

Algoritmi genetici - proprietăți și aplicații

noiembrie 2003

1 Analiza algoritmilor genetici

La proiectarea unui algoritm genetic trebuie să răspundem la o serie de întrebări, cum ar fi:

- Care este cea mai potrivită codificare ? Ce dimensiune să aibă populația ?
- Ce mecanism de selecție este mai adecvat ?
- Ce tip de încrucișare/mutație conduce la un algoritm eficace și ce valori să luăm pentru parametrii de control (p_c și p_m) ?

Ideal ar fi să existe rezultate teoretice care să stabilească o legătură între caracteristicile algoritmului și abilitatea/ eficiența de a rezolva problema pentru care a fost proiectat. Din păcate o astfel de teorie generală a algoritmilor genetici nu există. Absența unei teorii generale nu înseamnă că nu există *modele matematice* și rezultate teoretice privind anumite aspecte ale comportării algoritmilor genetici. Acestea dau răspunsuri (parțiale) la întrebări de genul: ce influență are un operator genetic (selecție, încrucișare, mutație) asupra structurii populației ? cum se comportă populația din punct de vedere asimptotic ?

În practică aceste rezultate se folosesc în combinație cu *experiența dobândită* în rezolvarea prin tehnici evolutive a unor probleme similare. Din acest motiv proiectarea algoritmilor genetici este în continuare o "artă" necesitând din partea expertului uman o bună cunoaștere a problemei precum și intuiție (componenta euristică joacă un rol important). Aceasta nu înseamnă că un algoritm genetic "standard" nu ar putea fi aplicat pentru o anumită problemă, însă s-ar putea să nu fie eficace.

Pe de altă parte în ultimul timp a devenit din ce în ce mai mult acceptată ideea că nu există un algoritm de optimizare (evolutiv sau nu) "universal" valabil, care să fie eficient pentru orice clasă de probleme. Mai mult în [5] se arată în manieră formală că toți algoritmi de optimizare se comportă similar *în medie* (în raport cu toate clasele de probleme). Aceasta înseamnă că nu există un algoritm care să fie mai bun decât toți ceilalți în raport cu toate clasele de probleme. Altfel spus orice modificare a algoritmului care duce la o îmbunătățire a comportării lui pentru o clasă de probleme provoacă și o reducere a eficienței în cazul altei clase de probleme. Aceste rezultate conduc la ideea inexistenței unui algoritm general ci doar a unor algoritmi genetici adecvați pentru fiecare clasă de probleme.

Există două abordări teoretice importante ale algoritmilor genetici:

- Abordarea bazată pe conceptul de *schemă* introdus de Holland care permite o analiză cu caracter static a comportării algoritmului (influența operatorilor genetici asupra structurii populației în cadrul unei generații).

- Abordarea bazată pe utilizarea *lanțurilor Markov* în modelarea algoritmilor genetici, ce permite o analiză cu caracter dinamic a acestora (comportarea lor asimptotică).

Nici una dintre aceste abordări nu oferă o imagine completă asupra comportării algoritmilor genetici, o serie de aspecte fiind încă probleme deschise.

1.1 Teoria schemelor

În ciuda criticilor care i-au fost aduse în ultimii ani, teoria schemelor reprezintă una dintre primele abordări teoretice sistematice a studiului algoritmilor genetici fiind adecvată codificării discrete. Ea permite formarea unei imagini intuitive despre ceea ce se "întâmplă" în cadrul populației în urma aplicării mecanismelor de selecție, încrucișare și mutație.

Pentru a ilustra ideile acestei teorii vom considera un algoritm genetic standard caracterizat prin:

- Populația este constituită din m elemente, fiecare fiind un vector cu r componente binare (din $\{0, 1\}$).
- Selecție proporțională a părinților și supraviețuirea tuturor fiilor.
- Încrucișare cu un singur punct de tăietură și probabilitatea de aplicare, p_c .
- Mutație aplicată, cu probabilitatea p_m asupra unei singure gene din fiecare cromozom.

Conceptele fundamentale ce intervin în teoria schemelor sunt prezentate în continuare.

Schemă. O schemă este un element din $\{0, 1, *\}^r$ care descrie de fapt o submulțime a lui $\{0, 1\}^r$. Fiecare element al populației aparține uneia sau mai multor scheme. De exemplu, pentru $r = 3$ exemple de scheme sunt: $(*, *, *)$, $(1, *, *)$, $(*, 0, 1)$, iar elementul $(1, 0, 1)$ aparține tuturor acestor scheme pe când elementul $(1, 0, 0)$ aparține doar primelor două scheme. Pozițiile din cadrul schemei care au asociate valorile 0 sau 1 se numesc *definite* (sau *specifice*) iar cele cărora le corespunde $*$ se numesc *ndefinite*. În cazul codificării binare, spațiul de căutare poate fi văzut ca un hipercub n -dimensional. Schemele corespund "muchiilor" și "fețelor" hipercubului, motiv pentru care schemele sunt numite și hiperplane. În cazul $r = 3$ hipercubul este un cub ale cărui componente sunt ilustrate în figura 1.

Ordinul unei scheme. Se definește ca fiind numărul de elemente definite (diferite de $*$) din schemă. În ilustrarea din figura 1 vârfulurile cubului sunt scheme de ordin 3, muchiile sunt scheme de ordin 2, fețele sunt scheme de ordin 1 iar întregul cub este de ordin 0. Notăm ordinul unei scheme S prin $o(S)$.

Lungimea utilă a unei scheme. Reprezintă diferența dintre ultima și prima poziție definită. De exemplu lungimea utilă a schemei $(*, 1, *, 0, *, *, 0, *)$ este $7 - 2 = 5$. Într-o schemă de dimensiune r cea mai mare lungime utilă poate fi $r - 1$ (când prima și ultima poziție sunt definite). O schemă de ordin mai mic sau egal decât 1 are lungimea utilă egală cu 0. Notăm lungimea utilă a unei scheme S prin $d(S)$.

Efect distructiv. Se consideră că un operator genetic are efect distructiv ("disruptive effect") asupra unei scheme dacă prin aplicarea lui asupra unui element ce aparține schemei rezultatul obținut nu mai aparține schemei respective. În felul acesta există posibilitatea ca schema să dispară din cadrul populației.

Cazul încrucișării. Să considerăm schemele $(0, 1, *, *)$ și $(1, 0, *, *)$ și încrucișarea folosind un punct de tăietură, k . Dacă $k = 1$ atunci prin încrucișare se obțin schemele $(0, 0, *, *)$ și $(1, 1, *, *)$

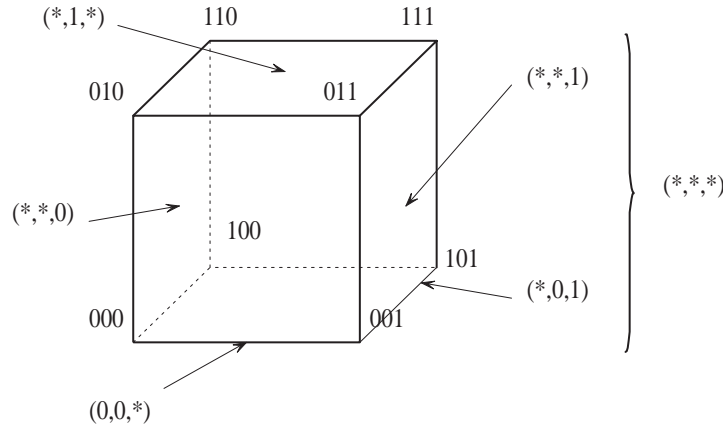


Figura 1: Ilustrarea schemelor printr-un cub 3-dimensional

în timp ce schemele inițiale au fost distruse. În schimb dacă $k = 3$ prin încrucișare se obțin aceleași scheme (de această dată încrucișarea nu mai are efect distructiv). Este ușor de observat că dacă punctul de tăietură nu este între două poziții definite încrucișarea nu are efect distructiv. În schimb dacă punctul de tăietură se află între două poziții definite atunci probabilitatea să se producă efect distructiv este mare (există însă și posibilitatea ca schema analizată să supraviețuiască). Rezultă că probabilitatea ca încrucișarea să aibă efect distructiv este cu atât mai mare cu cât lungimea utilă a schemei/schemelor este mai mare, întrucât probabilitatea ca punctul de tăietură să fie între două poziții definite este mai mare.

Cazul mutației. Modificarea prin mutație a unei poziții definite provoacă distrugerea schemei pe când a unei poziții nedefinite nu provoacă distrugerea schemei. De exemplu, dacă asupra schemei $(0, 1, *, *)$ se aplică mutația la nivelul celui de al doilea element se obține schema $(0, 0, *, *)$, iar dacă se aplică asupra celui de al treilea element schema rămâne nemodificată. Probabilitatea ca mutația să aibă efect distructiv este proporțională cu ordinul schemei.

Efect constructiv. Se consideră că un operator genetic are efect constructiv ("constructive effect") dacă prin aplicarea lui se pot crea scheme noi. Din exemplele de mai sus se observă că atât încrucișarea și mutația pot avea efect constructiv.

Probabilitate de supraviețuire. Se referă la șansa ca o schemă să rămână neschimbată după aplicarea operatorilor genetici. Supraviețuirea unei scheme este un fenomen opus celui de distrugere. Probabilitatea de supraviețuire a unei scheme după aplicarea încrucișării și/sau mutației este invers proporțională cu lungimea utilă a schemei și/sau ordinul acesteia.

Teorema schemelor. Fie $P(t)$ populația de la generația t și S o schemă. Notăm cu $m(S, t)$ numărul de elemente ale populației $P(t)$ care aparțin schemei S . Teorema schemelor stabilește o legătură între valoarea medie a numărului de elemente din populația $P(t + 1)$, $\langle m(S, t + 1) \rangle$ și $m(S, t)$. Fiecare dintre mecanismele evolutive are o influență specifică asupra lui $m(S, t)$.

Influența selecției. Considerăm că fiecărui element, $x_i(t)$, al populației i se asociază valoarea fitness-ului, $f_i = f(x_i(t))$. Valoarea medie a fitnessului pentru întreaga populație este $\bar{f}(t) = \sum_{i=1}^m f(x_i(t))/m$. Presupunem că elementele populației sunt numerotate astfel încât $x_i(t) \in S$ pentru $i = 1, m(S, t)$ (elementele ce aparținând schemei S sunt primele din populație). Utilizând selecția proporțională, rezultă că numărul mediu de apariții ale elementului x_i în populația $P(t + 1)$

este:

$$m_i = m \frac{f_i}{\sum_{j=1}^m f_j} = \frac{f_i}{\bar{f}(t)}.$$

Numărul mediu de elemente din populația $P(t+1)$ care aparțin schemei S este:

$$\langle m(S, t+1) \rangle = \sum_{i=1}^{m(S,t)} m_i = \sum_{i=1}^{m(S,t)} \frac{f_i}{\bar{f}(t)} = m(S, t) \frac{\bar{f}(S, t)}{\bar{f}(t)},$$

unde $\bar{f}(S, t)$ reprezintă valoarea medie a fitnessului elementelor din populație ce aparțin schemei. Rezultatul

$$\langle m(S, t+1) \rangle = m(S, t) \frac{\bar{f}(S, t)}{\bar{f}(t)}$$

sugerează că prin selecție, numărul (mediu) de elemente ale unei scheme de fitness "mare" crește iar cel al unei scheme de fitness "mic" scade. Cu alte cuvinte, proliferează schemele adecvate. Calificativele "mare" și "mic" sunt relative la populația curentă (fitnessul mediu al schemei este calculat pe baza populației curente fiind astfel doar o estimare a celui global).

Influența încrucișării. După selecție populația este supusă încrucișării (se aplică cu probabilitatea p_c asupra a m perechi de elemente). Pentru a estima numărul mediu de elemente ale populației care aparțin schemei este suficient să determinăm probabilitatea de supraviețuire a schemei, $p_{s,c}$. Aceasta poate fi exprimată prin probabilitatea de distrugere $p_{d,c}$ folosind relația $p_{s,c} = 1 - p_{d,c}$. Probabilitatea de distrugere este determinată de probabilitatea ca punctul de tăietură k să fie între două poziții definite și anume $d(S)/(r-1)$. Cum nu în toate situațiile un punct de tăietură între două poziții definite provoacă distrugerea schemei și cum încrucișarea se aplică cu probabilitatea p_c rezultă că $p_d \leq p_c d(S)/(r-1)$, prin urmare $p_s \geq 1 - p_c d(S)/(r-1)$. Astfel după selecție și încrucișare numărul mediu de reprezentanți ai schemei S respectă relația:

$$\langle m(S, t+1) \rangle \geq m(S, t) \frac{\bar{f}(S, t)}{\bar{f}(t)} \left(1 - p_c \frac{d(S)}{r-1}\right).$$

Influența mutației. La fel ca și în cazul încrucișării estimăm probabilitatea de supraviețuire după mutație $p_{s,m}$. Întrucât mutația este un proces aleator care se aplică independent fiecărei poziții (gene) și cum efectul distructiv apare doar dacă este aplicată unei poziții definite rezultă că probabilitatea de supraviețuire constă în produsul probabilităților ca pozițiile definite să nu fie supuse mutației: $p_{s,m} = (1 - p_m)^{o(S)}$. Prin urmare după selecție, încrucișare și mutație numarul mediu de reprezentanți ai schemei S verifică:

$$\langle m(S, t+1) \rangle \geq m(S, t) \frac{\bar{f}(S, t)}{\bar{f}(t)} \left(1 - p_c \frac{d(S)}{r-1}\right) (1 - p_m)^{o(S)}.$$

Aceasta este o variantă a *teoremei schemei*. O variantă ușor simplificată se obține ținând cont de faptul că în practică p_m se alege foarte mic (de ordinul 10^{-3}) astfel că este acceptabilă aproximarea: $(1 - p_m)^{o(S)} \simeq 1 - p_m o(S)$.

Blocuri constructive. Teorema schemei afirmă că de la o generație la alta proliferează schemele a căror *adecvare este peste medie*, au *ordin mic* și *lungime utilă mică*. Schemele care satisfac aceste trei criterii sunt denumite în teoria algoritmilor genetici blocuri constructive ("building blocks"). Teoria blocurilor constructive susține că prin încrucișare scheme adecvate, de lungime utilă mică pot conduce la scheme de fitness mai bun. Detalii privind teoria blocurilor constructive pot fi găsite în [2]

1.2 Modelarea cu lanțuri Markov

Întrucât populația de la generația curentă depinde doar de populația de la generația anterioară iar transformările au caracter aleator, un model matematic adecvat pentru descrierea procesului de evoluție este cel al lanțurilor Markov. În cazul algoritmilor genetici este vorba despre lanțuri Markov cu număr finit de stări. Folosind acest model au fost determinate condiții suficiente pentru ca probabilitatea ca populația să atingă optimul global să tindă către 1 (algoritmul converge aproape sigur). Fără a intra în detalii amintim că astfel de condiții suficiente sunt: (i) mutația permite trecerea într-un număr finit de generații de la o configurație la oricare alta; (ii) orice element al populației curente poate fi selectat pentru a participa la construirea generației următoare (probabilitatea de selecție a oricărui element este nenulă) și cu o probabilitate nenulă rămâne nealterat prin încrucișare; (iii) Orice element (fie din generația curentă fie obținut în procesul de reproducere) are o probabilitate nenulă de supraviețuire; (iv) cel mai bun element al populației supraviețuiește prin selecție în generația următoare (această proprietate a selecției este numită *elitism*). Pentru detalii se poate consulta [3].

2 Aplicații

Algoritmii genetici au aplicabilitate în rezolvarea unor probleme cu caracter științific și ingineresc. Câteva dintre acestea sunt [2]:

- Optimizare. Probleme de optimizare în domenii continue și discrete cu aplicații în proiectarea sistemelor (de exemplu a formelor aerodinamice), proiectarea circuitelor electronice, planificarea activităților (de exemplu generarea automată a orarelor).
- Clasificarea datelor și învățare automată. Dezvoltarea evolutivă a unor reguli de învățare pentru sistemele de clasificare sau a regulilor de producție utilizate în sistemele expert.
- Proiectarea unor sisteme computaționale. Algoritmii genetici pot fi utilizați pentru determinarea regulilor de calcul la automate celulare precum și în stabilirea arhitecturii rețelelor neuronale.
- Modelare și predicție. Modelarea sistemelor dinamice și efectuarea de predicții (de exemplu, predicția vremii, predicția structurii proteinelor. Modelarea sistemelor economice și efectuarea de predicții privind fenomene cu caracter economic (de exemplu evoluția pieței financiare, dezvoltarea unor strategii de ofertă).
- Modelarea unor sisteme biologice. Algoritmii genetici pot fi utilizați pentru modelarea unor aspecte ale sistemului imunitar natural sau pentru modelarea unor fenomene ecologice (cum sunt coevoluția gazdă-parazit, simbioză etc.)
- Control. Determinarea regulilor ce permit controlul unui sistem neliniar. Ideea este de a nu folosi un model matematic asociat sistemului.
- Dezvoltare de strategii. Determinarea prin algoritmi genetici a strategiilor de "caching" și replicare în serverele Web.

Referințe

- [1] Beyer H.G.; An Alternative Explanation for the Manner in which Genetic Algorithms Operate, 1996.
- [2] M. Mitchell; An Introduction to Genetic Algorithms, MIT Press, 1996.
- [3] G. Rudolph; Finite Markov Chain Results in Evolutionary Computation: A Tour dorizon, Fundamenta Informaticae, 1-22, 1998.
- [4] Spears W.M.; Evolutionary Algorithms. The Role of Mutation and Recombination, Springer Verlag, 2000.
- [5] Wolpert D., Macready W.; No free lunch theorem for optimization, IEEE Trans. on Evolutionary Computation, 1(1), 67-82, 1997.