CZU: 004.9:575 DOI: 10.46727/c.steam-2023.p285-291

STUDIEREA ALGORITMULUI GENETIC DIN PERSPECTIVA STEAM

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 Pedagogică de Stat „Ion Creangă” din Chișinău

Abstract. În acest articol autorii abordează unele aspecte metodico-didactice care țin de calculul evolutiv și studierea Algoritmului Genetic din perspectiva STEAM. Este examinat la modul practic procesul de formare a populațiilor de cromozomi, cât și procedeele de aplicare ale operatorilor genetici: selecție, crossover-ul și mutația. La fel, se cercetează procesul de optimizare a funcției fitness și problema amplasării vârfurilor unui graf pe o riglă liniară, mizând pe îmbunătățirea soluției odată cu trecerea calitativă de la o populație la alta. Sunt evidențiate tehnicile și procedeele care pot fi aplicate eficient atât de biologi cât și de matematicieni și informaticieni la rezolvarea problemelor respective.
Cuvinte cheie: calcul evolutiv, algoritm genetic, concept STEAM.

Abstract. In this article the authors address some methodological-didactic aspects related to the evolutionary calculation and the study of the Genetic Algorithm from the STEAM perspective. The process of forming chromosome populations is examined in a practical way, as well as the application procedures of genetic operators: selection, crossover and mutation. In the same way, the optimization process of the fitness function and the problem of locating the vertices of a graph on a linear ruler are researched, counting on the improvement of the solution with the qualitative transition from one population to another. The techniques and procedures that can be effectively applied both by biologists and mathematicians and computer scientists to solve the respective problems are highlighted.
Keywords: evolutionary computation, genetic algorithm, STEAM concept.

1. Dezvoltarea calculului evolutiv

Abordarea privind aplicarea principiilor evolutive (calculul evolutiv) în soluționarea automată a problemelor (Problem Solving) datează cu mult timp mai înainte comparativ cu apariția și dezvoltarea calculatoarelor moderne. Alan Turing, încă în anul 1948, lansează o abordarea nouă aplicată la soluționarea problemelor numită abordare evolutivă ori genetică. Ulterior, în anii 1960, informaticienii Lawrence J. Fogel, A. J. Owens și M. J. Walsh introduceau și dezvoltau conceptul de programare evolutivă. În aceeași perioadă Holland se concertează pe algoritmii genetici, iar Rechenberg şi Schwefel lansează și dezvoltă strategiile evolutive ca modalităţi alternative de soluționare automată a problemelor. Ceva, mai târziu, în anii 1990, J. R. Koza dezvoltă programarea genetică, o nouă tehnică de căutare a soluţiilor.

Așa dar, calculul evolutiv este un domeniu al informaticii moderne, cu accent puternic în matematică, inspirat din procesul evoluţiei naturale, iar conceptul de bază care

ține de calculul evolutiv este interconexiunea evoluţie naturală – tehnica de rezolvare a problemelor de tip experiment-eroare.

În contextul celor menționate, în prezent, un domeniu important de cercetare al informaticii îl reprezintă calculul evolutiv. După cum se știe, domeniul respectiv își ”trage seva”, adică își are rădăcinile în procesul evoluţiei naturale. Algoritmii care apar și se dezvoltă în acest domeniu se numesc algoritmi evolutivi şi ei includ subdomenii importante de mare perspectivă precum [1-10]:

- *Programarea evolutivă;*
- *Strategii evolutive;*
- *Programare genetică;*
- *Algoritmi genetici.*

Algoritmii evolutivi presupun existența într-un mediu examinat, a unor indivizii constituiţi într-o populaţie, care intră în competiţie pentru a supravieţui, a se adapta şi a se reproduce. Abilitatea indivizilor de a se adapta în mediul în care activează este corelată cu şansele lor de supravieţuire şi reproducere şi determină evoluţia populaţiei în timp. În contextul modalităţii de rezolvare a problemelor populaţia este modelată de operatorii genetici. Indivizii cel mai bine adaptați, după mai multe iterații a generațiilor, vor determina soluțiile problemei examinate. Mai jos descriem succint fiecare din domeniile menționate mai sus.

Programarea evolutivă. Programarea evolutivă este una dintre cele patru paradigme majore de algoritm evolutiv. Este similar cu programarea genetică, dar structura programului de optimizat este fixă, în timp ce parametrii numerici ai acestuia sunt lăsați să evolueze. Programarea evolutivă a fost folosit pentru prima dată de Lawrence J. Fogel în SUA în 1960 pentru a utiliza evoluția simulată ca proces de învățare care urmărește generarea de inteligență artificială. Fogel a folosit mașini cu stări finite ca predictori și le-a dezvoltat. În prezent, a devenit mai greu de distins de strategiile evolutive. Principalul operator al programării evolutive este mutația.

Strategii evolutive. Strategiile evolutive sunt algoritmi evolutivi care datează din anii 1960 și care sunt cel mai frecvent aplicați problemelor de optimizare a cutiei negre în spațiile de căutare continuă. Inspirat de evoluția biologică, formularea lor originală se bazează pe aplicarea mutației, recombinării și selecției în populații care reprezintă soluţii candidate. Din punct de vedere algoritmic, strategiile evolutive sunt metode de optimizare care eșantionează noi soluții candidate stocastic, cel mai frecvent dintr-o probabilitate normală multivariată distribuție. Cele mai proeminente două principii de proiectare ale lor sunt *imparțialitatea și controlul adaptiv* al parametrilor a distribuției eșantionului.

Programarea genetică. Programarea genetică este o metodă independentă de domeniu care generează genetic o populație de programe pe calculator pentru a rezolva o

problemă. Mai exact, programarea genetică transformă în mod iterativ o populație de programe pe calculator într-o nouă generație de programe prin aplicarea operațiilor genetice care apar în mod natural. În inteligența artificială, programarea genetică (GP) se consideră o tehnică evolutivă a programelor, pornind de la o populație de programe mai puțin adecvate (de obicei aleatoare), potrivite pentru o anumită sarcină prin aplicarea unor operațiuni similare proceselor genetice naturale populației de programe. Operațiunile aplicate sunt: selectarea celor mai potrivite programe pentru reproducere (încrucișare), replicare și/sau mutație în dependență de funcția de fitness predefinită. Operația de încrucișare implică schimbarea părților specificate ale perechilor selectate (părinți) pentru a produce descendenți noi și diferiți care devin parte din noua generație de programe. Unele programe care nu sunt selectate pentru reproducere sunt copiate din generația curentă în generația nouă. Mutația implică înlocuirea unei părți aleatoare a unui program cu o altă parte aleatorie a unui alt program. Apoi selecția și alte operațiuni sunt aplicate recursiv noii generații de programe. Elementele populației sunt de cele mai multe ori structuri arborescente. În mod obișnuit, membrii fiecărei noi generații sunt mai apți decât membrii generației precedente, iar programul cel mai bun din generație este adesea mai bun decât programele cele mai bune din generațiile anterioare. Încetarea evoluției are loc de obicei atunci când un program individual atinge un nivel predefinit de competență sau de fitness.

Algoritm genetic. Algoritmul genetic este o metodă care se referă la rezolvarea problemelor de optimizare și care se bazează pe selecția naturală, încrucișări și mutații, procesul care repetă evoluția biologică. Algoritmul genetic modifică în mod repetat o populație de soluții individuale. Referitor la algoritmul genetic, pe parcursul acestei lucrări, vom examina în profunzime mai multe aspecte care ține de dezvoltarea și aplicarea algoritmului.

În genere, ținem să menționăm că un algoritm evolutiv este considerat o componentă a calculului evolutiv în inteligența artificială. Un algoritm evolutiv funcționează prin intermediul procesului de selecție în care sunt excluși membrii cei mai puțin adaptați (cei mai puțin în formă) din grupul de populație examinat, în timp ce membrii apți supraviețuiesc și sunt luați în considerație până când sunt determinate soluții mai bune. Cu alte cuvinte, algoritmii evolutivi sunt aplicații computerizate care imită procesele biologice pentru a rezolva probleme complexe. În timp, membrii de succes evoluează pentru a prezenta soluția optimizată a problemei. Algoritmii evolutivi utilizează concepte în biologie, cum ar fi selecția, reproducerea și mutația.

2. Abordări didactice în studierea algoritmului genetic

Un algoritm evolutiv are o populație de soluții candidate, spre deosebire de metodele clasice, care încearcă să mențină o singură soluție cât mai bună. Există numeroase beneficii asociate cu algoritmi evolutivi. Unul dintre cele mai mari avantaje

vine în câștigurile de flexibilitate, deoarece majoritatea conceptelor de algoritm evolutiv sunt adaptabile chiar și la probleme complexe. Există câteva dezavantaje asociate algoritmilor evolutivi. Pentru una, soluția oferită de un algoritm evolutiv este mai bună decât în comparație cu alte soluții cunoscute. Ca atare, algoritmul nu poate dovedi că nici o soluție este total optimă, doar că este optimă în comparație cu celelalte rezultate.

Faptul că matematica și informatica se aplică pe larg în diverse științe, inclusiv în biologie, este un lucru cunoscut și apreciat. Reciprocitatea în acest sens nu întotdeauna are loc. De exemplu, în știința modernă nu se întâlnesc așa de multe situații când mecanismele, conceptele și noțiunile de bază din biologie se utilizează pe larg și eficient în matematică și informatică.

Algoritmul genetic este în acest sens un exemplu elocvent și convingător. Algoritmii genetici reprezintă tehnici adaptive de căutare euristică, care se implementează mizând pe principiile selecțiile naturale și geneticii. Mecanismele de realizare ale Algoritmului Genetic este asemănător evoluției naturale și mizează pe principiul enunțat de Charles Darwin, ”supraviețuiește nu cel mai puternic ori inteligent, dar cel mai bine adaptat”. Astfel, Algoritmul Genetic reprezintă un model informatic-matematic care imită modelul biologic evoluționist pentru a soluționa probleme de căutare și optimizare, care poate fi interpretat și ca un model STEAM. Algoritmul Genetic este determinat dintr-un set de elemente care reprezintă o populație, formată din cromozomi (șiruri binare) și un set de operatori genetici (selecția, încrucișare și mutația) care influențează structura populației.

Algoritmul genetic este des utilizat pentru probleme în care găsirea soluției optime nu este simplă ori cel puțin ineficientă datorită caracteristicilor căutării probabilistice. Algoritmii genetici codifică o posibilă soluție la o problemă specifică într-o unică structură de date numită „cromozom” și aplică operatorii genetici la aceste structuri astfel încât să mențină informațiile critice. Aplicarea algoritmilor genetici pornesc de la o ”mulțime inițială de soluții posibile” ale problemei examinate (aleasă de obicei aleator) numită în literatura de specialitate „populație”.

Fiecare individ din ”populația examinată” reprezintă o posibilă soluție a problemei și este numit ”cromozom”, care este un șir de simboluri, de obicei exprimat ca un șir de biți. Cromozomii examinați evoluează pe durata iterațiilor succesive efectuate numite simbolic – generații. În cadrul fiecărei generații, cromozomii respectivi sunt evaluați folosind unele măsuri de potrivire, numite – fitness.

Pentru generarea următoarei populații (generații) sunt selectați cei mai ”eficienți”, cei mai ”buni” cromozomi din generația curentă. Cromozomii noi sunt formați utilizând unul din cei trei (ori chiar toți trei) operatori genetici centrali: ***selecția, crossover și mutația***. Algoritmii Genetici au fost aplicați cu succes într-o varietate de aplicații NP-complete care necesită optimizarea globală a soluției și, în acest sens, nu există o metodă iterativă de rezolvare.

În Algoritmii Genetici indivizii dintr-o populație sunt reprezentați de cromozomi cu seturi codificate în ei, parametrii sarcinii, de exemplu soluții care altfel sunt numite puncte în spațiu de căutare ori puncte de căutare. În unele lucrări, indivizi se numesc organisme. În acest sens vom clarifica sensul următoarelor noțiuni biologice din perspectiva informaticii.

Conceptul de evoluție al lui Darwin este adaptat la funcționarea algoritmului genetic pentru a găsi soluții la o problemă exprimată prin intermediul funcției de fitness (funcție obiectivă ori funcție de adaptare). ***Funcția fitness*** *(fitness function)* 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).

Algoritmii genetici au fost aplicați cu succes într-o varietate de aplicații NP-complete care necesită optimizarea globală a soluției și , în acest sens nu există, o metodă iterativă de rezolvare [1-10]. Care sunt principalele avantaje și dezavantaje ale algoritmilor genetici?

Avantajele Algoritmilor Genetici:

- Algoritmii genetici sunt simplu de utilizat și nu cer proprietăți importante ale funcției obiectiv, precum continuitate, derivabilitate, convexitate, ca in cazul algoritmilor clasici. AG pot găsi soluții optime sau aproape optime cu o mare probabilitate.
- 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.
- Oferă întotdeauna o listă de soluții „bune" și nu doar o singură soluție.
- 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.
- Algoritmii genetici sunt mai robuști decât algoritmii clasici de optimizare și mai eficienți comparativ cu metodele de căutare dirijată.
- Metodele clasice de căutare de regulă acționează la un moment dat asupra unui singur punct din spațiul de căutare. În schimb algoritmii genetici mențin o mulțime de soluții posibile care se exprimă prin populație.
- Algoritmii genetici sunt algoritmi probabiliști ce îmbina căutarea aliatoare cu cea dirijată.

- Algoritmii genetici realizează un echilibru între explorarea spațiului stărilor și găsirea celor mai bune soluții.
- Algoritmii genetici nu acționează direct asupra spațiului de căutare ci asupra unei codificări a lui.

Dezavantajele aplicării Algoritmilor Genetici:

- Valorile funcției de fitness este calculată 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 calității soluției.
- Algoritmul genetic dacă nu este implementat corespunzător poate să nu conveargă către soluția optimă.
- Algoritmii genetici nu sunt potriviți pentru toate problemele, în special pentru problemele care sunt simple și pentru care sunt disponibile informații derivate.

3. Soluționarea problemelor practice cu aplicarea Algoritmului Genetic

Autorii, în contextul dat au examinat 2 tipuri de probleme.

1) ***Problema amplasării optime a vârfurilor unui graf neorientat pe o riglă liniară***. Problema amplasării optime a vârfurilor pe o riglă este o problemă clasică care necesită cunoștințe atât din matematică, informatică cât și, evident, care țin și de Algoritmul Genetic.

Problemă 1. Fie este dat graful G, unde $n = |G|$ - este numărul de vârfuri ale grafului G. Se cere de găsit cea mai bună amplasare a vârfurilor grafului G pe o riglă liniară după realizarea algoritmului genetic în k - cicluri (k - generații) și $k < n$.

În acest sens a fost propus un Algoritm care soluționează problema prin intermediul Algoritmului Genetic.

2) ***Problema optimizării***. Când este necesar să rezolvăm o problemă de optimizare, aplicând algoritmii genetici, examinăm anumite soluții obținute la o anumită populație, care țin de o generație mai avansată, și care de obicei, sunt mai bune decât alte soluții obținute anterior. Spațiul tuturor soluțiilor fezabile se numește spațiul de căutare sau spațiul stărilor. Problemele abordate folosind algoritmi genetici sunt de obicei probleme pentru care căutarea în spațiul soluțiilor este o problemă complicată sau chiar, după cum am menționat mai sus, NP-completă.

Algoritmii genetici sunt eficienți în special la problemele unde spațiul de căutare este imens și găsirea determinstă a unei soluții este foarte costisitoare sau chiar imposibilă. Formularea problemei optimizării este următoarea:

Se dă funcția $g(x_1, x_2, \ldots, x_n)$: $S \rightarrow R$, unde $S = T_1 \times T_2 \times \ldots \times T_n$ este spațiul de căutare. Se cere să se găsească o soluție $s^* = (x_1^*, x_2^*, \ldots, x_n^*) \in S$ pentru care se atinge minimul (maximul) funcției g. Mulțimile $T_i \leq R$, unde $i = 1, \ldots, n$. Deci $S \leq R^n$.

De cele mai multe ori, nu este necesară soluția exactă s^*, ci o soluție optimă s_{optim}, o aproximare a soluției exacte, cu îndeplinirea condiției: $|s^* - s_{optim}| \leq \varepsilon$ ori $|g(s^*) - g(s_{optim})| \leq \varepsilon$, unde ε este gradul de precizie cerut pentru soluție.

Problema 2. Aplicând Algoritmul Genetic se cere de găsit maximul funcției

$$F(x) = -x2 + 16x - 15,$$

pe intervalul de numere întregi [1,…,15].

Autorii soluționează problema respectivă prin generare de noi generații de populații (soluții) care conduc la rezolvarea Problemei respective.

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