Fişa suspiciunii de plagiat / Sheet of plagiarism's suspicion	Indexat la: 00021/00

Opera suspicionată (OS) Suspicious work	Opera autentică (OA) Authentic work

OS	FILIMON, Maria Viorela. Securizarea accesului, interogarea şi sincronizarea bazelor de date distribuite utilizând tehnici de inteligență artificială. Teză de doctorat, Universitatea Tehnică din Cluj-Napoca. 2011. **Conducător ştiințific**: Prof.VĂLEAN, Honoriu (Universitatea Tehnică Cluj-Napoca). **Comisia de evaluare a tezei**: Prof.NEDEVSCHI, Sergiu (Universitatea Tehnică Cluj-Napoca), Prof.Univ. PROŞTEAN, Octavian (Universitatea Politehnica Timişoara), Prof.ILEANĂ, Ioan (Universitatea "1 Decembrie 1918" Alba Iulia), Prof.ABRUDEAN, Mihail (Universitatea Tehnică Cluj-Napoca).
OA	Rotar, C. *Modele naturale si algoritmi evolutivi*, 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

Fişa 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

MINISTERUL EDUCAȚIEI, CERCETĂRII, TINERETULUI ŞI SPORTULUI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE

Maria Viorela Filimon

TEZĂ DE DOCTORAT

SECURIZAREA ACCESULUI, INTEROGAREA ŞI SINCRONIZAREA BAZELOR DE DATE DISTRIBUITE UTILIZÂND TEHNICI DE INTELIGENȚĂ ARTIFICIALĂ

Conducător ştiințific,

Prof.dr.ing. Honoriu VĂLEAN

Comisia de evaluare a tezei de doctorat:

PREŞEDINTE: - Prof.dr.ing. ***Sergiu Nedevschi*** - Decan, Facultatea de Automatică şi Calculatoare, Universitatea Tehnică din Cluj-Napoca;

MEMBRI: - Prof.dr.ing. ***Honoriu Vălean*** - Conducător ştiințific, Facultatea de Automatică şi Calculatoare, Universitatea Tehnică din Cluj-Napoca;

- Prof.dr.ing. ***Octavian Proştean*** - Referent, Universitatea „Politehnica” din Timişoara;

- Prof.dr. ing. ***Ioan Ileană*** - Referent, Universitatea „1 Decembrie 1918” din Alba Iulia;

- Prof.dr.ing. ***Mihail Abrudean*** – Referent, Facultatea de Automatică şi Calculatoare, Universitatea Tehnică din Cluj-Napoca.

CUPRINS

1. Introducere

Lumea modernă trece printr-o transformare fundamentală, de la societatea industrială care a marcat secolul al XX-lea, la societatea informaţională a secolului al XXI-lea. Acest proces dinamic a produs schimbări fundamentale în toate aspectele vieţii noastre (diseminarea cunoştinţelor, interacţiunea socială, economie şi practici de afaceri, angajament politic, media, sănătate, odihnă şi agrement), dând naştere unei societăţi care se hrăneşte cu informaţie. Din nefericire, forma recreativă şi de divertisment a informaţiei, vine mai degrabă spre noi într-o formă crudă, de date înregistrate în depozite de mari dimensiuni.

Pentru că dimensiunea datelor stocate în depozite creşte (cercetătorii de la Universitatea din Berkeley au calculat că în fiecare an sunt generate şi salvate aproximativ un exabyte de date), ne găsim copleşiţi de date, şi realizăm că ne lipseşte puterea procesării acestora pentru a extrage informaţii folositoare din ele. Chiar şi cu ajutorul calculatorului, echipat cu cele mai recente sisteme de gestiune a bazelor de date şi cu tehnologii noi de comunicare, suntem neajutoraţi când ne confruntăm cu diversitatea şi complexitatea pe care această lume o prezintă. Ritmul rapid de creştere, cantitatea enormă de date, adunate şi stocate în mari şi numeroase baze de date, a depăşit cu mult capacitatea noastră umană de înţelegere. Abundenţa de date, cuplată cu nevoia de putere a instrumentelor de analiză a datelor a fost descrisă ca o bogăţie de date, dar ca o sărăcie a informaţiilor.

Ca şi rezultat, datele colectate în mari baze de date devin „morminte de date", date arhivate care arareori sunt vizitate. În consecinţă, deciziile importante sunt bazate nu pe informaţiile bogate stocate în arhive de date, ci mai degrabă pe intuiţie, pe factorul de decizie a acestuia. În plus, luându-se în considerare tehnologiile sistemelor expert, care de obicei se bazează pe utilizatori sau experţi, se introduc manual cunoştinţele în baze. Din nefericire, această procedură este predispusă deviaţiilor şi erorilor fiind o mare consumatoare de timp şi costuri. Instrumentele de extragere a cunoştinţelor efectuează analiza datelor şi pot descoperi modele de date importante, care contribuie foarte mult la strategiile de afaceri, baze de cunoştinţe, cercetări ştiinţifice şi medicale. Adâncirea diferenţelor dintre date şi informaţii cheamă pentru o dezvoltare sistematică a instrumentelor de extragere a cunoştinţelor, care vor transforma mormintele de date în „pepite" ale cunoaşterii.

O notă recentă a Gartner Group Advanced Technology Research, a poziţionat extragerea cunoştinţelor din date şi inteligenţa artificială în topul celor cinci domenii cheie ale tehnologiei, şi că acest lucru „va ave un impact major într-o arie largă de industrii în următorii 3-5 ani". Gartner, de asemenea a prezentat arhitecturile paralele şi data mining ca două dintre cele 10 noi tehnologii în care companiile vor investi în următorii 5 ani. În conformitate cu o notă recentă a Gartner HPC Research, „cu înaintarea rapidă în captarea datelor, transmisia şi stocarea acestora, utilizatorii marilor sisteme vor avea din ce în ce mai multă nevoie să pună în aplicare metode noi şi inovatoare pentru extragerea cunoştinţelor, care angajează procesarea masivă paralelă (*massively parallel processing - MPP*) a sistemelor pentru a crea noi surse de avantaj în afaceri".

Utilizând aceste noţiuni, acurateţea se defineşte ca fiind raportul dintre numărul instanţelor clasificate corect şi numărul total al instanţelor, atât clasificate corect, cât şi clasificate eronat [30]:

$$acuratetea = \frac{AP + AN}{AP + AN + FP + FN} \qquad (41)$$

4.2. Algoritmi evolutivi

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 reuşit să aducă în atenţia cercetătorilor ideea simulării selecţiei artificiale a organismelor. Interesul asupra fenomenelor evoluţiei, privite din perspectivă computaţională, a crescut în decadele următoare, culminând cu lucrarea de referinţă 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 forţa algoritmilor genetici în rezolvarea problemelor de căutare şi optimizare. În paralel cu dezvoltarea algoritmilor genetici se conturează şi celelalte direcţii majore ale Calculului Evolutiv: strategiile evolutive, programarea genetică, programarea evolutivă. Cu toate acestea, cea mai prolifică direcţie rămâne cea a algoritmilor genetici, popularitatea lor fiind justificată de simplitatea şi succesul înregistrat în rezolvarea multor probleme dificile.

Un alt punct de referinţă în istoria algoritmilor inspiraţi 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 şi formulate principiile algoritmilor genetici, dintr-o perspectivă matură asupra noului domeniu conturat. Goldberg furnizează o ipoteză interesantă dar controversată asupra abilităţii algoritmului genetic de a găsi soluţiile bune a multor probleme. Cu toate că Goldberg afirmă că *teorema schemelor* (Holland, 1973) susţine *ipoteza blocurilor constructive*, teoria formulată de Goldberg nu se verifică în totalitate şi nu reuşeşte să explice potenţialul algoritmilor genetici în rezolvarea problemelor.

Studiul teoretic şi aplicabilitatea algoritmilor genetici în rezolvarea multor probleme grele se datorează cercetătorilor amintiţi, astfel încât aceştia sunt recunoscuţi pe drept ca fiind "părinţii" Algoritmilor Genetici.

Modelul natural al algoritmilor genetici

Principala sursă de inspiraţie a algoritmilor genetici este în mod cert teoria evoluţiei naturale enunţată de Charles Darwin (1809-1882). Cu toate acestea, descoperirile din genetică au influenţat, de asemenea, structura algoritmilor genetici. Teoria Darwinistă reprezintă un ansamblu de observaţii asupra evoluţiei privită la nivel macroscopic şi asupra identificării factorilor responsabili de acest fenomen. Adaptarea unei populaţii este interpretată prin două principii majore: *selecţia naturală* – supravieţuirea celor mai adaptaţi indivizi şi *mutaţia* – variaţie a caracteristicilor indivizilor apărută ca răspuns la mediul înconjurător. Cercetările şi lucrările lui Darwin (ex. *Originea speciilor*, 1959) nasc controverse şi astăzi, însă afirmaţiile conform cărora evoluţia este călăuzită de principiul supravieţuirii celor mai performanţi indivizi şi a diversificării prin mutaţiile apărute, sunt fundamentele algoritmilor evolutivi.

Teoria neo-darwinistă, sau *teoria eredităţii* este un alt punct de vedere asupra evoluţiei, de data acesta interpretat prin transferul de material genetic de la părinţi la descendenţi. La sfârşitul secolul al XIX-lea, un preot şi om de ştiinţă austriac, Gregor Mendel (1822-1884) descoperă mai mult sau mai puţin accidental teoria eredităţii prin experimentele efectuate în vederea încrucişării speciilor diferite de mazăre. Acesta lasă posterităţii câteva însemnări în caietele sale de observaţii şi o lucrare publicată care a fost ignorată la vremea

aceea. Mai târziu, aceste observaţii enunţate de Mendel sunt recunoscute ca fiind legile eredităţii şi autorul acestora este considerat părintele geneticii moderne. Multe aspecte ale algoritmilor genetici sunt împrumutate din genetică. Dintre acestea remarcăm moştenirea trăsăturilor (genelor) părinţilor de către descendenţi, codificarea indivizilor prin secvenţe de *gene*, dar şi caracterul aleator al aportului trăsăturilor provenite de la părinţi.

Algoritmii genetici nu respectă strict ingredientele celor două teorii amintite, iar terminologia împrumutată nu acoperă semnificaţia originală; aceştia se formează ca o reţetă practică inspirată de evoluţie şi ereditate în care factorul uman a strecurat secvenţe de calcule şi parametri artificiali în scopul obţinerii unor rezultate mai bune. Privit la nivel superficial, un algoritm genetic simulează evoluţia unei populaţii, însă, în esenţă, fenomenul evoluţiei artificiale este ghidat de funcţii matematice şi controlat prin parametrii suplimentari pentru a genera o soluţie cât mai apropiată de optimul problemei. Avantajul imediat al algoritmilor genetici constă în gradul mare de generalitate şi paleta largă a problemelor abordate. Originalitatea, simplitatea şi sursa inedită de inspiraţie poate justifica parţial atenţia acordată dezvoltării acestor metaeuristici. Succesul real înregistrat în probleme de dificultate ridicată este argumentul forte al algoritmilor genetici [36].

Descrierea algoritmului genetic standard

Pe scurt, un algoritm genetic operează cu o mulţime de indivizi (denumiţi în mod uzual *cromozomi*) asupra cărora se aplică *operatorii genetici*. Fiecare individ reprezintă o soluţie posibilă din spaţiul de căutare. Maniera de codificare a unei soluţii este binară, reală sau specifică, în funcţie de opţiunea programatorului şi de contextul problemei. Mulţimea de cromozomi formează o *populaţie*. Evoluţia populaţiei este condusă de două elemente: operatorii genetici şi *funcţia de evaluare* a calităţii cromozomilor. *Selecţia*, *încrucişarea* şi *mutaţia* sunt operatorii genetici uzuali, fiind inspiraţi de procesele naturale care stau la baza evoluţiei.

Indiferent de natura problemei considerate, se poate construi cel puţin o funcţie de evaluare a soluţiilor posibile. Prin aceasta, fiecărui individ al populaţiei i se atribuie o valoare numerică reprezentând performanţa sa în raport cu cerinţele problemei. Ulterior, măsura calităţii indivizilor populaţiei curente este folosită în procesul de *selecţie* a acelor indivizi, denumiţi părinţi, asupra cărora se aplică operatorii de încrucişare şi mutaţie pentru obţinerea noii generaţii. Principiul este simplu: cu cât părinţii selectaţi sunt mai performanţi, cu atât şansa ca descendenţii obţinuţi să fie calitativ superiori este mai mare. Procesul se reia având ca populaţie curentă noua generaţie de cromozomi. Se observă de-a lungul evoluţiei populaţiei o creştere a calităţii indivizilor săi. După un număr considerabil de generaţii soluţia globală a problemei poate fi aproximată suficient de bine printr-un individ al ultimelor generaţii.

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

1. individul
2. populaţia
3. funcţia de evaluare
4. selecţia
5. operatorii de variaţie (încrucişare şi mutaţie)
6. condiţia de oprire a algoritmului

1) Individul

Analizând specificaţiile problemei, se poate identifica spaţiul de căutare. Orice punct al acestuia constituie o soluţie posibilă. În funcţie de specificul soluţiei posibile, se poate determina o manieră inspirată de codificare numerică sau nenumerică a acesteia. Codificarea unei soluţii formează un *cromozom* şi identifică un individ al

populaţiei curente. Fiecare cromozom este format dintr-un şir de valori ale unui alfabet dat. Valoarea de pe o poziţie oarecare a şirului codificării se numeşte *genă*. Alfabetul *A* al valorilor genelor este stabilit în prealabil şi poate fi format din:

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

Lungimea şirului de gene din codificarea cromozomului poate fi constantă sau variabilă şi depinde de numărul de trăsături definitorii ale unei soluţii posibile a problemei:

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

$$c = (\alpha_1, \alpha_2, ..., \alpha_n), \qquad (42)$$

$$\text{unde } \alpha_i \in A, \forall i \in \{1,2,...,n\}. \qquad (43)$$

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

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

2) Populaţia

O mulţime finită de cromozomi alcătuieşte populaţia. Dimensiunea populaţiei (numărul de indivizi care o formează) este, în general, o valoare constantă stabilită în prealabil şi reprezintă un parametru important al algoritmului dezvoltat. Într-o abordare mai flexibilă, dimensiunea populaţiei poate fi variabilă, un exemplu în acest sens ar fi descreşterea numărului de indivizi de-a lungul evoluţiei, o dată cu creşterea performanţei acestora, fapt care ar uşura determinarea soluţiilor finale din ultima generaţie produsă. Nu este exclusă utilizarea mai multor populaţii cooperante sau a unei populaţii suplimentare cu rol de memorie, în care sunt reţinute soluţii bune găsite în cadrul generaţiilor intermediare. Modelul standard al algoritmului genetic presupune exploatarea unei populaţii unice de dimensiune constantă. Formal, o astfel de populaţie se descrie astfel:

$$P = \{c_1, c_2, ..., c_m\}, \qquad (44)$$

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

Prima generaţie se numeşte *populaţie iniţială* şi construirea acesteia se face în principal prin generarea aleatoare a unor soluţii posibile în domeniul de căutare. Dacă sunt cunoscute zone ale spaţiului care conţin soluţii bune ale problemei, populaţia iniţială poate fi îmbogăţită cu indivizi care codifică puncte din aceste zone. În această manieră se accelerează convergenţa populaţiei, oferindu-se şansa ca încă de la primele generaţii să fie obţinute soluţii optime ale problemei. Există puţine situaţii în care procedeul descris mai sus poate fi aplicat deoarece din specificaţiile problemei rareori putem deduce zonele promiţătoare ale spaţiului de căutare. Modelul standard al algoritmului genetic presupune generarea aleatoare a populaţiei iniţiale.

Cu cât dimensiunea populaţiei este mai mare, cu atât şansa obţinerii rapide a unei bune aproximări a soluţiilor problemei creşte. De asemenea, distribuţia indivizilor populaţiei joacă un rol important în economia construirii algoritmului genetic: o distribuţie echilibrată a populaţiei iniţiale măreşte substanţial viteza de identificare a zonelor promiţătoare ale spaţiului de căutare. Acest fapt aduce cu sine o convergenţă mai bună înspre soluţiile problemei.

În concluzie, o populaţie de dimensiune suficient de mare şi având un grad de diversitate considerabil reprezintă unul dintre factorii majori ce conduc la obţinerea soluţiilor dorite.

3) Funcţia de evaluare

Una dintre condiţiile fundamentale ale funcţionării eficiente ale unui algoritm genetic o reprezintă alegerea inspirată a funcţiei de evaluare. În cazul unei probleme de optimizare numerică (ex. determinarea maximului/minimului global al unei funcţii matematice), funcţia de evaluare se poate alege ca fiind chiar funcţia criteriu sau o funcţie construită pe baza funcţiei criteriu.

Pentru o mai bună înţelegere prezentăm în continuare un exemplu:

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

$$f(x) = \sin(x) \qquad (45)$$

Se doreşte determinarea maximului funcţiei *f*.

Criteriul care ghidează căutarea soluţiei globale este maximizarea funcţiei 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 populaţiei corespunzător unei valori reale *x* va fi construit ca un vector de cifre binare ale codificării punctului *x*. Se impune observaţia că o abordare mai flexibilă permite ca reprezentarea cromozomială a unui punct *x* al spaţiului de căutare să fie dată de un vector particular cu o singură componentă. Unica genă a codificării reprezintă chiar valoarea reală a punctului corespunzător.

În acest caz particular, funcţia de evaluare este dată de însăşi funcţia criteriu. Fie cromozomul *c* reprezentarea (*binară* sau *reală*) a punctului *x*. Funcţia de evaluare este construită astfel:

$$performanta(c) = \sin(x) \qquad (46)$$

Valorile calculate ale funcţiei de performanţă califică şi diferenţiază indivizii populaţiei. Astfel, este posibilă identificarea indivizilor performanţi ai populaţiei, cei cărora li se oferă ulterior o şansă mai mare de a produce descendenţi.

Există situaţia în care este dificil de exprimat matematic criteriul de optimizare al problemei date. În aceste situaţii, utilizatorul va construi funcţii de evaluare prin care se va încerca calificarea cât mai coerentă a soluţiilor 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 doreşte determinarea optimelor multiple ale unei funcţii (optimizare multimodală) sau cele în care se impun anumite restricţii (optimizare cu restricţii). Pentru rezolvarea acestor probleme, funcţia de evaluare va fi atent construită pentru a respecta restricţiile impuse, respectiv, pentru a permite populaţiei să conveargă înspre toate optimele funcţiei criteriu. Un alt caz în care construirea funcţiei de evaluare necesită un efort mai mare este cel al problemelor de optimizare multiobiectiv. În discuţia referitoare la funcţia de evaluare ne-am referit strict la probleme de optimizare. Justificarea acestui fapt rezidă din afirmaţia că orice problemă de căutare a soluţiilor în spaţiul soluţiilor posibile, căutare ghidată de unul sau mai multe criterii, poate fi reinterpretată ca o problemă de optimizare: din mulţimea soluţiilor posibile se caută acelea care optimizează cel mai bine criteriile problemei. Deşi principala aplicabilitate a algoritmilor genetici constă în rezolvarea problemelor de optimizare, afirmaţia precedentă ne permite extinderea ariei de aplicabilitate a algoritmilor genetici şi înspre probleme de alt gen.

4) Selecţia

Evoluţia populaţiei, obţinerea generaţiilor de indivizi mai performanţi decât predecesorii lor, este cauzată de operatorul de selecţie şi de operatorii de variaţie (recombinarea şi mutaţia) aplicaţi.

Unul dintre factorii majori ai evoluţiei este selecţia naturală. O interpretare simplificată a acestui fenomen ne permite înţelegerea procesului prin care o populaţie poate converge înspre anumite puncte ale spaţiului de căutare, respectiv, înspre soluţiile unei probleme date. Pornind de la observaţia acceptată că indivizii performanţi ai unei specii dau naştere unor descendenţi asemănători şi de performanţe apropiate şi, în schimb, indivizii slab calificaţi produc indivizi noi ale căror performanţe sunt, în general, slabe, putem considera două scenarii:

- în primul scenariu, indivizii unei populaţii suferă încrucişări şi mutaţii pur aleatoare, fără a se aplica principiul selecţiei;
- al doilea scenariu implică o selecţie prealabilă a părinţilor noii generaţii, selecţie făcută pe baza valorilor de performanţă a indivizilor populaţiei curente.

Cele două populaţii ale scenariilor descrise se monitorizează de-a lungul evoluţiei. Se observă că, dacă în primul caz absenţa selecţiei induce o stagnare în procesul evoluţiei populaţiei înspre indivizi de calitate superioară, în a doua situaţie, selecţia favorizează o creştere accelerată a performanţei medii a generaţiilor produse. Comparând ultimele generaţii ale ambelor scenarii putem deduce importanţa selecţiei în evoluţia unei populaţii.

Implementarea operatorului de selecţie se face în diverse maniere. Unul dintre cei mai populari algoritmi de selecţie este cel al *selecţiei proporţionale* inspirat de algoritmul ruletei. Selecţia proporţională presupune alegerea părinţilor noii generaţii în mod proporţional cu valoarea de performanţă a acestora. Altfel spus, probabilitatea de selectare a unui individ performant este mai mare decât a unuia slab calificat, şi această probabilitate este direct proporţională cu valoarea performanţei individului. Acest procedeu favorizează o convergenţă rapidă a populaţiei, însă în unele cazuri prezintă dezavantaje majore, fapt pentru care se optează pentru implementarea altor variante de selecţie.

5) Operatorii de variaţie (încrucişare şi mutaţie)

Selecţia propriu-zisă nu este suficientă pentru a induce dinamica unei populaţii. Indivizii selectaţi vor suferi modificări prin operatorii genetici de variaţie pentru a permite obţinerea unor descendenţi diferiţi, respectiv pentru a permite o explorare bună a spaţiului de căutare. Principalii operatori de variaţie sunt încrucişarea şi mutaţia.

Varianta standard a operatorului de încrucişare se aplică asupra a doi părinţi selectaţi anterior şi produce unul sau doi descendenţi care moştenesc trăsăturile părinţilor în proporţii diferite. Prezentăm în continuare un operator de încrucişare aplicabil pentru codificarea binară. Tipul operatorului este de 2:2, respectiv, din 2 părinţi se produc 2 descendenţi. Datorită specificului său, operatorul descris mai jos este cunoscut sub denumirea de *încrucişare cu un punct de tăietură*.

Fie c_1 şi c_2 doi indivizi oarecare ai populaţiei curente:

$$c_1 = (\alpha_1, \alpha_2, ..., \alpha_n) \qquad (47)$$

$$c_2 = (\beta_1, \beta_2, ..., \beta_n) \qquad (48)$$

Se generează aleator o valoare naturală $t \in \{1,2,...,n\}$, având semnificaţia punctului de tăietură. Ambii părinţi se vor diviza, obţinându-se secvenţele:

$$c_1^1 = (\alpha_1, \alpha_2, ..., \alpha_t),\ c_1^2 = (\alpha_{t+1}, \alpha_{t+2}, ..., \alpha_n) \qquad (49)$$

$$c_2^1 = (\beta_1, \beta_2, ..., \beta_t),\ c_2^2 = (\beta_{t+1}, \beta_{t+2}, ..., \beta_n) \qquad (50)$$

Descendenţii d_1 şi d_2 ai părinţilor c_1 şi c_2 se obţin prin concatenarea secvenţelor obţinute:

$$d_1 = c_1^1 + c_2^2 \qquad (51)$$

$$d_2 = c_2^1 + c_1^2 \qquad (52)$$

Observaţie: Operatorul + are în această descriere formală semnificaţia concatenării secvenţelor.

Se obţin astfel noii indivizi:

$$d_1 = (\alpha_1, ..., \alpha_t, \beta_{t+1}, ..., \beta_n) \text{ şi} \qquad (53)$$

$$d_2 = (\beta_1, ..., \beta_t, \alpha_{t+1}, ..., \alpha_n) \qquad (54)$$

Prin aplicarea încrucişării, noii indivizi moştenesc genele părinţilor, fapt pentru care aceştia sunt asemănători indivizilor care îi produc. Se întâlneşte deseori situaţia în care o anumită soluţie a spaţiului de căutare nu poate fi obţinută prin aplicarea operatorului de încrucişare descris. Există, de asemenea, suficiente argumente în favoarea aplicării altor tipuri de operatori de încrucişare (exemplu: *încrucişarea cu puncte multiple de tăietură*). Pentru asigurarea unei bune explorări a spaţiului de căutare, un alt operator genetic este introdus: *mutaţia*. Teoria evoluţiei naturale menţionează faptul că selecţia nu este unicul factor responsabil al fenomenului evoluţiei. Mutaţia naturală, definită prin micile modificări la nivelul codificării cromozomiale, este cea care, în combinaţie cu mecanismul selecţiei, asigură adaptarea la mediu. Implementarea procesului natural al mutaţiei se face prin *operatorul de mutaţie*, care, la rândul său, cunoaşte variate forme descrise în lucrări de specialitate.

Prezentăm în continuare o variantă simplă a operatorului de mutaţie. Acest operator este de tipul 1:1, respectiv, dintr-un părinte selectat se obţine 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 populaţiei curente:

$$c = (\alpha_1, \alpha_2, ..., \alpha_k, ..., \alpha_n) \qquad (55)$$

Se generează aleator o poziţie *k* din secvenţa binară a codificării.

Individul mutant $c' = (\alpha_1, \alpha_2, ..., \beta_k, ..., \alpha_n)$ se obţine prin copierea nealterată a genelor părintelui *c*, cu excepţia genei de pe poziţia *k*. Valoarea genei mutate se obţine prin inversare:

Dacă $\alpha_k = 1$ **atunci**

$$\beta_k \leftarrow 0$$

Altfel

$$\beta_k \leftarrow 1$$

SfDacă

Alegerea operatorilor folosiţi 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. Specificaţiile problemei sunt cele care ne pot oferi informaţiile 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 procedeelor de încrucişare şi mutaţie, clasificate pe tipul codificării pre-fixate, fac dificilă alegerea inspirată a celor mai eficiente variante. Preferinţele se îndreaptă înspre acei operatori care se dovedesc a fi eficienţi în rezolvarea unor probleme similare. Lipsa constrângerilor

legate de implementarea operatorilor genetici, permite utilizatorului să îşi construiască operatori specifici problemei sau clasei de probleme pe care le abordează.

6) Condiţia de oprire a algoritmului genetic

În esenţă, algoritmul genetic este o procedură repetitivă prin care o populaţie de soluţii posibile, prin aplicarea operatorilor genetici (selecţie, încrucişare, mutaţie) produce noua generaţie. Procesul se reia pentru noua populaţie obţinută până când o condiţie de terminare este îndeplinită. Condiţia de terminare a algoritmului se referă în general la atingerea unui număr maxim prestabilit de generaţii. Există şi situaţii în care utilizatorul poate să impună o cu totul altă condiţie de încheiere a evoluţiei, condiţie independentă de numărul populaţiilor generate. Un exemplu în acest sens ar fi înregistrarea unui anumit grad de uniformitate în cadrul populaţiei. În situaţia în care indivizii populaţiei devin asemănători putem considera că diversitatea slabă a populaţiei nu mai permite o explorare eficientă a spaţiului de căutare. Mai mult, putem concluziona că indivizii populaţiei s-au stabilit în zona cea mai promiţătoare, corespunzătoare soluţiei globale. Experimentele arată că nu întotdeauna această concluzie este adevărată. Populaţia poate fi atrasă de un punct de optim local, producându-se fenomenul de convergenţă prematură. Convergenţa prematură este unul dintre fenomenele nedorite ale unui algoritm genetic şi evitarea acestui neajuns comportă câteva modificări în structura algoritmului: alegerea unei alte scheme de selecţie, reiniţializarea aleatoare unei secţiuni din populaţie sau accentuarea mutaţiilor produse. O altă soluţie oferită este cea prin care sunt monitorizate ultimele *k* generaţii obţinute. Pentru a nu cădea în posibila capcană a scăderii temporare a gradului de diversitate a populaţiei putem considera un număr mai mare de generaţii analizate din punct de vedere al asemănării indivizilor. În această manieră obiectivitatea deciziei de terminare a evoluţiei creşte substanţial. Parametrul *k* reprezintă în acest caz o valoare numerică naturală prestabilită.

Revenind la condiţia de terminare, ca şi în cazul celorlalte componente ale algoritmului genetic, alegerea acesteia se face de către proiectantul algoritmului după o atentă observare a comportamentului populaţiei de-a lungul generaţiilor. O bună experienţă a proiectantului, cât şi concluziile experimentelor efectuate, sunt factorii principali ai construirii unui algoritm genetic valoros.

Înainte de a furniza schema algoritmului genetic standard, este important să afirmăm că acest algoritm nu funcţionează ca un şablon de rezolvare a oricărei probleme. Detalii legate de implementare, alegerea operatorilor, construirea funcţiei de evaluare, precum şi stabilirea parametrilor impliciţi reprezintă aspecte care diferenţiază algoritmii obţinuţi din punct de vedere al aplicabilităţii 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 $P(0)$ **- *populaţia iniţială*;** $t = 0$ **;**

Câttimp (NOT *condiţie de terminare*)

***Evaluare*(** $P(t)$ **)**

$P_S(t)$ **=*Selecţie* (** $P(t)$ **)**

$P_R(t)$ **=*Încrucişare*(** $P_S(t)$ **)**

$P(t+1)$ **=*Mutaţie* (** $P_R(t)$ **)**

t:=t+1

SfCâttimp

SfAlgoritm

Notăm:

t – numărul generaţiei curente.

P_S - mulţimea indivizilor selectaţi

P_R - mulţimea indivizilor obţinuţi prin încrucişare

Soluţia finală a algoritmului genetic este dată de cel mai performant individ al ultimei generaţii produse. Se impune observaţia că această variantă de extragere a rezultatului nu este cea mai eficientă. Considerăm următorul scenariu: un individ performant, extrem de apropiat de soluţia globală a problemei) este obţinut într-una dintre generaţiile intermediare. Principiul selecţiei ne conduce la supoziţia că probabilitatea acestuia de a fi selectat în vederea aplicării operatorilor de variaţie este mare. Operatorul de încrucişare nu asigură păstrarea părinţilor performanţi în noua generaţie. Astfel, individul considerat în scenariul nostru va fi distrus, neavând certitudinea că descendenţii săi sunt cel puţin la fel de performanţi ca şi el. Observăm astfel că indivizii calificaţi ai generaţiilor intermediare pot dispărea pe parcursul evoluţiei populaţiei.

Remedierea acestui fenomen poate fi făcută extrem de simplu prin suplimentarea algoritmului iniţial cu o formă de *elitism*. Prin elitism, cel mai bun, sau cei mai buni *k* indivizi ai generaţiei curente vor fi copiaţi nemodificaţi în noua generaţie. Se evită în acest mod pierderea soluţiei globale dacă aceasta este obţinută într-o etapă intermediară a evoluţiei.

Funcţionarea algoritmului genetic

Intuitiv, algoritmul genetic va conduce populaţia de soluţii posibile înspre soluţiile optime ale problemei de rezolvat. De cele mai multe ori acest fenomen decurge în mod firesc, fără anomalii, şi putem spune că algoritmul converge înspre optimele problemei. Există şi situaţia în care indivizii populaţiei se blochează în zone sub-optimale ale spaţiului de căutare, evoluţia lor ulterioară fiind îngreunată sau imposibilă. În aceste situaţii rezultatul oferit de algoritm este eronat. Cauzele acestui anomalii sunt multiple. Una dintre cele mai frecvente surse de eşec este alegerea neinspirată a funcţiei de evaluare şi a operatorului de selecţie, rezultând pierderea diversităţii populaţiei şi incapacitatea de a determina soluţiile globale. Fenomenul nedorit, descris ca obstrucţionarea explorării spaţiului de căutare şi blocarea populaţiei în zone corespunzătoare optimelor locale, este denumit *convergenţă prematură*.

În esenţă, convergenţa prematură este cauzată de pierderea diversităţii genetice a indivizilor într-o etapă timpurie a evoluţiei, fapt care generează o stagnare a evoluţiei ulterioare. Probabilitatea ca indivizii asemănători ai populaţiei curente să genereze descendenţi diferiţi este redusă, ceea ce induce o rezistenţă sporită la dislocarea indivizilor din zonele de blocaj.

În scopul evitării convergenţei premature au fost dezvoltate diferite strategii de menţinere a diversităţii populaţiei. Dintre acestea enumerăm:

- stabilirea unei dimensiuni mai mari a populaţiei;
- aplicarea unor operatori genetici care s-au dovedit benefici evoluţiei şi menţinerii diversităţii (ex. încrucişarea uniformă);
- strategii de împerechere care evită obţinerea de indivizi similari prin aplicarea încrucişării;

- aplicarea mutaţiei cu o mai mare probabilitate;
- evitarea folosirii operatorului de selecţie proporţională care, în anumite situaţii, va favoriza evoluţia doar a unei mici fracţiuni de indivizi din populaţie, conducând la omogenizarea populaţiei.

Un alt aspect al funcţionării algoritmului genetic este cel al *elitismului*. În multe situaţii, o soluţie bună a problemei (codificată ca individ al populaţiei) poate să apară într-una dintre generaţiile intermediare; având o performanţă mare, şansa de a fi selectată în vederea supunerii acesteia la acţiunea operatorilor de variaţie (mutaţie şi încrucişare) este mare. Aplicarea operatorilor genetici poate să distrugă soluţia performantă, fapt pentru care evoluţia populaţiei este încetinită. Remedierea acestui fenomen constă în aplicarea unor strategii de salvare a indivizilor performanţi determinaţi în populaţiile intermediare. În mod sugestiv, aceşti indivizi cu performanţe ridicate se denumesc *elită* şi procedeul prin care elita este menţinută de-a lungul populaţiilor se denumeşte *elitism*. Cea mai simplă strategie de elitism constă în copierea elitei în noua generaţie. Acest transfer garantează protejarea soluţiilor calificate de-a lungul evoluţiei şi o mai bună convergenţă a algoritmului genetic.

Funcţionarea algoritmului genetic este strâns legată de două concepte duale: *explorarea* şi *exploatarea* spaţiului de căutare. Primul concept se referă la capacitatea populaţiei de a acoperi spaţiul soluţiilor posibile prin căutare globală şi este strâns legat de proprietatea de diversitate a populaţiei. Cel de-al doilea concept se referă la capacitatea populaţiei de a asigura căutarea locală în zonele promiţătoare ale spaţiului şi corespunde unei îmbunătăţiri a indivizilor în vederea obţinerii unei aproximări mai bune a soluţiilor finale. Cele două fenomene definite de conceptele sus-menţionate sunt duale în sensul în care, accentuarea unuia conduce în mod gradual la inhibarea apariţiei celuilalt. Astfel, o *explorare* asiduă a spaţiului de căutare, marcată prin menţinerea unei populaţii diverse, îngreunează procesul de rafinare a soluţiilor bune şi cauzează a stagnare în evoluţia populaţiei. Contrar, accentuarea *exploatării* zonelor promiţătoare conduce la o explorare defectuoasă a spaţiului de căutare şi poate conduce la fenomenul de convergenţă prematură. Ambele situaţii descrise trebuie evitate. Pentru buna funcţionare a algoritmului genetic din perspectiva celor două concepte, este necesară stabilirea unui echilibru *explorare-exploatare* obţinut, în principal, prin aplicarea unui mecanism de evaluare şi selecţie bine ales.

Algoritmii genetici au fost folosiţi atât în clasificare [64], [99], cât şi în alte probleme de optimizare [100], [101], [102]. În *data mining* aceştia sunt utilizaţi pentru a evalua fitness-ul altor algoritmi.

4.3. Clasificatori bazaţi pe reguli de decizie şi algoritmi evolutivi pentru detecţia intruşilor

Întrucât regulile de decizie pot fi reprezentate cu uşurinţă ca şi *bit-string*-uri de lungime fixă, se pot utiliza împreună cu algoritmi genetici cu scopul creşterii performanţei clasificatorului. În contextul *data mining*-ului, populaţia reprezintă spaţiul de căutare a soluţiei, unde o soluţie este reprezentată de o regulă DACĂ - ATUNCI. Un cromozom este o regulă DACĂ – ATUNCI, iar o genă reprezintă o sub-condiţie a regulii. Bazându-se pe noţiunea de supravieţuire a celui mai performant individ, o nouă populaţie este formată din cele mai bune reguli ale populaţiei curente, dar şi din urmaşii acestor reguli. În mod tipic, fitness-ul unei reguli este reprezentat de acurateţea clasificării instanţelor de antrenare.

Algoritmul propus funcţionează în felul următor [28], [62], [103]:

1. *Încarcă setul de date care urmează a fi procesat;*
2. *Parsează regula DACĂ – ATUNCI în forma antecedentă şi forma consecventă;*
3. *Calculează distribuţia clasei pentru fiecare regulă;*