Prace Naukowe Wyższej Szkoły Pedagogicznej w Częstochowie
Matematyka V, Częstochowa 1997

Algorytmy genetyczne

Artur Gola

Algorytmy genetyczne oraz programy ewolucyjne umożliwiają rozwiązanie wielu zadań, których nie można efektywnie rozwiązać przy pomocy metod heurystycznych. Algorytmy genetyczne są algorytmami losowymi wykorzystującymi mechanizmy doboru naturalnego oraz dziedziczenia potrzebne do tworzenia coraz lepszych zbiorów rozwiązań. Połączenie ewolucyjnej zasady przeżycia najlepiej przystosowanych z systematyczną wymianą informacji umożliwiają określenie nowych obszarów poszukiwań o spodziewanej podwyższonej wydajności.

Zbiór potencjalnych rozwiązań zwany populacją składa się z „osobników" zwanych chromosomami, które reprezentują pojedynczy element przestrzeni rozwiązań. Podstawowym problemem przy tworzeniu algorytmu genetycznego jest określenie kodu chromosomu, który musi być zapisany za pomocą łańcuchów o skończonej długości w pewnym określonym alfabecie.

W klasycznych algorytmach genetycznych używa się chromosomów zakodowanych przy pomocy wektorów binarnych oraz dwóch podstawowych operatorów: krzyżowania i mutacji. Nieco inne podejście do sposobu reprezentacji chromosomów zastosowano w programach ewolucyjnych. Zrezygnowano z kodowania na rzecz „naturalnej" reprezentacji potencjalnych rozwiązań. Wymagało to także zastosowania odpowiednich operatorów genetycznych zapewniających efektywne przekazywanie informacji.

W swoim referacie chciałbym przybliżyć problem doboru struktury danych i operatorów genetycznych do danego zadania, gdyż te czynniki w znaczny sposób wpływają na efektywność algorytmów genetycznych.

Algorytmy są opisem metody, sposobem postępowania prowadzącym do rozwiązania danego zadania.
Przykładem takich metod są algorytmy genetyczne. Wykorzystują one wiedzę z dziedziny genetyki dotyczącą:

- procesów adaptacyjnych;
- metod doboru naturalnego;
- dziedziczenia;
- ewolucji.

Również terminologia używana w teorii algorytmów genetycznych pochodzi z tej dziedziny nauki. Algorytmy genetyczne są losowymi algorytmami przeszukiwania i optymalizacji. Ich zadaniem jest wybór najlepszych elementów, należących do przestrzeni rozwiązań danego zadania według założonych kryteriów.

Tradycyjne algorytmy przeszukują daną przestrzeń rozwiązań analizując po kolei poszczególne jej elementy, albo korzystają z metod analitycznych, które z kolei są uzależnione od spełnienia pewnych warunków (np. istnienia pochodnych, ciągłości funkcji itp.). Metody te są też szczególnie wrażliwe na rozmiar przestrzeni rozwiązań i w wielu przypadkach są nieefektywne z tego powodu.

W przeciwieństwie do metod tradycyjnych algorytmy genetyczne nie operują na pojedynczych elementach przestrzeni rozwiązań lecz na ich podzbiorach – populacjach. Również udało się wyeliminować wiele ograniczeń dzięki kodowaniu potencjalnych rozwiązań i przeprowadzaniu operacji na tak zakodowanych elementach. Rolę populacji pełnią przedstawiciele przestrzeni rozwiązań natomiast środowisko oceniające osobniki jest zastąpione przez funkcję dopasowania. Jej zadaniem jest określenie stopnia dopasowania każdego osobnika w populacji. Osobniki najlepsze mają wyższe oceny i większe prawdopodobieństwo udziału w tworzeniu nowej populacji. Metoda ta nie determinuje słabszych osobników, które mogą być użyte w procesie reprodukcji (z małym prawdopodobieństwem), a które mogą być etapem przejściowym do utworzenia optymalnego rozwiązania.

Algorytm genetyczny dla każdego zadania musi zawierać następujące elementy:

- podstawową reprezentację potencjalnych rozwiązań,
- sposób tworzenia początkowej populacji,

- funkcję dopasowania, która gra rolę środowiska i ocenia rozwiązania według ich dopasowania,
- operatory genetyczne,
- wartości parametrów używanych w algorytmach genetycznych (rozmiar populacji, prawdopodobieństwo użycia operatorów genetycznych itp.),
- warunkach zakończenia algorytmu.

Przykład. Algorytm szukający maksimum funkcji $y = x^2$ w przedziale $< 0, 31 >$. Do reprezentacji potencjalnych rozwiązań można użyć kodu 5 - cio bitowego umożliwiającego operacje na liczbach całkowitych z przedziału od 0(00000) do 31(11111), np. 23 = 10111. Kod ten jest zapisem liczby w systemie dwójkowym.

Funkcję oceniającą będzie stanowiła funkcja, której maksimum szukamy. W programie użyjemy dwóch podstawowych operatorów genetycznych: krzyżowania i mutacji odpowiednio z prawdopodobieństwem 0,25 i 0,01.

Struktura programu genetycznego przedstawia się następująco:

```
program Program_Ewolucyjny ;
begin
    t = 0;
    Populacja (t);                          - ustalenie populacji początkowej
    Ocena (Populacja (t));                  - ocena populacji
    while (not Warunek_Zakończenia) do
    begin
        t = t + 1;
        Populacja (t)= Wybór (Populacja (t - 1);   - reprodukcja
        Zmień (Populacja (t));              - operacje krzyżowania i mutacji
        Ocena (Populacja (t));              - ocena populacji
    end;
end.
```

Pierwszym krokiem algorytmu jest utworzenie nowej populacji. Zazwyczaj jest ona tworzona w sposób losowy, jednak można użyć w tym celu innej znanej metody rozwiązania danego problemu.

Algorytmy genetyczne wymagają aby chromosomy były kodowane za pomocą łańcuchów symboli skończonej długości z pewnego określonego, skończonego alfabetu. Najczęściej przyjmuje się notację binarną. Jest ona łatwa do analizy oraz są dla niej określone różnorodne operacje genetyczne

Nr ciągu	Populacja początkowa	Wartość x	Funkcja dopasowania $f(x_i) = x_{i2}$	Prawdopod. wyboru każdego chromosomu $p_i = f(x_i)/F$	Dystrybuanta $q_i = \sum_{j=1}^{n} p_j$
1	01001	21	441	0,47	0,47
2	01110	14	196	0,21	0,68
3	00101	5	25	0,03	0,71
4	01001	9	81	0,09	1,0
			$F = \sum f(x)$ 743 $F_{sr} =$ 185,75	n–liczeb. popul.	

Procedura *Ocena* określa stopień dopasowania poszczególnych chromosomów w populacji. Następne kroki algorytmu są wielokrotnie powtarzane aż do jego zbiegnięcia się. Należą do nich:

- wybór nowej populacji z poprzedniego pokolenia (procedura *Wybór*). Proces reprodukcji oparty jest na „obrocie koła ruletki" n razy i wyborze za każdym razem jednego chromosomu. Przebiega on następująco:
  - generujemy liczbę losową z przedziału $(0, 1)$
  - jeżeli jest ona mniejsza od (dystrybuanty) g_1 to wybieramy pierwszy chromosom, jeżeli nie to chromosom x_i dla którego wygenerowania liczba jest w przedziale $(g_{i-1}, g_i >$. Jeżeli wygenerowano liczby $0,97; 0,65; 0,21; 0,43$ to do nowej populacji wybrano osobniki: $x_1' = x_4, x_2' = x_2, x_3' = x_1, x_4' = x_1$
- przetworzenie populacji z wykorzystaniem operatorów genetycznych (*Zmień*). Dwie podstawowe operacje to krzyżowanie i mutacja. Krzyżowanie polega na zamianie części kodów pomiędzy (dwoma lub więcej) osobnikami np. dla osobników a i b:

$$a = a_1 a_2 \ldots a_k a_{k+1} \ldots a_n$$

$$b = b_1 b_2 \ldots b_k b_{k+1} \ldots b_n$$

jeżeli punkt krzyżowania wypadnie po k–tym genie to otrzymamy dwa nowe chromosomy a', b':

$$a' = a_1 a_2 \ldots a_k b_{k+1} \ldots b_n$$

$$b' = b_1 b_2 \ldots b_k a_{k+1} \ldots a_n$$

będące w bliskim otoczeniu chromosomów wyjściowych.

W przypadku krzyżowania dla każdego chromosomu losujemy liczbę losową z przedziału $(0, 1)$ i jeżeli jest ona mniejsza od prawdopodobieństwa krzyżowania (p_c) to wybieramy rozpatrywany chromosom do krzyżowania. Np. wylosowano następujące liczby losowe: 0,81; 0,24; 0,30; 0,11. Do krzyżowania wybrano x'_2 i x'_4. Punkt krzyżowania jest losową liczbą z przedziału $(1, n)$ np. 2. Wówczas z pary:

01|110 otrzymamy 01101

10|101 10110

Natomiast mutacja zmienia wartość pojedynczego genu w chromosomie. Operacja ta wprowadza pewną dodatkową zmienność w aktualnej populacji. Wielkości określające numery osobników wytypowanych do reprodukcji, krzyżowania, punkt krzyżowania, numer genu do mutacji są losowe.
Z operacją mutacji postępujemy analogicznie tzn. dla każdego genu losujemy liczbę z przedziału $(0, 1)$ i jeżeli jest mniejsza od prawdopodobieństwa mutacji (p_m) to zmieniamy wartość genu na przeciwny. Przeważnie mutacja odgrywa drugorzędną rolę w algorytmach gentycznych stąd małe prawdopodobieństwo jej użycia.

- ocena populacji – nadanie osobnikom wartości funkcji dopasowania (*Ocena*).

Nr ciągu	Populacja początkowa	Wartość x	Funkcja dopasowania $f(x_i) = x_{i2}$	Prawdopod. wyboru każdego chromosomu $p_i = f(x_i)/F$	Dystrybuanta $q_i = \sum_{j=1}^{n} p_j$
1	01001	9	81	0,06	0,06
2	01101	13	169	0,12	0,18
3	10101	21	441	0,3	0,48
4	10110	22	484	0,33	1,0
			$F = \sum f(x)$ 1175 $F_{sr} =$ 293,75		

Istota działania programów genetycznych polega na tworzeniu coraz lepszych zbiorów rozwiązań (populacja podlega stymulowanej ewolucji), a co za tym idzie osobników reprezentujących rozwiązania leżące blisko optymalnych. Każda następna populacja tworzona jest z przedstawicieli poprzedniego pokolenia z preferencją najlepszych osobników. Część osobników podlega jeszcze dodatkowym przekształceniom zwanym krzyżowaniem i mutacją. Po pewnym czasie populacja zbiega się w okolice szukanego rozwiązania.

Tradycyjne algorytmy genetyczne wymagają modyfikacji problemu do postaci odpowiedniej dla nich. Oznacza to wybór reprezentacji binarnej, który czasami może okazać się bardzo skomplikowany, gdyż elementy przestrzeni rozwiązań mogą być w naturalny sposób reprezentowane przez bardziej skomplikowane struktury danych, takie jak macierze, drzewa, grafy. Również tradycyjne algorytmy genetyczne nie zdają egzaminu w wielu zagadnieniach praktycznych a główną tego przyczyną jest niezależność od rozpatrywanej dziedziny. Dlatego w wielu przypadkach stosuje się zmodyfikowane algorytmy genetyczne zwane programami ewolucyjnymi.

Programy ewolucyjne wymagają reprezentacji chromosomów potencjalnych rozwiązań przy użyciu „naturalnych" struktur danych i zastosowania odpowiednich operatorów genetycznych. W wyniku takiego postępowania programy ewolucyjne korzystają z wbudowanej w struktury danych chromosomów wiedzy specyficznej dla zadania. Taka naturalna reprezentacja nie powoduje modyfikacji przestrzeni rozwiązań, która w wyniku kodowania mogłaby nie odpowiadać rzeczywistej przedstrzeni.

Przykład. Zagadnienie komiwojażera: Hipotetyczny komiwojażer ma za zadanie objechać wszystkie miasta z określonego zbioru, tak by zminimalizować przebytą drogę.

Podstawowym problemem przy tworzeniu algorytmów genetycznych jest dobór kodu chromosomu oraz operacji genetycznych. Powinny one uwzględniać specyfikę oraz ograniczenia nałożone na zadanie.

Zadanie to należy do zadań NP – trudnych, które nie dają się rozwiązć w czasie wielomianowym. Idealne rozwiązanie można otrzymać jedynie poprzez zbadanie wszystkich możliwych tras a ich liczba wynosi $(N-1)!$, gdzie N – liczba miast. Dla odpowiednio dużych wartości N ten sposób jest zdecydowanie nieefektowny.

W rozpatrywanym zagadnieniu można zastosować różne kody chromosomów opracowując do niech odpowiednie operacje genetyczne. W reprezentacji przyległościowej trasa jest opisywana jako ciąg miast. Miasto a znajduje się na pozycji b wtedy i tylko wtedy, gdy z miasta a prowadzi trasa do miasta b. Na przykład wektor (2 4 1 5 3) przedstawia trasę $1-2-4-5-3$. Na pierwszej pozycji mamy 2, czyli $1-2$, druga pozycja wskazuje na 4 $(1-2-4)$, na 4 pozycji jest 5 $(1-2-4-5)$, która z kolei wskazuje na 3 a na 3 pozycji jest 1 która kończy ciąg. Każdy chromosom przedstawia jedną trasę przyległościową, mogą jednak wystąpić trasy niedopuszczalne (trasy niepełne). W podanej reprezentacji nie można użyć klasycznych operacji.

W wielu przypadkach najlepsze wyniki otrzymuje się przy użyciu struktur, które w wyniku ewolucji tworzą niedopuszczalne rozwiązania. W takim przypadku ograniczenia, których nie wolno przekroczyć można uwzględnić

przez nakładania wysokich lub umiarkowanych kar na osobniki, które je naruszają lub też przez utworzenie mechanizmów naprawczych. Kara polega na zmniejszeniu wartości funkcji dopasowania o pewną wartość, przez co maleje szansa powielania niedopuszczalnych rozwiązań. Wysokie kary mogą być przyczyną preferowania słabych ale dopuszczalnych osobników w przypadku, gdy utworzenie lepszych rozwiązań wymaga skorzystania ze struktur przejściowych składających się z osobników nieprawdłowych. Natomiast umiarkowane kary mogą powodować rozwinięcie się osobników naruszających ograniczenia, ale ocenianych lepiej od tych, które ich nie naruszają, ponieważ główna część funkcji oceniającej może mieć przy umiarkowanej karze decydujące znaczenie. Jeżeli zaś będziemy korzystali z mechanizmów naprawy to może się okazać, że proces naprawy osobników niedopuszczalnych może być tak samo trudny, jak rozwiązanie początkowego zadania. Możliwa jest też strategia wykluczania osobników niedopuszczalnych. Jednak ma ona zastosowanie w przypadkach, gdy przestrzeń rozwiązań dopuszczalnych jest dostatecznie dużą częścią całej przeszukiwanej przestrzeni.

Funkcja dopasowania powinna pozwolić połączyć wymaganie zbieżności z koniecznością zbadania jak największego obszaru przestrzeni rozwiązań. Przy tworzeniu funkcji dopasowania należy często uwzględnić bardzo wiele ograniczeń określających dopuszczalne rozwiązania.

Parametry najczęściej używane w algorytmach genetycznych służą do określenia wielkości populacji, prawdopodobieństwa użycia operatorów genetycznych czy ilość tworzonych populacji (czas działania algorytmu). Powinny być one dopasowane do konkretnego zadania. Warunki zakończenia algorytmu określa się poprzez liczbę powstałych populacji lub też braku istotnych zmian informujących o zbieżności algorytmu.

O zbieżności populacji decyduje przede wszystkim podobieństwo w strukturze dobrze przystosowanych osobników. Pojęcie schematu jest kluczowym zagadnieniem w algorytmach genetycznych. Jest to wzorzec opisujący podzbiór ciągów podobnych ze względu na ustalone pozycje. Otrzymujemy go poprzez wprowadzenie do alfabetu dodatkowego uniwersalnego symbolu *. Na przykład w alfabecie dwójkowym $\{0, 1\}$ dodanie symbolu * umożliwi określenie ciągów, w których na odpowiednich miejscach jedynce odpowiada jedynka, zero zeru natomiast * dowolny z tych symboli. Dla ciągu *111 mamy podzbiór złożony z dwóch ciągów $\{0111, 1111\}$, natomiast schemat **11 określa podzbiór $\{00111, 0111, 1011, 1111\}$. W podanym przykładzie należy rozpatrywać 3^4 schematów (ogólnie $(n+1)^k$ schematów, gdzie n – ilość symboli w alfabecie; k – długość ciągu) pomimo, że istnieje 2^4 schematów. Związane to jest z faktem, że wykorzystuje się informację zawartą w populacji jako całości a nie w jej poszczególnych osobnikach. In-

formacja ta jest uwzględniana przy tworzeniu nowej populacji powodując stały wzrost najlepszych reprezentantów. Nowe elementy może wprowadzić do populacji przede wszystkim operacja krzyżowania. Schemat pozostaje nienaruszony, jeżeli operacja krzyżowania nie doprowadzi do jego przecięcia, w przeciwnym razie może on ulec zniszczeniu. Szczególnie narażone na zniszczenie są schematy, w których ustalone pozycje znajdują się w znacznych odległościach od siebie (np. 1*****0). Natomiast bardzo trudno „rozerwać" schematy o niewielkiej liczbie ustalonych pozycji w dodatku leżących blisko siebie. Wynika stąd, że krzyżowanie nie narusza schematów o małej „rozpiętości", a reprodukcja zapewnia ich dobre tempo propagacji. Natomiast mutacja o niedużym natężeniu rzadko powoduje zniszczenie konkretnego schematu. Wynika z tego, że schematy o wysokim przystosowaniu i małej rozpiętości rozpowszechniają się z pokolenia na pokolenie w rosnących wykładniczo porcjach.

Rzędem schematu H, który oznaczamy przez $o(H)$, nazywa się liczbę ustalonych pozycji we wzorcu (np. dla 011***11* rząd wynosi 5).

Rozpiętość schematu ($\delta(H)$) jest to odległość pomiędzy dwiema skrajnymi ustalonymi pozycjami (np. dla 011***11** rozpiętość wynosi 7).

Jeżeli w chwili t w populacji $P(t)$ znajdzie się $m = m(H,t)$ reprezentantów danego schematu H i ciągi podczas reprodukcji podlegają replikacji z prawdopodobieństwem proporcjonalnym do wskaźnika przystosowania ($p_i = \frac{f_i}{\Sigma f_j}$ gdzie f_i wartość funkcji dopasowania i – tego osobnika). Po utworzeniu nowej populacji możemy oczekiwać $m(H,t+1)$ reprezentantów schematu H, przy czym zachodzi wzór:

$$m(H,t+1) = m(H,t) \cdot n \cdot \frac{f(H)}{\Sigma f_j}$$

gdzie $f(H)$ określa średnie przystosowanie ciągów będących reprezentantami schematu H w chwili t. Średnie przystosowanie całej populacji można wyrazić jako $f^* = \frac{\Sigma f_j}{n}$ a powyższy związek można zapisać w postaci:

$$m(H,t+1) = m(H,t) \cdot \frac{f(H)}{f^*}$$

Oznacza to, że schematy o przystosowaniu wyższym niż średnie z populacji będą miały większą liczbę reprezentantów w następnym pokoleniu, podczas gdy schematy o przystosowaniu niższym niż średnie otrzymają mniej reprezentantów niż dotychczas. Wszystkie schematy w populacji rozprzestrzeniają się lub znikają odpowiednio do swego średniego przystosowania.

Jeżeli jakiś schemat H przewyższa średnią o wielkość $c \cdot f^*$, gdzie c – stała to:

$$m(H, t+1) = m(H, t) \cdot \frac{(f^* + c \cdot f^*)}{f^*} = m(H, t) \cdot (1 + c)$$

Wychodząc z populacji $t = 0$ otrzymujemy zależność:

$$m(H, t) = m(H, 0) \cdot (1 + c)^t$$

Algorytm genetyczny stosujący jedynie reprodukcję najczęściej osiągałby lokalne ekstremum kończąc w ten sposób przeszukiwanie przestrzeni rozwiązań. Użycie operatorów genetycznych umożliwia zbadanie chromosomów poza aktualnie penetrowanym obszarem wokół ekstremum lokalnego. Prawdopodobieństwo zniszczenia schematu po operacji krzyżowania wynosi $p_d = \frac{\delta(H)}{(l-1)}$ gdzie l – długość ciągu), a przeżycia $p_s \geq 1 - \frac{\delta(H)}{(l-1)} \cdot p_c$, gdzie p_c – prawdopodobieństwo krzyżowania. Natomiast w przypadku mutacji otrzymujemy $(l - p_m)^{o(H)}$ gdzie $o(H)$ oznacza liczbę pozycji ustalonych, p_m prawdopodobieństwo mutacji (mutacje na poszczególnych pozycjach są statystycznie niezależne). Przy $p_m << 1$ powyższy wzór przybiera postać: $1 - o(H) \cdot p_m$.
Uwzględniając wpływ operacji genetycznych w ogólnym wzorze otrzymamy nierówność:

$$m(H, t+l) \geq m(H, t) \cdot \frac{f(H)}{f^*} \cdot [1 - \frac{\delta(H)}{(l-1)} \cdot p_c - o(H) \cdot p_m]$$

Analiza powyższej nierówności pozwala sformułować podstawowe twierdzenie algorytmów genetycznych – twierdzenie o schematach.

Twierdzenie. *Krótkie, niskiego rzędu i oceniane powyżej średniej schematy uzyskują wykładniczo rosnącą liczbę łańcuchów w kolejnych pokoleniach.*

Bibliografia

[1] Cytowski J. *Algorytmy genetyczne. Podstawy i zastosowanie*, Akademicka Oficyna Wydawnicza, Warszawa, 1996

[2] Goldberg D.E. *Algorytmy genetyczne i ich zastosowania*, Wydawnictwo Naukowo - Techniczne, Warszawa, 1995

[3] Michalewski Z. *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, Wydawnictwo Naukowo - Techniczne, Warszawa, 1996