

Darko Grundler

Evolucijski algoritmi (I)

Pobude i načela

UDK 519.8:575.8
 IFAC IA 2.0;1.1.0

Pregledni članak

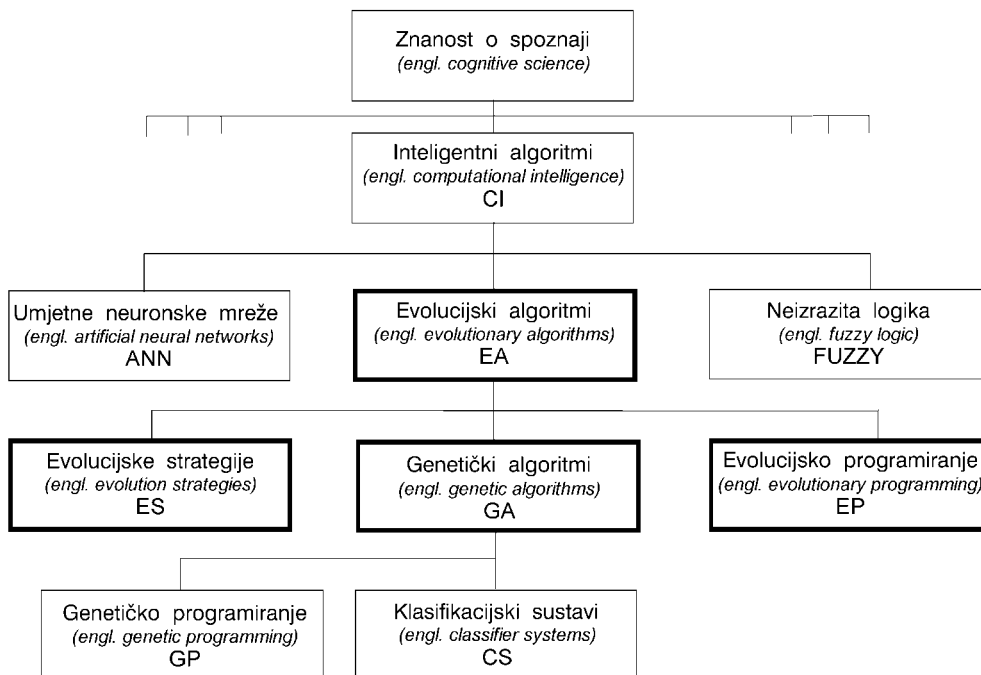
U tekstu je sažeto prikazan povijesni razvoj i osnovna zamisao primjene mehanizma prirodne evolucije pri optimiranju, učenju i modeliranju. Usporedo su objašnjeni glavni pojmovi i postupci prirodne i formalno predočene evolucije poznate pod skupnim imenom evolucijski algoritmi (skraćeno EA). Navedena je podjela inteligentnih algoritama u koje spadaju i EA, te navedene njihove značajke, prednosti i područja primjene. Jednostavnim primjerom ilustrirana je praktična predodžba pojmova i postupci EA.

Ključne riječi: evolucija, evolucijski algoritmi, fenotip, genotip, izbor, jedinka, optimiranje, populacija, primjerenost

1. UVOD

Evolucijski algoritmi (engl. *evolutionary algorithms, evolutionary computing*) su postupci optimiranja, učenja i modeliranja koji se temelje na mehanizmu prirodne evolucije. To su formalni sustavi koji nastoje biti izomorfni s prirodnom evolucijom. Evolucijski algoritmi nastali su iz dviju pobuda: želje za boljim razumijevanjem prirodne evolucije i

pokušaja primjene načela prirodne evolucije pri rješavanju različitih zadataka. Pedesetih godina javljaju se prve zamisli o primjeni evolucijskih načela pri rješavanju tehničkih zadataka [1, 2, 3, 4], ali su zbog skromnih mogućnosti računala tog doba te zamisli ostale neprimijenjene i nepoznate široj znanstvenoj javnosti. Šezdesetih godina neovisno su razvijena tri postupka zasnovana na načelima biološke evolucije:



Sl. 1. Podjela inteligentnih algoritama

evolucijsko programiranje (engl. *evolutionary programming*, EP) koga su razvili L. J. Fogel, Owens, i Walsh (1966.), *evolucijske strategije* (engl. *evolution strategies*, ES) koje su razvili Rechenberg (1973.) i Schwefel (1977.) i *genetički algoritmi* (engl. *genetic algorithms*, GA) koje je razvio J. H. Holland (1975.). Katkad se kao posebne inačice navode: genetičko programiranje (engl. *genetic programming*, GP) koje je razvio Koza (1980.) i klasifikacijski sustavi (engl. *classifier systems*, CS) koje je razvio J. H. Holland (1975.) [5, 6, 7, 8, 9]. Devedesetih godina evolucijski algoritmi se naglo razvijaju i primjenjuju u mnogim područjima, što se može potkrijepiti brojem objavljenih radova, konferencija, časopisa [10] i lokacija na Internetu [11, 12].

Evolucijski algoritmi spadaju u šire područje znanosti o spoznaji (engl. *cognitive science*), a u uže područje inteligentnih algoritama (engl. *computational intelligence*) [23], kako je to prikazano na slici 1.

2. BIOLOŠKA EVOLUCIJA

Evolucija je kamen temeljac moderne biologije [13] i smatra se znanstveno utemeljenom teorijom, kojoj nema ravnopravne teorije o razvoju života na Zemlji s gledišta suvremene znanosti. Među prvima koji su se proučavanjem razvoja života ozbiljnije znanstveno bavili su: Pierre de Maupertuis (1698.–1759.), Erasmus Darwin (1731.–1802.), Chevalier de Lamarck (1744.–1829.), Gregor Johann Mendel (1822.–1884.) i Alfred Russel Wallace (1823.–1913.). Posebno treba istaći Charlesa Darwina (1809.–1882.) čija je zasluga dvojaka: prvi je prikupio mnoštvo dokaza da evolucija postoji i prvi je ponudio prihvatljivo objašnjenje njezinog uzroka. Najutjecajnije je djelo u tom pogledu Darwinova knjiga *O porijeklu vrsta* (naslov originala *On the Origin of Species by Means of Natural Selection; Preservation of Favoured Races in the Struggle for Life*), koja je objavljena 1859. godine [14, 15]. Za daljnji razvitak evolucijske teorije, njezinu formalizaciju i matematičko modeliranje zaslužni su: Sir R. Fisher (1890.–1962.), J. B. S. Haldane (1892.–1964.), S. Wright (1889.–1988.) i dr.

Teorija biološke evolucije temelji se na nekoliko opće prihvaćenih pretpostavki.

a) Postoji naraštaj (populacija, engl. *population*) koji može stvarati potomke.

U biologiji *naraštaj* je skup živih organizama iste vrste koji borave na ograničenom području u određenom trenutku. Za mehanizam evolucije nužno je postojanje naraštaja više jedinki koje se mogu razmnažati. Kako se evolucija temelji na prirodnom izboru, nužno je postojanje naraštaja s više jedinki različitih značajki. Kod evolucijskih algoritama populacija je skup neovisnih matematičkih objekata (npr. skup realnih brojeva, skup binarnih brojeva, skup vektora i sl.).

b) Naraštaj se razmnožava (stvara potomke, rekombinira se, engl. *recombine*).

U biologiji novi naraštaj nastaje iz postojećeg postupkom razmnožavanja (rekombinacije). Pri tom novi naraštaj nasljeđuje značajke svojih prethodnika. Kod evolucijskih algoritama *rekombinacija* (razmnožavanje) je stvaranje novih matematičkih objekata na temelju postojećih uporabom prikladnih matematičkih operatora.

c) Događaju se slučajne promjene značajki jedinki koje uzrokuju varijacije, tj. nejednakosti među jedinkama (mutacija, engl. *mutation*).

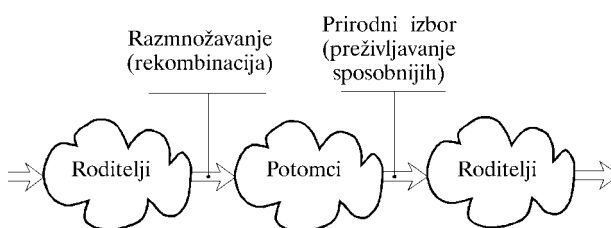
U biologiji se nasljedne značajke jedinke mogu mijenjati samo i jedino mutacijom, tj. slučajnom promjenom koja nastaje pri nasljeđivanju. Zbog *mutacije* može se dogoditi da neki od potomaka imaju značajke koje nema niti jedan od njihovih prethodnika. Kod evolucijskih algoritama *mutacija* je slučajna promjena matematičkog objekta koja nastaje stohastičkom primjenom odgovarajućeg matematičkog operatora na jedinku.

d) Razmnožavati se, a time i prenijeti nasljeđivanjem svoje značajke, mogu samo neke, a nikako sve jedinke.

To se zove *prirodni izbor* (engl. *natural selection*) ili preživljavanje sposobnijih (engl. *survival of the fittest*). U biologiji okoliš dopušta preživljavanje samo dijela, a ne svih jedinki naraštaja. Pretpostavlja se da će preživjeti one koje su bolje prilagođene okolišu. Pod preživljavanjem se podrazumijeva prosljeđivanje nasljednih značajki u sljedeći naraštaj. Kod evolucijskih algoritama izbor je odbacivanje nekih matematičkih objekata populacije. Okoliš kod evolucijskih algoritama jest zadaća koja se rješava.

e) Postupak nastanka novog naraštaja se opetuje, tj. postupak promjene je iterativan.

U biologiji promjene značajki cijelog naraštaja i stvaranje novih vrsti nastaje tijekom mnogih naraštaja, tj. opetovanog postupka razmnažanja i prirodnog izbora (slika 2). Kod evolucijskih algoritama postupak rekombinacije, mutacije i izbora se opetuje, sve dok se ne postigne zadovoljavajuće rješenje postavljene zadaće ili ne zadovolji neki drugi kriterij za završetak algoritma.



Sl. 2. Shematski prikaz evolucije

Rašireno je mišljenje da je evolucija napredak. To je pogrešno mišljenje jer evolucijom nastaje naraštaj koji po svojim značajkama nije ni bolji ni lošiji od prethodnog u općem smislu te riječi, već je samo prilagođeniji svom okolišu.

2.1. Genotip i fenotip

Britanski biolog William Bateson (1861.–1926.) utemeljio je i nazvao znanost poznatu danas pod nazivom genetika. Genetika je znanost koja proučava na koji se način prenose fizička, biokemijska i psihička svojstva s roditelja na potomke. *Genotip* nije fizički objekt već je to informacija, svojstvo, značajka, nacrt budućeg organizma. Slikovito se može reći da je genotip uputa ili recept o tome kako treba izgledati organizam koji se razvija na temelju te informacije. U širem smislu genotip je i genetska građa organizma: broj, vrsta i raspored gena jedinice. I u tom smislu genotip je informacija na temelju koje se može zaključiti o značajkama organizma. Značajke sadržane u genotipu redovito se razlikuju od značajki vanjske pojavnosti organizma, koja je uvjetovana i utjecajima okoliša [16, 17, 18, 19].

Zamislili se kromosom kao poredani niz gena, tada je svaki pojedini gen određen svojim položajem u tom nizu i svojim kemijskim sastavom. Označka položaja gena zove se *lokus* (engl. *locus*) gena, a njegov oblik, tj. kemijski sastav *alela* (engl. *allele*). U biologiji alela je naziv za različit sastav gena za istu značajku organizma, primjerice kod čovjeka boja očiju, boja kose itd. *Gen* je jednoznačno određen svojim položajem u kromosomu (lokusom) i oblikom (alelom).

Pojam gena i genotipa može se zorno ilustrirati analogijom s magnetskim diskom računala. Treba jasno razlučiti pojam gena, koji su fizički objekti, kao što su to primjerice magnetske čestice u magnetskom disku i genotipa koji je informacija, kao što su to npr. podaci pohranjeni na magnetskom disku. Iako su podaci pohranjeni kao magnetizirane čestice na površini diska, genotip u užem smislu nisu te čestice, već informacija koju one predočuju.

Važna značajka genotipa jest sposobnost prijenosa informacije između generacija, tj. s roditelja na potomka. Analogija nasljeđivanju bi kod magnetskog diska bilo kopiranje podataka s jednog diska na drugi. Takvim postupkom kopija (potomak) nasljeđuje ili poprima značajke originala (svog roditelja) s gledišta pohranjenih podataka.

U biologiji *fenotip* je naziv za vanjska obilježja (osobine, svojstva, odlike, karakteristike) nekog organizma po kojima se razlikuje od drugih pripadnika iste vrste. To su obilježja koja se mogu izvana opažati, mjeriti i sl. To su morfološke, fiziološke, biokemijske i druge značajke vezane za fizičku pojavnost organizma, primjerice, veličina, boja, temperatura i

sl. Iako pojam fenotipa obuhvaća i značajke ponašanja, primjerice potištenost, sve su te značajke posljedica fizičke građe ili kemijskog sastava organizma.

Fenotip je rezultat međusobnog djelovanja nasljeđa (genotipa) i vanjskih čimbenika. Moguće je primjerice da se dva organizma istog genotipa zbog različite prehrane razviju u organizme različitih vanjskih obilježja, tj. različitog fenotipa. Poslužimo li se analogijom s magnetskim diskom, jedan te isti program snimljen na disku (genotip) različito će se izvršiti ovisno o računalu na kojem se izvodi (okoliš). Primjerice, na jednom računalu će prikaz biti u boji, a na drugom jednobojni, na jednom računalu će se program izvršavati brže, a na drugom sporije, na nekim vrstama računala se program neće uopće moći izvršiti, itd. Jedan te isti program (genotip) će se ovisno o okolišu (računalu) korisniku sasvim različito prikazati (fenotip).

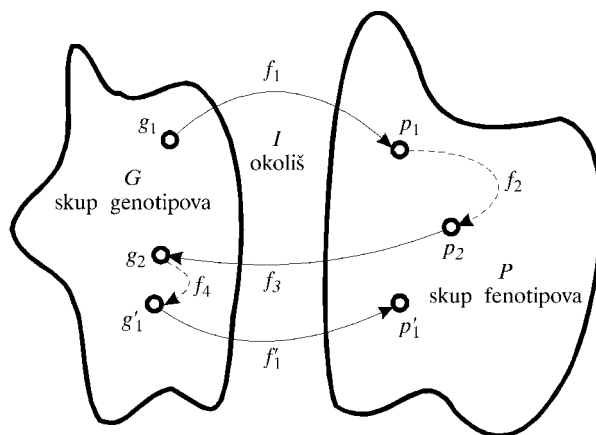
Lewontin [20] je formalno prikazao razliku genotipa i fenotipa definirajući dva skupa: skup informacija G (skup genotipova naraštaja) i skup pojavnosti P (skup fenotipova naraštaja), kako je to prikazano na slici 3. Četiri funkcije upravljaju elementima skupova G i P . Prema [21] odnos genotipa i fenotipa se može formalno prikazati na sljedeći način:

$$f_1: I \times G \rightarrow P \quad (1)$$

$$f_2: P \rightarrow P \quad (2)$$

$$f_3: P \rightarrow G \quad (3)$$

$$f_4: G \rightarrow G \quad (4)$$



Sl. 3. Formalni prikaz genotipa i fenotipa

Funkcija f_1 – epigeneza preslikava genotip $g_i \in G$ u skup P kao određeni skup fenotipova p_i koji ovise o utjecaju okoliša (1). To je označeno indeksom i , gdje su $(i_1, i_2, \dots, i_n) \in I$; I je skup svih utjecaja okoliša na preslikavanje. To znači da će se genotip g_1

preslikati u fenotip p_1^1 uz utjecaj okoliša i_1 ili u fenotip p_1^2 ako je utjecaj okoliša i_2 , itd. Preslikavanje je ovisno o genotipu i utjecajima okoliša.

Funkcija f_2 – izbor preslikava skup fenotipova p_1 u skup fenotipova p_2 (2). Funkcija f_2 predočuje izbor jedinki koje će preživjeti u sljedeću generaciju. Kako izbor djeluje samo u skupu fenotipova [22], funkcija f_2 djeluje samo unutar skupa P .

Funkcija f_3 – genotipsko preživljavanje predočuje razmnožavanje i nasljeđivanje, tj. preživljavanje genotipa (3). Funkcija f_3 preslikava fenotip u genotip koji će sudjelovati kod stvaranja potomaka.

Funkcija f_4 – mutacija preslikava genotip $g_2 \in G$ u genotip $g'_1 \in G$ (4). Ova funkcija predočuje sve genetičke promjene koje mogu nastati na genotipu, najčešće uzrokovanih slučajnom i nepredvidivom promjenom gena. Stvaranjem nove generacije genotipova g'_1 završava prethodna generacija i nastaje nova. Evolucijska prilagodba nastaje uzastopnim kružnim ponavljanjem opisanih koraka preslikavanja.

2.2. Prirodni izbor

Ovisno o prilagodbi okolišu neke jedinke naraštaja će stvoriti više potomaka od drugih. Ako se to ponavlja tijekom više generacija, povećat će se zastupljenost genotipova jedinki koje stvaraju relativno više potomaka. Ta se pojava naziva *prirodni izbor* ili preživljavanje sposobnijih. Prirodni izbor je jedini mehanizam evolucije koji vodi k boljoj prilagodbi naraštaja okolišu. Na prilagodbu prirodnim izborom može se gledati kao na proces kojim fenotipovi daju prednost nasljeđivanju onih gena koji poboljšavaju fenotipove s gledišta sposobnosti preživljavanja u svom okolišu i tako unapređuju gene s gledišta prilagodbe okolišu. Prirodni izbor djeluje pomoću gena ali posredstvom fenotipova.

3. ZNAČAJKE EVOLUCIJSKIH ALGORITAMA

Evolucijski algoritmi spadaju u širu skupinu inteligentnih algoritama. Glavne značajke inteligentnih algoritama su [24]:

- prilagodljivost okolišu;
- robustnost;
- numerička ili znakovna predodžba znanja;
- oponašanje postupaka iz prirode;
- moгуćnost dobivanja dobrih približnih rješenja uz prihvatljiv utrošak vremena i sklopovlja;
- moгуćnost rješavanja zadataka teško rješivih determinističkim postupcima;
- široko područje primjene.

Uz navedene značajke, za evolucijske algoritme je svojstveno:

- Implicitni paralelizam. Evolucijski algoritmi pretražuju prostor rješenja za sve parametre odjednom, tj. implicitno su paralelni.
- Stohastička narav algoritma. Za pronalaženje rješenja evolucijski algoritmi rabe stohastičke a ne determinističke postupke. Evolucijski algoritmi iterativnim postupkom traže novo, potencijalno bolje rješenje na temelju postojećeg rješenja uporabom operatora izbora, mutacije i rekombinacije rabeći pri tom stohastičke postupke. Stohastičnost je važna značajka koja bitno razlikuje evolucijske algoritme od determinističkih postupaka optimiranja.
- Pronalaženje globalnog ekstrema. Svojstvo pronalaženja globalnog ekstrema neovisno o vrsti zadatka koji se rješava jedna je od temeljnih odlika evolucijskih algoritama, a posljedica je stohastične naravi algoritma. To je svojstvo inherentno evolucijskom algoritmu i nisu potrebni posebni zahvati niti promjena algoritma za pronalaženje globalnog ekstrema. Kod neevolucijskih metoda osnovni se postupci proširuju kako bi se omogućilo pronalaženje globalnog ekstrema (npr. ponavljanje postupka s različitim polaznih točaka).
- Jednostavno sučelje i proširivost. Evolucijske algoritme jednostavno je sučeliti postojećim matematičkim modelima, simulacijama i stvarnim procesima, a jednostavno ih je proširivati i dograđivati. U načelu sučeljavanje se sastoji od predodžbe rješenja u obliku prihvatljivom evolucijskom algoritmu i obrnutom postupku pretvorbe te predodžbe nakon provedbe evolucijskog algoritma u oblik prihvatljiv izvornoj predodžbi zadatka.
- Moguća hibridizacija. Evolucijske algoritme jednostavno je povezati s drugim postupcima optimiranja u hibridne sustave. Hibridizacija se obično provodi tako da se evolucijski algoritam rabi za pronalaženje približnog globalnog rješenja, a zatim se pronalaženje točnijeg rješenja u okolišu tog približnog rješenja prepušta nekom od determinističkih postupaka.
- Optimiranje izvorne funkcije. Evolucijski algoritmi rade izravno s izvornom funkcijom a ne s nekim izvedenim svojstvom funkcije. Pod izvornom funkcijom podrazumijeva se ovdje izvorni matematički model u kome je zadatak predočen. Mnogi deterministički postupci rabe neko od svojstava funkcije, primjerice derivaciju funkcije.
- Algoritam je jednostavan za primjenu i bez većih izmjena je primjenjiv na raznovrsne zadatke, što neki autori smatraju glavnim razlogom popularnosti evolucijskih algoritama [25, 26]. Konceptijska jednostavnost i jednostavno ostvarenje pomoću računala važne su prednosti evolucijskih algoritama. Evolucijski algoritmi primjenjivi su praktično na svaki zadatak koji se može formulirati kao optimiranje.

h) Nema jamstva da će algoritam pronaći apsolutno najbolje rješenje (apsolutni ekstrem). Evolucijski algoritmi pronalaze približno rješenje, što je posljedica stohastičnosti i temeljnog mehanizma algoritama. Taj nedostatak je za mnoge praktične zadatke relativan. Naime, ako ostali postupci ne mogu uopće pronaći rješenje, onda je približno rješenje evolucijskog algoritma najbolje raspoloživo rješenje.

Pri izboru metode optimiranja treba uzeti u obzir činjenicu da svaki postupak optimiranja ima područje kome je primjeren i područje u kome nije primjeren. Zadaci koji nisu prikladni za rješavanje evolucijskim algoritmom su [27]:

- zadaci s malo parametara koji su zadovoljavajuće rješivi determinističkim postupcima;
- zadaci koji su rješivi posebno razvijenim postupcima i daju dobre rezultate (takav namjenski postupak je redovito djelotvorniji od rješavanja pomoću evolucijskog algoritma).

Uz to valja imati na umu da je za zadatke koji ne spadaju u dvije navedene skupine potrebno pronaći odgovarajuću predodžbu zadatka prikladnu za rješavanje genetičkim algoritmom, te da je potrebno izabrati parametre genetičkog algoritma: veličinu populacije, način rekombinacije i izbora, način i učestalost mutacije, itd. Postoje samo neke načelne smjernice kako to učiniti, pa to može biti dodatni izvor poteškoća pri primjeni evolucijskih algoritama.

4. TEMELJNI ALGORITAM

Postoji mnogo inačica evolucijskih algoritama, koje se razlikuju s gledišta formaliziranja spoznaja o prirodnoj evoluciji. Sve se one mogu predočiti pseudokodom [26]:

```
t = 0;
initialize P(t);
evaluate P(t);
while not terminate do
    P'(t) = variation [P(t)];
    evaluate [P'(t)];
    P(t+1) = select [P'(t) ∪ Q(t)];
    t = t+1;
od
```

gdje je:

- $t \in \mathbb{N}$, oznaka (indeks) generacije (iteracije);
- $P(t)$ populacija generacije t ; skup koji ima μ jedinki u iteraciji t (skup roditelja);
- $P'(t)$ populacija potomaka generacije t potekla iz populacije $P(t)$, skup koji ima λ jedinki u iteraciji t (skup potomaka);

$Q(t) \subseteq P(t)$, skup jedinki roditelja koje mogu biti birane za sljedeću generaciju (samo neke jedinke populacije $P(t)$ mogu biti birane kao roditelji sljedeće populacije $P(t+1)$);

initialize operator stvaranja početne populacije;

evaluate operator procjene primjerenosti jedinki populacije;

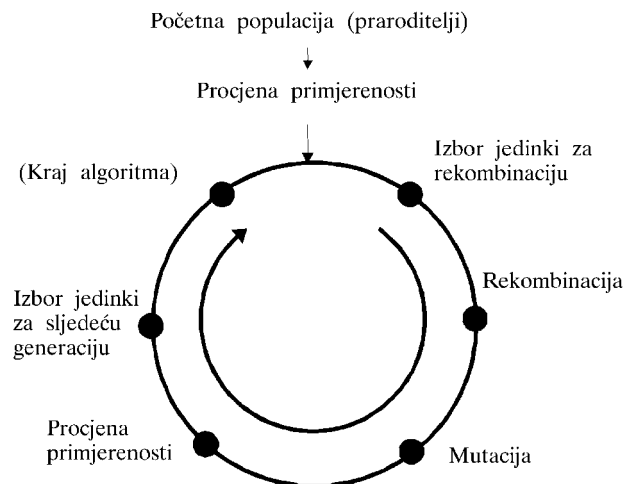
terminate uvjet završetka algoritma;

variation operator promjene genotipa jedinki;

select operator izbora jedinki za sljedeću generaciju.

Pojedine vrste evolucijskih algoritama razlikuju se uglavnom u:

- predodžbi jedinki;
- operatorima initialize, variation i select. Evolucijski algoritam može se prikazati iteracijskom petljom prikazanom na slici 4. U petlji je operator variation raščlanjen na: operator izbora jedinki za rekombinaciju, operator rekombinacije i operator mutacije.



Sl. 4. Iteracijska petlja evolucijskog algoritma

4.1. Jedinka

Kod evolucijskih algoritama *jedinka* populacije (engl. *individual*) predočena je neovisnim matematičkim objektom, primjerice: realnim brojem, višedimenzijalnim vektorom ili uređenim nizom znakova. Izbor predodžbe jedinke bitno utječe na djelotvornost algoritma ili čak na mogućnost pronalaženja zadovoljavajućeg rješenja. Četiri su glavne skupine predodžbi jedinke.

- Predodžba jedinke prilagođena zadatku. Kod ove vrste predodžbe polazi se od pretpostavke da predodžba jedinke treba biti što prirodnija zadatku koji se rješava.

- b) Predodžba jedinke prilagođena evolucijskom algoritmu. Time se postiže općenitost algoritma jer je u načelu algoritam neovisan o zadatku.
- c) Posebna predodžba. Ta se mogućnost koristi kad rješenje zadatke nije moguće prikazati brojanom vrijednosti. To je, primjerice, zadaća čije je rješenje određena kombinacija stanja ili određeni raspored znakova.
- d) Prilagodljiva predodžba jedinke. Jedna od mogućnosti koja se još istražuje jest prilagodljiva predodžba, tj. takva predodžba koja se mijenja tijekom algoritma.

Primjerice, neka je zadatak pronaći maksimum dvodimenzionalne Rastrigin funkcije:

$$F(x_1, x_2) = 2 + x_1^2 - \cos(18 \cdot x_1) - \cos(18 \cdot x_2), \quad (5)$$

koja ima mnogo lokalnih ekstrema i globalni ekstrem $x_{\min} = (0,0)$, $F(0,0) = 0$. Jedinica bi u tom primjeru bila dvodimenzionalni vektor čije su komponente realni brojevi $\mathbf{x} = [x_1 \ x_2]$. Te komponente mogu biti realni brojevi, pa neka jedinka može npr. biti $\mathbf{x}_a = [1,2 \ 0,8]$. U tom je primjeru vrijednost varijable $x_1 = 1,2$ i vrijednost varijable $x_2 = 0,8$. Moguća je i drukčija predodžba vrijednosti varijabli, a jedna od često rabljenih je predodžba binarnim brojevima. Jedinica čije su komponente binarni brojevi može biti npr. $\mathbf{x}_a = [11011 \ 10001]$.

4.2. Populacija

U biologiji se pod *naraštajem* (populacijom) najčešće podrazumijeva ukupan broj ljudi koji borave unutar određenog područja u određenom trenutku. To je, primjerice, ukupan broj stanovnika nekog grada, države ili Zemlje u određenom trenutku. U širem smislu to je skup živih organizama iste vrste koji borave na ograničenom području u određenom trenutku. Obično se pod istom vrstom podrazumijevaju živa bića koja se mogu međusobno razmnažati. Naraštaj je zemljopisno ili razlikama u anatomiji i ponašanju u određenoj mjeri odvojen od ostalih jedinki svoje vrste. Npr., populacija Europe je u davnjoj prošlosti bila odvojena od populacije Afrike i jedinke te dvije populacije su malokad imale prilike stvarati potomstvo.

Kod evolucijskih algoritama populacija je konačni skup jedinki (matematičkih objekata):

$$P_\mu = (x_1, \dots, x_\mu), \quad x_i \in X, \quad i = 1, 2, \dots, \mu, \quad (6)$$

gdje je:

- P populacija,
- μ veličina populacije (broj jedinki populacije),
- x_i jedinka populacije (matematički objekt),
- i indeks jedinke,
- X skup svih jedinki (redovito skup istovrsnih matematičkih objekata).

Populacija (6) je skup jedinki (roditelja, engl. *parent*) iz kojih mogu nastati nove jedinke (potomci, engl. *offspring*). Jedinke populacije x_i najčešće su istovrsne glede strukture, no mogu biti i međusobno različite.

Veličina populacije se označava sa μ ($\mu \geq 1$, $\mu \in N$) i konačna je. Može biti nepromjenljiva tijekom algoritma (engl. *steady state*, *steady population*) ili se može mijenjati (engl. *dynamic population*) [28, 29]. Zbog jednostavnosti algoritma kod većine evolucijskih algoritama veličina populacije je konstantna i u rasponu $2 \leq \mu \leq 10\,000$.

Primjerice, neka je zadatak pronaći maksimum Rastrigin funkcije (5) i neka je veličina populacije $\mu = 5$. Nasumce stvorena početna populacija bi mogla biti:

$$\{\mathbf{x}_a = [1,2 \ 0,8], \mathbf{x}_b = [0,3 \ 1,1], \mathbf{x}_c = [2,2 \ 2,4], \mathbf{x}_d = [0,9 \ 0,0], \mathbf{x}_e = [3,6 \ 0,1]\}. \quad (7)$$

4.3. Generacija

Evolucijski algoritam je iterativni postupak čije se iteracije ili generacije mogu poredati počevši od početne ili nulte pa do završne ili zadnje:

$$P_\mu^0 \Rightarrow P_\mu^1 \Rightarrow \dots \Rightarrow P_\mu^n, \quad (8)$$

gdje oznaka u eksponentu označava redoslijed generacije, a oznaka \Rightarrow označava da generacija P_μ^{i+1} proizlazi iz generacije P_μ^i . Početna, nulta ili inicijalna populacija P_μ^0 može nastati na dva načina: slučajnim generiranjem jedinki populacije ili generiranjem jedinki na temelju spoznaja o zadatku koju treba riješiti evolucijskim algoritmom [30, 31].

U primjeru Rastrigin funkcije (5) to bi se moglo ilustrirati sljedećim primjerom:

$$\begin{aligned} &\{\mathbf{x}_a = [1,2 \ 0,8], \mathbf{x}_b = [0,3 \ 1,1], \mathbf{x}_c = [2,2 \ 2,4], \\ &\mathbf{x}_d = [0,9 \ 0,0], \mathbf{x}_e = [3,6 \ 0,1]\}^0 \Rightarrow \\ &\{\mathbf{x}_a = [2,1 \ 4,2], \mathbf{x}_b = [7,1 \ 9,1], \mathbf{x}_c = [0,4 \ 3,0], \\ &\mathbf{x}_d = [5,5 \ 2,9], \mathbf{x}_e = [3,6 \ 3,1]\}^1 \Rightarrow \\ &\{\mathbf{x}_a = [2,1 \ 6,4], \mathbf{x}_b = [4,3 \ 3,1], \mathbf{x}_c = [0,2 \ 4,4], \\ &\mathbf{x}_d = [1,5 \ 1,6], \mathbf{x}_e = [0,7 \ 2,5]\}^2 \Rightarrow \\ &\text{itd.,} \end{aligned}$$

gdje oznaka u eksponentu vitičaste zagrade označava redni broj slijednih generacija. Početna ili nulta populacija je nastala nasumce. Način na koji jedna generacija proizlazi ili nastaje iz prethodne ovisi o vrsti i parametrima evolucijskog algoritma, a u ovom trenutku dovoljno je znati da generacija $\{\}^{n+1}$ nastaje na temelju generacije $\{\}^n$.

4.4. Primjerenost

U biologiji je *primjerenost* (prikladnost, doličnost, shodnost, engl. *fitness*, performance index) značajka jedinke koja opisuje njezinu prilagođenost okolišu. Primjerenost živih organizama ovisi o okolišu, o pripadnicima iste vrste i drugih vrsta koje ga okružuju.

Primjerenost jedinke kod evolucijskih algoritama jest značajka jedinke koja opisuje kvalitetu rješenja zadaje uporabom te jedinke. Primjerenost jedinke se najčešće izražava realnim brojem, a računa se pomoću funkcije primjerenosti f_p :

$$f_p: P \Rightarrow \mathfrak{R}, \quad (9)$$

gdje je P populacija. Postupak izračunavanja primjerenosti zove se *prosudba jedinke* (engl. *evaluation*, *trial*).

Izbor funkcije primjerenosti je presudan i bitno određuje rezultat algoritma. Funkcija primjerenosti predočuje zadaću koja se želi riješiti algoritmom. Ako se najpovoljnija jedinka glede rješenja zadaje proglasi optimalnom jedinkom, a postupak traženja te jedinke optimiranja, funkcija primjerenosti jest kriterij optimiranja.

U primjeru Rastrigin funkcije (5) primjerenost jest vrijednost Rastrigin funkcije. Kako se traži minimum funkcije, primjerenost je to bolja što je manja vrijednost Rastrigin funkcije. Primjerice za jedinku $\mathbf{x}_a = [1,2 \ 0,8]$ primjerenost je $F(\mathbf{x}_a) = F([1,2 \ 0,8]) = 4,6$. Uobičajeno je funkciju primjerenosti urediti tako da poprma samo pozitivne vrijednosti i da veća vrijednost znači bolju primjerenost. U našem primjeru takva bi funkcija primjerenosti f_p mogla biti:

$$f_p = e^{-F(x_1, x_2)} \quad (10)$$

Tako je osigurano da je vrijednost funkcije primjerenosti uvijek pozitivna i ograničena $0 < f_p < 1$. U našem primjeru $f_p(\mathbf{x}_a) = 0,0101$. Moguće je izabrati u neku drugu funkciju primjerenosti, no svakako treba imati na umu da ona mora odražavati mjeru kvalitete rješenja zadatka.

4.5. Mutacija

U biologiji *mutacija* je prirodno nastala ili umjetno proizvedena promjena nasljednih svojstava organizma. Prirodno nastala mutacija redovito je slučajna. Kod evolucijskih algoritama mutacija je stohastička promjena jedinki populacije. To je bilo koja matematički izvediva operacija nad jedinkama koja se naziva operator mutacije. Mutacija djeluje samo na neke jedinke ili na cjelokupnu populaciju. Bitna je značajka mutacije njezina stohastička narav.

Postoje različite preporuke glede vjerojatnosti pojave mutacije p_m , primjerice $p_m = 0,001$ [32], $0,005 <$

$< p_m < 0,01$ [33], $p_m = 1/k$, gdje je k broj znamenki binarnog broja koji predočuje jedinku [34].

U primjeru Rastrigin funkcije (5) i jedinke $\mathbf{x}_a = [1,2 \ 0,8]$ mutacija može biti dodavanje nasumce izabranog realnog broja svakoj od komponenata vektora, npr:

$$\mathbf{x}_a + \mathbf{m} = [1,2 \ 0,8] + [-0,5 \ 1,1] = [0,7 \ 1,9],$$

gdje je \mathbf{m} vektor čije su komponente nasumce izabrane.

Mutacija nikad nije usmjerena, što znači da nije ni na koji način uvjetovana prethodno pronađenim rješenjima ili prethodnim postupcima. To je uvijek u potpunosti stohastičan postupak na koji se može utjecati izborom učestalosti primjene mutacije (na sve ili samo neke jedinke) i iznosom promjene (npr. nasumičan izbor komponenti vektora \mathbf{m} prema normalnoj razdiobi).

4.6. Rekombinacija

Rekombinacija je u biološkom smislu razmnožavanje ili reprodukcija. Kod evolucijskih algoritama *rekombinacija* (engl. *recombination*) je postupak stvaranja novih jedinki populacije na temelju jedne ili više postojećih jedinki. Matematička operacija kojom nastaju nove jedinke na temelju postojećih zove se operator rekombinacije. Operator rekombinacije može se primijeniti na sve jedinke populacije, ili samo na stohastički izabrane jedinke populacije. U potonjem slučaju vjerojatnost primjene rekombinacije p_r je redovito u rasponu $0,7 < p_r < 0,95$ [35].

Neka su u primjeru Rastrigin funkcije (5) nasumce izabrane dvije jedinke za rekombinaciju: $\mathbf{x}_a = [1,2 \ 0,8]$ i $\mathbf{x}_b = [0,3 \ 1,1]$. Jedan od načina stvaranja nove jedinke (potomka) na temelju postojeće dvije jedinke (roditelja) je postupak križanja, nalik na izmjenu gena kod živih bića. Potomak križanjem nastaje tako da potomak od svakog roditelja naslijedi jednu komponentu: $\mathbf{x}_{ab} = [1,2 \ 1,1]$ ili $\mathbf{x}_{ba} = [0,3 \ 0,8]$. Postoji mnogo postupaka rekombinacije, a svima je zajedničko da potomak mora naslijediti dio genetičkog materijala jednog ili više roditelja (jednog u slučaju tzv. bespolne rekombinacije koja se primjenjuje kod npr. evolucijskih strategija).

4.7. Izbor

Temeljna pretpostavka biološke evolucije je činjenica da mogu preživjeti i razmnažati se samo neke, a nikako sve jedinke. Koje će to jedinke biti, ovisi o njihovim značajkama i okolišu. U prirodi preživljavaju one jedinke koje su bolje prilagođene okolišu. Ako se mjera prilagodbe nazove primjerenost, tada će veću vjerojatnost preživljavanja imati jedinke bolje primjerenosti. Pod većom vjerojatnošću preživljavanja podrazumijeva se da će primjerenije jedinke

imati relativno više potomaka u sljedećoj generaciji, tj. da će njihov genotip biti zastupljeniji. Izbor je postupak kojim se određuje koje će jedinke preživjeti, tj. imati potomstvo u sljedećoj generaciji. U tom smislu preživljavanjem se naziva prijenos određenog genotipa u sljedeću generaciju.

Kod evolucijskih algoritama *izbor* (engl. *selection*) je biranje jedinki za sljedeću iteraciju algoritma sukladno njihovoj primjerenosti. Ako se primjerenost može izraziti brojem, izbor se temelji na numeričkoj usporedbi primjerenosti jedinki. Za razliku od ostalih operatora evolucijskih algoritama, koji djeluju na genotip, operator izbora djeluje na temelju fenotipskih značajki jedinke. Da li će jedinka preživjeti u sljedećoj generaciji, ovisi samo o njezinoj primjerenosti, tj. o tome koliko je rješenje zadatka dobro uz uporabu te jedinke. Kako je rješenje zadatka fenotip jedinke, jedinke koje će preživjeti biraju se uvijek na temelju svog fenotipa.

Postoji mnogo načina na koji je moguće birati roditelje sljedeće generacije. Primjerice, jedan od načina je binarni turnirski izbor. Nasumce se iz populacije biraju dvije jedinke, pa se zatim ona od njih koja ima povoljniju primjerenost prosljeđuje sljedećoj generaciji. Postupak se ponavlja tako dugo dok se ne popuni generacija novih roditelja. Treba uočiti da takav postupak uključuje stohastičnost (nasumični izbor dvije jedinke) i da daje prednost primjerenijoj jedinki (roditelj postaje primjerenija od dvije jedinke). Važna je značajka takvog postupka da niti jedna jedinka, bez obzira na svoju primjerenost, nema jamstva preživjeti. To je važno zato što se na taj način izbjegava usmjeravanje algoritma ka lokalnom ekstremu kome može pripadati naj-primjerenija jedinka trenutne generacije.

Neka je populacija u primjeru Rastrigin funkcije (5):

$$\{\mathbf{x}_a = [2,1 \ 4,2], \mathbf{x}_b = [7,1 \ 9,1], \mathbf{x}_c = [0,4 \ 3,0], \\ \mathbf{x}_d = [5,5 \ 2,9], \mathbf{x}_e = [3,6 \ 3,1]\}.$$

Neka se primjenjuje binarni turnirski izbor i neka su pri prvom nasumičnom izboru izabrane jedinke \mathbf{x}_a i \mathbf{x}_c . U sljedeću generaciju prosljeđuje se kao roditelj jedinka povoljnije primjerenosti. Uz primjenu funkcije primjerenosti f_p prema (10):

$$f_p(\mathbf{x}_a) = 0,0122 < f_p(\mathbf{x}_c) = 0,0940, \\ \text{pa se u sljedeću generaciju prosljeđuje jedinka } \mathbf{x}_c.$$

Za popunu populacije potrebno je postupak ponoviti još četiri puta (zbog veličine populacije $\mu = 5$):

$$f_p(\mathbf{x}_a) = 1,22 \cdot 10^{-02} < f_p(\mathbf{x}_c) = 9,40 \cdot 10^{-02}, \\ \text{pa se u sljedeću generaciju prosljeđuje jedinka } \mathbf{x}_c \\ f_p(\mathbf{x}_d) = 8,96 \cdot 10^{-15} < f_p(\mathbf{x}_e) = 4,96 \cdot 10^{-07}, \\ \text{pa se u sljedeću generaciju prosljeđuje jedinka } \mathbf{x}_e$$

$$f_p(\mathbf{x}_a) = 1,22 \cdot 10^{-02} > f_p(\mathbf{x}_d) = 8,96 \cdot 10^{-15}, \\ \text{pa se u sljedeću generaciju prosljeđuje jedinka } \mathbf{x}_a$$

$$f_p(\mathbf{x}_b) = 3,56 \cdot 10^{-23} < f_p(\mathbf{x}_c) = 9,40 \cdot 10^{-02}, \\ \text{pa se u sljedeću generaciju prosljeđuje jedinka } \mathbf{x}_c$$

Navedene jedinke za međusobnu usporedbu su nasumce birane, a konačna populacija nastala izborom je:

$$\{\mathbf{x}_a = [2,1 \ 4,2], \mathbf{x}_c = [0,4 \ 3,0], \mathbf{x}_c = [0,4 \ 3,0], \\ \mathbf{x}_c = [0,4 \ 3,0], \mathbf{x}_e = [3,6 \ 3,1]\}.$$

Treba uočiti da određena jedinka može biti višestruki roditelj u sljedećoj generaciji, a neke jedinke mogu izostati.

4.8. Genotip i fenotip

Kod evolucijskih algoritama *genotip* je matematički objekt koji predstavlja rješenje zadatka. Drugim riječima, genotip je vrijednost koja je rješenje zadatka. Analogno definicijama kod biološke evolucije, kromosom je matematički objekt, tj. predodžba jedinke. Locus je oznaka položaja gena u kromosomu, a alela oblik gena, obično numerička vrijednost [35]. Kod evolucijskih algoritama *fenotip* je vrijednost ocjene kvalitete rješenja zadatka za određeni genotip. Usporedni pojmovi biološke evolucije i evolucijskih algoritama prikazani su u tablici 1.

Tablica 1. Usporedba prirodne evolucije i EA

Biološka evolucija	Evolucijski algoritmi
Jedinka	Rješenje zadatka (numerička ili znakovna predodžba)
Populacija	Skup mogućih rješenja zadatka, skup jedinki
Primjerenost	Kvaliteta rješenja zadatka
Kromosom	Predodžba jedinke, matematički objekt
Gen	Element matematičkog objekta, parametar
Razmnožavanje i mutacija	Matematički operatori pretraživanja prostora mogućih rješenja
Prirodni izbor	Višestruko korištenje dobrih mogućih rješenja zadatka; uklanjanje loših rješenja zadatka iz populacije

U primjeru Rastrigin funkcije (5) genotip jest jedinka, npr: $\mathbf{x}_a = [2,1 \ 4,2]$, a fenotip te jedinke jest vrijednost funkcije primjerenosti

$$f_p(\mathbf{x}_a) = f_p([2,1 \ 4,2]) = 1,22 \cdot 10^{-02}.$$

Analogno živim bićima fenotip je uvjetovan genotipom (vektor \mathbf{x}_a) i okolišem (funkcija primjerenosti f_p). Iz istog genotipa (vrijednost vektora \mathbf{x}_a) mogu proizaći različiti fenotipovi, ovisno o okolini (različitim funkcijama primjerenosti f_p).

LITERATURA

- [1] G. E. P. Box, **Evolutionary operation: A Method of Increasing Industrial Productivity**. Applied Statistics, Vol. 6, pp. 81–101, 1957.
- [2] A. S. Fraser, **Simulation of Genetic Systems by Automatic Digital Computers**. Australian Journal of Biological Science, 10, pp. 484–491, 1957.
- [3] R. M. Friedberg, **A Learning Machine: Part I**. IBM J., Vol. 2, No. 1, pp. 2–13, 1958.
- [4] H. J. Bremermann, **Optimization Through Evolution and Recombination, in Self-Organizing Systems**. M. C. Yovits et. al. eds., Washington, Spartan, 1962.
- [5] L. J. Fogel, A. J. Owens, M. J. Walsh, **Artificial Intelligence Through Simulated Evolution**. New York, Wiley Publishing, 1966.
- [6] J. H. Holland, **Adoption an Natural and Artificial Systems**. The University of Michigan Press, Ann Arbor, Usa, 1975.
- [7] J. R. Koza, **Genetic Programming II: Automatic Discovery of Reusable Programs**. MIT Press, 1994.
- [8] T. Bäck, **Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms**. Oxford University Press, 1996.
- [9] D. Fogel, **Evolutionary Computation: The Fossil Record**. IEEE Press, 1998.
- [10] J. T. Alander, **EC bibliography, Report, University of Vaasa, Department of Information Technology and Production Economics**. Vaasa, Finland, available via anonymous FTP from <ftp://garbo.uwasa.fi/pc/research/2500GArefs.ps.gz>
- [11] ..., **EvoNet, Network of Excellence in Evolutionary Computing**. <http://www.dcs.napier.ac.uk/evonet/>
- [12] J. Heitkötter, D. Beasley, eds., **The Hitch-Hiker's Guide to Evolutionary Computation**. Issue 8.2, released 20 September 2000, Copyright © 1993-2000 by J. Heitkötter and D. Beasley, all rights reserved, A list of Frequently Asked Questions (FAQ), USENET: comp.ai.genetic, available via anonymous FTP from <ftp://rtfm.mit.edu/pub/usenet/news.answers/ai-faq/genetic/>
- [13] C. Colby, **Introduction to Evolutionary Biology**. Version 2, Last Update: January 7, 1996, <http://www.talkorigins.org/faqs/faq-intro-to-biology.html>
- [14] C. Darwin, **The Origin Of Species**. Penguin Classics, London, 1985.
- [15] C. Darwin, **The Origin of Species**. <ftp://ftp.de.uu.net/pub/research/softcomp/EC/etc/darwin/origin-of-species.gz>
- [16] R. Dawkins, **The Extended Phenotype, The Gene as a Unit of Selection**, Oxford, Oxford University Press, 1982.
- [17] R. Dawkins, **The Blind Watchmaker**. New York, W.W. Norton, 1986.
- [18] R. Dawkins, **The Selfish Gene**. Oxford, Oxford University Press, 1989.
- [19] K. Sigmund, **Games of Life: Explorations in Ecology, Evolution and Behaviour**. Oxford: Univ. Press, 1993.
- [20] R. C. Lewontin, **The Genetic Basis of Evolutionary Change**. Columbia University Press, New York, 1974.
- [21] W. Atmar, **On the Rules and Nature of Simulated Evolutionary Programming**. In Proc. of the First Ann. Conf. On Evolutionary Programming, D. B. Fogel and W. Atmar. La Jolla ed., CA: Evolutionary Programming Society, pp. 17–26, 1992.
- [22] D. L. Hartl, A. G. Clark, **Principles of Population Genetics**. 2nd ed. Sinauer, Sunderland, MA, 1989.
- [23] N. Cercone, G. McCalla, **The Years of Computational Intelligence, Computational Intelligence**. Vol. 10, Nr. 4, pp. i-iv, 1994.
- [24] J. C. Bezdek, **What is Computational Intelligence, Computational Intelligence: Imitating Life**. J. M. Zurada, R. J. Marks, Ch. J. Robinson, Eds., New York, IEEE Press, pp. 1–12, 1994.
- [25] D. B. Fogel, **The Advantages of Evolutionary Computation**. Bio-Computing and Emergent Computation 1997, D. Lundh, B. Olsson, and A. Narayanan (eds.), Sköve, Sweden, World Scientific Press, Singapore, pp. 1–11, 1997.
- [26] T. Bäck, U. Hammel, H. P. Schwefel, **Evolutionary Computation: Comments on the History and Current State**, IEEE Trans. on Evolutionary Computation, Vol. 1. No. 1, pp. 3–17, 1997.
- [27] G. E. Liepins, M. R. Hilliard, **Genetic Algorithms: Foundations and Applications**. Annals Of Operations Research, 21(1–4), pp. 31–58, 1989.
- [28] R. E. Smith, **Adaptively Resizing Populations: An Algorithm and Analysis**. Tega Report No. 93001, The University of Alabama, Tuscaloosa, USA, pp. 1–14, 1993.
- [29] G. G. Robertson, **Population Size in a Classifier System**. Proc. Fifth Int. Conf. On Machine Learning, Morgan–Kaufmann, pp. 142–152, 1988.
- [30] J. J. Grefenstette, **Incorporating Problem-Specific Knowledge into Genetic Algorithm, Genetic Algorithms and Simulated Annealing**, Pitman, London, pp. 42–60, 1987.
- [31] G. E. Liepins, M. R. Hilard, M. Palmer, M. Morrow, **Greedy Genetics, Genetic Algorithms and their Appl.** in Proc. Sec. Int. Conf on GA, Lawrence Erlbaum Associates, pp. 90–99, 1987.
- [32] K. A. De Jong, **Analysis of the Behavior of a Class of Genetic Adaptive Systems**. Ph.D. Thesis, Dpt. of Computer and Communication Sciences, Univ. Of Mich., 1975.
- [33] J. D. Schaffer, R. A. Caruna, L. J. Eshelman, R. Das, **A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization**. In Proc. of the Third Intl. Conf. on Genetic Algorithms, Morgan Kaufman, Pub., pp. 51–60, 1989.
- [34] H. Mühlenbein, (R. Maenner and B. Manderick, Editors), **How Genetic Algorithms Really Work: I. Mutation And Hillclimbing**. Parallel Problem Solving From Nature 2, North Holland, 1992.
- [35] D. E. Goldberg, **Genetic Algorithm in Search, Optimization, and Machine Learning**. Addison-Wesley, Reading MA, 1989.

Napomena. Ovo je prvi od dva dijela članka. U drugom dijelu bit će prikazan primjer tipičnog predstavnika evolucijskih algoritama, uz praktične upute za njegovu primjenu. Bit će navedena područja uspješne primjene evolucijskih algoritama, te izvori podataka (knjige, časopisi, konferencije, udruge, dostupni podaci posredstvom Interneta i dr.) za one koji žele detaljnije proučiti to područje.

Evolutionary Algorithms (I) – Inspirations and Principles. Text shortly presents the basic idea and historical development of application of a natural evolution mechanism for optimizing, learning and modeling. The main concepts and methods of natural and formal representation of evolution known under the common name »evolution algorithms« (EA for short) are explained side by side. Types of intelligent algorithms including EA are mentioned and its characteristics, advantages and fields of application are given. Simple examples illustrate the practical representation of EA notions and methods.

Key words: evolution, evolutionary algorithms, evolutionary computing, fitness, genotype, individual, optimization, phenotype, population, selection

NASLOV AUTORA:

Doc. dr. sc. Darko Grundler
Sveučilište u Zagrebu
Tekstilno tehnološki fakultet
Prilaz baruna Filipovića 30
10000 Zagreb
E-mail: darko.grundler@sk.tel.hr
WWW: <http://public.srce.hr/~dgrund>

Primljeno: 2001–10–05