nam podajata novo formulo za reprodukcijsko rast sheme:

$$E(S, t+1) \geq E(S, t) \cdot \frac{u(S, t)}{F(t)} \cdot \left(1 - p_c \cdot \frac{\delta(S)}{m-1}\right).$$

Iz neenačbe je razvidno, da se bo število osebkov, ki pripadajo kratkim nadpovprečnim shemam, povečevalo.

Vpliv mutacije Shema preživi mutacijo, če njeni fiksni biti ostanejo nespremenjeni, torej ne podležejo mutaciji. Verjetnost, da določen bit preživi, je $1 - p_m$, pri čemer je p_m verjetnost mutacije. Ker je mutacija določenega bita neodvisna od mutacij preostalih bitov in ker shema preživi mutacijo le, če preživijo vsi njeni fiksni biti (število teh določa red sheme $o(S)$), je verjetnost, da določena shema preživi mutacijo, enaka

$$p_s(S) = (1 - p_m)^{o(S)}.$$

Iz tega lahko takoj sklepamo, da imajo sheme z nižjim redom večjo možnost preživetja mutacije. Za p_m je značilno $p_m \ll 1$, iz česar sledi naslednja ocena:

$$p_s(S) \approx 1 - o(S) \cdot p_m.$$

Če združimo sedaj vpliv operatorjev selekcije, križanja in mutacije, dobimo končno neenačbo reprodukcijske rasti sheme:

$$E(S, t+1) \geq E(S, t) \frac{u(S, t)}{F(t)} \left(1 - p_c \cdot \frac{\delta(S)}{m-1}\right) (1 - o(S) \cdot p_m). \quad (10)$$

To nas pripelje do izreka o shemah.

Izrek 1 (Izrek o shemah). *Število osebkov, ki pripadajo kratkim, nadpovprečnim shemam z nizkim redom, se s časom povečuje.*

Izrek zanemari majhno vendar neničelno verjetnost, da je niz, ki bo ustrezal shemi S , možno ustvariti tudi z mutacijami začetnega niza, ki originalno ni pripadal shemi S v prejšnjih generacijah. Zato podaja samo spodnjo mejo pričakovanega števila nizov, ki se bodo ujemali s shemo S v naslednji generaciji.

Iz izreka o shemah lahko sklepamo, da genetski algoritem preiskuje prostor s kratkimi, nadpovprečnimi shemami nizkega reda. Še več kot to, naslednja hipoteza pravi, da je genetski algoritem sposoben takšne sheme sestavljati v uspešnejše.

Hipoteza 2 (Hipoteza o gradnikih). *Genetski algoritem išče optimalno rešitev s sestavljanjem kratkih nadpovprečno uspešnih shem nizkega reda, imenovanih gradniki.*

Izrek o shemah postavlja teoretično osnovo za delovanje genetskih algoritmov, zmotno pa je misliti, da govori o kakovosti dobljenih rešitev. Pogost in napačen sklep je tudi ta, da osebk, ki pripadajo nadpovprečnim shemam, pripomorejo h kreiranju uspešnejših potomcev. Na to nas opominja naslednji primer iskanja igle v senu.

Primer 15. Naj bo preiskovani prostor igla v kupu sena. Uspešnostna funkcija vrne visoko uspešnost v primeru igle in zelo nizko uspešnost v primeru sena. Predpostavimo, da smo iglo že našli in se nahaja v trenutni populaciji. Selekcija bo poskrbela, da se bo množično kopirala v naslednjo populacijo, medtem ko bosta sledeča križanje in mutacija ustvarjala le še več sena. Sheme, katerim pripada igla, bodo povečevale število elementov v generacijah, vendar se uspešnost teh osebkov ne bo večala.

Vse, kar izrek o shemah pravi, je, da se število osebkov, ki pripadajo kratkim nadpovprečnim shemam z nizkim redom, povečuje. Kar pomeni, da se genetski algoritem osredotoča na iskanje rešitev v uspešnejših območjih (nadpovprečnih shemah) in s tem povečujejo verjetnost odkrivanja uspešnejših osebkov. Izrek o shemah torej ne povezuje uspešnosti staršev z uspešnostjo potomcev, čeprav je korelacija nakazana s hipotezo o gradnikih. V nadaljevanju, v razdelku 3.2, bo z uporabo markovskih verig dokazano, da izrek o shemah zgolj komentira delovanje genetskega algoritma, ne moremo pa nič sklepati o njegovi konvergenci.

3.2 Markovske verige in konvergenca

Na začetku tega razdelka bodo predstavljene osnove markovskih verig, pri čemer bomo sledili viru [7] in [8]. Začeli bomo z obravnavo slučajnih procesov z diskretnim časom. Zapisani bodo kot zaporedja slučajnih spremenljivk X_0, X_1, \dots , ki lahko zavzamejo vrednosti iz končne množice stanj S z močjo $|S| = n$.

Definicija 6. Stohastičen proces X_0, X_1, \dots zadošča markovski lastnosti, če zanj velja

$$P(X_m = s_m | X_0 = s_0, X_1 = s_1, \dots, X_{m-1} = s_{m-1}) = P(X_m = s_m | X_{m-1} = s_{m-1}),$$

$s_i \in S$.

Opomba 1. Da stohastičen proces zadošča markovski lastnosti pomeni, da na novo stanje X_m vpliva samo zadnje predhodno stanje X_{m-1} .

V nadaljevanju naj velja, da so stanja v množici S označena z $1, 2, \dots, n$, to je $S = \{1, 2, \dots, n\}$.

Slučajni proces z markovsko lastnostjo imenujemo *markovska veriga z diskretnim časom*. Za stanji $i, j \in S$ naj bo

$$p_{ij}(m) = P(X_m = j | X_{m-1} = i)$$

prehodna verjetnost iz stanja i v stanje j v m -tem koraku. V primeru, da je prehodna verjetnost neodvisna od časa m , to je $p_{ij}(m) = p_{ij}$ za vsak $m \in \mathbb{N}$, pravimo, da je markovska veriga *homogena*. Prehodne verjetnosti homogene končne markovske verige lahko združimo v prehodno matriko P , ki je v primeru n možnih stanj oblike

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}.$$

Zanjo velja, da je $p_{ij} \in [0, 1]$ in $\sum_{j=1}^n p_{ij} = 1, i = 1, 2, \dots, n$. Matrikam s tema dvema lastnostima pravimo *stohastične matrike*. Naj bo π_0 vrstični vektor začetne porazdelitve verjetnosti,

$$\pi_0 = [P(X_0 = j)]_{j=1}^n = [P(X_0 = 1), P(X_0 = 2), \dots, P(X_0 = n)],$$

pri čemer je $P(X_0 = j)$ verjetnost, da se markovska veriga začne v stanju j . S pomočjo te porazdelitve in prehodne matrike pridemo do porazdelitve na prvem koraku $\pi_1 = [P(X_1 = j)]_{j=1}^n = \pi_0 \cdot P$, saj je

$$P(X_1 = j) = \sum_{i=1}^n P(X_1 = j | X_0 = i) \cdot P(X_0 = i) = \sum_{i=1}^n p_{ij} \cdot P(X_0 = i).$$

Porazdelitev verige po k -tem koraku lahko torej zapišemo kot $\pi_k = \pi_0 P^k$. Potemtakem je homogena končna markovska veriga določena s parom (π_0, P) .

Kanonični genetski algoritem si lahko predstavljamo kot markovsko verigo. Množica stanj je oblike $S = \{0,1\}^{\ell \cdot N}$, kjer je N velikost populacije in ℓ dolžina dvojiškega niza, ki predstavlja posamezen osebek. Vsak element $s \in S$ lahko identificiramo z naravnim številom i , ki ustreza danemu dvojiškemu nizu dolžine $\ell \cdot N$, to je $(i)_2 = s$. Pišimo kar $i \in S$. Ker je vsak niz sestavljen iz N nizov dolžine ℓ , definiramo projekcijo $proj_k(i)$, ki vsakemu $i \in S$ priredi k -ti dvojiški segment dolžine ℓ . S to projekcijo določimo posamezne osebe v populaciji, ki jo predstavlja element $i \in S$. V prehodno matriko P zapišemo vse spremembe genov, ki se pojavijo zaradi genetskih operatorjev. Matrika P se na naraven način razcepi v produkt treh stohastičnih matrik K, M in S , $P = KMS$, ki predstavljajo spremembe osebkov zaradi križanja, mutacije in selekcije.

V nadaljevanju bodo homogene končne markovske verige uporabljene za dokaz konvergence različice kanoničnega genetskega algoritma, pri kateri se skozi generacije ohranja najuspešnejši osebek. Prav tako bo dokazano, da osnoven kanonični genetski algoritem ne konvergira h globalnemu optimumu ne glede na krmilne parametre, začetno populacijo in uspešnostno funkcijo. Pred njeno resnejšo obravnavo pa se je potrebno bolje spoznati z mehaniko markovskih verig in zapisati nekatere definicije.

Definicija 7. Nenegativna matrika A je

- *primitivna*, če obstaja tak $k \in \mathbb{N}$, da je A^k pozitivna,
- *reducibilna*, če obstaja permutacijska matrika P , da velja

$$PAP^T = \begin{bmatrix} C & 0 \\ R & T \end{bmatrix},$$

kjer sta matriki C in T kvadratni in je 0 ničelna matrika,

- *ireducibilna*, če ni reducibilna.

Definicija 8. Stohastična matrika A je

- *stabilna*, če ima enake vrstice,
- *stolpično dopustna*, če vsak stolpec vsebuje vsaj en pozitiven element.

Izrek 3 ([4, stran 133]). Naj bo P primitivna stohastična matrika. Potem zaporedje matrik P^k konvergira, ko $k \rightarrow \infty$, k pozitivni stabilni stohastični matriki P^∞ , to je matriki, ki ima vse elemente pozitivne in vse vrstice enake. Pri tem vrstice predstavljajo limitno porazdelitev verige π_∞ , za katero velja, da ni odvisna od začetne porazdelitve verjetnosti π_0 .

Lema 4. Naj bosta A in B stohastični matriki. Potem je tudi $C = AB$ stohastična matrika.

Dokaz. Velja $C = [c_{ij}]_{i,j=1}^n = AB = [\sum_{k=1}^n a_{ik}b_{kj}]_{i,j=1}^n$. Sledi

$$\sum_{j=1}^n c_{ij} = \sum_{j=1}^n \sum_{k=1}^n a_{ik}b_{kj} = \sum_{k=1}^n \left(a_{ik} \sum_{j=1}^n b_{kj} \right) = \sum_{k=1}^n a_{ik} = 1,$$

kar dokazuje, da je $\sum_{j=1}^n c_{ij} = 1$ za vsak i .

Sledi obravnava konvergence, pri čemer sledimo članku [7].

Lema 5. *Naj bodo K , M in S stohastične matrike. Naj za M dodatno velja, da je pozitivna in S stolpično dopustna matrika. Potem je njihov produkt KMS pozitivna matrika.*

Dokaz. Naj bo $A = KM$ in $B = AS$. Ker je K stohastična, obstaja vsaj en pozitiven element v vsaki vrstici. Iz tega sledi, da je $a_{ij} = \sum_{l=1}^n k_{il}m_{lj} > 0$ za vse $i, j \in \{1, \dots, n\}$. Torej je A pozitivna matrika. Na podoben način dobimo, da je $b_{ij} = \sum_{l=1}^n a_{il}s_{lj} > 0$ za vse $i, j \in \{1, \dots, n\}$, ker je matrika S stolpično dopustna. Matrika KMS je torej pozitivna.

Izrek 6. *Prehodna matrika kanoničnega genetskega algoritma $P = KMS$ z verjetnostjo mutacije $p_m \in (0,1)$, verjetnostjo križanja $p_c \in (0,1)$ in proporcionalno selekcijo je primitivna.*

Dokaz. Matrika K je stohastična, saj je posamezen element k_{ij} verjetnost, da se populacija iz stanja i preslika v populacijo v stanju j . Podobno lahko sklepamo tudi za matriki M in S . Verjetnost, da se osebek i z mutacijo preslika v osebek j , je enaka $p_m^{H(i,j)}(1-p_m)^{\ell \cdot N - H(i,j)} > 0$ za vse $i, j \in S$, kjer je $H(i,j)$ Hammingova razdalja med dvojiškima predstavitevama osebkov i in j . Sledi, da je M pozitivna. Verjetnost, da selekcija ne vpliva na populacijo, se da omejiti z

$$s_{ii} \geq \frac{\prod_{k=1}^N \text{eval}(\text{proj}_k(i))}{(\sum_{k=1}^N \text{eval}(\text{proj}_k(i)))^N} > 0$$

za vse $i \in S$, kar pomeni, da je S stolpično dopustna. Iz leme 5 sledi, da je matrika $P = KMS$ pozitivna. Trditev je s tem dokazana, saj je vsaka pozitivna matrika primitivna.

Združeno znanje izrekov 3, 4 in 6 nam pove, da začetna porazdelitev verjetnosti π_0 nima vpliva na obnašanje markovske verige v limiti. Torej je izbira začetne populacije popolnoma poljubna in ne vpliva na končni rezultat. Poglejmo si sedaj natančno definicijo konvergence genetskega algoritma.

Definicija 9. Naj bo $Z_t = \max\{\text{eval}(\text{proj}_k(i_t)), k = 1, \dots, N\}$ zaporedje slučajnih spremenljivk, ki predstavljajo najboljši uspeh v populaciji, podani s stanjem i_t ob času t . Genetski algoritem konvergira h globalnemu optimumu natanko tedaj, ko

$$\lim_{t \rightarrow \infty} P(\{Z_t = \text{eval}^*\}) = 1,$$

pri čemer je $\text{eval}^* = \max\{\text{eval}(b); b \in \{0,1\}^\ell\}$ globalni optimum problema, ki ga rešujemo.

Izrek 7. *Kanonični genetski algoritem s parametri iz izreka 6 ne konvergira h globalnemu optimumu.*

Dokaz. Naj bo $i \in S$ stanje, za katerega velja

$$\max\{\text{eval}(\text{proj}_k(i)); k = 1, \dots, N\} < \text{eval}^*$$

in $\pi_t(i)$ verjetnost, da je genetski algoritem v stanju i ob času t . Očitno velja

$$P(\{Z_t \neq \text{eval}^*\}) \geq \pi_t(i) \iff P(\{Z_t = \text{eval}^*\}) \leq 1 - \pi_t(i).$$

Iz izreka 3 vemo, da verjetnost $\pi_t(i)$ konvergira k $\pi_\infty(i) > 0$. Sledi

$$\lim_{t \rightarrow \infty} P(\{Z_t = \text{eval}^*\}) \leq 1 - \pi_\infty(i) < 1,$$

kar se ne sklada z definicijo konvergence iz 9.

S tem smo dokazali, da osnoven kanonični genetski algoritem ne konvergira h globalni optimalni rešitvi. Sedaj si pogledjmo, kaj se zgodi s konvergenco, če populaciji i dodamo nek super osebek $proj_0(i)$, ki ne bo sodeloval v nadaljnjem procesu evolucije. Vseh možnih super osebkov je 2^ℓ , zato se bo prostor stanj povečal na $S = \{0,1\}^{\ell \cdot (N+1)}$. Super osebkke razporedimo glede na vrednost funkcije eval in jih označimo z $o_1, o_2, \dots, o_{2^\ell}$, pri čemer velja

$$\text{eval}(o_1) \geq \text{eval}(o_2) \geq \dots \geq \text{eval}(o_{2^\ell}).$$

V prehodni matriki $P^+ \in \mathbb{R}^{\ell \cdot (N+1) \times \ell \cdot (N+1)}$ naj vrstice od 1 do $2^{\ell N}$ ustrezajo stanjem, ki smo jim dodali super osebek o_1 . Posledično bodo vrstice od $2^{(i-1)\ell N} + 1$ do $2^{i\ell N}$ ustrezale stanjem z dodanim super osebkom o_i ; $i = 2, \dots, 2^\ell$. Ker super osebek ne sodeluje pri selekciji, križanju in mutaciji, se lahko razširjene matrice teh operatorjev zapišejo kot bločno diagonalne matrice

$$K^+ = \begin{bmatrix} K & & & \\ & K & & \\ & & \ddots & \\ & & & K \end{bmatrix}, \quad M^+ = \begin{bmatrix} M & & & \\ & M & & \\ & & \ddots & \\ & & & M \end{bmatrix}, \quad S^+ = \begin{bmatrix} S & & & \\ & S & & \\ & & \ddots & \\ & & & S \end{bmatrix}$$

z 2^ℓ bloki, kjer so matrice K , M in S velikosti $2^{N-\ell} \times 2^{N-\ell}$. Sledi

$$K^+ M^+ S^+ = \begin{bmatrix} KMS & & & \\ & KMS & & \\ & & \ddots & \\ & & & KMS \end{bmatrix},$$

pri čemer velja $KMS > 0$. Operacija kopiranja najboljšega osebkka v naslednjo generacijo je predstavljena z matriko $U \in \mathbb{R}^{\ell \cdot (N+1) \times \ell \cdot (N+1)}$, ki na mesto super osebkka postavi najuspešnejši osebek v populaciji, če se tam še ne nahaja. Matrika U je stohastična matrika, ki ima v vsaki vrstici na enem mestu vrednost 1, na ostalih pa 0. Naj

$$\tilde{k} = \arg \max\{\text{eval}(proj_k(i)); k = 1, \dots, N\}$$

določa najboljši osebek $proj_{\tilde{k}}(i)$ v populaciji i , pri čemer je super osebek izvzet. Če velja $\text{eval}(proj_0(i)) \geq \text{eval}(proj_{\tilde{k}}(i))$, potem naj bo $u_{ii} = 1$ in $u_{ij} = 0$ za $j \neq i$. V nasprotnem primeru je $u_{ij} = 1$ za $j = (proj_{\tilde{k}}(i), proj_1(i), \dots) \in S$ in $u_{ik} = 0, k \neq j$. Sledi, da je v vsaki vrstici matrice U natanko ena 1, kar pa ne velja za stolpce, saj za vsako stanje $j \in S$, tako da $\text{eval}(proj_0(j)) < \max\{\text{eval}(proj_k(j)); k = 1, \dots, N\}$, velja $u_{ij} = 0$ za vse $i \in S$. Matrika U ima bločno obliko

$$U = \begin{bmatrix} U_{11} & & & \\ U_{21} & U_{22} & & \\ \vdots & \vdots & \ddots & \\ U_{2^\ell,1} & U_{2^\ell,2} & \dots & U_{2^\ell,2^\ell} \end{bmatrix},$$

kjer so podmatrice velikosti $2^{N-\ell} \times 2^{N-\ell}$. Zaradi vrstnega reda stanj, pri čemer so populacije z najuspešnejšim super osebkov zapisane na začetek, je matrika U_{11} enaka identiteti. Sledi zapis

razširjene matrike verjetnosti prehodov stanj:

$$P^+ = \begin{bmatrix} P & & & \\ & P & & \\ & & \ddots & \\ & & & P \end{bmatrix} \begin{bmatrix} U_{11} & & & \\ U_{21} & U_{22} & & \\ \vdots & \vdots & \ddots & \\ U_{2^\ell,1} & U_{2^\ell,2} & \cdots & U_{2^\ell,2^\ell} \end{bmatrix} = \\ = \begin{bmatrix} P & & & \\ PU_{21} & PU_{22} & & \\ \vdots & \vdots & \ddots & \\ PU_{2^\ell,1} & PU_{2^\ell,2} & \cdots & PU_{2^\ell,2^\ell} \end{bmatrix}.$$

Podmatrike oblike PU_{a1} za $a \geq 2$ lahko združimo v matriko $R \neq 0$.

Izrek 8 ([4, stran 126]). Naj bo P reducibilna stohastična matrika oblike

$$P = \begin{bmatrix} C & 0 \\ R & T \end{bmatrix},$$

kjer je C primitivna stohastična matrika velikost $m \times m$, matriki R in T pa sta neničelni. Potem je

$$P^\infty = \lim_{k \rightarrow \infty} P^k = \begin{bmatrix} C_\infty & 0 \\ R_\infty & 0 \end{bmatrix}$$

stabilna stohastična matrika, katere oblika vrstic je $[\pi_\infty(1), \dots, \pi_\infty(m), 0, \dots, 0]$, pri čemer $\pi_\infty(i) > 0$ za $i = 1, 2, \dots, m$.

Izrek 9. Kanonični genetski algoritem, podan v izreku 6, ki skozi generacije ohranja najboljšo rešitev dobljeno **po** selekciji, konvergira h globalnemu optimumu.

Dokaz. Podmatrika $PU_{11} = P > 0$ vsebuje prehodne verjetnosti stanj, ki vsebujejo globalno optimalno rešitev. Ker je P primitivna stohastična matrika in R nenegativna, nam izrek 8 zagotavlja, da verjetnost, da bi obtičali v lokalnem maksimumu, konvergira proti 0. Torej verjetnost, da smo v globalno optimalnem stanju, konvergira k 1. Sledi $\lim_{t \rightarrow \infty} P(\{Z_t = \text{eval}^*\}) = 1$.

V primeru, da najuspešnejši osebek ohranimo pred selekcijo, velja naslednji izrek.

Izrek 10. Kanonični genetski algoritem, podan v 6, ki skozi generacije ohranja najboljšo rešitev dobljeno **pred** selekcijo, konvergira h globalnemu optimumu.

Dokaz. Prehodna matrika je v tem primeru $P^+ = T^+US^+$, kjer je $T^+ = K^+M^+$. Naj bo $T = KM$. Sledi

$$P^+ = \begin{bmatrix} T & & & \\ & T & & \\ & & \ddots & \\ & & & T \end{bmatrix} \begin{bmatrix} U_{11} & & & \\ U_{21} & U_{22} & & \\ \vdots & \vdots & \ddots & \\ U_{2^\ell,1} & U_{2^\ell,2} & \cdots & U_{2^\ell,2^\ell} \end{bmatrix} \begin{bmatrix} S & & & \\ & S & & \\ & & \ddots & \\ & & & S \end{bmatrix} = \\ = \begin{bmatrix} TU_{11}S & & & \\ TU_{21}S & TU_{22}S & & \\ \vdots & \vdots & \ddots & \\ TU_{2^\ell,1}S & TU_{2^\ell,2}S & \cdots & TU_{2^\ell,2^\ell}S \end{bmatrix},$$

pri čemer je $TU_{11}S = TS = P > 0$. Podmatrike $TU_{i1}S$, kjer je $i \geq 2$, lahko zberemo v neničelno matriko R . Z enakimi argumenti kot pri dokazu izreka 9 dobimo $\lim_{t \rightarrow \infty} P(\{Z_t = \text{eval}^*\}) = 1$.

Vrnimo se k izreku o shemah 1 in z znanjem o markovskih verigah pokažimo, da izrek res ne govori o konvergenci algoritma. Naj bo S množica stanj oziroma vseh možnih populacij. Kot prej označimo populacijo ob času t z i_t . Definirajmo indikator

$$I_t = \begin{cases} 1, & \text{če } \text{eval}^* \in \{\text{eval}(\text{proj}_1(i_t)), \text{eval}(\text{proj}_2(i_t)), \dots, \text{eval}(\text{proj}_N(i_t))\}, \\ 0, & \text{sicer} \end{cases},$$

ki spremlja pojavitev globalno optimalnega osebka v populaciji ob času t . Označimo ga z b^* . Naj bo $g(b^*, i_t)$ funkcija, ki šteje pojavitve optimalnega osebka b^* v populaciji ob času t . Potem se enačba (10) prevede na

$$E(b^*, t+1) \geq g(b^*, i_t) \left(1 - p_c \cdot \frac{\delta(b^*)}{m-1}\right) (1 - o(b^*) \cdot p_m).$$

Očitno je desna stran enačbe manjša od $g(b^*, i_t)$, kar ne nakazuje na konvergenco. Po drugi strani pa tudi ne moremo dokazati nekonvergenca. Poglejmo si naslednji lemi.

Lema 11. Velja: $\lim_{t \rightarrow \infty} E[I_t] = 1 \iff \lim_{t \rightarrow \infty} P(\{Z_t = \text{eval}^*\}) = 1$.

Dokaz. Vemo, da $\{I_t = 1\} \iff \{Z_t = \text{eval}^*\}$. Naj bo $1_A(x)$ indikatorska funkcija, kjer je $A = \{I_t = 1\}$. Če enačbo $E(1_A) = P(A)$ pogledamo v limiti, dobimo ravno želeno ekvivalenco.

Lema 12. Velja: $\lim_{t \rightarrow \infty} E[I_t] = 1 \implies \lim_{t \rightarrow \infty} E(b^*, t) \geq 1$.

Dokaz. Očitno velja neenakost $g(b^*, i_t) \geq I_t$. Sledi

$$\lim_{t \rightarrow \infty} E(b^*, t) = \sum_{i_t=1}^{|S|} g(b^*, i_t) \cdot \pi_\infty(i_t) \geq \sum_{i_t=1}^{|S|} I_t \cdot \pi_\infty(i_t) = \lim_{t \rightarrow \infty} E(I_t) = 1,$$

kjer je π_∞ limitna porazdelitev markovske verige.

Obrat leme 12 ne velja vedno. Naj bo $S = \{00, 01, 10, 11\}$, kjer naj 1 predstavlja globalno optimalno rešitev. Naj bo $\pi_\infty = [0, 0, 1, 0]$. Potem

$$\lim_{t \rightarrow \infty} E(b^*, t) = \sum_{i_t=1}^{|S|} g(b^*, i_t) \cdot \pi_\infty(i_t) = 0 \cdot 0,01 + 1 \cdot 0,25 + 1 \cdot 0,25 + 2 \cdot 0,49 = 1,48 > 1$$

in

$$\lim_{t \rightarrow \infty} E(I_t) = \sum_{i_t=1}^{|S|} I_t \cdot \pi_\infty(i_t) = 0 \cdot 0,01 + 1 \cdot 0,25 + 1 \cdot 0,25 + 1 \cdot 0,49 = 0,99 < 1.$$

Iz primera je razvidno, da kljub povečevanju pričakovanega števila optimalnih osebkov v populaciji, to še ne pomeni, da bo algoritem skonvergiral k optimalni rešitvi.

Preko uporabe homogenih končnih markovskih verig je bilo v tem razdelku dokazano, da osnoven kanonični genetski algoritem ne bo nikoli skonvergiral h globalni rešitvi, ne glede na izbiro začetne populacije. Teorija markovskih verig je bila uporabljena tudi v izreku o shemah (izrek 1), pri čemer je bilo dokazano, da izrek ne govori o konvergenci genetskega algoritma. Dokazali smo konvergenco kanoničnega genetskega algoritma, ki skozi generacije ohranja najboljše osebke. Takim genetskim algoritmom pravimo tudi *elitistični genetski algoritmi (EGA)*. Pri tem je bila uporabljena predpostavka, da je stohastična matrika mutacij M pozitivna. Vredno je opozoriti, da elitističnega genetskega algoritma ne smemo zamešati z uporabo elitistične selekcije. Pri slednji različici kanoničnega genetskega algoritma se najboljši osebek uporabi pri generiranju potomcev, medtem ko se ga pri uporabi elitističnega genetskega algoritma zgolj shranjuje v obliki dodatnega osebka v populaciji, ki pa ni prisoten pri kreiranju populacije potomcev.

4. Zaključek

V članku je bil predstavljen genetski algoritem, ki se pri iskanju rešitve opira na tri operatorje - selekcijo, križanje in mutacijo. Takemu genetskemu algoritmu pravimo tudi enostavni genetski algoritem (SGA). S preprostimi primeri smo opozorili na nekatere njegove lastnosti, ki jih moramo imeti v mislih pri reševanju optimizacijskih problemov. V članku je veliko pozornosti namenjeno konvergenci k optimalni globalni rešitvi pri zelo specifičnih različicah enostavnih genetskih algoritmov.

V splošnem so genetski algoritmi zelo robustni, računsko preprosti ter dovtetni za najrazličnejše nadgradnje in modifikacije. Eden izmed načinov, kako izboljšati njihovo robustnost, je z dodajanjem novih operatorjev in mehanizmov (npr. migracija, izbris, segregacija, dominanca, inverzija, itd.). Hibridni genetski algoritem (HGA) je primer modifikacije, kjer genetski algoritem odkrije okolico globalnega maksimuma, sam globalni maksimum pa določimo s pomočjo poljubnega determinističnega algoritma za iskanje lokalnega maksimuma (npr. z gradientno metodo). Obstaja tudi način za reševanje večkriterijskih optimizacijskih problemov in še veliko drugih posebnih načinov uporabe algoritma. S časom se število vrst genetskih algoritmov le še povečuje, prav tako pa narašča tudi število objav na temo analize delovanja genetskih algoritmov, kar potrjuje dejstvo, da so genetski algoritmi zanimivi in uporabni za široko paleto problemov.

LITERATURA

- [1] M. Črepinšek, M. Mernik in V. Žumer, *Evolucijski algoritmi*, Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut za računalništvo, Maribor, 2003.
- [2] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley publishing company, USA, 1989.
- [3] J. Holland, *Adaptation in natural and artificial systems*, The University of Michigan Press, USA, 1975.
- [4] M. Iosifescu, *Finite markov processes and their applications*, Chichester, Wiley, 1980.
- [5] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*, Springer-Verlag, New York, 1996.
- [6] G. Mitsuo, Y. Xinjie, *Introduction to evolutionary algorithms*, Springer-Verlag, London, 2010.
- [7] G. Rudolph, *Convergence analysis of canonical genetic algorithms*, IEEE Transactions on neural networks, 5(1), 96–101, 1994.
- [8] E. Seneta, *Non-negative matrices and markov chains*, Springer, New York, 1981.