

UTICAJ BINARNOG KODIRANJA NA GENETSKE ALGORITME ZA NALAZENJE EKSTEMNIH VREDNOSTI

V. Filipović¹, J. Kratica, S. Radojević, M. Vugdelija²

Rezime:

U radu se daje kratak pregled Genetskih algoritama (GA). Opisuju se koraci algoritma, prednosti, mane i navode neki problemi na koje je GA sa uspehom primenjen.

Prikazuju se rezultati dobijeni poređenjem dva najpopularnija oblika kodiranja (klasično i Grej) u GA za nalaženje ekstremne vrednosti funkcija koje pripadaju De Jongovom skupu test funkcija.

Ključne reči: *genetski algoritmi, kodiranje.*

INFLUENCE OF BINARY CODING ON GENETIC ALGORITHMS FOR FUNCTION OPTIMIZATION

Summary:

The short review of Genetic algorithms (GA) is shown. The algorithm steps, advantages and some domains of application are described.

Results obtain by comparasion between two the most popular ways of coding (canonical and Gray) are presented. Algorithms for function optimization are tested on the De Jong's test bed functions.

Key words: *genetic algorithms, coding.*

1. GENETSKI ALGORITMI

Iako Genetski algoritmi (u daljem tekstu GA) postoje od 1975. godine, činjenica je da oni u poslednje vreme doživljavaju sve veću popularnost. Evoluciona paradigma koju eksploatišu GA postaje široko prihvaćena kao put za rešavanje određenih klasa problema.

Mada su i pre 70.-tih, naročito u Nemačkoj, predlagana rešenja raznovrstnih problema koja emuliraju ponašanje, odnose i veze kao u prirodi, ipak se nastanak GA vezuje za 1975. godinu i Holland-ovu knjigu "Adaptation in natural and artificial systems". Holland je GA razvio radi proučavanja procesa prilagođavanja kod prirodnih sistema i radi razvoja sistema veštačke inteligencije koji oponašaju modele prilagođavanja. Treba istaći da je u prethodno pomenutoj knjizi napravljen i veliki pomak u matematičkom zasnivanju osnova GA

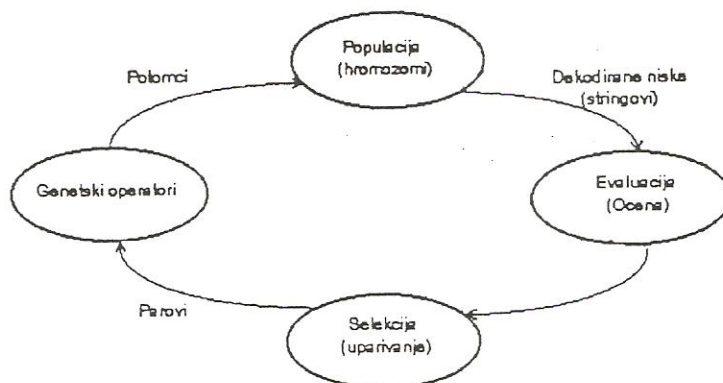
¹ Vladimir Filipović, Matematički fakultet, 11000 Beograd, ul. Studentski trg 16.

² Mr. Jozef Kratica, Mr. Slobodan Radojević, Milan Vugdelija, Mašinski fakultet, 11000 Beograd, ul. 27. marta 80.

(definicija shema, hipoteza gradivnih blokova, teorema o implicitnoj paralelnosti) i da GA povećanje popularnosti u velikoj meri duguju i postojanju matematičke aparature čijim se korišćenjem pojedini rezultati mogu i predvideti i objasniti.

GA se primenjuje kod raznih optimizacionih problema - kako problema iz domena teorije (raspoređivanje poslova, problem trgovačkog putnika) tako i onih praktičnih (finansijsko modeliranje, projektovanje gasovoda); kod mašinskog učenja; kod teorije igara; kod neuralnih mreža; kod prepoznavanja oblika itd.

Odnos elemenata GA može biti iskazan sledećim dijagramom:



Karakteristike GA:

- ♦ GA koriste kodiranje parametara, a ne same parametre.
- ♦ GA ne pretražuju jedan čvor pretraživačkog prostora, već populaciju čvorova (robustnost).
- ♦ GA koriste probabilistička pravila prelaska.
- ♦ GA (u svojoj čistoj formi) tokom procesa nalaženja rešenja ne eksploatišu eventualne dodatne informacije o prirodi problema (npr. glatkost funkcije kod problema optimizacije funkcije, itd.).

U realizaciji GA, radi povećavanja njihove efikasnosti, koriste se raznovrsne modifikacije: pri formiranju niske koja reprezentuje jedinku (haploidna i diploidna reprezentacija); pri kodiranju, odnosno preslikavanju pretraživačkog prostora u prostor niski (npr. kanonično kodiranje, Grej kodovi, itd.); pri računanju ocenjivačke funkcije (razna skaliranja); pri izboru genetskih operatora (npr. za diploidnu reprezentaciju se uvodi dominacija); pri utvrđivanju verovatnoća koje određuju primenu genetskih operatora; pri utvrđivanju kriterijuma konvergencije (npr. da li se najbolja jedinka promenila tokom fiksiranog broja generacija, da li više od pola jedinki u populaciji poklapa sa najboljom, da li se prosečna vrednost jedinki malo razlikuje od najbolje, itd.).

Dok je originalni GA baziran na potpunoj smeni generacija, u nekim modifikacijama se dopušta da roditelj, naravno ukoliko je njegova ocena zadovoljavajuća, nastavlja da živi naporedo sa svojim potomcima. U konkretnim realizacijama GA se često eksploatiše međurešenje: najbolje prilagođena jedinka iz prethodne generacije nastavlja svoj život, i to je poznato pod imenom "elitist strategija". Nadalje, u raznim realizacijama se parametri koji određuju stohastičku kontrolu genetskih operatora menjaju tokom samog rada algoritma.

Familija strategija rada GA, koje se razlikuju po vrednosti stohastičkih parametara nazivaju se reproduktivnim planom. Termin reproduktivni plan potiče od De Jong-a.

2. NAČINI KODIRANJA

GA su nastavili da se razvijaju i mnogi autori su predlagali razna poboljšanja, tj. razne nove reproduktivne planove. Neke od predloženih izmena su bile skoncentrisane i efikasno primenljive samo na problem kojim se autor-predlagač bavio, dok su drugi predlozi bili opšteprimenjivi.

U praksi se dešva (dakle, empirijski je utvrđeno) da se određena kodiranja za neke tipove problema mnogo bolje "ponašaju" tj. da su rezultati primene GA sa takvim kodiranjem osetno bolji. Ta činjenica ima svoje teoretsko razjašnjenje, koje se zasniva na hipotezi gradivnih blokova. Gradivni blokovi su sheme malog reda a velikog fitness-a (ocene), koje se eksponencionalno šire kroz prostor niski. Ispostavlja se da su bolja kodiranja kod kojih bliskim tačkama pretraživačkog prostora odgovaraju kodovi tj. niske koji se razlikuju na malom broju bitovnih mesta, odnosno niske sa malim Hemingovim rastojanjem.

Predloga i poboljšanja kod načina kodiranja ima mnogo, ali u radu su opisani kanoničko kodiranje, Grej kodiranje i dinamičko kodiranje parametara.

2.1. Kanoničko kodiranje

Ovo je istorijski najstariji način kodiranja. Za problem optimizacije funkcija se preslikavanje realnog broja iz nekog intervala u nisku bitova (dužine n) se obavlja na sledeći način: interval se podeli na ekvidistantne tačke (broj tačaka je 2^n), svakoj tački se pridruži binarni ceo broj koji označava redni broj te tačke u intervalu, i realnom broju se dodeli redni broj najbliže donje tačke podele.

2.2. Grej kodiranje

Samo kodiranje se vrši na sličan način kao u prethodnom podpoglavlju, s tim što se podeonim tačkama ne dodeljuje binarni ekvivalent rednog broja, već Grej kod koji odgovara rednom broju tačke u intervalu.

Grej kod je dobio naziv po Frenku Grej-u, koji je 1953. patentirao njegovo korišćenje za određene kodere i dekodere. Grej kodiranje će svaki od celih brojeva iz skupa $\{0, 1, 2, \dots, 2^n - 1\}$ predstaviti kao binarnu nisku dužine n na takav način da susedni celi brojevi imaju Grej reprezentacije koje se razlikuju u jednoj bitovnoj poziciji. Tako je jedno (ne i jedino) Grej kodiranje za skup $\{0, 1, 2, 3, 4, 5, 6, 7\}$ dato sa 000,001,011,010,110,111,101,100. Ovakav način kodiranja se najčešće koristi i naziva se binarno-reflektovani Grej kod.

Prosta šema za generisanje Grej kod sekvenci glasi: "počni sa svim nulama, sukcesivno menjaj najdesniji bit čijom se izmenom može proizvesti nova niska".

2.3. Dinamičko kodiranje parametara

Ovde šema kodiranja ne ostaje konstantna tokom genetičke pretrage. Tako i dobijeni rezultat može biti precizniji od $1/2^n$, jer se na početku koristi vrlo gruba previznost, sve dok GA ne prokonvergira. U tom trenutku se rezultujuća populacija (odnosno podinterval koji ona zauzima) ponovo kodira u isti skup bitova, tako da je pretraga sužena na ovaj podinterval, pa se GA ponovno startuje. Ovaj bi se proces mogao nastaviti sve dok rezultat ne dobije željenu preciznost. U stvari, dinamičko kodiranje parametara to i radi, i to autonomno i paralelno za svaku realnu vrednost. Na ovaj način, žrtvovanje rezolucije pretage ne mora dovesti do istovetnog gubljenja preciznosti kod rezultata pretrage.

Kod ovog načina kodiranja mora postojati operator zumiranja.

2.4. Teoretska procena

Poglavje o kodiranju ne bi bilo kompletno ukoliko se ne navede i Vous-ov rezultat: da gradivni blokavi ne moraju biti invarijante problema koji se rašava Genetskim algoritmom. Dakle, rezultati GA za svako fiksno kodiranje pri rešavanju neke klase problema bi, ukoliko se usrednje po skupu svih instanci te klase problema, trebalo da daju otprilike iste rezultate.

Vous je, u drugom svom radu, još pokazao da je Grej kodiranje specijalan slučaj kodiranja indukovano fiksnom matricom.

3. DE JONGOV TEST

Godine 1975, u vreme izdavanja Hollandove knjige, De Jong završava svoju doktorsku disertaciju. Taj će rad imati važno, moglo bi se reći centralno, mesto u daljem razvoju GA. Disertacija je postala prekretnica u daljem razvoju GA kako zbog povezanosti sa Holland-ovom teorijom šema, tako i zbog pažljivih i preciznih računarskih eksperimenata, te pažljivo razmotrenih zaključaka.

De Jong je uvideo značaj pažljivo kontrolisanog eksperimentisanja u problemskom domenu koji nije nesređen. On je tako ogolio GA, okruženje i performanse na same osnove. Takvo uprošćavanje mu je omogućilo da izvrši eksperimente koji su postali bazna linija za sva dalja GA proučavanja i radove.

De Jongov skup uključuje test funkcije sa sledećim karakteristikama: Nепrekidne/Prekidne; Konveksne/Nekonveksne; Jednomodalne/Višemodalne; Kvadratne/Nekvadratne; Malodimenzionalne/Mnogodimenzionalne; Determinističke/Stohastičke

De Jongovo test-okruženje sadrži sledećih pet funkcija:

1) Sferna funkcija:

$$f_1(x_1, x_2, x_3) = \sum_{i=1}^3 x_i^2, \text{ pri čemu je } -5.12 \leq x_i \leq 5.12$$

2) Rozenbrokova funkcija:

$$f_2(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2, \text{ pri čemu je } -2.048 \leq x_1 \leq 2.048$$

3) Stepenična funkcija:

$$f_3(x_1, \dots, x_5) = \sum_{i=1}^5 [x_i], \text{ pri čemu je } -5.12 \leq x_i \leq 5.12$$

4) Četvorna funkcija sa šumom:

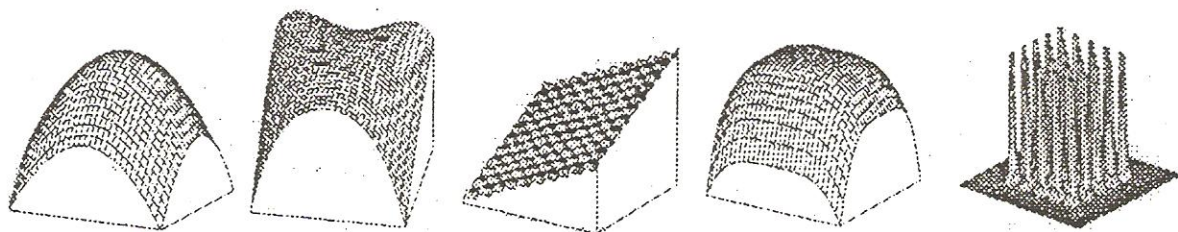
$$f_4(x_1, \dots, x_{30}) = \sum_{i=1}^{30} ix_i + \text{Gaus}(0,1), \text{ pri čemu je } -1.28 \leq x_i \leq 1.28$$

5) Šekelova funkcija:

$$f_5(x_1, \dots, x_{25}) = 0.0002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_i)^6}, \text{ pri čemu je } -65.536 \leq x_i \leq 65.536$$

Na slikama koje slede su predstavljene grafici dvodimenzionalnih ekvivalenata prethodno

definisanih funkcija:



De Jong je, radi kvantifikovanja efektivnosti svakog od različitih GA, smislio dve mere, jednu za merenje konvergencije, a drugu za merenje performansi toka. Prvu je nazvao of-lajn performansa, a drugu on-lajn performansa - po analogiji sa of-lajn i on-lajn aplikacijama.

- ♦ Of-lajn performansa (konvergencija) strategije s u okruženju e

$$x_o^*(s) = \frac{1}{T} \sum_{t=1}^T f_o^*(t)$$

, gde je $f_o^*(t)$ označena vrednost najboljeg u skupu $\{f_1(t), f_2(t), \dots, f_c(t)\}$.

- ♦ On-lajn performansa (tok) strategije s u okruženju e

$$x_o(s) = \frac{1}{T} \sum_{t=1}^T f_o(t)$$

, gde je $f_o(t)$ označena vrednost funkcije objekcije za okruženje e tokom probe t .

Of-lajn performansa je prosek dosad najboljih performansi u raznim generacijama. De Jong je predložio i opštiju verziju ovog kriterijuma, koja dopušta neuniformnu dodelu težina raznim pokušajima. Međutim, u daljem radu se isključivo radi sa uniformnom varijantom. On-lajn performansa je prosek svih funkcijskih evaluacija zaključno sa tekućim pokušajem. I za ovu meru je bila predložena i neuniformna varijanta, ali je u radu korišćena uniformna.

Pored ove dve karakteristike, De Jong je ispitivao i broj izgubljenih gena tokom probe t . Ova veličina ukazuje na širinu pretrage kroz pretraživački prostor.

De Jong je u svojim testovima koristio isključivo kanoničko kodiranje.

Treba napomenuti da, iako se pomoću GA lepo rešavaju određeni problemi optimizacije funkcija, ne treba smatrati da se GA samo zato koriste. Nedavno je izašao rad jednog od vodećih autoriteta u ovoj oblasti, Keneta De Jonga, pod naslovom: "Genetski algoritmi nisu optimizatori funkcija" u kom autor kritikuje taj, dosta često viđen, simplifikovan prilaz problematici GA, ukazuje na razlike koje se pojavljuju između GA za optimizaciju funkcija i GA za rešavanje drugih problema i ističe nedostatak čvršće matematičke teorije vezane za predložena poboljšanja.

4. TESTIRANJE

Evaluacija raznih varijacija GA i određivanje performansi je izvršeno na algoritamski orijentisanom GA sistemu, koji su razvili autori rada. Dizajn i osobine ovog GA sistema, komparativne prednosti i eventualni nedostaci u odnosu na preko Interneta dostupan PD softver kao i u odnosu na komercijalni GA softver su tema za sebe. Rasprava o tim pitanjima zahteva mnogo više prostora. Međutim, mišljenja smo da bi ovaj rad bio nekompletan ukoliko se ne navedu neke najvažnije karakteristike razvijenog GA sistema:

- ♦ Fleksibilnost - autori su, koristeći razvijeni softver, a zahvaljujući pažljivo izvršenom koncipiranju i realizovanju, kreirali GA zasnovane programe i "napadali" probleme raznovrsnih tipova, od kojih su neki vrlo egzotični (npr. genetski algoritam na dva

nivoa, normalnom i meta, a radi odredjivanja samih stohastičkih veličina koje definišu rad GA; NP teški kombinatorni problemi, itd.).

- ♦ Ponovno korišćenje razvijenog koda (reusability) - kod razvijen radi rešavanja jednog od problema se može, bez ikakvih izmena, čak i bez ponovnog prevođenja, koristiti i u druge svrhe (npr. računanje vrednosti De Jongovih performansi, napravljeno za potrebe ovog rada, može da se koristi, i koristi se, za ocene pervormansi novih varijacija GA itd.).
- ♦ Nezavisnost od operativnog sistema i računara- softer je razvijen u programskom jeziku C, uz strogo poštovanje ANSI standarda, tako da je GA sistem lako portovati na najraznovrsnije platforme. Nezavisnost od platforme se ne može u potpunosti postići kada npr. treba eksploatisati eksplicitnu paralelnost GA, ali i tada ostali bitni elementi u svim drugim modulima ostaju nepromenjeni.
- ♦ Efikasnost - način koncipiranja i način realizacije ovog softvera omogućili su da iako je softver veoma fleksibilan i potpuno nezavistan od platforme, njegove performanse ne budu bitno degradirane.

Veličine parametara (generacijski jaz, verovatnoće ukrštanja i mutacije) su pri testiranju postavljene na vrednosti koje je De Jong (uz Greffenstate-ova preciziranja) utvrdio kao najbolje (potpuna smena generacija tj. $G=1$, verovatnoća ukrštanja 0.6, verovatnoća mutacije 0.01).

Rezultati dobijeni poređenjem ukazuju da je, za ovaj skup test-funkcija, Grej kodiranje bolje od kanonskog (u mislu poboljšanja off-lajn i on-lajn performansi) prosečno za desetak procenata, pri čemu je kod "normalnijih" funkcija to poboljšanje primetnije.

5. LITERATURA

- [1] Holland John H. "Adaptation in Natural and Artificial Systems", The University of Michigan Press, Ann Arbor, 1975.
- [2] Goldberg David E. "Genetic Algorithms in Search, Optimization & Machine Learning", Addison-Wesley Publishing Company Inc., Reading, Massachusetts, 1989.
- [3] Forsyth Richard "Machine Learning - Principles and techniques", Chapman and Hall, London.
- [4] De Jong Kenneth A. "Genetic Algorithms are NOT Function Optimizers", in Foundation of Genetic Algorithms 2, pg. 5-19, edited by Whitley Darrel, Morgan Kaufmann Publishers, San Mateo, California, 1992.
- [5] Schraudolph Nicol N, Belew Richard K. "Dynamic Parameter Encoding for Genetic Algorithms, Machine Learning, July 1992.
- [6] Hancock Peter J. B. "An empirical comparison of selection methods in evolutionary algorithms", in Evolutionary Computing, pg. 80-94, edited by Fogarty Terece, AISB Workshop Leeds, April 1994.
- [7] Vose Michael D. "Generalizing the notion of schema in genetic algorithms", Artificial Intelligence 50, 1991.
- [8] Battle David, Vose Michael D. "Isomorphisms of genetic algorithms", Artificial Intelligence 60, 1993.