

CZU: 004.9:575

DOI: 10.36120/2587-3636.v33i3.31-42

DEZVOLTAREA ALGORITMILOR GENETICI DIN PERSPECTIVE INTER/TRANSDISCIPLINARE

Liubomir CHIRIAC, dr. hab., prof. univ.

<https://orcid.org/0000-0002-5786-5828>

Natalia LUPAȘCO, dr., conf. univ.

<https://orcid.org/0000-0002-3854-2521>

Maria PAVEL, dr., conf. univ.

<https://orcid.org/0000-0003-4803-6398>

Universitatea de Stat din Tiraspol / Universitatea Pedagogică de Stat
„Ion Creangă” din Chișinău

Rezumat. În această lucrare sunt tratate fundamentele teoretice-practice ale Algoritmilor Genetici, care sunt construiți pe principiul „supraviețuiește cel mai bine adaptat”, enunțat de Charles Darwin. În lucrare sunt descrise caracteristicile de bază ale algoritmului genetic, evidențind avantajele și dezavantajele acestuia. Sunt examinate problemele care cad sub incidența algoritmului genetic. Algoritmul Genetic este examinat din perspectiva examinării problemelor în care găsirea soluției optime nu este simplă ori cel puțin ineficientă datorită caracteristicilor căutării probabilistice. Sunt arătate etapele în care Algoritmii Genetici codifică o posibilă soluție la o problemă specifică într-o unică structură de date numită „cromozom” și pregătesc terenul pentru a aplica operatorii genetici la aceste structuri astfel încât să mențină informațiile critice.

Cuvinte cheie: algoritm genetic, cromozomi, operatori genetici, selecție, crossover, mutație.

DEVELOPMENT OF GENETIC ALGORITHMS FROM INTER/TRANSDISCIPLINARY PERSPECTIVES

Abstract. The theoretical-practical foundations of Genetic Algorithms, which are built on the principle of "survival of the fittest", enunciated by Charles Darwin, are dealt with in this paper. The paper describes the basic characteristics of the genetic algorithm, highlighting its advantages and disadvantages. Genetic algorithm problems are examined. The Genetic Algorithm is examined from the perspective of examining problems in which finding the optimal solution is not simple or at least inefficient due to the characteristics of the probabilistic search. The steps are shown in which Genetic Algorithms encode a possible solution to a specific problem in a single data structure called a "chromosome" and set the stage for applying genetic operators to these structures in order to maintain critical information.

Keywords: genetic algorithm, chromosomes, genetic operators, selection, crossover, mutation.

1. Inteligența Artificială în Slujba Binelui

În data de 7 iulie, 2023, la Geneva s-a desfășurat summit-ul „AI for Good” - „Summitul Global de Inteligență Artificială în Slujba Binelui”, organizat de Uniunea Internațională a Telecomunicațiilor (UIT, engl. - International Telecommunication Union - ITU), o instituție din cadrul ONU, specializată în noile tehnologii. În cadrul acestui summit au participat circa 3000 de persoane, alături de umanoizii echipați cu inteligență artificială, pentru a discuta dacă ar putea roboții inteligenți să conducă lumea mai bine decât oamenii.

Menționăm faptul că summit-ul global „AI for Good” este principala platformă a Națiunilor Unite de promovare a Inteligenței Artificiale (IA) pentru soluționarea la nivel global a diferitor priorități, precum problemele climaterice, egalitatea de gen, sănătatea, incluziunea socială și infrastructura durabilă. Acest summit a fost organizat de UIT, în colaborare cu aproximativ 40 de agenții ONU și guvernul Elveției.

Pe parcursul desfășurării summit-ului, experții în domeniul AI au discutat despre necesitatea elaborării unor reguli clare, care ar putea să garanteze că inteligența artificială va fi utilizată doar în scopuri benefice pentru omenire, precum lupta împotriva foametei sau împotriva schimbărilor climatice.

Secretarul general al UIT, Doreen Bogdan-Martin a avertizat că „Realizările și cercetările din domeniul IA, în special cele care privesc inteligența artificială generativă, sunt în progres, iar ONU a cerut crearea unor reguli și limitări, pentru ca aceste tehnologii să aducă beneficii pentru omenire, fără să o pună în pericol. În absența unor astfel de reglementări, IA riscă să ne facă să trăim un adevărat coșmar”.

Totodată, roboții umanoizi consideră că oamenii ar trebui să dea dovadă de prudență în fața inteligenței artificiale și au recunoscut că nu stăpânesc deocamdată emoțiile umane.

Sophia, un robot dezvoltat de compania „Hanson Robotics”, consideră că capacitatea roboților umanoizi de a conduce lumea ar putea fi la un nivel de eficiență mai înalt față de capacitatea oamenilor, deoarece în activitatea roboților nu sunt implicate emoțiile, prejudecățile, ce ar influența adoptarea deciziilor și în plus aceștia pot analiza un volum mare de date cu viteză foarte înaltă.

Ideile scoase în evidență de umanoidul Sophia sunt, în linii mari, ideile și caracteristicile tuturor roboților umanoizi. Cum s-a ajuns la o așa etapă importantă în dezvoltarea IA?

O serie de probleme care cad sub incidența inteligenței artificiale sunt prea complexe pentru a fi soluționate cu tehnici directe și în contextul dat se aplică metode de căutare coresponzătoare, evident, împreună cu tehnicile directe de ghidare a căutării disponibile. Tehnicile respective pot fi descrise independent de orice domeniu de probleme.

Un rol important în dezvoltarea inteligenței artificiale, și în special în domeniul învățării automate (Machine Learning) îl au Algoritmii Genetici. Algoritmii genetici reprezintă tehnici adaptive de căutare euristică, ce se bazează pe principiile geneticii și ale selecției naturale și care pornesc de la idea că „cel mai bine adaptat – supraviețuiește”. Mecanismul este similar procesului de evoluție naturală, proces ce se bazează pe principiul că „doar speciile care se adaptează mai bine la mediu sunt capabile să supraviețuiască și să evolueze în timp, iar acelea care, în urma selecției naturale, nu reușesc să se adapteze vor dispărea”. Șansele că o specie să supraviețuiască în timp, peste generații, este în dependență directă de gradul de adaptare. Dacă ar fi să trecem în limbaj matematic aceasta ar însemna că cu cât este mai mare gradul de adaptare cu atât soluția se apropie de optim. Așa dar, un

algoritm genetic reprezintă un model matematic care imită modelul biologic evoluționist pentru a soluționa probleme de optimizare ori căutare. Cum susținea Charles Darwin: „*Speciile care supraviețuiesc nu sunt speciile cele mai puternice și nici cele mai inteligente, ci cele care se adaptează cel mai bine la schimbare*”. În contextul dat, mai jos, vom explica funcționarea unuia din cei mai eficienți algoritmi în inteligența artificială – algoritmi genetici.

2. Algoritmii genetici. Generalități

Algoritmii genetici (AG), sunt algoritmi de calcul evoluționist, inspirați din Teoria Evoluției lui Darwin. În anul 1960, Ingo Rechenberg (20 noiembrie 1934 - 25 septembrie 2021) a introdus ideea calculului evoluționist în lucrarea intitulată „*Evolution strategies*”. Ingo Rechenberg a fost un cercetător și profesor german în domeniul bionicii. Rechenberg a fost un pionier în domeniile calculului evolutiv și al evoluției artificiale. În anii 1960 și 1970 el a inventat mai multe metode de optimizare cunoscute sub numele de strategii de evoluție (din germană *Evolutionsstrategie*). Echipa de cercetători coordonată de el a aplicat cu succes algoritmi elaborați la probleme de optimizare, inclusiv la proiectarea aerodinamică a aripilor. Acestea au fost primele aplicații tehnice serioase ale evoluției artificiale, un compartiment important care ține de dezvoltarea bionicii și inteligenței artificiale.

În anul 1975, John Henry Holland (2 februarie 1929 – 9 august 2015) un om de știință american și profesor de psihologie și inginerie electrică și informatică la Universitatea din Michigan a introdus și a analizat un model matematic, care prin intermediul unor proceduri adaptive, pentru a găsi soluția unei probleme de optimizare se bazează pe un mecanism de selecție naturală și evoluție genetică numit algoritm genetic. El a fost un pionier în ceea ce a devenit cunoscut sub numele de algoritmi genetici.

Deseori, în cazul problemelor pentru care soluția optimă implică căutări printre toate combinațiile, permutările sau aranjările probabilistice, proces foarte complex și uneori ineficient, se utilizează algoritmul genetic. Acesta va implementa structuri de date specifice numite „cromozomi” pentru a codifica cu ajutorul operatorilor genetici soluția posibilă a unei probleme particulare, păstrând informațiile importante.

De obicei, pentru soluționarea unei probleme ce implică un algoritm genetic, se identifică așa numita „populație” ce se construiește aleatoriu pe baza „mulțimii inițiale de soluții posibile”. Fiecare individ sau „cromozom” (un șir de caractere, exprimat printr-o consecutivitate de biți), din „populația” examinată, reprezintă o soluție posibilă a problemei. Prin parcurgerea unor iterații consecutive, are loc evoluția „cromozomilor” ce se realizează la nivel de „generație”, validarea fiecăreia din care se face printr-o funcție de evaluare, numită *fitness*. Utilizând unul din cei trei operatori genetici principali: *selecția*, *crossover*-ul și *mutația*, se generează „cromozomi” noi, pentru viitoarea populație,

identificați din generația curentă, ca cei mai „eficienți”. Astfel, exact ca și în biologie, cei mai „puternici cromozomi”, cu o mai mare probabilitate, sunt *selectați* din generația dată, pentru a transmite prin copiere caracteristicile lor (valorile „funcției de evaluare”, conform cerințelor problemei) generației viitoare, astfel asigurând perpetuarea întregului proces. Cu ajutorul operatorului genetic de încrucișare (*crossover*) are loc combinarea informațiilor a doi indivizi („părinți”) din populația curentă pentru a genera unul sau câțiva descendenți. *Mutația* în schimb, permite schimbarea aleatorie a unei gene sau a unei secțiuni mici din „cromozom” pentru a asigura diversitatea la viitoarea populație.

Astfel, succesul algoritmilor genetici este asigurat de implementarea lor la soluționarea unui șir de probleme NP-complete, a căror soluții nu pot fi identificate prin metode iterative, ci prin obținerea soluției optime la nivel global [1-10].

Din cele expuse mai sus, pot fi scoase în evidență următoarele *avantaje* ale algoritmilor genetici:

- algoritmii genetici sunt mai puternici în raport cu algoritmii clasici de optimizare și mai eficienți comparativ cu metodele de căutare dirijată.
- algoritmii genetici identifică așa numita „populație”, construită din mulțimea soluțiilor posibile, pe când în mod tradițional căutarea se răsfrânge asupra unei soluții din mulțimea de căutare.
- algoritmii genetici interferează căutarea în mod aleatoriu cu căutarea în mod controlat și potrivit teoriei probabilităților pot fi numiți „algoritmi probabiliști”.
- algoritmii genetici echilibrează cercetarea „spațiului stărilor” și identificarea soluțiilor optime.
- algoritmii genetici se răsfrâng asupra codului „spațiului de căutare” și nu direct asupra acestuia.
- algoritmii genetici prevalează prin facilitatea utilizării asupra algoritmilor tradiționali ce solicită îndeplinirea unui șir de caracteristici importante ale funcției de evaluare (funcție continuă, derivabilă, convexă etc.). Probabilitatea de identificare a soluției optime de către algoritmii genetici este mare.
- algoritmii genetici nu necesită informații derivate (care ar putea să nu fie disponibile pentru multe probleme din lumea reală).
- algoritmii genetici sunt mai rapizi și mai eficienți în comparație cu metodele tradiționale. Optimizează atât funcțiile continue și discrete, cât și problemele multi-obiective.
- algoritmii genetici oferă întotdeauna o listă de soluții „bune” și nu doar o singură soluție.
- algoritmii genetici oferă întotdeauna un răspuns la problemă, care se îmbunătățește în timp și este util atunci când spațiul de căutare este mare și există un număr mare de parametri implicați.

Totuși algoritmi genetici presupun și unele **dezavantaje**, precum:

- Algoritmii genetici nu sunt potriviți pentru toate problemele, în special pentru problemele care sunt simple și pentru care sunt disponibile informații derivate.
- Valorile funcției de fitness sunt calculate la fiecare populație, în mod repetat, ceea ce poate fi costisitor din punct de vedere computațional pentru unele probleme.
- Fiind o metodă stocastică, nu există garanții asupra optimității sau calității soluției.
- Algoritmii genetici dacă nu este implementat corespunzător poate să nu convergă către soluția optimă.

3. Terminologia utilizată

În algoritmi genetici indivizii dintr-o populație sunt reprezentați de cromozomi cu seturi codificate de parametri ai sarcinii, de ex. soluții, care altfel sunt numite puncte în spațiu de căutare (puncte de căutare). În unele lucrări, indivizi se numesc organisme. În acest sens vom clarifica următoarele noțiuni biologice, împrumutate de informaticieni din perspectiva algoritmilor genetici:

- ✓ **Cromozomii** (lanțuri sau secvențe de cod) sunt secvențe ordonate de gene.
- ✓ O **genă** (numită și proprietate, semn sau detector) este elementul atomic al genotipului, în special al cromozomilor.
- ✓ **Genotipul** sau structura este setul de cromozomi ai unui individ dat. În consecință, indivizii unei populații pot fi genotipuri sau cromozomi unici (într-un caz destul de comun, când genotipul este format dintr-un singur cromozom).
- ✓ **Fenotipul** este un set de valori care corespund unui anumit genotip. Ori, structura decodificată sau set de parametri ai sarcinii {soluție, punct de spațiu de căutare}.
- ✓ O **alelă** este valoarea unei anumite gene, definită și ca valoarea proprietății sau varianta proprietății.
- ✓ **Locusul** sau poziția indică locația unei anumite gene în cromozom (lanț). Mulțimea de poziții ale genelor reprezintă locusuri.
- ✓ **Genom** – totalitatea materialului genetic al unui organism sau al unei specii, care determină dezvoltarea, funcționarea și transmiterea trăsăturilor ereditare ale organismului de la o generație la alta.
- ✓ **Individ** – o entitate unică care are un set specific de cromozomi moșteniți de la părinți. În cadrul algoritmilor genetici un individ reprezintă o posibilă soluție sau o combinație a parametrilor care poate fi optimizată în timp prin procesele de selecție și recombinare și care este evaluată în cadrul unei probleme specifice.
- ✓ **Populație** – un grup de indivizi ce împărtășesc un set comun de caracteristici genetice ce ocupă un anumit tip de mediu. Variația genetică în cadrul populațiilor este importantă pentru adaptarea la schimbările de mediu.

- ✓ **Mapping** – funcție esențială de evaluare ce are rolul de a atribui o valoare numerică fiecărui individ din populația dată, reflectând calitatea sau adecvarea soluției respective în cadrul problemei de optimizare.

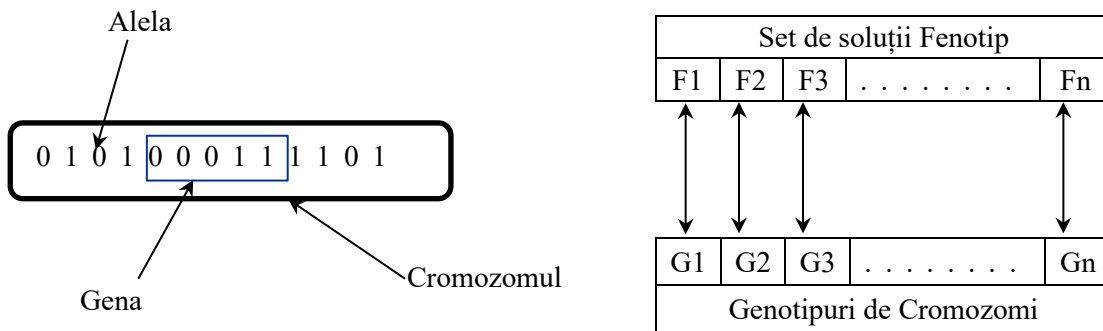


Figura 1. Schema genelor și alelor

Un concept foarte important în algoritmi genetici este funcția care măsoară gradul de adaptivitate cunoscută și sub denumirea de **funcție de fitness** (*fitness function*).

Funcția fitness reprezintă o măsură a adaptabilității unui individ dat în cadrul fiecărei generații. Această caracteristică permite evaluarea gradului de adaptare al indivizilor din populație și se alege dintre cei mai adaptați, adică pe cei cu cele mai mari valori ale funcției de fitness, în conformitate cu evoluția principiul supraviețuirii celui mai „puternic” (cel mai bine adaptat). Funcția de fitness și-a primit și numele direct din genetică. Are un impact puternic asupra funcționării algoritmilor genetici și trebuie să aibă o precizie și definiție corectă. În problemele de optimizare, funcția de fitness, de regulă, este optimizată (maximizată ori minimizată) și se numește funcție obiectiv. În probleme de minimizare funcția obiectiv este transformată și problema se reduce la maximizare. În teoria controlului, funcția de fitness poate lua forma unei funcții de eroare, iar în teoria jocurilor - funcția de cost. La fiecare iterație a algoritmului genetic gradul de adaptivitate (fitnessul) a fiecărui individ dintr-o anumită populație este estimat cu ajutorul funcției de fitness și, pe această bază, se generează următoarea generație (populație de indivizi) care va constitui posibila mulțime de soluții în cazul problemei examinate.

4. Aplicații ale algoritmilor genetici

Algoritmii genetici sunt adesea utilizați în informatică, inteligență artificială, etc., pentru a găsi soluții complexe și neevidente la probleme de optimizare și de căutare algoritmică. Mai jos vom puncta unele aplicații ale algoritmilor genetici.

- A) Probleme de optimizare.** Procesul de optimizare se referă la găsirea celei mai bune soluții sau a celei mai bune configurații dintr-un set posibil de opțiuni, în scopul maximizării sau minimizării unei funcții obiectiv sau criterii date. Optimizarea este implementată într-o gamă largă de domenii, cum ar fi matematică, inginerie, științe naturale, economie, informatică și multe altele, pentru a rezolva probleme complexe

sau pentru a îmbunătăți performanța unui sistem sau proces. Pentru formularea problemei de optimizare se parcurg mai multe *etape*:

- se selectează una sau mai multe variabile de optimizare;
- se alege funcția obiectiv;
- se identifică setul de constrângeri.

Sunt utilizate pe scară largă și alte probleme de optimizare, precum: programarea lucrărilor, optimizarea calității sunetului, problema comis-voiajorului etc.

B) Învățare automata. Cercetătorul John McCarthy definește IA, încă în anul 1956, în felul următor: „IA implică calculatoare care pot îndeplini sarcini ce sunt caracteristice inteligenței umane”. Domeniul care ține de Machine Learning este o modalitate de a realiza Inteligența Artificială. O definiție foarte potrivită a învățării automate (ML), formulată de Arthur Samuel, datează încă din anul 1959: Machine Learning este „domeniul de studiu care oferă calculatoarelor capacitatea de a învăța fără a fi programate în mod explicit”. Problemele tipice rezolvate de Machine Learning se referă la: *regresie, clasificare, segmentare, analiza rețelei*. În ultima perioadă Inteligența artificială, pentru a soluționa unele probleme de importanță majoră, are nevoie de instrumente și tehnici de intervenție care țin de Big Data și Machine Learning. Și în contextual dat Algoritmii Genetici joacă un rol extrem de important. Algoritmii genetici sunt folosiți pentru a rezolva problemele legate de clasificare, predicție, crearea de reguli pentru învățare și clasificare.

C) Modelul sistemului imunitar. Algoritmii genetici sunt utilizați și pentru modelarea diferitelor aspecte ale sistemului imunitar pentru genele individuale și familiile multi-genice în timpul evoluției.

D) Alte exemple de probleme rezolvate cu ajutorul algoritmilor genetici se referă la: identificarea soluției optime sau aproximative pentru problema proiectării oglinzilor folosite în sistemele de concentrare solară, optimizarea formei și performanței antenelor în comunicații wireless, proiectarea de dispozitive complexe în domeniul Computer-Aided Design, dezvoltarea mișcărilor realiste ale personajelor sau obiectelor animate în mediul virtual în scopul optimizării parametrilor mișcărilor astfel încât acestea să arate cât mai natural și să îndeplinească obiectivele specifice ale animației, proiectarea optimă a corpurilor aerodinamice în câmpuri de curgere complexe. De asemenea, algoritmii genetici se aplică în procesul de prelucrare și restaurarea a imaginilor, dar și un rol important în ultima perioadă îl joacă în procesul de prognoză a piețelor financiare.

5. Codificarea cromozomilor și problema de optimizare

Căutarea soluțiilor optime sau aproximative pentru o problemă specifică prin intermediul unui algoritm genetic implică un șir de componente ale acestuia, printre care, principale sunt:

- 1) *codul problemei*
- 2) *funcția fitness (de potrivire, sau de evaluare).*

Aceste două componente diferă de la o problemă la alta, în dependență de cerințele acesteia. Așa cum s-a arătat mai sus, cromozomii conțin structuri de date (informații) ce permit identificarea soluției problemei și elaborarea codului acestei soluții.

Cromozomii pot lua diverse forme, sau coduri, printre care:

- A) **Codificarea binară:** indivizii sau cromozomii sunt reprezentați sub formă de șiruri de biți (0 și 1) care reprezintă binar soluția problemei.

1	0	0	1	1	1	0	1
---	---	---	---	---	---	---	---

- B) **Codificarea reală** (sau prin valori): indivizii sunt reprezentați prin valori numerice reale, stocate într-un vector.

0.2	0.1	0.4	0.5	0.7	0.8	0.6	0.9
-----	-----	-----	-----	-----	-----	-----	-----

De exemplu, în situația când este necesar să codificăm Nord, Sud, Est ori Vest, vom utiliza doar 4 valori întregi {1, 2, 3, 4}. Astfel:

1	3	4	2	1	4	3	2
---	---	---	---	---	---	---	---

- C) **Codificare cromozomilor prin intermediul permutărilor.** În multe probleme, soluția ține de o reprezentare de ordine a elementelor. În astfel de cazuri, reprezentarea prin intermediul permutării este cea mai potrivită. Un exemplu clasic al acestei reprezentări ține de problema comis-voiajorului. Astfel, comis-voiajorul trebuie să facă un tur al tuturor orașelor, vizitând fiecare oraș exact o dată și să revină în orașul de plecare. Distanța totală a turului trebuie redusă la minimum. Soluția în acest caz reprezintă o ordonare sau o permutare a tuturor orașelor numerotate și, prin urmare, utilizarea reprezentărilor care reprezintă permutări are sens pentru această problemă.

1	3	4	6	5	2	7	1
---	---	---	---	---	---	---	---

6. Formularea matematică a problemei

Când este necesar să rezolvăm o problemă, aplicând algoritmi genetici, examinăm anumite soluții „mai bune” obținute la o anumită populație, care țin de o generație mai avansată, față de soluțiile identificate anterior. Setul soluțiilor posibile (fezabile) pentru o anumită problemă constituie „spațiul de căutare” sau „spațiul stărilor”. Fiecare individ din populația algoritmului genetic reprezintă o stare sau o soluție candidată din acest spațiu al stărilor. În esență, spațiul stărilor cuprinde toate configurațiile și combinațiile posibile ale soluțiilor care pot fi evaluate folosind funcția de fitness. În cazul când simpla căutare în spațiul imens al stărilor unei probleme specifice presupune un algoritm foarte complicat, costisitor, atunci se recurge la algoritmi genetici, care și-au demonstrat eficiența în acest sens.

Sub aspect matematic, o astfel de problemă se formulează astfel:

Fie funcția $g(x_1, x_2, \dots, x_n): S \rightarrow R$, unde $S = T_1 \times T_2 \times \dots \times T_n$ reprezintă spațiul stărilor. Să se identifice soluția $s^* = (x_1^*, x_2^*, \dots, x_n^*) \in S$ astfel încât pentru aceasta, funcția g va atinge minimum (maximum). Mulțimile $T_i \leq R$, unde $i = 1, \dots, n$. Deci $S \leq R^n$.

Deseori, este suficient să se identifice soluția optimizată s_{optim} sau cea aproximativă, nu neapărat soluția exactă s^* , astfel încât: $|s^* - s_{optim}| \leq \varepsilon$ ori $|g(s^*) - g(s_{optim})| \leq \varepsilon$, unde ε reprezintă exactitatea solicitată pentru soluție.

Exemplul 1. Pentru a determina maximum funcției $f(x) = x^2 - 1$, unde x ia valori întregi de la 0 la 15, este necesar de clarificat esența matematică a mai multor concepte, din perspectiva algoritmului genetic, examinate anterior.

Astfel, mulțimea $\{0, 1, \dots, 15\}$ constituie spațiul de căutare și în același timp - un set de soluții potențiale la problemă. Fiecare dintre cele 16 valori ce aparțin acestei mulțimi reprezintă punct al spațiului de căutare, decizie, valoare a parametrului, sau fenotip. De remarcat că se numește soluția care optimizează funcția cea mai bună sau soluția optimă. Codificăm valorile x de la 0 la 15 după cum urmează:

0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
8 1000	9 1001	10 1010	11 1011	12 1100	13 1101	14 1110	15 1111

Aceasta este o metodă de codificare binară bine cunoscută, asociată cu scrierea cifrelor zecimale în sistemul binar. Secvențele de cod prezentate mai sus sunt numite și **lanțuri** sau **cromozomi**. În exemplul dat, aceștia acționează și ca **genotipuri**. Fiecare dintre cromozomi este format din 4 gene (altfel putem spune că secvențele binare constau din 4 biți). Valoarea genei într-o poziție specifică se numește **alela** care ia valoarea 0 sau 1. Populația este formată din indivizi selectați dintre acești 16 cromozomi.

Un exemplu de populație egală cu 6 poate fi, de exemplu, o multitudine de cromozomi $\{0101, 0111, 1001, 1100, 1110, 1111\}$, care este o formă codificată a următoarelor fenotipuri: $\{5, 7, 9, 12, 14, 15\}$. Funcția de fitness în acest exemplu este dată de expresia $f(x) = x^2 - 1$. Fitness-ul cromozomilor dintr-o populație este determinată de valoarea acestei funcții. Valorile respectivilor cromozomi corespund anumitor genotipuri. Apropo, maximum funcției $f(x) = x^2 - 1$, unde x ia valori întregi de la 0 la 15 este 224.

7. Algoritm Genetic

În literatura de specialitate, Algoritm Genetic presupune realizarea unor pași (propuși inițial de Holland), care ar trebui să finalizeze cu o soluție optimă a problemei examinate.

Pasul 1. Crearea/generarea populației inițiale;

Pasul 2. Evaluarea valorii funcției de fitness (funcției de adecvare) a fiecărui individ (mecanism utilizat pentru a măsura și evalua starea unui cromozom);

Pasul 3. Selecția – luând în considerare caracteristicile fiecărui individ, în cadrul acestei etape, unii indivizi se pot reproduce mai des decât alții;

Pasul 4. Crossover-ul (Încrucișarea);

Pasul 5. Mutația;

Pasul 6. Substituirea populației vechi de cromozomi cu noua populație;

Pasul 7. Găsirea celei mai bune soluții (dar, în cazul în care criteriile de optimizare nu sunt îndeplinite, atunci metoda impune întoarcerea la pasul 2 și selectarea în final a celui mai bun individ ca soluție finală).

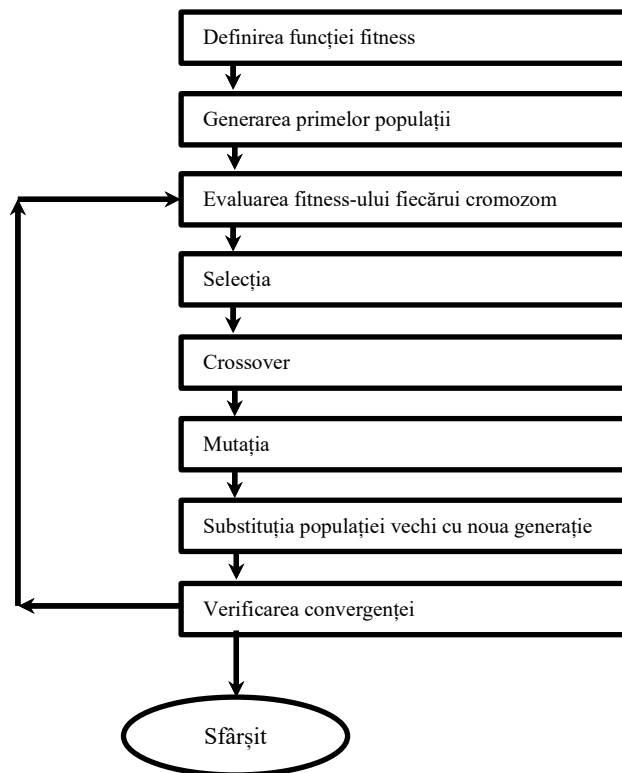


Figura 2. Schema logică a Algoritmului Genetic

Se inițiază astfel un fond genetic aleatoriu prin crearea unui set de cromozomi conform unui șablon predefinit, unde valorile tuturor genelor sunt selectate în mod aleatoriu pentru fiecare cromozom. Acești cromozomi inițiali corespund indivizilor din populația inițială. De regulă, la diferite generații, constant rămâne numărul de indivizi (și implicit de cromozomi) din populație, fapt ce nu este valabil mereu. Algoritmul Genetic pornește cu un set de soluții admisibile, numit „populație” (creată, după cum am menționat, în mod arbitrar), fiecare dintre ele reprezentând o potențială soluție a problemei, numită „cromozom”.

După ce populația este stabilită, aceasta evoluează spre soluții mai bune, prin diferite procese genetice (*selecția, încrucișarea, mutația*) care conduc la o mai bună valoare a funcției de adecvare, utilizată pentru a evalua starea fiecărui cromozom.

Algoritmii genetici generează o nouă populație, formată din indivizi cu caracteristici mai bune și mai adecvați mediului decât cei ai populației anterioare. Schema logică care ține de implementarea Algoritmului Genetic este prezentată în figura 2.

8. Mecanisme de generare a populației

Algoritmii genetici operează pe o populație de indivizi, în care fiecare individ poate să reprezinte o posibilă soluție la o problemă examinată.

Populația se evaluează din perspectiva măsurii proprii „aptitudinii” de a se adapta „condițiilor impuse”. Astfel, în procesul de evaluare se clarifică cât de eficient este un organism din populația respectivă în competiția pentru resursele existente. Indivizii ineficienți din acest punct de vedere pier, iar cei eficienți supraviețuiesc și se „reproduc”. Această afirmație, reformulată în termeni matematici, este echivalentă cu următoarele: cât de „bună”, „optimă” este soluția identificată problemei care îi corespunde.

Prin intermediul „încrucișării” cu alți indivizi din populația examinată, indivizii eficienți sunt capabili să „reproducă” descendenți, care pregătesc terenul pentru apariția unei noi generații de indivizi care, totodată, păstrează ori combină unele din caracteristicile moștenite de la părinți.

În cursul evoluției indivizii „slabi”, cei mai puțin apti, vor avea o probabilitate mai mică să se poată „reproduce”. În așa mod, în procesul de trecere de la o generație la alta, indivizii respectivi vor dispărea treptat din populație, iar proprietățile pe care le posedau vor dispărea treptat din caracteristicile populației. În anumite situații apar mutații sau modificări spontane ale genelor.

Încrucișarea celor mai eficienți și adaptați indivizi duce la faptul că cele mai promițătoare caracteristici ale spațiului de căutare sunt moștenite. În așa fel, din generație în generație, caracteristicile „bune”, aplicând concepte și noțiuni din biologie, informatică și matematică, sunt distribuite în întreaga populație.

De la o generație la alta, populația de indivizi va converge treptat către o soluție optimă a problemei. Avantajul unui Algoritm Genetic este că găsește soluții optime aproximative într-un timp relativ scurt.

Articol realizat în cadrul proiectului de cercetări științifice „Metodologia implementării TIC în procesul de studiere a științelor reale în sistemul de educație din Republica Moldova din perspectiva inter/transdisciplinarității (concept STEAM)”, inclus în „Program de stat” (2020-2023), Prioritatea IV: Provocări societale, cifrul 20.80009.0807.20, cu suportul financiar oferit de Agenția Națională pentru Dezvoltare și Cercetare

Bibliografie

1. BEASLEY, D.; BULL, D. R.; MARTIN, R. R. An Overview of Genetic Algorithms: Part 1. Fundamentals. In: *University Computing*, volume 15(2), pp. 58- 69, 1993.
2. DUMITRESCU, D. *Algoritmi genetici și strategii evolutive - Aplicații în inteligența artificială și în domenii conexe*. Cluj-Napoca: Editura Albastra, 2000.
3. GAREY, M.R.; JOHNSON, D.S. *Computers and Intractability: A Guide to NP-completeness*. New York: W.H. Freeman and Company, 1978.
4. GOLDBERG, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison -Wesley, Reading, MA,1989.
5. HOLLAND, J.H. *Adaptation in Natural and Artificial Systems*. Ann. Arbor: University of Michigan Press, 1975. 183 p.
6. KOZA, J.R. *Genetic Programming*. Cambridge: MIT Press, MA, 1992.
7. MITCHELL, M. *An Introduction to Genetic Algorithms*. Cambridge: MIT Press, 1996.
8. MITCHELL, M. Genetic Algorithms: An Overview. In: *Complexity*, 1995. Nr. 1(1), pp. 31-39.
9. OLTEAN, M. *Proiectarea și implementarea algoritmilor*. Cluj-Napoca: Computer Libris Agora, 2000.
10. RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. Second Edition. Prentice Hall, 2003.