

ALGORITMI GENETICI, ÎNTRE CERTITUDINI ȘI SPERANȚE

de Florin Buzuloiu

Institutul de Cercetări în Informatică

Rezumat: În articol se trec în revistă principiile generale de construire a Algoritmilor Genetici, operatorii folosiți și modul de lucru în cadrul modelării unei probleme ca algoritm genetic, deci ca un program de evoluție. Este prezentată apoi rezolvarea unei probleme de planificare a experiențelor din domeniul cercetării experimentale printr-un algoritm genetic elitist cu imigrare/emigrare și concluziile simulărilor efectuate. În final, se propun câteva domenii în care autorul consideră că abordarea prin tehnica algoritmilor genetici ar fi benefică.

Cuvinte cheie: Algoritmi Genetici (AG), programe de evoluție, statistică matematică, simulare, euristică.

1. Introducere

În prima jumătate a decadei a 8-a a acestui secol apar și se dezvoltă algoritmi genetici ca o metodă de rezolvare a problemelor combinatoriale grele [1]. De-a lungul următoarei decade ei au rămas mai curând neobservați: în ultimul timp însă, cercetători din diverse domenii, din ce în ce mai numeroși, prezintă investigații cu rezultate remarcabile. Amintim în acest sens că, în anul 1993, a avut loc cea de a 5-a Conferință Internațională asupra Algoritmilor Genetici - ele ținându-se din doi în doi ani. Atractivitatea acestor metode de cercetare și această "creștere de popularitate" se explică prin faptul că, algoritmi genetici au un foarte mare grad de aplicabilitate de la problema clasică a comis-voiajorului, probleme de joc și, în general, probleme de cercetare operațională și până la modele complicate de studiu și control al fenomenelor chimice, biologice, electrodinamice (rezultatele sunt performante) și o relativă ușurință în abordare. În special, algoritmi genetici își găsesc aplicabilitatea acolo unde rezolvarea problemei nu se poate face cu metode tradiționale sau rezolvarea prin aceste metode solicită un timp de calcul foarte mare, cunoașterea soluției exacte sau optime nemotivând economic efortul.

Din articolele sau cărțile care tratează algoritmi genetici și care prezintă latura lor practică, puține sunt cele care prezintă o formalizare a acestora; semnalăm, totuși, un articol în care într-un mod foarte riguros se face legătura între ceea ce autorii denumesc algoritm genetic abstract (AGA), noțiune care include ca particularizări, atât orice algoritm genetic clasic, cât și orice algoritm de cristalizare imulată și lanțurile Markov din teoria probabilităților [3].

2. Prezentare generală, noțiuni specifice

2.1. Origini

Algoritmul genetic este o "abstractizare metaforică" a evoluției biologice naturale, fiind, deci, un algoritm de căutare și optimizare, similar cu ceea ce se întâmplă în genetica naturală. Conceptele de bază ale evoluției naturale, care se regăsesc în orice algoritm genetic sunt: selecția naturală, recombinarea și mutația genetică. Cu ajutorul acestor trei procese, denumite aici operatori, la care se adaugă operațiile de decodificare se construiește întregul algoritm genetic.

Procesul evolutiv pentru algoritmi genetici, ca și în natură, se desfășoară în cromozomi care sunt lanțuri de biți, în care fiecare porțiune, uneori chiar bitul, reprezintă gene, valorile acestora fiind alele; în edificiul biologic cromozomii sunt aglomerări de materie organică în care este conținut materialul genetic codificat și care determină structura și comportamentul ființelor vii.

Identificând un cromozom cu un individ dintr-o populație, principiul darwinist al evoluției naturale va caracteriza aici procesul evolutiv prin atribuirea unei probabilități de reproducere superioară cromozomilor cu capabilitate mai mare, această capabilitate fiind rezultatul decodificării materialului genetic. Similar cu ceea ce se întâmplă în cazul speciilor, o dată selectați pentru reproducere pe baza acestei valori care poate să fie funcția obiectivă a unui model matematic oarecare, cromozomii se recombină schimbând material genetic și creând urmași asemănători sau complet diferiți de ei. În acest proces de trecere de la o generație la alta are loc cu o anumită probabilitate și fenomenul de mutație, prin schimbarea în interiorul unei gene a valorii unui bit, adică modificarea alelei. Procesul evolutiv continuă până ce se parcurge un număr predeterminat de generații, rezultând în generația finală cromozomi cu o capabilitate satisfăcătoare, reprezentând soluția optimă sau aproape optimă a problemei.

Algoritmi genetici au fost definiți de John Holland, colaboratorii și studenții săi de la Universitatea Michigan [2], iar dezvoltarea lor ulterioară se poate regăsi în cartea lui D.E. Goldberg "Genetic Algorithms in Search, Optimization and Machine Learning" din 1989.

Algoritmi genetici (AG) diferă de metodele tradiționale de optimizare sau de procedurile de căutare prin următoarele patru caracteristici principale:

- (1) AG lucrează cu codificări ale mulțimii parametrilor, nu cu parametrii înșiși;
- (2) AG caută o populație de puncte, nu un singur punct;
- (3) AG folosesc informația dată de funcția obiectivă, nu derivate ale acesteia sau cunoștințe auxiliare;
- (4) AG folosesc legi de transfer probabiliste, nu reguli deterministe.

Structura unui algoritm genetic simplu (SGA) este aceea a unui program de evoluție și anume, schema:

procedure evolution program

begin

t=0;

initialize P(t); evaluate P(t); while (not termination condition) do

begin

t=t+1;

select P(t) from P(t+1); recombine P(t);
evaluate P(t);

end; end.

În natură dezvoltarea unei populații oarecare depinde de doi factori majori: cum se nasc indivizii și cum mor ei; copiii sunt produși prin recombinarea genelor parentale, noul genotip determinând un nou individ. Schematizarea matematică pentru situația biologică în care, o dată cu apariția descendenților (care pot fi chiar identici cu părinții), părinții dispar, este existența unui spațiu de căutare S și a unei funcții obiectiv f ; parcurgerea spațiului în scopul găsirii extremelor lui f implică generarea și eliminarea elementelor lui S . Pentru algoritmi genetici acest spațiu de căutare este mulțimea perechilor de cromozomi selectați pentru reproducere. Ideea majoră, dar adeseori neclară sau nesubliniată efectiv în timpul folosirii unui algoritm genetic este aceea de ereditate. De cele mai multe ori se presupune că indivizii puternici vor avea mai mulți urmași decât cei slabi și că acest principiu acționează de-a lungul întregului proces evolutiv, în fiecare generație; acesta este, de fapt, mecanismul care conduce sistemul către un optim. Ameliorarea indivizilor poate fi considerată drept adaptarea întregii populații la cerințele mediului.

2.2. Descriere.

Un algoritm genetic pentru orice fel de problemă trebuie să aibă următoarele cinci componente:

- o reprezentare genetică pentru soluțiile potențiale ale problemei;
- un procedeu de creare a populației de soluții potențiale;
- o funcție de evaluare prin care să se poată aprecia potența de reproducere a individului, capacitatea cromozomului;
- operatori genetici, care modifică urmașii de-a lungul procesului de reproducere;
- valori pentru diferiți parametri pe care algoritmul genetic îi utilizează (mărimea populației, numărul de generații, probabilitățile de aplicare a operatorilor folosiți etc.).

Reprezentarea genetică este cea de cromozom, adică un lanț de biți inițial generați aleator și atâți cromozomi cât este valoarea parametrului care stabilește mărimea populației. Lungimea cromozomului depinde de natura problemei, înțelegând prin aceasta chiar și gradul de

aproximație dorit [3]. Spre exemplu: dacă avem de optimizat funcția simplă de o variabilă $f(x) = x \cdot \sin(10x) + 1$ pe intervalul $[-1, 2]$ cu o aproximație la a șasea zecimală, intervalul de lungime 3 va trebui împărțit în $3 \cdot 10^6$ subintervale și din relația $3 \cdot 10^6 < 2^{22}$ rezultă că avem nevoie de 22 de biți pe cromozom. Cromozomul (1000101110110101000111) va reprezenta numărul 2288967 adică valoarea lui $x = -1 + 3.2288967 / (2^{22} - 1) = 0,637197$. Evident cromozomii (0000...0000) și (1111...1111) reprezintă capetele intervalului de variație, -1 respectiv 2.

Întreaga populație de cromozomi va fi generată aleator.

Procedeu de evaluare vizează decodificarea cromozomului, pe porțiuni dacă este cazul, și, în final, ca valoare de funcție obiectiv. Apreciem că această componentă, codificarea formală a soluției potențiale ca un cromozom și decodificarea acestuia împreună cu aplicarea operatorilor specifici constituie kernelul algoritmilor genetici.

Principalii operatori genetici sunt cei amintiți mai înainte: de selecție, de combinare și de mutație.

Selecția cromozomilor pentru determinarea perechilor de părinți în vederea reproducerii, precum și aplicarea operatorilor crossover se poate face determinist, euristic sau aleator. Există o multitudine de posibilități de manipulare, rezultate din combinații ale modului de selectare și crossover, folosite fie pentru a feri algoritmul de prematura convergență, fie pentru a-i determina caracterul elitist [6]. De asemenea, mutația se poate aplica supravegheat, cu o probabilitate variabilă sau ca operator unic și ea schimbă valoarea unui bit modificând gena corespunzătoare; fenomenul natural biologic este cel de mutație genetică și, prin extensie, asimilat cu cel de imigrare [6].

Parametrii algoritmului reglează procesul de convergență: dimensiunea populației precum și numărul de generații parcurse trebuie stabilite în mod judicios, ele fiind, de regulă, rezultate ale unui proces de simulare.

2.3. Modul de lucru

Populația inițială de cromozomi se generează aleator, biți cu biți, folosind o subrutină existentă în orice mediu de programare. Se face apoi calculul capacității fiecărui cromozom prin decodificarea lui ca soluție potențială a problemei și se aplică unul din operatorii de selectare.

Fiecărei perechi de părinți selectați în vederea recombinației se aplică operatorul crossover. Pentru ilustrare: fie cromozomii părinți $(a_1, b_1, c_1, d_1, e_1)$ și $(a_2, b_2, c_2, d_2, e_2)$ unde a, \dots, e sunt cele cinci presupuse gene, operatorul crossover aplicat după a doua genă va transforma cei doi părinți în urmașii $(a_1, b_1, c_2, d_2, e_2)$ și $(a_2, b_2, c_1, d_1, e_1)$.

Mutația, modificarea valorii unui bit sau, eventual, a mai multora intervine cu o probabilitate egală cu rata mutație genetice, un parametru stabilit în inițializări și e.

funcționează asupra unuia sau a mai multor cromozomi tabiliți aleator.

ambii operatori, de mutație și crossover, semnificând modificarea indivizilor de la o generație la alta nu sunt decât modalități de explorare a spațiului soluțiilor.

Obținând noua generație prin înlocuirea perechilor de părinți cu perechea de urmași sau prin păstrarea unor părinți aloroși (marcați) și înlocuirea doar a unor perechi de părinți, aceștia îi vor fi aplicați din nou operatorii de selecție, mutație și crossover, după schema generală.

Parametrii algoritmului pot fi modificați și în timpul execuției lui, execuție care poate fi efectuată aleator, în numite limite de variație sau euristic, prevalând cunoștințele în domeniu ale specialistului.

Pentru exemplificare, să considerăm cazul când avem de optimizat funcția de mai multe variabile $f(x_1, \dots, x_k): R^k \Rightarrow R$ unde fiecare variabilă x_i poate lua valori într-un domeniu $D_i = [a_i, b_i] \in R$, funcția fiind strict pozitivă pe $\prod D_i$. Dorim să avem șase cifre zecimale și pentru o asemenea precizie fiecare domeniu D_i va fi divizat în $(b_i - a_i) \cdot 10^6$ subdiviziuni egale, m_i fiind cel mai mic întreg pentru care $(b_i - a_i) \cdot 10^6 \geq 2^{m_i} - 1$, reprezentarea lui x_i va fi:

$$x_i = a_i + (101011\dots 101)_{10} \cdot (b_i - a_i) / (2^{m_i} - 1)$$

Pentru fiecare cromozom, ca soluție potențială a problemei, va fi

un lanț de biți de lungime $\sum_{i=1}^k m_i = m$, fiecare grupă de m_i

biți fiind o valoare în intervalul $[a_i, b_i]$.

Pentru a crea populația inițială stabilim un număr natural care să reprezinte semnificația de dimensiune a populației (pop_size) și generăm aleator bit cu bit atâtția cromozomi. Restul algoritmului este următorul: în fiecare generație evaluăm fiecare cromozom prin capacitatea lui, funcția f de optimizat, se selectează noua populație corespunzător repartiției de probabilitate, determinată pe baza acestor valori, și se recombina cromozomii în noua populație prin operatorii mutație și crossover. După un număr de generații când nu se observă nici o îmbunătățire, cel mai bun cromozom va reprezenta soluția (posibil globală) optimală; în mod frecvent însă, algoritmul va fi oprit după un număr înainte fixat de generații, parametru care se stabilește în dependență cu resursele de calcul.

Selecția cromozomilor se face folosind un procedeu numit ruletă cu fante proporționale. Aria fantelor sau cercele de cerc sunt proporționale cu valorile probabilităților cromozomilor. Un asemenea procedeu funcționează în felul următor:

- Se calculează valoarea capacității pentru fiecare cromozom, $eval(v_i)$;
- Se calculează valoarea totală a capacității pentru populație $F = \sum_{i=1}^{pop_size} eval(v_i)$
- Se calculează probabilitatea de selecție p_i pentru

fiecare cromozom, $p_i = eval(v_i)/F$

- Se calculează probabilitatea cumulată q_i pentru

$$q_i = \sum_{j=1}^i p_j$$

Procesul de selecție constă din generarea de numere aleatoare de atâtea ori cât este mărimea populației, de fiecare dată selectând un singur cromozom pentru noua populație în modul următor:

- se generează un număr aleator în intervalul $[0, 1]$;
- dacă acest număr, r , este strict mai mic decât q_1 se selectează primul cromozom, v_1 ; în caz contrar, se selectează al i -lea cromozom, v_i , astfel ca $q_{i-1} < r < q_i$, bineînțeles cromozomii fiind ordonați după valoarea lui q_i .

Evident unii cromozomi vor fi selectați de mai multe ori, ceea ce este în concordanță cu principiul evoluției naturale amintit.

Un alt parametru al algoritmului genetic este probabilitatea operatorului crossover. Această probabilitate ne dă numărul cromozomilor cărora li se aplică operatorul crossover, și anume: pentru fiecare cromozom din noua populație

- se generează aleator un număr r în intervalul $[0, 1]$;
- dacă $r < p_c$ se selectează cromozomul pentru crossover.

Se vor împerechea cromozomii selectați în mod aleator: pentru fiecare pereche de cromozomi cuplați se va genera un număr întreg "pos" în intervalul $[1, m-1]$ care reprezintă poziția punctului de crossing.

Următorul operator de recombinare, mutația, acționează în conformitate cu un alt parametru stabilit al algoritmului, probabilitatea mutației, p_m , și anume: biții de pe poziția $p_m \cdot m - pop_size$ din toți cromozomii populației au aceeași șansă de a suferi mutația din 0 în 1 sau vice versa. Se procedează în felul următor:

- se generează un număr aleator r în intervalul $[0, 1]$;
- dacă $r < p_m$, biții respectivi suferă mutația.

Ulterior selecției, crossover și mutației, noua populație este pregătită pentru reevaluare care va da distribuția de probabilitate pentru următorul proces de selectare, construirea ruletei cu fante proporționale etc. Restul algoritmului se desfășoară ciclic.

3. Un algoritm genetic în cercetarea experimentală

3.1. Obiectul studiului

Scopul cercetărilor experimentale este cunoașterea și conducerea fenomenelor, de la cele naturale până la procesele de producție complexe. De cele mai multe ori legitatea fenomenului se prezintă sub forma unei hipersuprafețe de regresie, care trebuie determinată pe baza

observațiilor statistice asupra valorilor factorilor de influență, și a valorii variabilei dependente, ce caracterizează fenomenul studiat.

Metodologia Cercetării Experimentale (Experiment Design) își concentrează atenția asupra stabilirii matricilor de experiență optimale, conform cu diferite criterii, însă în multe cazuri matricile de experiență clasice nu se pot obține și este necesară construirea unor matrici de experiență particulare. Pentru prelucrarea acestor matrici se folosesc algoritmi de schimb (Mitchel, Feodorov etc.), într-un anumit fel similari simplexului, având criterii de intrare și de ieșire din bază; aceștia presupun extragerea unor mulțimi reduse de "puncte candidat", care trebuie să verifice anumite restricții de calitate. În cazul când există un număr foarte mare de factori și care posedă un număr mare de niveluri posibile, numărul de puncte candidat devine de ordinul milioane, iar algoritmi clasici de schimb nu se mai pot aplica eficient.

În cele ce urmează vom descrie construirea unui algoritm genetic pentru determinarea matricilor D-optimale, care intervin în stabilirea unui plan experimental pentru o dependență lineară în trei variabile.

Presupunem, deci, că avem trei factori de influență x, y, z ale căror valori determină în experiențe succesive un răspuns măsurat, u .

Scopul este determinarea unui set minim de experiențe pe baza cărora să se poată calcula valoarea lui u ca o dependență lineară de forma:

$$u = ax + by + cz + d$$

Aceasta este o problemă clasică de analiză de regresie și se rezolvă prin metoda celor mai mici pătrate, folosind pseudoinversa matricii:

$$A = \begin{pmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ \dots & \dots & \dots & \dots \\ 1 & x_n & y_n & z_n \end{pmatrix}$$

n fiind numărul de experiențe efectuate. Avem

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = (A^T A)^{-1} A^T \begin{pmatrix} u_1 \\ \dots \\ u_n \end{pmatrix}$$

Chestiunea care se pune este optimizarea matricii A și pentru aceasta se folosesc câteva criterii, cum ar fi: D-optimalitatea, adică determinantul matricii $(A^T A)^{-1}$ să fie maxim, λ -optimalitatea, adică raportul dintre valorile proprii $\lambda_{\max}/\lambda_{\min}$ ale matricii $(A^T A)$ să fie minim, T-optimalitatea, adică urma matricii $(A^T A)^{-1}$ să fie minimă, criterii care nu sunt simultan îndeplinite, iar nici

unul dintre algoritmi de schimb nu folosesc metodă multicriterială. Noi am ales D-optimalitatea respectând tradiția Laboratorului de Metodologie a Cercetării Experimentale (L.M.R.E.) din Universitatea Aix-Marseille III unde s-a efectuat aplicația [7].

3.2. Construirea algoritmului genetic

Cei trei factori de influență considerați sunt presupuși că pot lua valori între 1 și 2,5 cu determinare de 0,1, deci 1 valori posibile pentru fiecare dintre ei; simplificarea este doar aparentă și convențională, generalizarea constituind numai o problemă de alocare de memorie și decodificare. Cromozomul considerat va reprezenta planul experimental minim, adică patru determinări ale celor trei factori de influență și, deci, va fi un lanț de biți, constituit din 48 biți cu valori 0 sau 1 generate aleator. Spre exemplu cromozomul:

010100011110..... 10110111

va reprezenta matricea experimentală

$$\begin{pmatrix} 1 & 1,5 & 1,1 & 2,4 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 1 & \dots & 2,1 & 1,7 \end{pmatrix}$$

Numărul acestor cromozomi posibili este imens: $2^{48} - 1 \cong 28$ de milioane de milioane.

Algoritmul genetic folosit este unul de tip elitist, cu emigrare și imigrare. Pe scurt, are următoarea structură:

- Se generează aleator o populație de cromozomi și pentru fiecare cromozom, acesta se decodifică sub formă de matrice experimentală, A , se calculează produsul $A^T A$ și determinantul inverse: $\det((A^T A)^{-1})$. În populația inițială sunt reținuți doar acei cromozomi care au valoarea acestui determinant mai mare decât o valoare specificată în procedura de inițializare unde, de asemenea, se declară și dimensiunea populației (de exemplu, între 30 și 50).
- Se determină cromozomul cu valoare determinantului maximă și se marchează. Pentru ceilalți cromozomi, conform mecanismului de selecție, se aleg aleator perechi de cromozomi părinți, cărora li se aplică operatorul de crossover în o poziție de bit determinată aleator, obținându-se astfel o pereche de cromozomi urmași. Mulțimea de cromozomi urmași împreună cu cromozomul marcă constituie populația din generația următoare. Există posibilitatea ca, în cazul când ambii cromozomi urmași sunt inferiori cromozomilor părinți, ei să nu intre în generația următoare, în locul lor să fie păstrați părinții; când cel puțin unul dintre urmași este superior părinților, ambii urmași intră în

generația următoare. Aceasta devine generația actuală.

- Pentru generația actuală se reiau calculele anterioare. Dacă unul dintre cromozomi are valoarea determinantului, superioară celui marcat, acesta îi ia locul.
- Se procedează astfel pentru un număr predeterminat de generații.
- În locul operatorului mutație am preferat folosirea emigrării/imigrării. Astfel, după numărul de generații de care am amintit, o parte din cromozomii care constituie populația finală și anume cei mai puțin valoroși vor fi eliminați și înlocuiți cu alți cromozomi generați aleator și care sunt supuși aceleiași restricții inițiale sau a uncia îmbunătățite în mod aleator. Numerele acestor indivizi care emigrează sau imigrează sunt și ele generate aleator și, în general, vor fi negale, astfel încât dimensiunea populațiilor poate să nu rămână constantă de la un proces de emigrare/imigrare la altul.
- Pentru trecerea de la o generație la alta am considerat și posibilitatea combinării, prin crossover, a cromozomului marcat cu ceilalți cromozomi din populație în mod aleator.

Programul pentru algoritmul genetic descris, scris în turbo PASCAL vers. 7 este următorul:

```
begin init_data;init_pop;{monogamique}text;
if condition_1 then

begin ordering;dreamchild;end else begin
ordering;crossover;

immigration;text;if condition_2 then ordering;
dreamchild;end;

report_mono;{poligamique} harem;report poli;end.
```

4. Rezultate și concluzii

4.1. Referitor la algoritmul experimentat

În procesul de simulare s-a ajuns la rezultate suficient de apropiate de optimul general, cunoscut, adică matricea de experiență:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2,5 & 1 & 2,5 \\ 1 & 1 & 2,5 & 2,5 \\ 1 & 2,5 & 2,5 & 1 \end{pmatrix}$$

ceea ce probează viabilitatea algoritmului și posibilitatea aplicării ideilor care au stat la baza modelării pentru determinarea de planuri experimentale pentru o multitudine de dependențe, chiar și pentru cele nelinearizabile.

S-a constatat că nu există o diferență esențială între aplicarea operatorului crossover la nivel de bu, față de nivelul de genă sau chiar la nivelul setului de gene, care reprezintă o determinare experimentală.

Are o mare importanță alegerea judicioasă a limitei de selecție pentru populația inițială: se mărește timpul pentru generarea populației inițiale, dar procesul de convergență este mult mai rapid.

În general, cam 85% din cazuri, cromozomii urmași a doi părinți "valoroși" sunt inferiori acestora, ceea ce motivează opțiunea pentru elitism.

4.2. Observații de sinteză și propuneri

În sintagma "algoritmi genetici" noțiunea de algoritm este într-un fel extinsă, folosindu-se doar nota caracteristică de recursivitate; nu există, până în prezent, și nici nu s-a pus problema unui program, chiar parametrizat, ca pentru simplex, drum critic sau alți algoritmi folosiți în cercetarea operațională, pentru o problemă abordată printr-un algoritm genetic.

Așa cum am mai amintit, în cazul folosirii algoritmilor genetici, este esențială modelarea problemei pentru a i se putea aplica operatorii geneticii specifici, de selecție, mutație, crossover, imigrare, emigrare, reciclare etc.

Prematura convergență a algoritmului poate fi evitată printr-o alegere judicioasă a parametrilor ca: mărimea populației, probabilitățile de aplicare a operatorilor genetici, numărul de generații etc. ceea ce necesită simulări multiple. În termeni matematici, prematura convergență este echivalentă cu intrarea procesului într-o stare absorbantă, alta decât optimul absolut. Nu s-a demonstrat faptul că algoritmul genetic ar fi un proces ergodic.

În afara celor amintite în 2.1., subliniem caracterul euristic al celor mai multe dintre aplicațiile întâlnite în literatură [2,5,6]. Noi înșine (laboratorul 1.10 din ICI) suntem preocupați de găsirea unor domenii unde modelarea prin algoritmi genetici combinați cu euristica ar fi benefică.

Referitor la această ultimă afirmație considerăm că un algoritm genetic pentru o problemă de decizie multinivel, care să folosească o bază de cunoștințe în care sunt înmagazinate reguli euristice, ar rezolva problema de strategie a conducerii proceselor (și ne gândim, în special, la cele ecologice) în felul următor:

- anumite zone de gene din cromozom reprezintă, codificat, baza de cunoștințe, mulțimea regulilor euristice la nivelurile la care se dorește controlul procesului; în cadrul procedurilor de generare și de reproducere se vor selecta doar acei cromozomi care prezintă compatibilități între reguli, aceasta fiind sarcina specialistului utilizator; sau
- algoritmul genetic este el însuși parte din sistemul de simulare, bazat pe cunoștințe, în special, având rol important în construirea modelelor orientate pe obiecte.

* * *

Pentru realizarea aplicației prezentate țin să mulțumesc prof. Roger Phan Than Luu din IPRAI - Universitatea Aix-Marseille III fără ale cărei sugestii și sprijin material ea nu ar fi fost posibilă. De asemenea, amintesc cu plăcere discuțiile științifice purtate cu Dr. ing. Florin Stănciulescu din care au rezultat interesante posibilități de aplicare a algoritmilor genetici, în special pentru probleme ecologice.

Bibliografie

1. **DE JONG K.A.:** Genetic Algorithms - A 10 Years Perspective. In: Proceedings of the first Int. Conf. on GA Lavrance Erlbaum Associates, N.Y., SUA, 1985.
2. **GOLDBERG, D.E.:** Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing London, England, 1989.
3. **MICHILEWICZ, Z.:** Genetic Algorithms + Data Structure = Evolution Programs, Springer, Berlin, 1992.
4. **GOLDBERG, D.E., RUDNIK, M.:** Genetic Algorithms and the Variance of Fitness. In: Complex Systems, No.5, iunie, 1991.
5. **BRYNN HIBBERT, D.:** Genetic Algorithms in Chemistry. In: Chemometrics and Intelligent Laboratory Systems, vol.19, no.3, iulie, 1993.
6. **POTTS, J.C., GIDDENS, T.D., YADAV, S.B.:** The Development and Evaluation of an Improved Genetic Algorithm based on Migration and Artificial Selection. In: IEEE Transactions on SMC, ian. 1994.
7. **MATHIEU, D., PHAN THIAN LUU, R.:** Methodologie de la Recherche Experimentale, vol. 1-7 IPRAI -IUT Marseille 1983-1993.