

Algoritmi genetici – scurtă prezentare

Algoritmii genetici fac parte din clasa algoritmilor evolutivi, care au proprietatea de a deveni mai performanți chiar în timp ce rulează, prin adaptarea și ”șlefuirea” soluției în mod iterativ până se ajunge la o soluție suficient de bună sau performantă, chiar dacă ea nu este soluția exactă a problemei.

Algoritmii genetici copiază practic modul cum teoria darwiniană presupune că evoluează speciile pe parcursul a mai multe generații. Mecanismul propus de Charles Darwin implică două proprietăți de bază:

- supraviețuirea indivizilor bine adaptați la mediu. Toți indivizii sunt constrânși să trăiască într-un mediu, cu care interacționează constant și care le pune la încercare trăsăturile morfogenetice. Unii indivizi prezintă, din întâmplare, acele trăsături care le conferă un avantaj la supraviețuire (fiind de exemplu, moștenite de la părinți). Astfel de indivizi vor ajunge să-și transmită genele urmașilor lor, pasându-le deci noii generații, iar urmașii lor vor manifesta, preponderent, aceleași caracteristici. Indivizii care nu supraviețuiesc până la vârsta reproducerii, evident, nu își transmit genele descendenților. Astfel, în timp – pe parcursul a mai multe generații, are loc un proces de *filtrare* a genelor care facilitează supraviețuirea indivizilor în mediu, iar genele nepotrivite sunt eliminate. Bineînțeles, supraviețuirea individului ”cel mai bine adaptat la mediu” nu este întotdeauna garantată, la fel cum nu este garantată nici eliminarea celui mai puțin adaptat. În definitiv, și pe cel mai bun atlet îl poate lovi un camion ! Putem spune deci că legea evoluției este oarecum aleatoare, dar numai într-o anumită măsură, pentru că are totuși un grad ridicat de determinism: prezența genelor ”bune” înclină puternic balanța spre supraviețuire, dar nu o garantează.
- aparitia unor modificări genetice aleatoare, care îmbogățesc bazinul genetic și introduc noi caracteristici în populația care evoluează. Chimic, aceasta se întâmplă în urma unor *mutații* aleatoare a ADN-ului la unii indivizi. Cauza acestor modificări poate fi atribuită ”sorții”, unii argumentând că ar fi vorba de radiații cosmice, caracteristica nedeterministă a unor procese chimice, sau chiar intervenția divină.

Teoria evoluționistă a lui Darwin susține că, prin repetarea succesivă a celor doi pași descriși anteriori, populațiile de indivizi evoluează și se adaptează încontinuu la mediul înconjurător. În ceea ce privește algoritmii genetici, corespondența cu teoria darwiniană este astfel: mediul – problema dată, caracteristicile ”câștigătoare” ale indivizilor – soluția problemei. De cele mai multe ori, nu se cunoaște de la început cum arată individul cel mai bine adaptat la mediu, dar se cunoaște modul cum afectează mediul (problema) capacitatea de supraviețuire a fiecărui individ (soluție posibilă).

1.1. Formularea problemei din punct de vedere matematic

Datorită caracteristicii aleatoare dar ghidată de mediu, algoritmii genetici sunt pretabili în special la problemele unde spațiul de căutare este imens și găsirea deterministă a unei soluții este sau imposibilă sau foarte costisitoare. Formularea problemei matematice este următoarea:

Se dă funcția $f(x_1, x_2, \dots, x_n): S \rightarrow R$, unde $S = M_1 \times M_2 \times \dots \times M_n$ este spațiul de căutare, se cere să se găsească $s' = (x_1', x_2', \dots, x_n') \in S$ pentru care se atinge minimul funcției f . Multimile $M_i \leq R$, deci $S \leq R^n$.

De obicei, spațiul de căutare nu este R^n , ci este limitat la un subset al acestuia, S . De asemenea, 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:

$$|s' - s_{optim}| \leq \varepsilon \quad \text{sau} \quad |f(s') - f(s_{optim})| \leq \varepsilon,$$

unde ε este gradul de precizie cerut de problemă, aplicat fie pe soluție direct, fie pe performanța soluției.

1.2. Noțiuni și concepte

Definim în continuare câteva concepte și noțiuni utile în domeniul algoritmilor genetici. Majoritatea noțiunilor sunt împrumutate din biologie, unde au înțeles similar.

- *Genă* – acea parte constituentă (codificată) a cromozomului care determină o anumită caracteristică în organismul care o conține. De exemplu, există gena care determină culoarea ochilor, lungimea brațelor sau masa musculară.
- *Alelă* – valoarea conținută în genă. În funcție de valoarea genei, caracteristica asociată va avea o proprietate sau alta. Alelele se aleg dintr-o mulțime finită (în biologie, dintr-un set de proteine).
- *Genotip* – mulțimea tuturor alelelor posibile. De exemplu, la organismul uman, valoarea alelelor conținute în gene este întotdeauna aleasă dintr-o mulțime de 4 tipuri de proteine: adenină (A), guanină (G), citozină (C), timină (T).
- *Fenotip* – caracteristicile comportamentale pe care fiecare individ le manifestă, fiecare dintre ele fiind codificată de alela genei corespunzătoare
- *Cromozom* – o mulțime cu un anumit număr de gene, care conține toată informația genetică aferentă unui individ.
- *Genom* – toată informația genetică care codifică fenotipul (deci mulțimea comportamentelor manifestate de o specie întreagă)
- *Individ* – organismul care trăiește în mediu și care este perfect caracterizat la nivel comportamental de cromozomi. Se consideră că pentru un individ, toți cromozomii săi sunt identici, deci putem vorbi despre o relație unu la unu între individ și cromozom.
- *Populație* – o mulțime de indivizi care trăiesc în mediu. În cadrul unei populații indivizii sunt diferiți și deci și gradul lor de adaptare la mediu este diferit.
- *Mapping* – o funcție de transformare care descrie modul cum dintr-un cromozom se poate obține individul pe care îl codifică. Procesul de *mapping* se mai numește și *morfogeneză*.
- *Funcția de performanță (funcția obiectiv)* – de obicei notată cu J , indică predispoziția fiecărui individ de a supraviețui, în funcție de fenotipul său. Practic, funcția de performanță reflectă mediul în care indivizii trăiesc.

1.3. Algoritmii genetici

În figura 1 se prezintă schema logică a modului cum funcționează algoritmi genetici.

Pasul 1. Se stabilește șablonul sau modelul după care se construiesc cromozomii. Aici trebuie ținut cont de faptul că individul va reprezenta soluția finală a problemei, deci cromozomul trebuie să aibă destulă informație genetică încât poată codifica oricare dintre soluțiile posibile (ținând cont și de precizia impusă). O posibilă abordare este să calculăm numărul de biți de care este nevoie pentru a putea reprezenta tot spațiul de soluții, și apoi să ne asigurăm că modelul cromozomului conține cel puțin atâția biți. Trebuie deci să se decidă *numărul de gene* ale cromozomului și *genotipul* (ce valoare poate lua fiecare genă). Dacă individul prezintă mai multe caracteristici diferite, este recomandat ca ele să fie codificate de gene diferite. Mai multe gene însă pot codifica împreună o singură caracteristică a fenotipului. În acest pas se stabilește de asemenea și funcția de mapping astfel încât mulțimea tuturor cromozomilor posibili să acopere perfect întreg spațiul de căutare al soluției (prin transformarea dată de mapping).

Pasul 2. Se creează un fond genetic aleator prin construcția unui bazin de cromozomi pe baza șablonului, în care valorile tuturor genelor sunt aleatoare în fiecare cromozom. Indivizii corespunzători acestor cromozomi inițiali formează *populația inițială*. De obicei, numărul de indivizi (deci cromozomi) în populație se păstrează constant de la o generație la alta, dar nu întotdeauna.

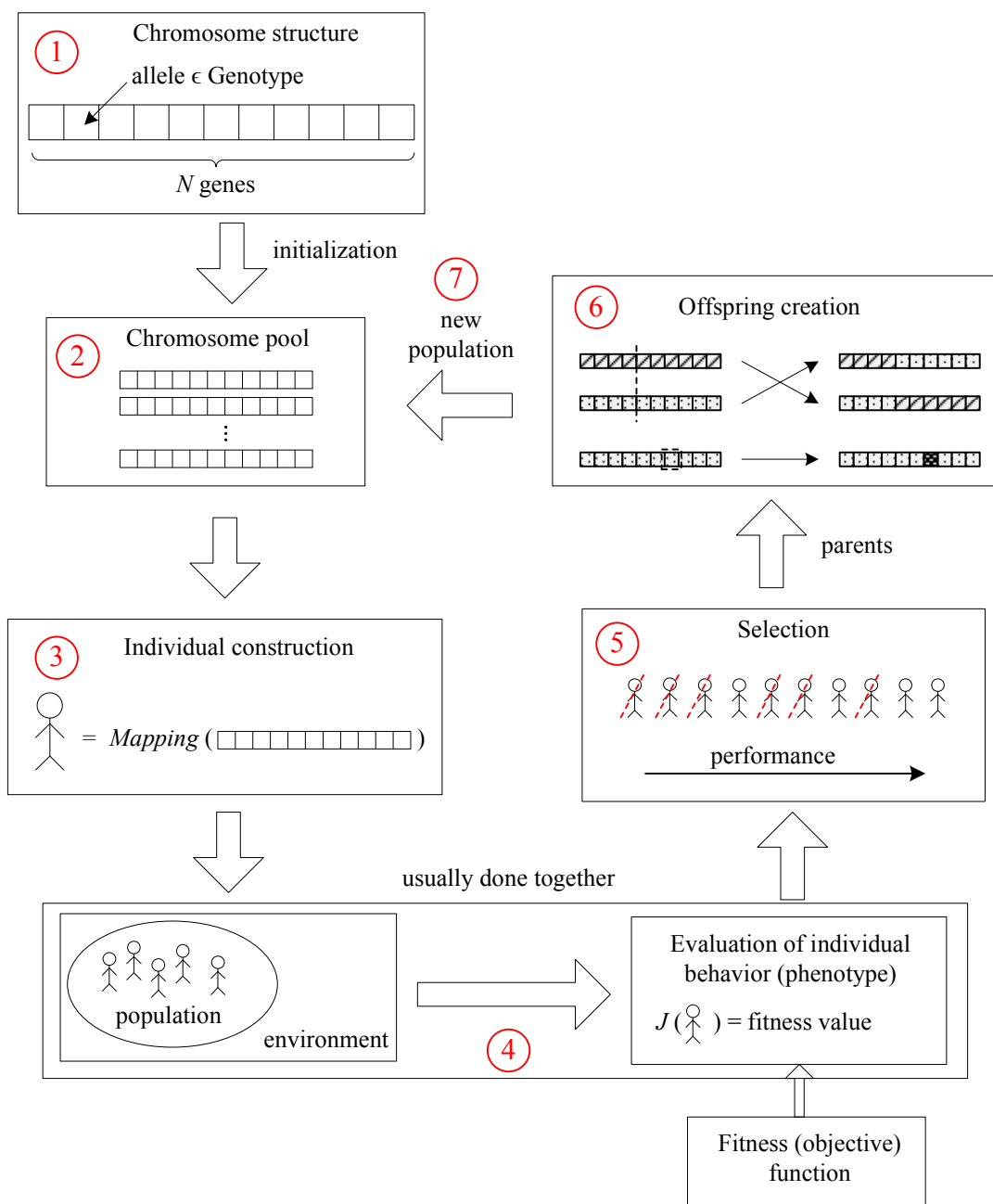


Figura 1. Schema logică a algoritmilor genetici

Pasul 3. Cu ajutorul funcției de *mapping*, se creează din fiecare cromozom individul corespunzător.

Pasul 4. Toată populația de indivizi se introduce în mediul înconjurător și se evaluează comportamentul fiecăruia folosind funcția de performanță (J). Funcția de performanță trebuie aleasă în așa fel încât să indice cât de bună este soluția posibilă găsită (individul) pentru problema dată. Cu cât este mai bună soluția, cu atât performanța ei este mai mare. De cele mai multe ori, se alege însă varianta în care o performanță ridicată

este reprezentată printr-un număr mic, dar pozitiv. Astfel, problema de optimizare este de a căuta acel individ pentru care funcția de performanță returnează valoarea cea mai mică.

Construirea funcției de performanță poate presupune acordarea de *bonusuri* pentru acele caracteristici care reprezintă o bună adaptare a individului la mediu, respectiv acordarea de *penalizări* pentru trăsăturile care scad probabilitatea individului de a supraviețui în mediu.

Pasul 5. După ce toți indivizii au fost evaluați, se face o selecție a celor mai potriviți candidați pentru a deveni părinții generației următoare, căreia îi vor transmite caracteristicile lor. Selecția celor mai buni candidați se poate face în mai multe moduri, cu efecte directe asupra modului de convergență a algoritmului spre soluție.

- *Selecția elitistă (elite selection)*: din întreaga populație se păstrează cei mai buni n indivizi (de exemplu, jumătate)
- *Selecția de tip "turneu" (tournament selection)*. Se alege aleator un grup mic de indivizi din populație și se păstrează numai cel mai bun individ din grup. Populația nouă este formată deci din câștigătorii acestor mici "turnee".
- *Selecția de tip ruletă (roulette selection)*. Probabilitatea fiecărui individ de a fi ales pentru crearea descendenților este direct proporțională cu performanța sa dată de funcția J (vezi figura 1).
- *Diferite combinații*: uneori se optează pentru păstrarea celor mai buni 2 sau 3 indivizi (elitist), iar ceilalți sunt selectați cu una din celelalte metode. Se observă că păstrarea unor indivizi mai neperformanți în fondul genetic poate fi uneori benefică pentru că altfel, algoritmul poate converge spre o zonă de minim local.

Pasul 6. Folosind cromozomii indivizilor selectați la pasul anterior se creează descendenții (urmasii) acestora, folosind o combinație a următorilor operatori:

- *Încrucișare (crossover)*: din 2 părinți se creează 2 descendenți diferiți astfel: se stabilește aleator un loc pe unde "se taie" amândoi cromozomii părinților, iar primul descendent va moșteni prima parte de la primul părinte, și a doua parte de la al doilea, iar al celălalt descendent – invers (vezi figura 2).

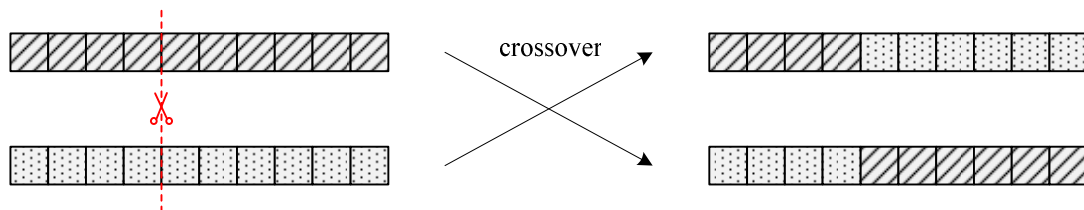


Figura 2. Încrucișare (crossover)

- *Mutație (mutation)*: dintr-un singur părinte se creează un descendent nou prin selecția aleatoare a unei gene și înlocuirea valorii ei cu o altă alelă din genotip (mulțimea alelelor posibile) (vezi figura 3).

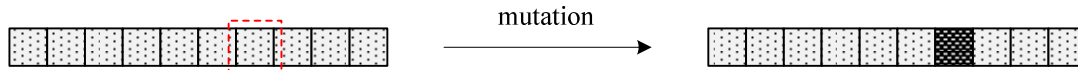


Figura 3. Mutație

- Uneori se folosește și operatorul de *reproducere directă* (*reproduction*) prin care un individ este "clonat" în mod identic în generația următoare.

Pentru a decide care dintre operatorii de mai sus se folosește la un moment dat se va folosi o componentă aleatoare.

Pasul 7. Mulțimea de descendenți creată la pasul 6 reprezintă noua populație (generația a doua).

Procesul descris de pașii 3 – 7 se reia de mai multe ori, iar cu fiecare generație care trece, populația de indivizi devine din ce în ce mai performantă, adică soluția cea mai bună se apropie de soluția optimă căutată. Condiția de oprire a algoritmului iterativ poate fi:

- un număr fix de iterații (generații)
- atingerea unei anumite valori a performanței (dată de funcția J)
- momentul în care progresul performanței de la o generație la următoarea nu este suficient de mare.