

Algorytmy ewolucyjne z AB-mutacją

Kazimierz Grygiel

Instytut Informatyki

Uniwersytetu Warszawskiego

email: grygiel@mimuw.edu.pl

Streszczenie

W artykule zaproponowano nowy rodzaj mutacji jako alternatywę dla “zwykłej” mutacji binarnej w połączeniu z kodem Graya, stosowanych tradycyjnie w algorytmach genetycznych do zadań optymalizacji parametrycznej. Wyprowadzono pesymistyczne oszacowanie czasu oczekiwania na znalezienie najlepszego przybliżenia maksimum globalnego dla klasy jednomodalnych funkcji ciągłych na odcinku $[0, 1]$. Przeprowadzona analiza sugeruje, że wbrew rozpowszechnionemu pogładowi skuteczność mutacji zależy w istotnym stopniu od jej zdolności do “oddziaływania na odległość”.

1 Wprowadzenie

Jednym z ważnych zastosowań algorytmów ewolucyjnych jest *optymalizacja parametryczna*. W ogólnym przypadku zadanie optymalizacji parametrycznej postawione jest następująco:

Dla zadanego zbioru $S \subset R^k$ i (skończonej) funkcji $f : S \rightarrow R$ znaleźć optimum (maksimum lub minimum) globalne f^ funkcji f na S .*

Zbiór S będziemy nazywali *zbiorem rozwiązań* albo *zbiorem poszukiwań*, funkcję f *funkcją celu*, a dowolny punkt $x^* \in S$ taki, że $f(x^*) = f^*$ — *rozwiązaniem optymalnym*.

W praktyce wystarcza wyznaczenie poszukiwanego optimum z dowolną zadaną dokładnością. Jeśli f jest funkcją lipschitzowską (ze znaną stałą Lipschitza), a S zbiorem zwartym, zadanie to można sprowadzić do wyznaczenia optimum globalnego f_ϵ^* funkcji f na skończonej ϵ -sieci S_ϵ pokrywającej dziedzinę S ¹ ([4]). Tak zdyskretyzowane zadanie można oczywiście rozwiązać w skończonym czasie, przeglądając po kolei wszystkie węzły skonstruowanej sieci i zapamiętując najlepszą (najmniejszą lub największą) znalezioną wartość funkcji. Podejście takie nie zawsze jednak jest opłacalne, a nawet praktycznie wykonalne, gdyż liczba węzłów sieci może być olbrzymia. Alternatywą może być zastosowanie algorytmu ewolucyjnego z dyskretną przestrzenią poszukiwań reprezentującą ϵ -sieć, tj. w istocie pewnego wariantu *algorytmu genetycznego* [2].

¹ Tj. na skończonym podzbiorze $S_\epsilon \subset S$ o tej własności, że dla każdego punktu $x \in S$ istnieje węzeł sieci $x' \in S_\epsilon$ taki, że $\|x - x'\| < \epsilon$.

Wstępna faza konstrukcji algorytmu genetycznego polega na wyborze metody “kodowania”, co jest równoważne dyskretyzacji zbioru rozwiązań. Procedura dyskretyzacji stosowana w “kanonicznych” algorytmach genetycznych polega na zastąpieniu współrzędnych rozwiązania ich skończoną reprezentacją binarną. Można w ten sposób otrzymywać ε -sieci dla pewnych szczególnych wartości parametru ε , ściśle związanych z *rozmiarem reprezentacji* l . Ponieważ parametrem algorytmu genetycznego jest rozmiar reprezentacji, a nie dokładność, zdyskretyzowany zbiór rozwiązań będziemy dalej oznaczać przez S_l .

Ogólny schemat algorytmu przedstawia się następująco:

Krok początkowy: Pobieramy N -elementową próbę losową X_0 (zwaną *populacją*) ze zbioru S_l , gdzie $N \geq 1$ jest parametrem algorytmu.

Krok iteracyjny: Jeśli bieżąca populacja X_k spełnia jest *warunek zatrzymania* (takim warunkiem może być np. przekroczenie limitu liczby iteracji), to wynikiem poszukiwania jest estymator optimum globalnego f_k^* , którym jest wartość optymalna funkcji w bieżącej populacji (wariantowo: we wszystkich dotychczasowych populacjach łącznie). W przeciwnym przypadku dokonujemy *transformacji ewolucyjnej* populacji X_k otrzymując nową populację X_{k+1} elementów S_l i wykonujemy ponownie krok iteracyjny.

Transformacja ewolucyjna, oparta na pewnych mechanizmach losowych, łączy w sobie aspekt *eksploracji* (wytwarzanie nowych elementów poprzez modyfikację aktualnych) oraz *selekcji* (zachowywanie lepszych rozwiązań, odrzucanie gorszych). Podstawowymi mechanizmami eksploracji są w algorytmach ewolucyjnych rozmaite warianty operacji *mutacji* i *rekombinacji*. Wśród badaczy utrzymują się znaczne różnice poglądów co do funkcji spełnianej przez oba te mechanizmy w procesie poszukiwania ewolucyjnego [1, 2]. Wyniki przedstawione w tym artykule rzucają, w przekonaniu autora, nowe światło na tę starą kontrowersję.

Tradycyjnie stosowaną w algorytmach genetycznych operacją mutacji jest *mutacja binarna*. Wiadomo jednak, że mutacja binarna (zwana dalej *B-mutacją*) w powiązaniu z “naturalnym” kodem pozycyjnym jest mało skuteczna w zadaniach optymalizacji parametrycznej (problem tzw. klifów Hamminga), dlatego też w praktyce zamiast kodu pozycyjnego zaleca się stosowanie kodu Graya. Ten ostatni znacznie jednak komplikuje analizę zachowania algorytmu i wiedza na ten temat ogranicza się w zasadzie do wyników empirycznych.

Alternatywą dla kodu Graya może być użycie innego mechanizmu mutacji, wykorzystującego “naturalną” strukturę dziedziny optymalizacji. W niniejszej pracy proponuje się nową operację *arytmetycznej mutacji binarnej* (w skrócie: *AB-mutacji*), uwzględniającą naturalne uporządkowanie odcinka. Operacja ta może być w szczególności włączona do “kanonicznego” algorytmu genetycznego, zastępując zwykłą B-mutację.

Wstępne testy algorytmów ewolucyjnych z AB-mutacją (dla dość szerokiego zestawu funkcji celu opisanych w literaturze) dały wyniki porównywalne z rezultatami otrzymywanymi przy użyciu algorytmów w których zastosowano B-mutację w powiązaniu z kodem Graya. Jako miernik wydajności algorytmu przyjęto przy tym średni czas oczekiwania (mierzony liczbą ewaluacji funkcji celu) na znalezienie rozwiązania optymalnego dla danego rozmiaru reprezentacji l .

Efektywne oszacowanie wydajności algorytmu ewolucyjnego dla dostatecznie ogólnej klasy zadań optymalizacyjnych nie wydaje się możliwe przy użyciu aktualnie dostępnych metod i narzędzi teoretycznych. Jak zatem objaśnić stwierdzoną empirycznie względną

skuteczność AB-mutacji? Wydaje się, że przynajmniej częściowo satysfakcjonujące wyjaśnienie można otrzymać analizując zachowanie jak najprostszego wariantu algorytmu dla przypadku “łatwej” z punktu widzenia optymalizacji klasy funkcji. Ponieważ algorytmy ewolucyjne charakteryzują się z natury rzeczy (jako algorytmy stochastyczne) dużą odpornością, więc można oczekiwać, że wraz ze wzrostem “trudności” zadania będzie następować “łagodna degradacja” wydajności.

W niniejszej pracy za “wzorcowe” zadanie optymalizacji przyjęto zadanie maksymalizacji jednomodalnej funkcji ciągłej jednej zmiennej na odcinku $[0, 1]$. “Jednomodalność” oznacza, że funkcja przybiera maksimum globalne w dokładnie jednym punkcie i nie ma poza tym maksimów lokalnych. Funkcje takie można optymalizować przy użyciu algorytmów lokalnych typu “wspinaczkowego”.

W dalszej części artykułu omówiono kolejno metodę dyskretyzacji binarnej, operacje B-mutacji i AB-mutacji oraz przedstawiono elementarny algorytm ewolucyjny z AB-mutacją i wyprowadzono pesymistyczne oszacowanie jego wydajności dla określonego wyżej zadania optymalizacji.

2 Reprezentacja rozwiązań

Niech $S = [0, 1]$. Dla zadanego rozmiaru reprezentacji l każda liczba x z przedziału $[0, 1)$ zostaje odwzorowana na sekwencję kodową złożoną z pierwszych l cyfr rozwinięcia dwójkowego x (po przecinku). (Prawy koniec odcinka, tj. liczba 1 nie ma swojego odpowiednika w tej reprezentacji). Każda sekwencja kodowa reprezentuje zatem pewien przedział (lewostronnie domknięty i prawostronnie otwarty) o długości 2^{-l} ; zbiór lewych końców wszystkich takich przedziałów tworzy sieć S_l o skoku 2^{-l} . Dokładniej, niech

$$\Omega_l = \{(g_0, g_1, \dots, g_{l-1}) : g_i \in \{0, 1\}, 0 \leq i \leq l-1\}$$

będzie zbiorem binarnych sekwencji kodowych długości l . Każdej sekwencji $g = (g_0, g_1, \dots, g_{l-1})$ odpowiada wówczas liczba rzeczywista

$$x(g) = \sum_{i=0}^{l-1} g_i 2^{-i-1}$$

z przedziału $[0, 1)$, będąca węzłem sieci S_l . W dalszym ciągu tam, gdzie to będzie wygodne będziemy utożsamiać sekwencję kodową g z węzłem $x(g)$, a zbiór sekwencji kodowych Ω_l ze zbiorem węzłów sieci S_l .

3 AB-mutacja

Mutacja jest operacją jednoargumentową, która polega na wprowadzeniu losowego zaburzenia do danego elementu zbioru poszukiwań. “Zwyczajna” mutacja binarna (B-mutacja) charakteryzuje się punktowym prawdopodobieństwem mutacji μ i może być opisana za pomocą pojęcia *maski mutacyjnej* [3]. Operacja B-mutacji przekształca sekwencję kodową

$g \in \Omega_l$ na losową sekwencję kodową $\xi = g \oplus M$, gdzie działanie \oplus oznacza “wektorowe” dodawanie modulo 2, a M jest losową maską mutacyjną, przybierającą wartość $m \in \Omega_l$ z prawdopodobieństwem $p_m = \mu^{|m|/l}(1 - \mu)^{l - |m|/l}$, przy czym $|m|$ oznacza sumę współrzędnych wektora m , tj. liczbę pozycji w m zawierających jedynkę. Maską mutacyjną zawierającą k jedynek zmienia zatem k pozycji sekwencji kodowej, na którą zostaje “nałożona”. W przypadku $\mu = 1/2$ wszystkie maski mutacyjne mają jednakowe prawdopodobieństwo; jeśli natomiast $0 < \mu < 1/2$, to prawdopodobieństwo maski maleje wraz ze wzrostem liczby jedynek.

Nietrudno zauważyć, że przy kodowaniu pozycyjnym punkty bliskie w sensie odległości na prostej mogą być odległe w sensie metryki Hamminga, tj. odległości w przestrzeni sekwencji kodowych, w wyniku czego w procesie poszukiwania ewolucyjnego powstają lokalne “pułapki”, z których trudno się wydostać przy użyciu samej B-mutacji. Wskazana wyżej anomalia nie występuje w przypadku arytmetycznej mutacji binarnej. Istota pomysłu sprowadza się do tego, by zamiast “nakładać” maskę mutacyjną na sekwencję kodową — dodawać jej wartość liczbową (ze znakiem dodatnim lub ujemnym) do odpowiedniego węzła sieci. Aby uniknąć przekroczenia zakresu, wystarczy przy tym “skleić” oba końce odcinka, co odpowiada dodawaniu modulo 1. Dokładniej, operacja AB-mutacji przekształca węzeł x na węzeł losowy

$$\xi = x \pm M \pmod{1},$$

gdzie maska mutacyjna M jest określona jak wyżej, przy czym znak (+ lub −) jest wybierany niezależnie od M z prawdopodobieństwem 1/2. Efektywna realizacja algorytmiczna opisaney operacji nie przedstawia większych trudności, przynajmniej w typowym zakresie długości reprezentacji. W przypadku optymalizacji funkcji wielu zmiennych operację AB-mutacji należy stosować osobno do każdego argumentu.

Kluczowa własność AB-mutacji wynika z następującego lematu:

Lemat 1: Niech $x, y \in S_l$, $x < y$. Wówczas istnieją maski mutacyjne $m', m'' \in S_l$ takie, że $|m'| = |m''| = 1$ oraz $y - x' < (y - x)/2$ i $x'' - x < (y - x)/2$, gdzie $x' = x + m'$, $x'' = x - m''$.

Prosty dowód lematu pomijamy. ■

Intuicyjnie, sens powyższego lematu można streścić następująco: dla każdych dwóch węzłów $x, y \in S_l$ istnieje maska mutacyjna składająca się z $l - 1$ zer i jedynki, która po “zastosowaniu” do x (dodaniu lub odjęciu) tworzy mutantą (x' lub x''), którego odległość od y nie przekracza połowy odległości x i y . Ponieważ dla odpowiednio dobranego parametru μ maski tej postaci wytwarzane są stosunkowo wydajnie, oznacza to, że AB-mutacja ma potencjalnie zdolność szybkiej (długozasięgowej) redukcji dystansu od węzła docelowego. Skuteczność tego mechanizmu w procesie poszukiwania ewolucyjnego uzależniona jest jednak od jego współdziałania z mechanizmem selekcji, na co z kolei decydujący wpływ ma postać funkcji celu. Można się spodziewać, że najlepsze wyniki (pomijając trywialny przypadek funkcji stałych) powinno się otrzymywać dla funkcji monotonicznych, gdyż wówczas każdy punkt pomiędzy węzłem bieżącym a węzłem docelowym jest “lepsz” niż węzeł bieżący. Niżej wykażemy, że podobne zachowanie uzyskuje się dla zadania maksymalizacji funkcji jednomodalnych.

4 Algorytm ewolucyjny

Niech $f : [0, 1] \rightarrow R$ będzie daną funkcją celu oraz l — rozmiarem reprezentacji binarnej dla zdyskretyzowanej wersji zadania maksymalizacji. Rozważmy następujący schemat iteracyjny:

Wybierz losowy punkt X_0 z S_l ;

$$X_{n+1} = \begin{cases} \xi_n & \text{jeżeli } f(\xi_n) > f(X_n) \\ X_n & \text{w przeciwnym przypadku} \end{cases}$$

gdzie ξ_n oznacza mutantą sekwencji X_n , utworzonego za pomocą operacji AB-mutacji z punktowym prawdopodobieństwem mutacji μ .

Schemat powyższy odpowiada algorytmowi ewolucyjnemu z populacją jednoelementową i selekcją “twardą” (bieżące rozwiązanie zostaje zastąpione nowym tylko w przypadku, gdy to ostatnie daje większą wartość funkcji celu). Jediną operacją “eksploracji” jest tu AB-mutacja.

5 Analiza

Klasę funkcji ciągłych jednomodalnych na odcinku $[0, 1]$ oznaczajmy dla skrótów przez F_1 . Dla $f \in F_1$ niech $f_l^* = \max\{f(x) : x \in S_l\}$. Wielkość ta jest najlepszym przybliżeniem maksimum globalnego funkcji f dla reprezentacji o rozmiarze l . Niech $W_l \subset S_l$ będzie zbiorem rozwiązań odpowiadających najlepszemu przybliżeniu f_l^* .

Nietrudno zauważyć, że jeśli $f \in F_1$, to f jest (ściśle) rosnąca na odcinku $[0, x^*]$ i (ściśle) malejąca na odcinku $[x^*, 1]$. Zbiór W_l składa się zatem z jednego lub dwóch punktów; jeśli istnieją dwa, to ich odległość jest równa 2^{-l} i pomiędzy nimi nie znajduje się żaden punkt ze zbioru S_l . Wprowadzimy oznaczenia: $x_l^- = \min\{x : x \in W_l\}$ oraz $x_l^+ = \max\{x : x \in W_l\}$.

Jak łatwo stwierdzić, ciąg zmiennych losowych $\{X_n, n \geq 0\}$ zdefiniowany w poprzednim paragrafie tworzy pewien skończony łańcuch Markowa. Można bez trudu wykazać, że dla $0 < \mu < 1$ jest to łańcuch pochłaniający, w którym zbiorem stanów pochłaniających jest W_l . Wynika stąd, że ciąg $\{f(X_n), n \geq 0\}$ jest z prawdopodobieństwem 1 zbieżny do f_l^* w skończonym czasie.

Przejdziemy teraz do zbadania średniego czasu pochłonięcia $\tau_l = E\{T_l\}$, gdzie $T_l = \min\{k \geq 0 : X_k \in W_l\}$.

Powiemy, że w chwili $n > 0$ nastąpiła *maksymalna redukcja lewostronna* jeżeli $X_{n-1} \leq X_n \leq x_l^-$ oraz $x_l^- - X_n \leq (x_l^- - X_{n-1})/2$. Analogicznie, powiemy, że w chwili n nastąpiła *maksymalna redukcja prawostronna* jeżeli $x_l^+ \leq X_n \leq X_{n-1}$ oraz $X_n - x_l^+ \leq (X_{n-1} - x_l^+)/2$.

Lemat 2: Dla dowolnej funkcji $f \in F_1$ prawdopodobieństwo zajścia maksymalnej redukcji lewostronnej lub prawostronnej w chwili $n \geq 1$ jest nie mniejsze niż $p_r = \mu(1 - \mu)^{l-1}/2$.

Dowód. Załóżmy, że $X_{n-1} < x_l^-$. Wówczas na mocy Lematu 1 istnieje maska mutacyjna m' składająca się z $l - 1$ zer i jedynki taka, że $X_{n-1} < (X_{n-1} + m') \leq x_l^-$ oraz $x_l^- -$

$(X_{n-1} + m') \leq (x_l^- - X_{n-1})/2$. Prawdopodobieństwo wygenerowania takiej maski jest równe $\mu(1 - \mu)^{l-1}/2$. Ponieważ $f(X_{n-1} + m') > f(X_{n-1})$, więc zgodnie z regułą selekcji punkt ten zostanie zaakceptowany jako nowe rozwiązanie X_n . Przypadek $X_{n-1} > x_l^+$ rozpatruje się analogicznie. Jeśli natomiast $X_{n-1} \in W_l$, to teza lematu spełniona jest w sposób trywialny. ■

Jako wniosek z Lematu 2 otrzymujemy następujące

Twierdzenie 1: Dla dowolnej funkcji $f \in F_1$ średni czas oczekiwania na znalezienie najlepszego przybliżenia maksimum globalnego dla reprezentacji o rozmiarze l spełnia nierówność

$$\tau_l \leq \frac{4l}{\mu(1 - \mu)^{l-1}} \quad (1)$$

Zarys dowodu. Łączna liczba maksymalnych redukcji lewo- lub prawostronnych gwarantująca osiągnięcie jednego z punktów x_l^- lub x_l^+ nie przekracza $2l$. Średni czas oczekiwania na zajście maksymalnej redukcji lewo- lub prawostronnej jest nie większy niż $1/p_r$, skąd wynika teza twierdzenia. ■

Chociaż Twierdzenie 1 daje tylko oszacowanie pesymistyczne, można je wykorzystać w celu “dostrojenia” algorytmu ewolucyjnego. Największą wartość mianownika w (1) otrzymuje się dla $\mu = 1/l$; wówczas $(1 - \mu)^{l-1} = (1 - 1/l)^{l-1} \approx e^{-1} = O(1)$ dla dostatecznie dużego l . Stąd wynika asymptotyczne oszacowanie

$$\tau_l = O(l^2) \quad (2)$$

Przy użyciu nieco bardziej zaawansowanych metod wynik ten można dalej uogólnić na przypadek separowalnych funkcji n zmiennych o składnikach z F_1 , otrzymując dla $\mu = 1/nl$ oszacowanie $O(nl^2)$.

6 Podsumowanie i wnioski

W artykule opisano operację arytmetycznej mutacji binarnej (AB-mutacji) wykorzystującą naturalne uporządkowanie odcinka. Zbadano zachowanie elementarnego algorytmu ewolucyjnego z populacją jednoelementową, arytmetyczną mutacją binarną i twardą selekcją dla zadania optymalizacji jednomodalnych funkcji ciągłych na odcinku $[0, 1]$ wykazując, że przy odpowiednio dobranym punktowym prawdopodobieństwie mutacji średni czas potrzebny do znalezienia najlepszego przybliżenia optimum jest rzędu $O(l^2)$, gdzie l jest rozmiarem reprezentacji. Z przeprowadzonej analizy wynika, że efektywność AB-mutacji jest konsekwencją jej zdolności do “długozasięgowej” penetracji przestrzeni poszukiwań, co zdaje się przeczyć rozpowszechnionemu przekonaniu o roli mutacji jako “operatora lokalnego” w procesie poszukiwania ewolucyjnego.

Bibliografia

- [1] Beyer, H.-G., Rudolph, G. (1997). Theoretical Foundations and Properties of Evolutionary Computations. Local performance measures, in: *Handbook of Evolutionary Computation*. IOP Publishing and Oxford University Press.
- [2] Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.
- [3] Vose, M. D., Wright, A. H. (1994). Simple Genetic Algorithms with Linear Fitness, *Evolutionary Computation* 2(4), 347-368.
- [4] Zieliński, R., Neumann, P. (1986). Stochastyczne metody poszukiwania minimum funkcji. WNT, Warszawa.