

Algoritmi genetici

Calculul cognitiv denotă o familie de metode de rezolvare a problemelor care imită inteligența “găsită” în natură. Obiectivul comun al acestor metode este de a soluționa situațiile ce au rezistat abordărilor de tip analitic. Din această categorie fac parte rețelele neuronale, logica fuzzy, automatele celulare, algoritmi genetici, fractalii, teoria haosului. Fiecare din aceste metode încercă să “simuleze soft” lumea reală și să proceseze datele de o manieră “cât mai naturală”.

Algoritmi genetici sunt strategii de rezolvare a problemelor cu ajutorul computerelor bazate pe teorii evoluționiste. Soluțiile potențiale sunt forțate să conlucreze într-un anumit mediu, având ca rezultat soluțiile cele mai bune în timp. Unul din principalele beneficii ale algoritmilor genetici este acela că pentru a rezolva o anumită problemă este nevoie de o cunoaștere a evoluției soluțiilor potențiale și nu construirea dinainte a uneia optime. Acest lucru devine evident cu cât complexitatea unei probleme se mărește, soluția optimă în cazul acest caz este mult mai greu de determinat, o astfel de abordare fiind mult mai bună. În plus metodele utilizate de algoritmi genetici sunt utilizate și în rezolvarea problemelor de către oameni.

Se poate spune că algoritmi genetici sunt proceduri de căutare bazate pe mecanisme naturale de selecție naturală și genetică. Algoritmi genetici au fost dezvoltati de John Holland în 1960 pentru a permite contemporanilor săi să găsească soluții pentru probleme dificile, cum ar fi funcțiile de optimizare și inteligența artificială.

Astfel, algoritmi genetici s-au dovedit utili în rezolvarea problemelor NP-complete. De exemplu, una dintre cele mai importante probleme în proiectarea compilatoarelor este să optimizeze codul generat dintr-un program scris într-un limbaj oarecare. În această privință, ne interesează să paralelizăm operațiile matematice de bază (adunări, scăderi, înmulțiri, împărțiri) pe mai multe unități de execuție sau pe mai multe procesoare. Această problemă este NP-completă, adică nu există algoritmi polinomiali care să o rezolve. O abordare posibilă ar fi folosirea *algoritmilor genetici*.

Algoritmi genetici

Algoritmi genetici sunt tehnici de rezolvare a problemelor NP-complete (dar nu numai) aplicând principii ale selecției naturale: se folosesc *populații* din care se selectează anumiți *indivizi* pentru *încrucișare* (*cross-over*), se evaluează indivizii populațiilor păstrându-se numai cei care convin, se aplică *mutații* pe indivizi selectați arbitrar, astfel încât populația evoluează, iar individul cel mai bun din populația finală este fie chiar soluția cea mai bună a problemei, fie o altă soluție foarte apropiată de cea optimă. Așadar, un algoritm genetic nu garantează găsirea soluției optime, dar conduce la găsirea unei soluții apropiate de optim.

În continuare sunt prezentate punctele principale prin care un AG diferă de tehnicile convenționale de căutare și optimizare. Acestea sunt:

1. AG sunt "orbi"

Un AG tratează problema de rezolvat ca pe o "cutie neagră", necesitând foarte puține informații despre aceasta. Din acest punct de vedere, un AG reprezintă o metodă slabă de căutare. Dar se observă în practică o foarte mare eficiență, putere de rezolvare a problemei, ceea ce este o caracteristică a metodelor puternice de căutare. Deci la acest punct apare un aparent paradox: AG sunt metode slabe de căutare cu eficiență foarte ridicată. După cum este argumentat în lucrările de specialitate, AG sunt metode slabe evoluționiste ce reprezintă proprietatea de inteligență emergentă, puterea lor venind dintr-o exploatare sofisticată a informațiilor obținute cu un efort de căutare relativ limitat (Grefenstette, 1987).

2. AG folosesc codificări ale variabilei problemei

Fiind dat un spațiu de căutare, cu un spațiu multidimensional al soluțiilor posibile, o reprezentare "cod genetic" este aleasă astfel încât fiecare punct din spațiul de căutare e reprezentat de un șir de simboluri numit **genotip**. În mod uzual, variabilele sunt codificate într-un genotip de lungime fixă, AG nu acționează direct asupra elementelor din spațiul stărilor problemei.

3. AG folosesc populații de soluții potențiale

Aici apare o diferență importantă față de alte tehnici adaptive de căutare care lucrează de la punct la punct, folosind informații locale. Un AG explorează spațiul de căutare în mai multe puncte simultan. Aceste puncte explorate la un moment dat formează o populație de soluții potențiale. Această explorare a spațiului de căutare e făcută simultan cu exploatarea celor mai bune soluții la pasul curent.

4. AG folosesc operatori probabilistici

Populația de la pasul următor de căutare este dedusă prin aplicarea unor operatori de tip probabilistic. Această caracteristică face ca AG să fie o metodă de tip Monte-Carlo, dar trebuie să facem observația că această căutare realizată nu e pur aleatoare, ci o explorare "inteligentă" a spațiului de căutare. Se spune că AG realizează un echilibru aproape optimal între explorarea spațiului stărilor și exploatarea celor mai bune soluții găsite.

Reprezentarea soluției

Fiecare individ al unei populații este o reprezentare a soluției problemei. Indivizii sunt denumiți *cromozomi*. Unul din cele mai importante lucruri în rezolvarea unei probleme folosind algoritmi genetici este găsirea unei reprezentări a soluției. În general, soluția (adică un cromozom) se reprezintă ca un tablou uni- sau multidimensional ce conține biți, întregi, caractere sau chiar numere reale. Cel mai utilizat mod de codare pentru cromozomi este cu ajutorul unui șir de biți. Astfel, fiecare cromozom va fi un șir de biți. Fiecare bit din acest șir poate reprezenta o caracteristică a soluției sau șirul în întregul său poate reprezenta un număr. Exemplu:

Cromozom 1	1101100100110110
Cromozom 2	1101111000011110

Pe cromozomi se aplică două operații importante:

- încrucișarea și
- mutația.

Prima generează doi cromozomi fii din doi cromozomi părinți, iar a doua operație modifică un cromozom într-unul nou. Orice cromozom nou generat trebuie să fie o soluție validă a problemei, ceea ce impune alegerea cu mare atenție a reprezentării cromozomilor (deci a soluțiilor problemei). Uneori trebuie aplicată o funcție de corecție pe cromozomi pentru a-i aduce la forme valide.

Structura algoritmului

Există mai multe forme pentru un algoritm genetic, însă forma de la care s-a plecat și care este și cea mai obișnuită este următoarea:

Procedura AlgoritmGenetic

begin

t=0

generează populația inițială P(0)

repeat

 evaluatează populația P(t)

 t=t+1

 selectează din P(t) indivizii care se vor încrucișa

 aplică operatorul de încrucișare

 aplică operatorul de mutație

```
until t = t_max
end
```

Mai întâi, generarea populației inițiale se poate face creând un număr fix de cromozomi în mod aleator, cu condiția ca fiecare cromozom să fie o soluție validă a problemei de rezolvat. Evaluarea unei populații se face aplicând o funcție de evaluare asupra fiecărui individ. Funcția de evaluare (fitness sau adecvare) este aleasă de programator și ea trebuie să reflecte apropierea sau depărtarea unui individ de soluția optimă a problemei. De alegerea corespunzătoare a acestei funcții depinde buna funcționare a algoritmului genetic. Se mai spune că această funcție descrie acțiunea mediului.

Procesul de **selecție** e bazat pe "notele" ("evaluările") punctelor din populația curentă, în așa fel încât indivizii mai "dotați", cu "scoruri" mai ridicate contribuie mai mult la formarea unei populații intermediare. În mod uzual, soluția se face în mod probabilistic: genotipii au o probabilitate de a trece în populația intermediară ce este proporțională cu "calitatea" lor. (Altă metodă utilizează o metodă numită selecție turnir, în care se alege aleator o submulțime a acestor indivizi, din care doar cel mai adecvat este ales pentru operațiile de încrucișare și mutație.)

Din această populație intermediară de genotipi selectați, noua generație e dedusă prin aplicarea unor **operatori genetici**, precum **încrucișarea** și **mutația**.

În sfârșit, t_{\max} reprezintă numărul maxim de generații pentru care se execută algoritmul, acesta constituind un alt parametru al algoritmului genetic.

Operatorii

1. *Operatorul de încrucișare* se aplică pe doi cromozomi (indivizi) și generează alți doi, alăturând părți din cromozomii inițiali. Concret, se consideră doi genotipi "părinți", se generează aleator o poziție de încrucișare undeva de-a lungul genotipului și se consideră secțiunea din stânga poziției respective, a unuia din părinți pe care o lipește la secțiunea din dreapta poziției de încrucișare a celuilalt părinte. Astfel cei doi urmași obținuți moștenesc material genetic de la fiecare din părinți.

Presupunem că cei doi cromozomi sunt reprezentați sub forma a doi vectori:

a0 a1 a2 a3 a4 a5 a6 a7

și respectiv

b0 b1 b2 b3 b4 b5 b6 b7.

O metodă ar fi să se aleagă arbitrar un indice, de exemplu 2 (vectorii încep cu indicele 0) și rezultă următorii cromozomi:

a0 a1 a2 b3 b4 b5 b6 b7

și

b0 b1 b2 a3 a4 a5 a6 a7.

Pentru anumite reprezentări ale cromozomilor, această metodă poate să nu genereze indivizi valizi și trebuie aplicată altă tehnică.

De exemplu, (unde | reprezintă punctul de încrucișare):

Cromozom 1	11011 00100110110
Cromozom 2	11011 11000011110
C 1 – după încrucișare	11011 11000011110
C 2 – după încrucișare	11011 00100110110

2. *Operatorul de mutație* schimbă în general un element (ales arbitrar) al tabloului care reprezintă un cromozom. De fapt, se spune că mutațiile modelează apariția unor defecțiuni în procesul transmiterii materialului genetic. Acestea modelează procesul observat în natură și au rolul evitării blocării căutării în puncte de maxim local.

Exemplu:

Offspring 1	11011/1000011110
Offspring 2	1101100100110110
Offspring 1 - dupa mutatie	1101101000011110
Offspring 2 - dupa mutatie	1100100100111110

Aplicații

În continuare este prezentat un exemplu de aplicație a algoritmilor genetici și anume determinarea maximului unei funcții de gradul II (i.e. $f(x) = -x^2 + 2x - 1$). Pentru reprezentarea mulțimii de gene se folosește un șir de biți. Valorile reprezentate ca gene vor avea **NRBIȚI** biți. Din punct de vedere al funcției matematice, valorile acestui șir sunt privite ca întregi, ele reprezentând x (argumentul funcției).

Trecerea de la o generație la alta se va face prin următorii pași:

a. Pentru toate genele, se vor determina valorile așa numitei funcții de fitness care va determina apropierea genei respective de un punct de maxim. Acest maxim nu este maximum absolut al funcției, ci este punctul de maxim corespunzător genelor existente în generația curentă.

S-a ales funcția următoare:

$$\text{fit_val}(x) = 1 / (1 + \text{val_max} - f(x))$$

Această funcție este apropiată de 1 pentru genele convenabile (pentru care funcția este apropiată de valoarea de maxim) și e apropiată de 0 pentru genele neconvenabile.

b. Se crează nucleul noii generații

Nucleul noii generații ("new generation pool") va conține un număr fix de gene, cele mai convenabile din generația respectivă.

c. Se face numărul dorit de încrucișări. Aceste încrucișări se vor face selectând gene aleatoare, care nu fac parte din nucleu, și înlocuind aceste gene cu gene obținute prin încrucișarea a două gene alese la întâmplare din nucleul noii generații. Încrucișarea se va face prin selectarea unei poziții de tăiere, și crearea noii gene prin preluarea primilor biți de la prima genă și preluarea ultimilor biți de la a doua genă părinte.

d. Se face numărul dorit de mutații. Aceste încrucișări se vor face selectând gene aleatoare, care nu fac parte din nucleu, și aplicând asupra acestor gene o mutație. Această mutație se va face prin modificarea aleatoare a unui bit din gena respectivă.

Se va urmări aplicația max_ga.

- Să se ruleze programul din aplicație pe un exemplu de 300 de gene, 50 de gene în nucleul noii generații, 250 de gene obținute din încrucișări, 10 gene obținute prin mutații iar reprezentarea genelor pe 12 biți.
- Încercați să măriți numărul de biți din reprezentarea genelor.
- Încercați să măriți numărul de gene din nucleul noii generații.
- Dacă se micșorează numărul de gene noi (din încrucișări sau mutații) ce se întâmplă?

Aplicații dezvoltate

The Westinghouse Science & Technology Center (STC) este unul din liderii mondiali ai aplicațiilor care folosesc algoritmi genetici pentru a rezolva probleme din lumea reală. În timpul celei de a doua jumătăți a anului 1995, STC a fost contactată pentru a crea pentru prima dată un sistem de planificare a expedițiilor pentru compania Furtuna 100. STC a creat un sistem avansat de evidență a vânzărilor denumit "Transportation Management Industry" Acest produs este un sistem de evidență

a expedițiilor încorporând cel mai avansat soft de decizie, oferă companiei suport tactic, în timp real, pentru companii care au volume mari de expediții pe zi. Acest sistem client-server este primul care utilizează algoritmi genetici pentru luarea deciziilor în cadrul aplicațiilor pentru transport. Compania Fortuna 100 speră să economisească zeci de milioane de dolari folosind acest produs.

Aplicații ale algoritmilor genetici pentru probleme seismice dezvoltă noi probleme de test și compară rezultatele algoritmilor genetici cu metode locale de calcul heuristic.

Supraveghetor cu algoritmi genetici. Scopul este acela de a rescrie softul existent pentru algoritmi genetici în scopul realizării unui supraveghetor în domeniul transportului. Acest supraveghetor va fi comparat cu doua supraveghetoare deja existente.

Optimizarea rețelor. Una din primele aplicații ale algoritmilor genetici a fost în controlul distribuției de gaz printr-o conductă, în regim staționar și în regim tranzitoriu. De-a lungul conduitei, presiunea gazului descrește în mod natural și trebuie mărită cu ajutorul unor compresoare. Obiectivul constă în menținerea presiunii în punctele de livrare la nivelul dorit, cu minimizarea energiei folosite în compresoare și îndeplinirea altor restricții. De asemenea, este necesară detectarea, pe baza măsurării presiunii, a scurgerilor probabile, evitând, pe cât posibil, alarmele false. Analog acestei aplicații, mai sunt o serie de aplicații, de exemplu, în cazul rețelor de comunicație între stații aflate la mare distanță. În acest caz, una dintre restricțiile impuse era ca la orice stație să se poată ajunge pe cel puțin două căi, astfel încât nici o stație să nu fie izolată în urma căderii unei singure legături. O aplicație similară a fost realizată în reglarea presiunii într-o rețea de distribuire a apei. Deoarece pierderile datorate scurgerilor în conducte subterane cresc în mod considerabil odată cu presiunea, este important ca aceasta să fie menținută la un nivel cât mai mic, dar fără a periclita deservirea clienților. O metodă simplă de control constă în deschiderea sau închiderea unor supape pe traseu, modificând în acest fel topologia curgerii. În acest context, sarcina unui algoritm genetic este de a găsi o topologie corespunzătoare, care minimizează presiunea apei pe traseu, îndeplinind totodată cerințele clienților. Fiecare cromozom reprezintă un graf al topologiilor posibile într-un mod relativ direct, prin două mulțimi: un arbore de deducție (spanning tree) și un co-arbore de legături suplimentare. Se utilizează o strategie generativă, iar părinții produc descendenți prin intermediul unui operator de recombinare specializat care garantează validitatea rezultatelor. Valorile de fitness sunt determinate printr-o analiză completă a topologiei reprezentate. O primă comparație cu metode enumerative sugerează că algoritmi genetici sunt utili în rezolvarea acestei probleme, dar, având în vedere că majoritatea topologiilor generate aleatoriu au defectul de a lăsa o parte dintre clienți fără apă, este posibil să existe variante îmbunătățite ale tehnicilor convenționale care să dea rezultate cel puțin la fel de bune ca algoritmi genetici.

Optimizarea planificărilor. De exemplu, se utilizează algoritmi genetici pentru:

- planificarea examenelor,
- problema orarului,
- problemele de optimizare a fluxului de producție și a utilizării forței de muncă.

Aplicații în industrie. Se utilizează algoritmi genetici pentru rezolvarea problemei de determinare a traseelor în domeniul circuitelor VLSI. Astfel, în formularea cea mai simplă, se dă o placă dreptunghiulară având un număr de intrări pe latura de jos și un număr de ieșiri pe latura de sus, între care trebuie făcute anumite rețele de legături.

Link-uri utile

<http://cs.felk.cvut.cz/~xobitko/ga/>

<http://www.optiwater.com/GAsearch/>

<http://homepage.sunrise.ch/homepage/pglaus/gentore.htm#Applet>

http://techni.tachemie.uni-leipzig.de/jsga/applications/fktmax/index_eng.html

<http://oldeee.sec.ed.ac.uk/~rjt/ga.html>

http://cmp.ameslab.gov/cmp/CMP_Theory/gsa/gen2.html

Dicționar

Alele – stările în care se poate găsi o genă (valori ale caracteristicilor).

Criteriu de oprire: Criteriu prin care se stabilește când se oprește procesul evolutiv.

Cromozom – informație pe baza căreia este construit un model matematic ce este supus evaluării. Deci, un cromozom este utilizat pentru a referi o soluție potențială, și constă în toate informațiile necesare pentru a descrie o soluție. Se spune că cromozomii sunt purtătorii informației genetice;

Evaluare: Operațiune prin care este stabilit gradul în care modelul matematic asociat unui cromozom satisface un anumit criteriu de optimalitate.

Evoluție: Procedu prin care o populație se transformă într-o nouă populație prin aplicarea selecției și a operatorilor genetici.

Fenotip – obiectul corespunzător genotipului în problema dată. Acesta este obiectul matematic.

Fitness: Măsură obținută în urma evaluării unui cromozom.

Gena – unitate compozițională indivizibilă a unui cromozom. Genele poartă caracteristicile ereditare, o genă controlând una sau mai multe caracteristici.

Generație: O populație, în ansamblul ei, privită prin prisma procesului de evoluție.

Genotip – totalitatea cromozomilor unui individ dintr-o populație (fiecare individ al unei specii posedă un număr determinat de cromozomi). Reprezentarea internă a acestui tip poate fi un șir de biți.

Locus sau loci – poziția ocupată de o genă în cadrul cromozomului.

Operator genetic binar (incrucisare): operator prin care se obțin doi cromozomi copii din doi cromozomi părinți, eventual prin interschimbarea de gene.

Operator genetic unar (mutație): operator prin care se obține un cromozom copil dintr-un cromozom părinte prin modificarea unor gene a cromozomului părinte.

Operator genetic: procedu prin care un număr de cromozomi, numiți părinți, contribuie cu anumite gene pentru a se obține alți cromozomi, numiți cromozomi copii.

Populație – mulțime ce conține în mod tipic din mai mulți cromozomi. În mod normal există un număr limitat de cromozomi și acei cromozomi care sunt defecți sunt eliminați pentru a face loc altora cu performanțe mai bune;

Selecție: Procedu sistematic prin care sunt aleși cromozomii ce urmează a fi părinți în cadrul operatorilor genetici. În principiu, cromozomii cu fitness mai bun au șanse mai mari de a fi selecționați.