

Fișa suspiciunii de plagiat / Sheet of plagiarism's suspicion
**Indexat la:
00021/00**
Opera suspicionată (OS)
Opera autentică (OA)
Suspicious work
Authentic work

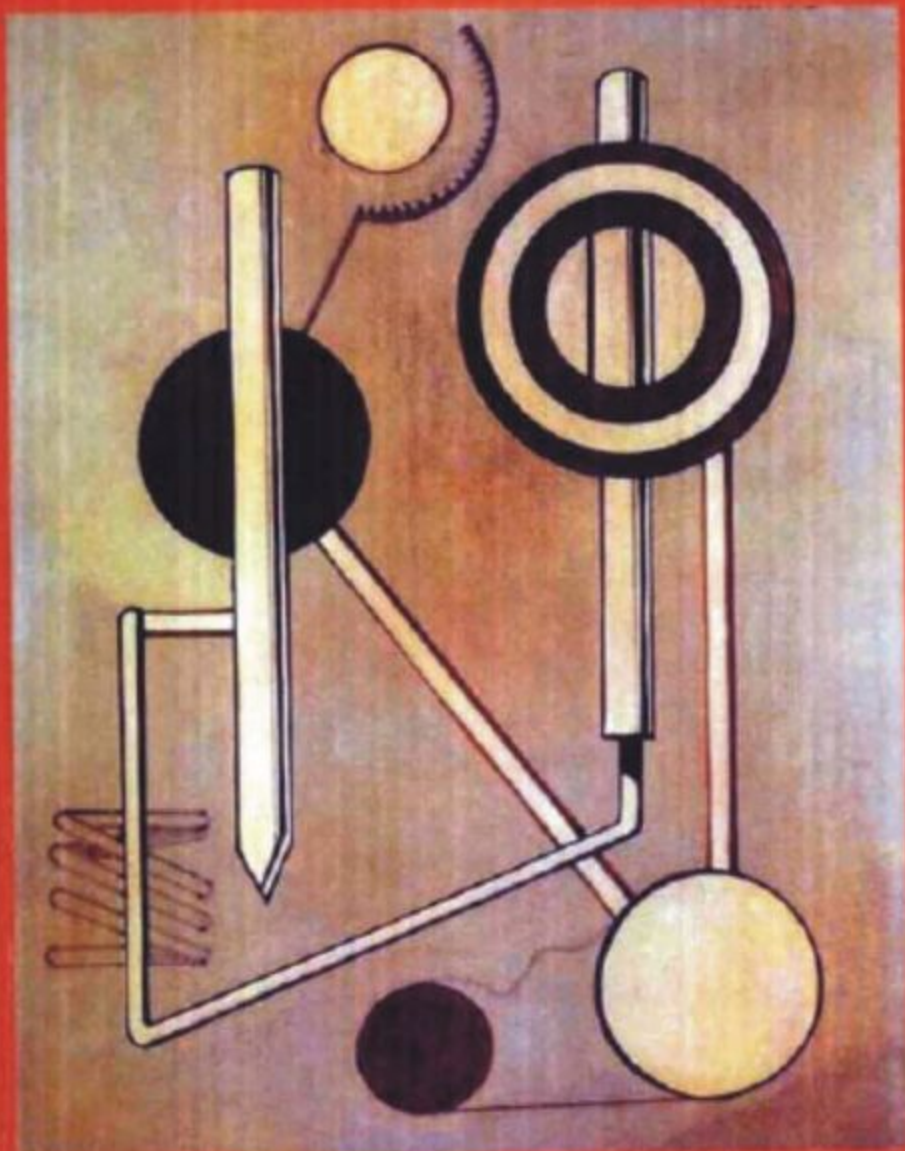
OS	FILIMON, Maria Viorela. Securizarea accesului, interogarea și sincronizarea bazelor de date distribuite utilizând tehnici de inteligență artificială. Teză de doctorat, Universitatea Tehnică din Cluj-Napoca. 2011. Conducător științific: Prof.VĂLEAN, Honoriu (Universitatea Tehnică Cluj-Napoca). Comisia de evaluare a tezei: Prof.NEDEVSCI, Sergiu (Universitatea Tehnică Cluj-Napoca), Prof.Univ. PROȘTEAN, Octavian (Universitatea Politehnica Timișoara), Prof.ILEANĂ, Ioan (Universitatea "1 Decembrie 1918" Alba Iulia), Prof.ABRUDEAN, Mihail (Universitatea Tehnică Cluj-Napoca).
OA	Rotar, C. <i>Modele naturale si algoritmi evolutivi</i> , Cluj-Napoca: Editura Accent. 2008.

Incidența minimă a suspiciunii / Minimum incidence of suspicion

p.46:5 – p.54:26

p.12:3 – p.26:15

Fișa întocmită pentru includerea suspiciunii în Indexul Operelor Plagiate în România de la
Sheet drawn up for including the suspicion in the Index of Plagiarized Works in Romania at
www.plagiate.ro



Corina ROTAR

MODELE NATURALE SI ALGORITMI EVOLUTIVI



ISBN 978-973-8915-67-1

© Editura Accent, 2008

Str. Dorobanților, nr. 98

Cluj-Napoca

tel. 0264-265922

e-mail: office@accentpublisher.com

2 Paradigma Algoritmilor Genetici

2.1 Istoric

Primele semnale privind posibilitatea simulării proceselor evolutive cu ajutorul calculatorului au fost înregistrate în anii 50. Unul dintre pionierii acestui domeniu, biologul Alex S. Fraser (1923-2002), prin lucrările sale (ex. "*Simulation of genetic systems by automatic digital computers*" – 1957), a reușit să aducă în atenția cercetătorilor ideea simulării selecției artificiale a organismelor. Interesul asupra fenomenelor evoluției, privite din perspectivă computațională, a crescut în decadele următoare, culminând cu lucrarea de referință a lui John Holland („*Adaptation in Natural and Artificial Systems*”, 1975) în care este descris algoritmul genetic standard. Holland elaborează *Teorema schemelor* în încercarea de a explica forța algoritmilor genetici în rezolvarea problemelor de căutare și optimizare. În paralel cu dezvoltarea algoritmilor genetici se conturează și celelalte direcții majore ale Calculului Evolutiv: strategiile evolutive, programarea genetică, programarea evolutivă. Cu toate acestea, cea mai prolifică direcție rămâne cea a algoritmilor genetici, popularitatea lor fiind justificată de simplitatea și succesul înregistrat în rezolvarea multor probleme dificile.

Un alt punct de referință în istoria algoritmilor inspirați de fenomene naturale îl constituie lucrările lui David E. Goldberg (vezi de ex. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1989) în care sunt revăzute și formulate principiile algoritmilor genetici, dintr-o perspectivă matură asupra noului domeniu conturat. Goldberg furnizează o

ipoteză interesantă dar controversată asupra abilității algoritmului genetic de a găsi soluțiile bune a multor probleme. Cu toate că Goldberg afirmă că *teorema schemelor* (Holland, 1973) susține *ipoteza blocurilor constructive*, teoria formulată de Goldberg nu se verifică în totalitate și nu reușește să explice potențialul algoritmilor genetici în rezolvarea problemelor.

Studiul teoretic și aplicabilitatea algoritmilor genetici în rezolvarea multor probleme grele se datorează cercetătorilor amintiți, astfel încât aceștia sunt recunoscuți pe drept ca fiind "părinții" Algoritmilor Genetici.

2.2 Modelul natural al algoritmilor genetici

Principala sursă de inspirație a algoritmilor genetici este în mod cert teoria evoluției naturale enunțată de Charles Darwin (1809-1882). Cu toate acestea, descoperirile din genetică au influențat de asemenea structura algoritmilor genetici. Teoria Darwinistă reprezintă un ansamblu de observații asupra evoluției privită la nivel macroscopic și identificarea factorilor responsabili de acest fenomen. Adaptarea unei populații este interpretată prin două principii majore: *selecția naturală* – supraviețuirea celor mai adaptați indivizi și *mutația* – variație a caracteristicilor indivizilor apărută ca răspuns la mediul înconjurător. Cercetările și lucrările lui Darwin (ex. *Originea speciilor*, 1959) nasc controverse și astăzi, însă afirmațiile conform cărora evoluția este călăuzită de principiul supraviețuirii celor mai performanți indivizi și a diversificării prin mutațiile apărute, sunt fundamentele algoritmilor evolutivi.

Teoria neo-darwinistă, sau *teoria eredității* este un alt punct de vedere asupra evoluției, de data acesta interpretat prin transferul de material genetic de la părinți la descendenți. La sfârșitul secolul al XIX-lea, un preot și om de știință austriac, Gregor Mendel (1822-1884) descoperă mai mult sau mai puțin accidental teoria eredității prin experimentele efectuate în vederea încrucișării speciilor diferite de mazăre. Acesta lasă posterității câteva însemnări în caietele sale de observații și o lucrare publicată care a fost ignorată la vremea aceea. Mai târziu, aceste observații enunțate de Mendel sunt recunoscute ca fiind legile eredității și autorul acestora este considerat părintele geneticii moderne. Multe aspecte ale algoritmilor genetici sunt împrumutate din genetică. Dintre acestea remarcăm moștenirea trăsăturilor (genelor) părinților de către descendenți, codificarea indivizilor prin secvențe

de *gene*, dar și caracterul aleator al aportului trăsăturilor provenite de la părinți.

Algoritmii genetici nu respectă strict ingredientele celor două teorii amintite iar terminologia împrumutată nu acoperă semnificația originală: aceștia se formează ca o rețetă practică inspirată de evoluție și ereditate în care factorul uman a strecurat secvențe de calcule și parametri artificiali în scopul obținerii unor rezultate mai bune. Privit la nivel superficial, un algoritm genetic simulează evoluția unei populații, însă, în esență, fenomenul evoluției artificiale este ghidat de funcții matematice și controlat prin parametrii suplimentari pentru a genera o soluție cât mai apropiată de optimul problemei. Avantajul imediat al algoritmilor genetici constă în gradul mare de generalitate și paleta largă a problemelor abordate. Originalitatea, simplitatea și sursa inedită de inspirație poate justifica parțial atenția acordată dezvoltării acestor metaeuristici. Succesul real înregistrat în probleme de dificultate ridicată este argumentul forte al algoritmilor genetici.

2.3 Descriere algoritm genetic

Pe scurt, un algoritm genetic operează cu o mulțime de indivizi (denumiți în mod uzual *cromozomi*) asupra cărora se aplică *operatorii genetici*. Fiecare individ reprezintă o soluție posibilă din spațiul de căutare. Maniera de codificare a unei soluții este binară, reală sau specifică, în funcție de opțiunea noastră și de contextul problemei. Mulțimea de cromozomi formează o *populație*. Evoluția populației este condusă de două elemente: operatorii genetici și *funcția de evaluare* a calității cromozomilor. *Selecția*, *încrucișarea* și *mutația* sunt operatorii genetici uzuali, fiind inspirați de procesele naturale care stau la baza evoluției.

Indiferent de natura problemei considerate se poate construi cel puțin o funcție de evaluare a soluțiilor posibile. Prin aceasta, fiecărui individ al populației i se atribuie o valoare numerică reprezentând performanța sa în raport cu cerințele problemei. Ulterior, măsura calității indivizilor populației curente este folosită în procesul de *selecție* a acelor indivizi, denumiți *părinți*, asupra cărora se aplică operatorii de încrucișare și mutație pentru obținerea noii generații. Principiul este simplu: cu cât părinții selectați sunt mai performanți, cu atât șansa ca descendenții obținuți să fie calitativ superiori este mai mare. Procesul se reia având ca populație curentă noua generație de cromozomi. Se observă de-a lungul evoluției populației o creștere a calității

indivizilor săi. După un număr considerabil de generații soluția globală a problemei poate fi aproximată suficient de bine printr-un individ al ultimelor generații.

Conceperea unui algoritm genetic de rezolvare a unei probleme concrete presupune evidențierea următoarelor componente:

1. individul
2. populația
3. funcția de evaluare
4. selecția
5. operatorii de variație (încrucișare și mutație)
6. condiția de oprire a algoritmului

1) Individul

Analizând specificațiile problemei, putem identifica spațiul de căutare. Orice punct al acestuia constituie o soluție posibilă. În funcție de specificul soluției posibile, putem determina o manieră inspirată de codificare numerică sau nenumerică a acesteia. Codificarea unei soluții formează un *cromozom* și identifică un individ al populației curente. Fiecare cromozom este format dintr-un șir de valori ale unui alfabet dat. Valoarea de pe o poziție oarecare a șirului codificării se numește *genă*. Alfabetul A al valorilor genelor este stabilit în prealabil și poate fi:

- valori binare: $A = \{0,1\}$
- valori reale: $A \subseteq R$
- alt alfabet (*codificare specifică*)

Lungimea șirului de gene din codificarea cromozomului poate fi constantă sau variabilă și depinde de numărul de trăsături definitorii ale unei soluții posibile a problemei:

Fie c un cromozom oarecare și A alfabetul genelor din reprezentarea sa. Cromozomul c se descrie prin vectorul de n elemente:

$$c = (\alpha_1, \alpha_2, \dots, \alpha_n),$$

unde $\alpha_i \in A, \forall i \in \{1, 2, \dots, n\}$.

În funcție de alfabetul genelor, codificarea cromozomială se clasifică în:

- codificare binară, $A = \{0,1\}$
- codificare reală, $A \subseteq R$
- codificare specifică – alta decât cea binară sau reală.

2) Populația

O mulțime finită de cromozomi alcătuiește populația. Dimensiunea populației (numărul de indivizi care o formează) este în general o valoare constantă pe care o stabilim în prealabil și reprezintă un parametru important al algoritmului dezvoltat. Într-o abordare mai flexibilă, dimensiunea populației poate fi variabilă, un exemplu în acest sens ar fi descreșterea numărului de indivizi de-a lungul evoluției, odată cu creșterea performanței acestora, fapt care ar ușura determinarea soluțiilor finale din ultima generație produsă. Nu este exclusă utilizarea mai multor populații cooperante (exemplu) sau a unei populații suplimentare cu rol de memorie în care sunt reținute soluții bune găsite în cadrul generațiilor intermediare (vezi de ex.). Modelul standard al algoritmului genetic presupune exploatarea unei unice populații de dimensiune constantă. Formal, o astfel de populație se descrie astfel:

$$P = \{c_1, c_2, \dots, c_m\},$$

unde: $\forall j \in \{1, 2, \dots, m\}$, c_j reprezintă un *cromozom*.

Prima generație se numește *populație inițială* și construirea acesteia se face în principal prin generarea aleatoare a unor soluții posibile în domeniul de căutare. Dacă sunt cunoscute zone ale spațiului care conțin soluții bune ale problemei, populația inițială poate fi îmbogățită cu indivizi care codifică puncte din aceste zone. În această manieră se accelerează convergența populației, oferindu-se șansa ca încă de la primele generații să fie obținute soluții optime ale problemei. Există puține situații în care procedeul descris mai sus poate fi aplicat deoarece din specificațiile problemei rareori putem deduce zonele promițătoare ale spațiului de căutare. Modelul standard al algoritmului genetic presupune generarea aleatoare a populației inițiale.

Cu cât dimensiunea populației este mai mare, cu atât șansa obținerii rapide a unei bune aproximări a soluțiilor problemei crește. De asemenea, distribuția indivizilor populației joacă un rol important în economia construirii algoritmului genetic: o distribuție echilibrată a populației inițiale mărește substanțial viteza de identificare a zonelor promițătoare ale spațiului

de căutare. Acest fapt aduce cu sine o convergență mai bună înspre soluțiile problemei.

În concluzie, o populație de dimensiune suficient de mare și având un grad de diversitate considerabil reprezintă unul dintre factorii majori ce conduc la obținerea soluțiilor dorite.

3) Funcția de evaluare

Una dintre condițiile fundamentale ale funcționării eficiente ale unui algoritm genetic o reprezintă alegerea inspirată a funcției de evaluare. În cazul unei probleme de optimizare numerică (ex. determinarea maximului/minimului global al unei funcții matematice), funcția de evaluare se poate alege ca fiind chiar funcția criteriu sau o funcție construită pe baza funcției criteriu.

Pentru o mai bună înțelegere prezentăm în continuare un exemplu:

Fie $f : D \rightarrow R$, $D = [-\pi, \pi]$, dată de formula:

$$f(x) = \sin(x)$$

Se dorește determinarea maximului funcției f .

Criteriul care ghidează căutarea soluției globale este maximizarea funcției date, respectiv, determinarea punctului x pentru care $\sin(x)$ este maxim.

Orice valoare reală din intervalul $[-\pi, \pi]$ poate fi reprezentată în limbaj binar, cu o anumită precizie impusă. Un cromozom al populației corespunzător unei valori reale x va fi construit ca un vector de cifre binare ale codificării punctului x . Asupra codificării binare vom reveni într-un paragraf următor. Se impune observația că o abordare mai flexibilă permite ca reprezentarea cromozomială a unui punct x al spațiului de căutare să fie dată de o un vector particular cu o singură componentă. Unica genă a codificării reprezintă chiar valoarea reală a punctului corespunzător.

În acest caz particular, funcția de evaluare este dată de însăși funcția criteriu. Fie cromozomul c reprezentarea (binară sau reală) a punctului x . Funcția de evaluare este construită astfel:

$$performanta(c) = \sin(x)$$

Valorile calculate ale funcției de performanță califică și diferențiază indivizii populației. Astfel, este posibilă identificarea indivizilor performanți ai populației, cei cărora li se oferă ulterior o mai mare șansă de a produce descendenți.

Există situația în care este dificil de exprimat matematic criteriul de optimizare al problemei date. În aceste situații, utilizatorul va construi funcții de evaluare prin care se va încerca calificarea cât mai coerentă a soluțiilor candidat, fapt pentru care rezultatele oferite de algoritmul dezvoltat sunt dependente de factorul uman. Mai mult, probleme de optimizare de dificultate ridicată sunt cele în care se dorește determinarea optimele multiple ale unei funcții (optimizare multimodală) sau cele în care se impun anumite restricții (optimizare cu restricții). Pentru rezolvarea acestor probleme, funcția de evaluare va fi atent construită pentru a respecta restricțiile impuse, respectiv, pentru a permite populației să convergă înspre toate optimele funcției criteriu.

Un alt caz în care construirea funcției de evaluare necesită un efort mai mare este cel al problemelor de optimizare multiobiectiv. Asupra acestor probleme vom reveni într-un capitol ulterior cu o prezentare detaliată.

În discuția referitoare la funcția de evaluare ne-am referit strict la probleme de optimizare. Justificarea acestui fapt rezidă din afirmația că orice problemă de căutare a soluțiilor în spațiul soluțiilor posibile, căutare ghidată de unul sau mai multe criterii, poate fi reinterpretată ca o problemă de optimizare: din mulțimea soluțiilor posibile se caută acelea care optimizează cel mai bine criteriile problemei. Deși principala aplicabilitate a algoritmilor genetici constă în rezolvarea problemelor de optimizare, afirmația precedentă ne permite extinderea ariei de aplicabilitate a algoritmilor genetici și înspre probleme de alt gen.

4) Selecția

Am vorbit în rândurile precedente de structura populației și calificarea indivizilor săi prin funcția de evaluare. Evoluția populației, obținerea generațiilor de indivizi mai performanți decât predecesorii lor, este cauzată

de operatorul de selecție și operatorii de variație (recombinarea și mutația) aplicați.

Unul dintre factorii majori ai evoluției este selecția naturală. O interpretare simplificată a acestui fenomen ne permite înțelegerea procesului prin care o populație poate converge înspre anumite puncte ale spațiului de căutare, respectiv, înspre soluțiile unei probleme date. Pornind de la observația acceptată că indivizii performanți ai unei specii dau naștere la descendenți asemănători și de performanțe apropiate și, în schimb, indivizii slab calificați produc indivizi noi ale căror performanțe sunt în general slabe, ne putem imagina două scenarii:

- în primul scenariu, indivizii unei populații suferă încrucișări și mutații pur aleatoare, fără a se aplica principiul selecției
- al doilea scenariu implică o selecție prealabilă a părinților noii generații, selecție făcută pe baza valorilor de performanță a indivizilor populației curente

Cele două populații ale scenariilor descrise se monitorizează de-a lungul evoluției. Se observă că, dacă în primul caz absența selecției induce o stagnare în procesul evoluției populației înspre indivizi de calitate superioară, în a doua situație, selecția favorizează o creștere accelerată a performanței medii a generațiilor produse. Comparând ultimele generații ale ambelor scenarii putem deduce importanța selecției în evoluția unei populații.

Implementarea operatorului de selecție se face în variate maniere. Unul dintre cei mai populari algoritmi de selecție este cel al *selecției proporționale* inspirat de algoritmul ruletei. Selecția proporțională presupune alegerea părinților noii generații în mod proporțional cu valoarea de performanță a acestora. Altfel spus, probabilitatea de selectare a unui individ performant este mai mare decât a unuia slab calificat, și această probabilitate este direct proporțională cu valoarea performanței individului. Acest procedeu favorizează o convergență rapidă a populației, însă în unele cazuri prezintă dezavantaje majore, fapt pentru care se optează pentru implementarea altor variante de selecție.

Algoritmii detaliați ai operatorilor de selecție vor fi prezentați într-un paragraf dedicat.

5) Operatorii de variație (încrucișare și mutație)

Selecția propriu-zisă nu este suficientă pentru a induce dinamica unei populații. Indivizii selectați vor suferi modificări prin operatorii genetici de variație pentru a permite obținerea unor descendenți diferiți, respectiv pentru a permite o explorare bună a spațiului de căutare. Principalii operatori de variație sunt încrucișarea și mutația.

Varianta standard a operatorului de încrucișare se aplică asupra a doi părinți selectați anterior și produce unul sau doi descendenți care moștenesc trăsăturile părinților în proporții diferite. Prezentăm în continuare un operator de încrucișare aplicabil pentru codificarea binară. Tipul operatorului este de 2:2, respectiv, din 2 părinți se produc 2 descendenți. Datorită specificului său, operatorul descris mai jos este cunoscut sub denumirea de *încrucișare cu un punct de tăietură*.

Fie c_1 și c_2 doi indivizi oarecare ai populației curente:

$$c_1 = (\alpha_1, \alpha_2, \dots, \alpha_n)$$

$$c_2 = (\beta_1, \beta_2, \dots, \beta_n)$$

Se generează aleator o valoare naturală $t \in \{1, 2, \dots, n\}$, având semnificația punctului de tăietură. Ambii părinți se vor diviza, obținându-se secvențele:

$$c_1^1 = (\alpha_1, \alpha_2, \dots, \alpha_t), \quad c_1^2 = (\alpha_{t+1}, \alpha_{t+2}, \dots, \alpha_n)$$

$$c_2^1 = (\beta_1, \beta_2, \dots, \beta_t), \quad c_2^2 = (\beta_{t+1}, \beta_{t+2}, \dots, \beta_n)$$

Descendenții d_1 și d_2 ai părinților c_1 și c_2 se obțin prin concatenarea secvențelor obținute:

$$d_1 = c_1^1 + c_2^2$$

$$d_2 = c_2^1 + c_1^2$$

Observație: Operatorul $+$ are în această descriere formală semnificația concatenării secvențelor.

Se obțin astfel noii indivizi:

$$d_1 = (\alpha_1, \dots, \alpha_t, \beta_{t+1}, \dots, \beta_n) \text{ și}$$

$$d_2 = (\beta_1, \dots, \beta_t, \alpha_{t+1}, \dots, \alpha_n)$$

Prin aplicarea încrucișării, noii indivizi moștenesc genele părinților, fapt pentru care aceștia sunt asemănători indivizilor care îi produc. Se întâlnește deseori situația în care o anumită soluție a spațiului de căutare nu poate fi obținută prin aplicarea operatorului de încrucișare descris. Există de asemenea suficiente argumente în favoarea aplicării altor tipuri de operatori de încrucișare (exemplu: *încrucișarea cu puncte multiple de tăietură*). Pentru asigurarea unei bune explorări a spațiului de căutare, un alt operator genetic este introdus: *mutația*. Teoria evoluției naturale menționează faptul că selecția nu este unicul factor responsabil al fenomenului evoluției. Mutăția naturală, definită prin micile modificări la nivelul codificării cromozomiale, este cea care, în combinație cu mecanismul selecției, asigură adaptarea la mediu. Implementarea procesului natural al mutației se face prin *operatorul de mutație*, care la rândul său cunoaște variate forme descrise în lucrări de specialitate.

Prezentăm în continuare o variantă simplă a operatorului de mutație. Acest operator este de tipul 1:1, respectiv, dintr-un părinte selectat se obține un unic descendent prin alterarea valorii unei singure gene. De asemenea, aplicabilitatea operatorului descris este strict limitată la *codificarea cromozomială binară*.

Fie c un cromozom al populației curente:

$$c = (\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_n)$$

Se generează aleator o poziție k din secvența binară a codificării.

Individul mutant $c' = (\alpha_1, \alpha_2, \dots, \beta_k, \dots, \alpha_n)$ se obține prin copierea nealterată a genelor părintelui c , cu excepția genei de pe poziția k . Valoarea genei mutate se obține prin inversare:

Dacă $\alpha_k = 1$ atunci

$$\beta_k \leftarrow 0$$

Altfel

$$\beta_k \leftarrow 1$$

SfDacă

Alegerea operatorilor folosiți joacă un rol decisiv în economia dezvoltării algoritmilor genetici. Utilizatorul optează pentru anumite forme ale operatorilor genetici pe baza unor criterii care sunt de cele mai multe ori subiective. Specificațiile problemei sunt cele care ne pot oferi informațiile necesare pentru a decide maniera de codificare a indivizilor. Odată stabilită forma de codificare (*binară, reală, specifică*), paleta operatorilor se restrânge semnificativ. Cu toate acestea, diversitatea procedeele de încrucișare și mutație, clasificate pe tipul codificării pre-fixate, fac dificilă alegerea inspirată a celor mai eficiente variante. Preferințele se îndreaptă înspre acei operatori care se dovedesc a fi eficienți în rezolvarea unor probleme similare. Lipsa constrângerile legate de implementarea operatorilor genetici, permite utilizatorului să își construiască operatori specifici problemei sau clasei de probleme pe care le abordează.

6) Condiția de oprire a algoritmului genetic

Paragrafele anterioare prezintă succint principalele ingrediente ale unui algoritm genetic. Se conturează astfel structura unui algoritm de acest tip. În sensul, algoritmul genetic este o procedură repetitivă prin care o populație de soluții posibile, prin aplicarea operatorilor genetici (selecție, încrucișare, mutație) produce noua generație. Procesul se reia pentru noua populație obținută până când o condiție de terminare este îndeplinită. Condiția de terminare a algoritmului se referă în general la atingerea unui număr maxim restabilit de generații. Există și situații în care utilizatorul poate să impună o altă condiție de încheiere a evoluției, condiție independentă de numărul populațiilor generate. Un exemplu în acest sens ar fi înregistrarea unui anumit grad de uniformitate în cadrul populației. În situația în care indivizii populației devin asemănători putem considera că diversitatea slabă a populației nu mai permite o explorare eficientă a spațiului de căutare. Mai mult, putem concluziona că indivizii populației s-au stabilit în zona cea mai promițătoare, corespunzătoare soluției globale. Experimentele arată că nu întotdeauna această concluzie este adevărată. Populația poate fi atrasă de un punct de optim local, producându-se fenomenul de convergență prematură. Convergența prematură este unul dintre fenomenele nedorite ale unui algoritm genetic și evitarea acestui neajuns comportă câteva modificări în structura algoritmului: alegerea unei alte scheme de selecție, reinițializarea unei secțiuni din populație sau accentuarea mutațiilor produse. O

altă soluție oferită este cea prin care sunt monitorizate ultimele k generații obținute. Pentru a nu cădea în posibilă capcană scăderii temporare a gradului de diversitate a populației putem considera un număr mai mare de generații analizate din punct de vedere al asemănării indivizilor. În această manieră obiectivitatea deciziei de terminare a evoluției crește substanțial. Parametrul k reprezintă în acest caz o valoare numerică naturală prestabilită.

Revenind la condiția de terminare, ca și în cazul celorlalte componente ale algoritmului genetic, alegerea acesteia se face de către proiectantul algoritmului la o atentă observare a comportamentului populației de-a lungul generațiilor. O bună experiență a proiectantului cât și concluziile experimentelor efectuate sunt factori principali ai construirii unui algoritm genetic valoros.

Înainte de a furniza schema algoritmului genetic standard, este important să afirmăm că acest algoritm nu funcționează ca un șablon de rezolvare a oricărei probleme. Detalii legate de implementare, alegerea operatorilor, construirea funcției de evaluare, precum și stabilirea parametrilor implicați reprezintă aspecte care diferențiază algoritmii obținuți din punct de vedere al aplicabilității acestora pe anumite clase de probleme.

Presupunem că cele 6 elemente enumerate anterior au fost stabilite. În acest caz putem formula structura algoritmului genetic:

Schema algoritmului genetic standard:

Algoritmul AG_standard este:

Fie $P(0)$ - populația inițială; $t = 0$;

Cât timp (NOT condiție de terminare)

Evaluare($P(t)$)

$P_S(t) = \text{Selectie} (P(t))$

$P_R(t) = \text{Încrucișare}(P_S(t))$

$P(t+1) = \text{Mutație} (P_R(t))$

$t := t + 1$

SfCât timp

SfAlgoritm

Notăm:

t - numărul generației curente.

P_S - mulțimea indivizilor selectați

P_R - mulțimea indivizilor obținuți prin încrucișare

Soluția finală a algoritmului genetic este dată de cel mai performant individ al ultimei generații produse. Se impune observația că această variantă de extragere a rezultatului nu este cea mai eficientă. Ne imaginăm următorul scenariu: un individ performant, extrem de apropiat de soluția globală a problemei) este obținut într-una dintre generațiile intermediare. Principiul selecției ne conduce la supoziția că probabilitatea acestuia de a fi selectat în vederea aplicării operatorilor de variație este mare. Operatorul de încrucișare nu asigură păstrarea părinților performanți în noua generație. Astfel, individul considerat în scenariul nostru va fi distrus, neavând certitudinea că descendenții săi sunt cel puțin la fel de performanți ca și el. Observăm astfel că indivizii calificați ai generațiilor intermediare pot dispărea pe parcursul evoluției populației.

Remediarea acestui fenomen poate fi făcută extrem de simplu prin suplimentarea algoritmului inițial cu o formă de *elitism*. Prin elitism, cel mai bun, sau cei mai buni k indivizi ai generației curente vor fi copiați nemodificați în noua generație. Se evită în acest mod pierderea soluției globale dacă aceasta este obținută într-o etapă intermediară a evoluției.

2.4 Funcționarea algoritmului genetic

Intuitiv, algoritmul genetic va conduce populația de soluții posibile înspre soluțiile optime ale problemei de rezolvat. De cele mai multe ori acest fenomen decurge în mod firesc, fără anomalii, și putem spune că algoritmul converge înspre optimele problemei. Există și situația în care indivizii populației se blochează în zone sub-optimale ale spațiului de căutare, evoluția lor ulterioară fiind îngreunată sau imposibilă. În aceste situații rezultatul oferit de algoritm este eronat. Cauzele acestui anomalii sunt multiple. Una dintre cele mai frecvente surse de eșec este alegerea neinspirată a funcției de evaluare și a operatorului de selecție, rezultând pierderea diversității populației și incapacitatea de a determina soluțiile globale. Fenomenul nedorit, descris ca obstrucționarea explorării spațiului de

căutare și blocarea populației în zone corespunzătoare optimelor locale, este denumit *convergență prematură*.

În esență convergența prematură este cauzată de pierderea diversității genetice a indivizilor într-o etapă timpurie a evoluției, fapt care generează o stagnare a evoluției ulterioare. Probabilitatea ca indivizii asemănători ai populației curente să genereze descendenți diferiți este redusă, ceea ce induce o rezistență sporită la dislocarea indivizilor din zonele de blocaj.

În scopul evitării convergenței premature au fost dezvoltate diferite strategii de menținere a diversității populației. Dintre acestea enumerăm:

- stabilirea unei dimensiuni mai mari a populației;
- aplicarea unor operatori genetici care s-au dovedit benefici evoluției și menținerii diversității (ex. încrucișarea uniformă)
- strategii de împerechere care evită obținerea de indivizi similari prin aplicarea încrucișării
- aplicarea mutației cu o mai mare probabilitate
- evitarea folosirii operatorului de selecție proporțională care în anumite situații va favoriza evoluția doar a unei mici fracțiuni de indivizi din populație, conducând la omogenizarea populației

Un alt aspect al funcționării algoritmului genetic este cel al *elitismului*. În multe situații, o soluție bună a problemei (codificată ca individ al populației) poate să apară într-una dintre generațiile intermediare; având o performanță mare, șansa de a fi selectată în vederea supunerii acesteia la acțiunea operatorilor de variație (mutație și încrucișare) este mare. Aplicarea operatorilor genetici poate să distrugă soluția performantă, fapt pentru care evoluția populației este încetinită. Remedierea acestui fenomen constă în aplicarea unor strategii de salvare a indivizilor performanți determinați în populațiile intermediare. În mod sugestiv, acești indivizi cu performanțe ridicate se numesc *elită* și procedeul prin care elita este menținută de-a lungul populațiilor se numește *elitism*. Cea mai simplă strategie de elitism constă în copierea elitei în noua generație. Acest transfer garantează protejarea soluțiilor calificate de-a lungul evoluției și o mai bună convergență a algoritmului genetic.

Funcționarea algoritmului genetic este strâns legată de două concepte duale: *explorarea* și *exploatarea* spațiului de căutare. Primul concept se referă la capacitatea populației de a acoperi spațiul soluțiilor posibile prin căutare

globală și este strâns legat de proprietatea de diversitate a populației. Cel de-al doilea concept se referă la capacitatea populației de a asigura căutarea locală în zonele promițătoare ale spațiului și corespunde unei îmbunătățiri a indivizilor în vederea obținerii unei aproximări mai bune a soluțiilor finale. Cele două fenomene definite de conceptele sus-menționate sunt duale în sensul în care, accentuarea unuia conduce în mod gradual la inhibarea apariției celuilalt. Astfel, o *explorare* asiduă a spațiului de căutare, marcată prin menținerea unei populații diverse, îngreunează procesul de rafinare a soluțiilor bune și cauzează o stagnare în evoluția populației. Contrar, accentuarea *exploatării* zonelor promițătoare conduce la o *explorare* defectuoasă a spațiului de căutare și poate conduce la fenomenul de convergență prematură. Ambele situații descrise trebuie evitate. Pentru buna funcționare a algoritmului genetic din perspectiva celor două concepte, este necesară stabilirea unui echilibru *explorare-exploatare* obținut în principal prin aplicarea unui mecanism de evaluare și selecție bine ales.

2.5 Aplicații

Paleta de aplicații ale Algoritmilor genetici este nelimitată. Concepți ca instrumente de optimizare, algoritmi genetici sunt aplicabili și în probleme de alt gen, prin reformularea problemelor respective ca probleme de căutare a soluțiilor optime în spațiul de căutare. Identificarea corectă a obiectivelor problemei date, delimitarea spațiului soluțiilor posibile, determinarea unei maniere de reprezentare a soluțiilor, construirea unei funcții de evaluare corespunzătoare și alegerea inspirată a operatorilor de variație, sunt câteva dintre ingredientele care asigură aplicabilitatea algoritmilor genetici. Literatura de specialitate este îmbogățită periodic cu cercetări dedicate aplicațiilor algoritmilor genetici în probleme reale. Enumerăm mai jos câteva dintre cele mai reprezentative probleme pentru care algoritmi genetici s-au dovedit eficienți:

- optimizarea numerică;
- proiectarea și optimizarea rețelelor neuronale;
- proiectarea automată a sistemelor;
- probleme de control;
- planificarea sarcinilor;
- probleme de transport;
- probleme din domeniul telecomunicațiilor;

- problema comis-voiajorului;
- etc.

2.6 Selecția

Responsabilitatea operatorului de selecție este aceea de a avantaja indivizii performanți ai populației, respectiv de a favoriza căutarea în apropierea acelor soluții care s-au dovedit a fi performante relativ la obiectivele problemei. Efectul operatorului de selecție poate fi controlat prin intermediul a doi parametri: *presiunea de selecție* și *timpul de acțiune*.

a) *Presiunea de selecție (selection pressure)* se definește ca fiind măsura în care indivizii performanți ai populației produc descendenți în noua generație. Valoarea mică a presiunii de selecție permite fiecărui individ al populației să fie selectat cu o probabilitate rezonabilă pentru a produce descendenți. Valorile apropiate ale probabilităților de selecție induc o slabă exprimare a preferinței înspre exploatarea indivizilor performanți. Contrar, valorile mari ale presiunii de selecție favorizează cei mai buni indivizi ai generației curente, fapt pentru care convergența populației este accelerată. Principalul neajuns al stabilirii unei presiuni mari de selecție constă în posibilitatea producerii efectului de convergență prematură datorat pierderii diversității populației.

b) *Timpul de acțiune (takeover time)* se definește astfel:

- fie un algoritm evolutiv înzestrat doar cu operator de selecție
- generația inițială conține o singură instanță a unei soluții optimale și dimensiunea populației este constantă.

Timpul de acțiune se exprimă prin numărul de generații în care un algoritmul produce o generație alcătuită exclusiv din instanțe ale soluției optimale inițiale.

Se poate observa că cei doi parametri ai operatorului discutat sunt duali: creșterea valorii presiunii de selecție produce descreșterea valorii timpului de acțiune și, reciproc, presiunea mare de selecție induce un număr mai mare de generații în care se produce o populație formată din copii ale soluției inițiale.

Prezentăm în continuare cele mai reprezentative variante ale operatorului de selecție.