Odredivanje preformansi Genetskih algoritama u teoriji i praksi

Vladimir Filipović
Matematički fakultet, Studentski trg 16, Beograd

1. Uvod

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 odredenih klasa problema ([9],[13]).

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 artifiicial systems". Holland je GA razvio radi proučavanja procesa prilagodavanja kod prirodnih sistema i radi razvoja sistema veštačke inteligencije koji oponašaju modele prilagodavanja. Treba istaći da je u prethodno pomenutoj knjizi napravljen i veliki pomak u matematičkom zasnivanju samog GA (definicija shema, teorema o shemama, teorema o implicitnom paralelizmu, hipoteza gradivnih blokova,) i da GA povećanje popularnosti u velikoj meri duguju i postojanju matematičke aparature čijim korišćenjem se pojedini rezultati mogu i predvideti i objasniti.

U realizaciji GA, radi povećavanja njihove efikasnosti, koriste se raznovrsne modifikacije: pri formiranju niske koja reprezentuje jedinku - npr. haploidna i diploidna reprezentacija; pri kodiranju, odnosno preslikavanju pretraživačkog prostora u prostor niski - npr. klasično kodiranje, Grejovi kodovi; pri računanju funkcije prilagodenosti (funkcije uklopljenosti ili fitness funkcije); pri izboru genetskih operatora -npr. za diploidnu reprezentaciju se uvodi dominacija; pri utvrdivanju verovatnoća koje odreduju primenu genetskih operatora; pri utvrdivanju kriterijuma konvergencije -npr. da li se najbolja jedinka promenila tokom fiksiranog broja generacija, da se li više od pola jedinki u populaciji poklapa sa najboljom, da li se prosecna 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 nivo njegove prilagodenosti zadovoljavajući, nastavlja da živi naporedo sa svojim potomcima. U konkretnim realizacijama GA se često eksploatiše medurešenje: najbolje prilagodena jedinka iz prethodne generacije nastavlja svoj život, i to je poznato pod imenom elitna strategija (elitist strategy). Nadalje, u raznim modifikacijama se parametri koji odreduju stohastičku kontrolu genetskih operatora menjaju tokom samog rada algoritma.

2. Teorija

Teoretsko razmatranje GA najčešće polazi od Holland-ovog doprinosa u ovoj oblasti, tj. od njegovih rezultata. I pre Holland-ovog rada su postojali evolutivno inspirisani programski sistemi, ali tek njegov rad daje odredeni uvid u to kako ova klasa algoritama dovodi do rešenja, i zašto su GA toliko efikasni pri rešavanju odredenih klasa problema.

2.1. Teorema o shemama

Pretpostavlja se da se proučava prosti (u literaturi se još naziva i kanonski) GA, sa sledećim operatorima: prosta tj. rulet (roulette) selekcija, jednopoziciono (one-point) ukrstanje, prosta (one-point) mutacija. Populacija u vremenskom trenutku t biće označena sa $a(t)$. Populacija se sastoji od niski (za elemente populacije koriste se još i nazivi genetski kod, string, nosilac genetskog materijala). U razmatranju koje sledi pretpostavlja se da su niske reči formirane nad dvočlanim alfabetom $\{0,1\}$ i da je njihova dužina l. Dakle,

Definicija 1. Niska, u oznaci α, je reč dužine l nad dvočlanim alfabetom $\{0,1\}$.

Pojam, tj. koncept sheme predstavlja osnovu u Hollandovom radu. Naime, umesto koncentrisanja na pojedine niske, koncentriše se na sličnosti između njih, tj. na pitanje koliko jedna niska može biti bliske (može ličiti) na neku drugu nisku. Šablon sličnosti koji opisuje podskup niski istih na određenim pozicijama naziva se shema.

Definicija 2. Shema je reč dužine l nad tročlanim alfabetom $\{0,1,*\}$. Simbol '*' je džoker simbol (nespecificirani simbol) - on je metasimbol za GA, tj. on se eksplicitno ne obrađuje genetskim algoritmom. Pozicije koje zauzimaju simboli '0' i '1' nazivaju se definišuće pozicije sheme.

Definicija 3. Shema odgovara određenoj niski ukoliko na svakoj lokaciji u shemi simbolu '1' odgovara '1' iz niske, simbolu '0' u shemi odgovara '0' iz niske, a simbolu '*' može odgovarati ma koji simbol.

Dakle, shema je mehanizam za prepoznavanje obrazaca. Ona daje moćan i kompaktan put za razmatranje svih dobro definisanih sličnosti među niskama konačne dužine nad konačnim alfabetom. Treba napomenuti da pored ovakvog, postoje i neki drugi načini definisanja shema. U njima se shema definiše bilo kao drvo izvođenja kod konteksno slobodnog izvođenja ([25]), bilo kao predikat ([23]), bilo kao ravni i hiperravni u višedimenzionalnom bitovnom prostoru ([9]). Ti drugačiji pogledi u nekim slučajevima daju bolju i precizniju sliku odnosno jasniji pogled na problematiku koja se proučava.

Stav 1. Postoji 2^l različitih niski dužine l i 3^l različitih shema iste dužine.

Dakle, uvođenjem simbola '*' se proučava veći broj reči nego kada nema ovog simbola. Ta novouvedena redundansa je od koristi zato što bi se, kada bi se posmatrale samo pojedinačne niske, posmatrali samo delovi informacije. Potpuna informacija, informacija koja omogućava praćenje načina rada (načina traženja rešenja) kod GA zahteva poznavanje sličnosti među niskama.

Pitanje broja informacija koje se posmatraju razmatranjem sličnosti se svodi na pitanje broja različitih shema koje sadrži populacija. Odgovor na ovo pitanje zahteva određivanje broja jedinki koje se sadrže u jednoj niski.

Stav 2. Niska dužine l sadrži 2^l različitih shema, tj. postoji 2^l različitih shema koje odgovaraju fiksiranoj niski dužine l.

Sada se donja i gornja granica broja shema koje se sadrže u populaciji od n jedinki trivijalno izvodi. Broj shema koje sadrži populacija zavisi od raznolikosti populacije.

Stav 3. Populacija od n jedinki (članova) sadrži ne manje od 2^l i ne više od $n\,2^l$ shema.

Sada, kada je utvrđen obim informacija koje koristi GA, postavlja se pitanje koliko efikasno GA eksploatiše ove informacije. Drugim rečima, utvrđena je gornja i donja granica za broj shema koje se sadrže u populaciji, pa se postavlja pitanje koliko shema među postojećim GA obrađuje na koristan način. Dakle, treba razmotriti efekte genetskih operatora selekcije, ukrštanja i mutacije na širenje i opadanje pojedinih shema iz generacije u generaciju.

Što se efekta selekcije tiče, niske sa najvećom prilagođenošću (koje se najbolje uklapaju, koje imaju najveći fitness) sa većom verovatnoćom preživljavaju, odnosno prelaze u sledeću generaciju. Zato će prosečan broj shema koje se odgovaraju najboljim niskama tokom vremena biti sve veći. Sama selekcija ne uključuje nove uzorke u prostor niski, tj. primenom ovog genetskog operatora ne kreira se nov genetski materijal.

Dejstvo operatora ukrštanja na shemu je nešto složenije. Naime, ukrštanje će ostaviti shemu nepromenjenom ukoliko je ne "raseče", a ukoliko dođe do "rasecanja" tokom ukrštanja shema će biti razorena (uništena, disrupted). Pri tome ne bivaju sve sheme "rasečene" sa istom verovatnoćom.

Primer 1. Uočimo sheme dužine pet: 1***1 i **01*. Prva od njih dve bi lakše mogla biti razorena ukrštanjem, dok bi se to za drugu relativno teže desilo (naravno, uz pretpostavku da su sve pozicije pri izboru tačke ukrštanja jednakoverovatne).

Dakle, neke će sheme sa manjom, a neke sa većom verovatnoćom biti razorene prilikom ukrštanja, tj. ukrštanje kod prostog GA olakšava širenje nekih shema, a razara neke druge.

Mutacija je, u tradicionalnom razmatranju prostih GA, operator sekundarne važnosti, obično se primenjuje sa malom verovatnoćom, pa će se zasada verovatnoću da shema bude prekinuta usled mutacije smatrati zanemarljivom.

Već se vidi da, iako GA manipulišu direktno i eksplicitno sa niskama u populaciji, rad GA prouzrokuje implicitno procesiranje mnogih shema tokom svake od generacija.

Da bi se moglo strože odrediti verovatnoća preživljavanja i verovatnoća uništavanja sheme pod dejstvom genetskih operatora, potrebno je definisati veličine koje karakterišu otpornost sheme na uništavanje:

Definicija 4. Red sheme H, u oznaci $o(H)$, je broj fiksnih pozicija, tj. broj nula i jedinica koje su prisutne u šablonu.

Definicija 5. Definišuća dužina sheme H, u oznaci $\delta(H)$, je rastojanje između prve i poslednje specificirane pozicije u niski (tj. između prve i poslednje pozicije u niski na kojoj se ne nalazi simbol '*'.

Primer 2. $o(011*1***) = 4$, $o(0*******) = 1$, $\delta(011*1***) = 5\text{-}1 = 4$, $\delta(0*******) = 1\text{-}1 = 0$.

Sada je moguće preciznije odrediti efekat selekcije na očekivani broj shema u populaciji.

Stav 4. Neka je $m(H,t)$ broj koji označava koliko se puta neka shema H sadrži u $\alpha(t)$ tj. u zadatoj populaciji u vremenskom trenutku t. Neka je sa $f(H)$ označena srednja vrednost prilagođenosti svih niski kojima odgovara shema H, a sa $f\,'$ srednja vrednost prilagođenosti svih niski u populaciji. Tada je $m(H,t+1) = m(H,t) * (f(H) / f\,')$.

Dokaz. Neka je a_i oznaka fiksirane niske u populaciji i neka je prilagođenost niske a_i označena sa f_i. Tada je verovatnoća izbora tj. prelaska u sledeću generaciju data sa $p_i = f_i \,/\, \sum f_j = f_i \,/\, nf\,'$. Iz ove jednakosti direktno sledi tvrđenje stava.

Dakle, brzina širenja određene sheme je jednaka količniku između prosečne prilagođenosti sheme i prosečne prilagođenosti populacije. Prema tome, sheme sa natprosečno dobrom prilagođenošću se šire, dok sheme sa ispodprosečnom prilagođenošću odumiru.

Stav 5. Neka određena shema H ostaje natprosečna sa faktorom c, tj. prosečna prilagođenost sheme H iznosi $f\,' + cf\,'$, pri čemu se pretpostavlja da je $f\,'$ konstanta, tj. da se tokom vremena populacija ne menja. Tada je $m(H,t) = m(H,0)*(1+c)^t$.

Dokaz. Na osnovu prethodnog stava je: $m(H,t+1) = m(H,t)*(f(H) / f\,') = m(H,t)*(\,(f\,'+cf\,') / f\,') = m(H,t)*(1+c)$

Iz ove jednakosti, primenom principa matematičke indukcije, direktno sledi tvrđenje stava.

Što se ukrštanja tiče, ono se posmatra kao struktuirana, ali i probabilistička razmena informacija među niskama. Ukrštanje kreira nove strukture, uz minimalno razaranje strategije alokacije shema, koje diktira sama selekcija.

Stav 6. Neka je pri jednopozicionom ukrštanju tačka ukrštanja slučajna promenljiva sa uniformnom raspodelom, i neka je p_c verovatnoća da je uopšte došlo do ukrštanja. Tada je verovatnoća da shema ne bude uništena tokom ukrštanja, u oznaci p_s data sledećom formulom: $p_s \geq 1 - p_c\ (\ \delta(H)\ /\ (l - 1)\ )$.

Još treba razmotriti i operator mutacije. Verovatnoća da shema preživi mutaciju je jednaka verovatnoći da ne bude zamenjeno nijedan od simbola na pozicijama koje su fikirane u shemi.

Stav 7. Neka je verovatnoća mutacije označena sa p_m. Tada je verovatnoća da shema preživi tokom mutacije data izrazom $(1 - p_m)^{o(H)}$.

Za slučaj kada je $p_m << 1$, tada je $(1 - p_m)^{o(H)} \sim 1 - p_m o(H)$, pa se u daljim aproksimacijama koristi ovaj izraz.

Uklapanjem prethodnih rezultata dobija se tzv. Teorema o shemama:

Stav 8. $m(H,t+1) \geq m(H,t)*(f(H)\ /\ f')*(\ 1 - p_c\ (\ \delta(H)\ /\ (l - 1)\ ) - p_m o(H)\ )$

Na osnovu teoreme o shemama može se zaključiti da sheme kratke definišuće dužine i niskog reda, čija je prilagođenost natprosečna, primaju eksponencijalno rastući broj pokušaja.

2.2. Teorema o implicitnom paralelizmu

Postavlja se pitanje procene donje granice broja uspešno obrađenih shema. Populacija se sastoji od n binarnih niski dužine l. Razmatraju se samo sheme koje preživljavaju sa verovatnoćom većom od p_s. Dakle, prihvataju se samo one sheme čiji je nivo greške manji od ε, pri čemu je $\varepsilon < 1 - p_s$. Ovo dovodi do razmatranja samo onih shema čija je dužina manja od l_s, pri čemu je $l_s < \varepsilon*(l - 1) + 1$.

Iz prethodne formule sledi da se donja granica broja različitih shema koje se obrađuju inicijalnom slučajnom populacijom - kao zbir broja shema dužine $1,2,3,\ldots,l_s$-1 koje se sadže u inicijalnoj slučajnoj populaciji.

Neka bude fiksirana dužina sheme - označimo je sa l_s. Da bi se minorirao broj shema koje obrađuje inicijalna slučajna populacija i čija je dužina manja od l_s, neophodno je odrediti koliko se najmanje shema mora sadržati u jednoj niski.

Primer 3. Pretpostvimo da treba prebrojati sheme dužine manje od $l_s = 5$ u fiksiranoj niski dužine $l = 10$, i da je niska oblika 1011100010. Postupak prebrojavanja je sledeći: Prvo se izračuna broj shema u podvučenom delu dužine 5, tj. u niski <u>10111</u>00010 i uz pretpostavku da je poslednji bit fiksiran. To je u ovom slučaju broj shema u formi %%%%1*****, pri čemu simboli % označava vrednost koja može stajati na nefiksiranoj poziciji - on iznosi $2^{5-1} = 16$. Potom se pet mesta od interesa klizno pomere za jednu poziciju udesno, kako bi se računao broj shema u niski 1<u>01110</u>0010 - opet se radi o broju $2^{5-1} = 16$. Kada se izračuna broj ovih shema, vrši se sledeće klizno pomeranje udesno, itd. dok se ne dođe do kraja niske. Pomeranja za po jedno mesto udesno ima 10-5+1 = 6.

Ukoliko se uopšti razmatranje iz prethodnog primera, lako se može pokazati da važi sledeći stav:

Stav 9. Neka je data niska dužine l. Broj shema dužine l_s u takvoj niski se može odozdo ograničiti sa izrazom $2^{l_s - 1}*(l - l_s+1)$.

Iz ovog rezultata direktno sledi procena donje granice broja shema koje obrađuje slučajno odabrana inicijalan populacija:

Stav 10. Neka je slučajno odabrana inicijalna populacija od n niski dužine l. Broj shema dužine l_s u takvoj niski se može odozdo ograničiti sa izrazom $n*2^{l_s - 1}*(l - l_s+1)$.

Može se primetiti da broj shema ima binarnu distribuciju, pa je broj shema čiji je red manji od ℓ_s / 2 jednak broju shema reda većeg od ℓ_s / 2. Ukoliko se broje samo sheme većeg reda, donje ograničenje njihovog broja dato je jednakošću $n_s = n * 2^{\ell_s - 2} * (\ell - \ell_s + 1)$

Nadalje, uvodi se pretpostavka da je broj jedinki u populaciji $n = 2^{(\ell_s / 2)}$. Ova pretpostavka unekoliko umanjuje opštost daljeg razmatranja, ali omogućuje da rezultati budu iskazani u preglednijoj formi:

Stav 11. Broj shema koje obradjuje populacija jei $n_s = (n^3 * (\ell - \ell_s + 1))$ / 4, tj. broj shema koje u datom trenutku bivaju obradene od strane GA je proporcionalan kubu veličine populacije.

Tvrdenje poslednjeg stava, zbog svog značaja, u literaturi ima svoje ime: Teorema o implicitnom paralelizmu.

Treba napomenuti da je ova procena u izvesnoj meri profinjena, a istovremeno i uopštena, tj da je odredena viša donja granica za broj shema koje se sadrže u populaciji ([4]). Kako je analitički izraz za funkciju koja predstavlja donju granicu kod nove procene duži i nepregledniji od Hollandovog rezultata, a već iskazani zaključak je isti, u literaturi se za procenu broja shema koje se sadrže u populaciji i nadalje češće koristi Holland-ov rezultat.

2.3. Hipoteza o gradivnim blokovima

Slika o performansama GA biva mnogo preglednija ukoliko se GA posmatra iz perspektive shema. Kratke sheme malog reda i velike prosečne prilagodenosti se biraju (koriste se još i izrazi uzorkuju, sampliraju), rekombinuju, ponovo biraju, ponovo rekombinuju itd. kako bi oformili niske sa potencijalno većom prilagodenošću. Takve sheme (kratke, malog reda i visokih performansi) nazivaju se gradivni blokovi. Dakle, pri radu GA, umesto da se niske visokih performansi grade pokušavajući sa svakom shvatljivom kombinacijom, grade se sve bolje i bolje niske od najboljih parcijalnih rešenja u prošlim biranjima. Hipoteza o gradivnim blokovima pretpostavlja da GA optimalna ili suboptimalna rešenja dobija na prethodno opisani način: rekombinovanjem gradivnih blokova.

Veliki broj empirijskih rezultata osnažuje ovu hipotezu.

Treba napomenuti i da postoje problemi kod kojih prosti GA zasigurno ne nalazi optimalno (ni suboptimalno) rešenje. Takvi problemi su nazvani GA-obmanjivački (GA-deceptive) i ova vrsta problema predstavlja važan objekt proučavanja medu autorima koji se bave evolucionim programiranjem, odnosno genetskim algoritmima ([15],[16]).

2.4. Noviji teoretski rezultati

Mnogi autori su, u vezi sa prethodno izvedenim rezultatima, naglasili izvesnu statičnost koju nameću polazne pretpostavke, odnosno istakli su da se pri ovakvoj analizu samog GA i njegovih performansi ne uvažava u dovoljnoj meri dinamičnost samog algoritma. Iako se pomoću GA lepo rešavaju odredeni problemi optimizacije funkcija, ne treba smatrati da se GA samo zato koriste. Kritikovan je, dosta često viđen, simplifikovan prilaz problematici GA, u kom se GA posmatra samo kao sredstvo za optimizaciju funkcija ([6]). Naglašava se da je GA i robustan adaptivan sistem sa velikim brojem najraznovrsnijih primena, pa se ukazuje na razlike koje postoje izmedu GA za optimizaciju funkcija i GA za rešavanje drugih problema.

Razvijen je metod analize konvergencije prostog GA koji se oslanja na verovatnosne matrice prelaska, odnosno na Markovljeve lance, i sa takvim pristupom su dobijeni odredeni dosta značajni rezultati ([12],[17],[21],[22]).

Od strane pojedinih autora se osporava tradicionalno tumačenje primarnosti genetskog operatora ukrštanja i sekundarnosti mutacije ([2],[19],[20]).

Hipoteza o gradivnim blokovima doživljava novo prevrednovanje. Od strane nekih autora biva kritikovana zbog svoje statičnosti, dok drugi revalorizuju njene stavove.

Uočena je tzv. kolateralna konvergencija (još se koristi i termin spora konvergencija), koja usložnjava proučavanje obmanjivačkih problema i samog obmanjivanja (desception). Pokazalo se da obanjivački problemi nisu jedini način da GA pri radu ne dođe do korektnog rezultata, tj. da se klasa GA-teških problema ne poklapa sa klasom GA-obmanjivačkoh problema ([10]). Ipak, proučavanje GA obmanjivačkih problema i dalje ima veliki značaj. Jer, dokazano je da svaki od GA-teških problema sadrži deo koji je GA-obmanjivački.

Pored ovog pristupa, pitanju rada genetskog algoritma i hipoteze o gradivnim blokovima se prilazi i preko pejsaža prilagođenosti (fitness landscape) i funkcije kraljevskog puta (royal road functions) tj. posmatra se sposobnost adaptiranja populacije kojom upravlja GA na pejsaž prilagođenosti ([7],[8]).

Neki od novijih rezultata pokazuju da određena poboljšanja GA, koja se uspešno koriste pri rešavanju mnogih problema nemaju opšti značaj.

Takav je, na primer, slučaj sa kodiranjem. Naime, pored već pomenutih operatora selekcije, ukrštanja i mutacije, GA je u ne manjoj meri određen i kodiranjem: - preslikavanjem jedinki realnog problema u odgovarajuće niske. Poželjno je da se to preslikavanje izvrši tako da se niske koje odgovaraju jedinkama sa približno istom prilagođenošću nalaze bliže nego niske koje odgovaraju jedinkama čija se prilagođenost u značajnoj meri razlikuje. Pokazano je, ipak, da je za ma koja dva fiksna kodiranja njihov prosek uspešnosti pri primeni na sve moguće funkcije, isti. Ako ovo primenimo na klasično i Grejovo kodiranje, to znači da postoji jednaki broj funkcija za koje će klasično kodiranje biti bolje od Grejovog, i funkcija za koje će Grejovo kodiranje biti bolje od klasičnog ([1],[3],[23],[24]).

NFL teorema je rezultat sa početka 1995. Njom se tvrdi da se svi algoritmi koji traže ekstremum funkcije cene ponašaju na isti način, kada se uproseče po svim mogućim funkcijama cene. Konkretno, ako algoritam A prevazilazi algoritam B za neke funkciju cene, tada mora postojati jednaki broj drugih funkcija u kojima B prevazilazi A ([27]).

3. Praksa

GA se razvijaju i mnogi autori su predlagali razna poboljšanja. Neke od predloženih izmena su bile skoncentrisane i efikasno primenljive samo na problem kojim se autor-predlagač bavio, dok su drugi predlozi bili opšteprimenjivi. Među istraživačima iz oblasti evolucionog programiranja je praksa da se kvalitet neke modifikacije algoritma procenjuje na osnovu rezultata koje tako modifikovani algoritam postiže u rešavanju problema optimizacije parametara.

Ovakvi empirijski i eksperimentalni prilazi poređenju algoritama imaju mnoge nedostatke, naročito kada se algoritmi dizajniraju tako da budu robusna oruđa za optimizaciju i pretragu. Jedna očigledna opasnost pri empirijskom evaluiranju algoritama pretraživanja je ta da rezultat tj. zaključak zavisi više zavisi od problema koji su uzeti za testiranje nego od algoritama koji se porede. Zbog ovoga se može dogoditi da se algoritmi podešavaju i testiraju da dobro rade na određenom, konkretnom skupu test-funkcija, a da se za druge funkcije ne ponašaju ni izbliza tako kvalitetno.

Dakle, neophodno je razviti metodologiju za empirijsko poređenje raznih GA kojima se rešava određeni problem, kao i za empirijsku proveru da li i gde predloženo poboljšanje zaista daje bolji rezultat.

Treba napomenuti da se u razvoju metodologije za empirijsko poređenje optimizacionih algoritama ne uzimaju u obzir problemi kombinatorne optimizacije. Za takve probleme obično od pre postoje dobro poznati test-primeri. Suštinska težina tih problema i njihova NP kompletnost je mnogo temeljnije dokumentovana nego težina najvećeg broja problema optimizacije parametara.

3.1. De Jongov rad

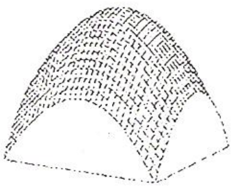
Pionir u zasnivanju te metodologije je Keneth De Jong. Godine 1975, u vreme izdavanja Hollandove knjige, De Jong završava svoju doktorsku disertaciju: "An Analysis of the Behavior of a Class of Genetic Adaptive Systems". Taj će rad imati važno, moglo bi se reći centralno, mesto u daljem razvoju GA.

Disertacija je postala prekretnica u daljem razvoju GA kako zbog povezanosti sa Holland-ovom teorijom shema, tako i zbog pažljivih i preciznih računarskih eksperimenata, te pažljivo razmotrenih zaključaka.

U to vreme se De Jong interesovao za primenu GA u dizajniranju struktura podataka, dizajniranju algoritama i adaptivne kontrole operativnog sistema. Uprkos privlačnosti ovih ezoteričnijih problema, 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 osnovica za sva dalja GA proučavanja i radove.

Već je naglašena potreba da se spreči specijalizacija, tj. prilagođavanje algorima konkretnom skupu test-funkcija. To se postiče izborom skupa test-funkcija (pored naziva test-funkcija koristićemo i naziv test-problem) tako da on bude istovremeno izazovan i raznovrstan. De Jongov skup uključuje test-funkcije sa sledećim karakteristikama: Neprekidne i Prekidne; Konveksne i Nekonveksne; Jednomodalne i Vishemodalne; Kvadratne i Nekvadratne; Determinističke i Stohastičke; Malodimenzionalne i Mnogodimenzionalne.

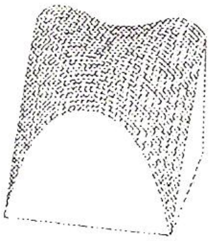
De Jongovo test-okruženje sadrži sledećih pet funkcija:



1) Sferna funkcija: grafik Sferne funkcije je prikazan na slici levo, a desno je prikazan analitički oblik funkcije i njen domen. Ova funkcija ima karakteristike neprekidnosti,

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

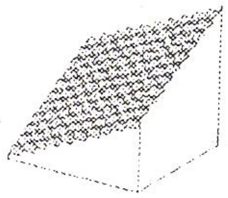
konveksnosti, jednomodalnosti, kvadrata, malodimenzionalnosti i determinističnosti. Dominantna karakteristika ove funkcije je jednomodalnost. Oznaka funkcije je F1.



2) Rozenbrokova funkcija: grafik Rozenbrokove funkcije je prikazan na slici levo, a desno je prikazan analitički oblik funkcije

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

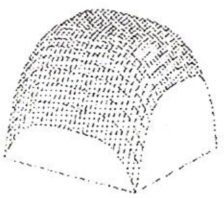
i njen domen. S obzirom na De Jongove karakteristike, ova funkcija je neprekidna, nekonveksna, višemodalna, nekvadratna, malodimenzionalna i deterministička. Dominantna karakteristika Rozenbrokove funkcije je nelinearnost. Oznaka ove funkcije je F2.



3) Stepenična funkcija: grafik dvodimenzionalnog ekvivalenta Stepenične funkcije prikazan je na slici levo, a desno je dat njen analitički oblik i domen. Dominantna karakteristika, (karakteristika zbog koje je ova

$$f_3(x_1, \ldots, x_5) = \sum_{i=1}^{5} \lfloor x_i \rfloor \quad \text{(pri čemu je } -5.12 \le x_i \le 5.12\text{)}$$

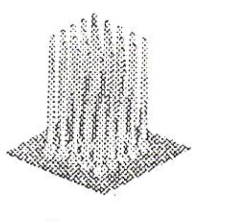
funkcija uključena u skup test-funkcija) je njena prekidnost. Oznaka ove funkcije je F3.



4) Četvorna funkcija sa šumom: grafik dvodimenzionalnog ekvivalenta ove funkcije prikazan je na slici levo, dok je desnom formulom dat njen analitički oblik i domen. Kao što joj samo

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

ime kaže, primarna karakteristika ove funkcije je šum. Veličina šuma je slučajna promenljiva koja ima normalnu raspodelu. Ova funkcija se označava sa F4.



5) Šekelova funkcija: grafik dvodimenzionalnog ekvivalenta ove funkcije prikazan je na slici levo, dok je desnom formulom dat njen analitički oblik i domen. Ova funkcija je uključena u skup test-funkcija zbog svoje višemodalnosti i činjenice da

$$f_5(x_1, x_2) = 0.0002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \quad \text{(pri čemu je } -65.536 \le x_i \le 65.536\text{)}$$

postoji veći broj lokalnih

optimuma. U literaturi se Šekelova funkcija označava sa F5.

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. Novouvedene performanse se definišu na sledeći način:

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

$$x_e^{*}(s) = \frac{1}{T}\sum_{t=1}^{T} f_e^{*}(t)$$

, gde je sa $f_e^*(t)$ označena vrednost najboljeg u skupu $\{f_1(t), f_2(t), \ldots, f_e(t)\}$.

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 (tok) strategije s u okruženju e

$$x_e(s) = \frac{1}{T}\sum_{t=1}^{T} f_e(t)$$

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

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.

- Broj izgubljenih gena tokom probe t. Ova veličina ukazuje na širinu pretrage kroz pretraživački prostor.

Ispitivane su razne varijacije genetskog algoritma. Reproduktivnim planom je DeJong nazvao familiju strategija koje se razlikuju po vrednosti stohastičkih parametara. U svojoj disertaciji, De Jong je razmatrao odnos nekoliko reproduktivnih planova i eksperimentalno određivao vrednosti stohstičih parametara za razne varijacije genetskih algoritama. Naravno, veličine tih parametara su vremenom bivale i ažurirane ukoliko su novi rezultati ukazivali da je tako nešto neophodno.

3.3. Novije metodologije empirijskih poređenja

Tokom godina su pored ovih pet funkcija, u skup test-funkcija uključene još neke funkcije ([5],[11],[14],[18]):

$$f_6(x_1, \ldots, x_N) = 10N + \sum_{i=1}^{N} (x_i^2 - 10\cos(2\pi x_i)) \quad \text{(pri čemu je } -5.12 \le x_i \le 5.11\text{)}$$

$$f_7(x_1, \ldots, x_N) = \sum_{i=1}^{N} -x_i \sin(\sqrt{|x_i|}) \quad \text{(pri čemu je } -5.12 \le x_i \le 5.11\text{)}$$

$$f_8(x_1, \ldots, x_N) = 1 + \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} \cos(x_i/\sqrt{i}) \quad \text{(pri čemu je } -5.12 \le x_i \le 5.11\text{)}$$

$$f_9(x_1, x_2) = 0.5 + \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{|1.0 + 0.1(x_1^2 + x_2^2)|^2} \quad \text{(pri čemu je } -100 \le x_i \le 100\text{)}$$

$$f_{10}(x_1, x_2) = \sqrt[4]{x_1^2 + x_2^2}\left(1.0 + \sin^2(50\sqrt[10]{x_1^2 + x_2^2})\right) \quad \text{(pri čemu je } -100 \le x_i \le 100\text{)}$$

U literaturi su ove funkcije su poznate pod nazivima: F6-Rastriginova (Rastrigin), F7-Švefelova (Schwefel), F8-Grivangkova (Griewangk), F9-sinusna ovojnica sinusni talas, F10-rašireno V sinusni talas.
Ovi test-problemi se često koriste radi poboljšavanja varijanti konkretnog evolutivnog algoritma, kao i u raspravama o prednosti jednog pristupa u odnosu na drugi. Kao što je već istaknuto, postoji opasnost da zbog takve prakse algoritmi mogu biti prilagođeni samo malom skupu test problema. Ta opasnost dobija na značaju ukoliko test-problemi ne predstavljaju tipove problema za koje se evolutivni algoritmi najviše koriste. To kod ovih test-funkcija i jeste najčešće slučaj - postoje drugi tipovi algoritama koji bi za prethodno pobrojane funkcije brže doveli do rešenja.
Učinjen je pokušaj da se, umesto dodavanja novih test-funkcija koje bi ispravile postojeći disbalans, definišu povoljen i nepovoljne osobine koje neka test funkcija treba da ima. Tako se predlaže ([26]) da skup test-funkcija obrazovan od starih i novih test-funkcija treba da zadovoljava sledeće osobine:

1) Skup test-funkcija (ili test-problema) treba sadžati probleme koji su otporni na hill-climbing. Kada je strategija hill-climbing uspešna, tada je ona obično brža od evolutivnog algoritma. Ovo ne znači da treba isključiti iz skupa test-problema one probleme koji se mogu rešiti pomoću hill-climbing-a, ali se pri analizi rezultata za takve probleme mora obratiti pažnja i na tu činjenicu.

2) Skup test-funkcija treba sadžati probleme koji su nelinearni, nerazdvojivi i nesimetrični. Problem se naziva nerazdvojivim ukoliko nije razdvojiv. Problem je razdvojiv (separable) ako nema nelinearne interakcije između promenljivih. Kod razdvojivih funkcija se optimalna vrednost za svaki parametar može izračunati nezavisno od svih ostalih parametara. Funkcija je simetrična ukoliko se zamenom dvaju promenljivih analitički izraz za funkciju ne menja. Za funkciju od N promenljivih može postojati N! ovakvih veza. Eksploatisanjem eventualne simetričnosti funkcije može bitno da se umanji veličina prostora pretraživanja.

3) Funkcije iz skupa test-funkcija trebaju biti skalabilne. Primeri skalabilnih funkcija su funkcije F7 i F8. Svojstvo skalabilnosti omogućava testiranja sa progresivno većim dimenzijama. Treba napomenuti da i nelinearna interakcija u test-funkcijama takođe treba biti osetljiva na skaliranje .

4) Skupovi test-fukcija trebaju sadržati probleme sa skalbilnom cenom evaluacije. Na primer, neki od problema seizmičke interpretacije podataka zbog promene jednog parametra moraju izvršiti $O(N^2)$ evaluacija, pri čemu je N dimenzija problema. Test funkcija treba da odslika i takve slučajeve.

5) Test-problemi trebaju da imaju kanonsku formu. Naime, kada se za testiranje koriste funkcije F1-F10, često se načini kodiranja razlikuju (npr. klasično binarno, Grejovo binarno, realno). Pri testiranju ne samo što treba navesti koje se funkcije koriste, već i kakvo se kodiranje koristi.

4. Literatura

[1] Antonisse Jim "A New Interpretation of Schema Notation that Overturns the Binary Encoding Constraint", in Proceedings of the Third International Conference on Genetic Alghorithms - ICGA '89, pg. 86-91, edited by Schaffer David J, Morgan Kaufmann Publishers Inc, San Mateo, California, 1989.

[2] Back Thomas, Hoffmeister Frank, Schwefel Hans-Paul "A Survey of Evolution Staregies", in Proceedings of the Fourth International Conference on Genetic Alghorithms - ICGA '91, pg. 2-9, edited by Belew Richard K. and Booker Lashon B, Morgan Kaufmann Publishers, San Mateo, California, 1991.

[3] Battle David, Vose Michael D. "Isomorphisms of genetic algorithms", Artificial Intelligence 60, pg. 155-165, 1993.

[4] Bertoni Alberto, Doringo Marco "Implicit parallelism in genetic algorithms", Artificial Intelligence 61, pg. 307-314, 1993.

[5] Davis Lawrence "Bit-Climbing, Representational Bias, and Test Suite Design", in Proceedings of the Fourth International Conference on Genetic Alghorithms - ICGA '91, pg. 18-23, edited by Belew Richard K. and Booker Lashon B, Morgan Kaufmann Publishers, San Mateo, California, 1991.

[6] De Jong Kenneth A. "Genetic Algorithms are NOT Function Optimizers", in Foundation of Genetic Algorithms 2 - FOGA 2, pg. 5-19, edited by Whitley Darrel, Morgan Kaufmann Publishers, San Mateo, California, 1992.

[7] Eshelman Larry J, Caruana Richard A, Schaffer David J. "Biases in Crossover Landscape", in Proceedings of the Third International Conference on Genetic Alghorithms - ICGA '89, pg. 2-9, edited by Schaffer David J, Morgan Kaufmann Publishers Inc, San Mateo, California, 1989.

[8] Forest Stephanie, Mitchel Melanie "Relative Building-Block Fitness and the Building-Block Hypothesis", in Foundation of Genetic Algorithms 2 - FOGA 2, pg. 109-126, edited by Whitley Darrel, Morgan Kaufmann Publishers, San Mateo, California, 1992.

[9] Goldberg David E. "Genetic Alghorithms in Search, Optimization & Machine Learning", Addison-Wesley Publishing Company Inc, Reading, Massachusetts, 1989.

[10] Grefenstette John J, Baker James E. "How Genetic Algorithms Work: A Critical Look at Implicit Parallelism", in Proceedings of the Third International Conference on Genetic Alghorithms - ICGA '89, pg. 20-27, edited by Schaffer David J, Morgan Kaufmann Publishers Inc, San Mateo, California, 1989.

[11] Hancock Peter J. B. "An empirical comparasion of selection methods in evolutionary algorithms", in Evolutionary Computing, pg. 80-94, edited by Fogarty Terece, AISB Workshop Leeds, april 1994.

[12] Herbert Dawid "A Markov Chain Analysis of Genetic Algorithms with a State Dependent Fitness Function", Internal report, Department of Operations Research and Systems Theory, Vienna University of Technology, 1996.

[13] Holland John H. "Adaptation in Natural and Artificial Systems", MIT Press, Cambridge, Massachusetts, 1992.

[14] Janikow Cezary Z, Michalewicz Zbigniew "An Experimental Comparasion of Binary and Floating Point Representations in Genetic Algorims", in Proceedings of the Fourth International Conference on Genetic Alghorithms - ICGA '91, pg. 37-44, edited by Belew Richard K. and Booker Lashon B, Morgan Kaufmann Publishers, San Mateo, California, 1991.

[15] Kalyanmoy Deb, Goldberg David E. "Analyzing Deception in Trap Functions", in Foundation of Genetic Algorithms 2 - FOGA 2, pg. 93-108, edited by Whitley Darrel, Morgan Kaufmann Publishers, San Mateo, California, 1992.

[16] Khuri Sami "Walsh and Haar Funcions in Genetic Algorithms", Internal report, San Jose State University, 1993.

[17] Reynolds David, Gomatam Jagannathan "Stohastic modelling of Genetic Algorithms", Artificial Inteligence 82, pg. 303-330, 1996.

[18] Schaffer David J, Caruana Richard A, Eshelman Larry J, Rajarshi Das "A Study Of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization", in Proceedings of the Third International Conference on Genetic Alghorithms - ICGA '89, pg. 51-60, edited by Schaffer David J, Morgan Kaufmann Publishers Inc, San Mateo, California, 1989.

[19] Schaffer David J, Eshelman Larry J. "On Crossover as Evolutionary Viable Strategy", in Proceedings of the Fourth International Conference on Genetic Alghorithms - ICGA '91, pg. 61-

68, edited by Belew Richard K. and Booker Lashon B, Morgan Kaufmann Publishers, San Mateo, California, 1991.

[20] Spears William M. "Crossover or mutation?", in Foundation of Genetic Algorithms 2 - FOGA 2, pg. 221-239, edited by Whitley Darrel, Morgan Kaufmann Publishers, San Mateo, California, 1992.

[21] Suzuki Joe "A Markov Chain Analysis on Simple Genetic Algorithms", IEEE Transactions on Systems, Man, And Cybernetics, vol.25, no. 4, pg. 655-659, april 1995.

[22] Suzuki Joe "Further Results on the Markov Chain Model of GA and their Application to SA-like Strategy", Internal report, Information Systems Labaratory, Stanford University, 1997.

[23] Vose Michael D. "Generalizing the notion of shema in genetic algorithms", Artificial Intelligence 50, pg. 385-396, 1991.

[24] Vose Michael D. "Modeling Simple Genetic Algorithms", in Foundation of Genetic Algorithms 2 - FOGA 2, pg. 63-73, edited by Whitley Darrel, Morgan Kaufmann Publishers, San Mateo, California, 1992.

[25] Whigham P. A. "A Generalised Schema Theorem for Fixed and Variable-Lenght Structures", Internal report, Department of Computer Science, University College, University of New South Wales, february 1996.

[26] Whitley Darel, Rana Soraya, Dzubera John, Mathias Keith E. "Evaluating evolutionary algorithms", Artificial Intelligence 85, pg. 245-276, 1996.

[27] Wolpert D.H, Macready W.G. "No Free Lunch Theorems for Search", Internal report, Santa Fe Instiute, 1995