

Poszukiwania efektywniejszych operatorów genetycznych dla strategii ewolucyjnych

Michał Rzewuski, Maciej Szreter, Jarosław Arabas
Instytut Podstaw Elektroniki, Politechnika Warszawska
ul. Nowowiejska 15/19, 00-665 Warszawa

Streszczenie— Niniejszy artykuł przedstawia nowe koncepcje specjalizowanych operatorów genetycznych. Mają one zastosowanie w problemach optymalizacji numerycznej. Przedstawione są dwa operatory. Pierwszy z nich - operator krzyżowania adaptacyjnego - jest rozwinięciem koncepcji mutacji gradientowej; drugim operatorem jest mutacja uwzględniająca gradient funkcji celu.

I. WSTĘP

Duża skuteczność algorytmów optymalizacji idzie często w parze z silnymi założeniami dotyczącymi rozwiązywanego problemu. Można np. postulować wypukłość funkcji celu, jej różniczkowalność, Lipschitzowskość itp., dzięki czemu istnieje możliwość skonstruowania wysoko specjalizowanych algorytmów optymalizacji (por. [6, 7]). Z drugiej strony, istnieje potrzeba narzędzi ogólnego przeznaczenia, działających przy słabych założeniach (algorytmami tymi mogą być algorytmy ewolucyjne). Często jednak „klasyczne” algorytmy ewolucyjne są zbyt ogólne, skutkiem czego nie da się wykorzystać posiadanej wiedzy o problemie. Stąd cała rodzina algorytmów ewolucyjnych posiadających wbudowaną wiedzę o problemie w postaci specjalizowanych schematów kodowania i operatorów genetycznych (por. [4]).

Pewną klasą słabo specjalizowanych algorytmów ewolucyjnych są strategie ewolucyjne, stosowane w optymalizacji numerycznej bez ograniczeń [5]. Ich niesłychanie ciekawą właściwością jest samoczynna adaptacja własnych parametrów, co umożliwia lepsze dostrajanie algorytmu do instancji rozwiązywanego problemu (por. [3]).

Niniejszy komunikat dotyczy dwóch aspektów poprawy efektywności strategii ewolucyjnej. Po pierwsze, opisany jest sposób adaptacji operatora krzyżowania. Temat ten poruszany już był w [1, 2], jednak wprowadzone koncepcje bazowały na obserwacji wektora parametrów mutacji. Opisana w tym artykule adaptacja krzyżowania przebiega niezależnie od mutacji. Po drugie, pokazana jest metoda uwzględniania informacji gradientowej w mutacji.

II. ADAPTACYJNY OPERATOR KRZYŻOWANIA

A. Samoczynna adaptacja zasięgu mutacji

Strategie ewolucyjne są algorytmami w których obserwuje się samoczynną adaptację zasięgu operatora mutacji zależnie od postępu ewolucji. Na początku działania algorytmu zasięg ten jest duży, podczas zbieżności obserwuje się jego stopniową redukcję. Dzięki temu osiągnięto w strategii ewolucyjnej płynne przejście od eksploatacji obszaru zainteresowania (mającej miejsce na początku działania algorytmu) do

eksploatacji najbardziej obiecujących obszarów, prowadzącej do zbieżności do rozwiązań optymalnych lub suboptymalnych. Przejście to nie jest charakterystyczne wyłącznie dla strategii ewolucyjnej, można je na przykład spotkać w algorytmie symulowanego wyżarzania. Cechą wyróżniającą strategię ewolucyjną jest fakt, że to stopniowe zawężanie zasięgu mutacji jest zjawiskiem wtórnym w algorytmie, wywołanym specyficzną metodą kodowania i operatorem mutacji: mutacja polega na dokonaniu perturbacji zmiennych niezależnych wektorem losowym o rozkładzie normalnym z zerową wartością oczekiwaną, którego standardowe odchylenia są kodowane w chromosomie i wraz z nim podlegają mutacji. Adaptacja zasięgu mutacji jest więc efektem współgrania schematu mutacji, dokonującego zmiany zasięgu operatora, z selekcją, preferującą osobniki lepiej przystosowane, które zarazem mają odpowiedni zasięg mutacji.

Wprowadzenie operatora krzyżowania arytmetycznego zaburza naszkicowany powyżej schemat samoczynnej adaptacji. Niezależnie bowiem od zasięgu mutacji, krzyżowanie ma charakter eksploracyjny lub eksploatacyjny, zależnie od położenia osobników podlegających krzyżowaniu. Gdy są one położone blisko siebie (tzn. w obszarze przyciągania jednego minimum lokalnego), wówczas krzyżowanie jest operatorem eksploracyjnym; w przypadku krzyżowania osobników leżących w obszarach przyciągania różnych minimów, operator ten prowadzi do eksploracji. Łatwo wyobrazić sobie sytuację, w której populacja zbiega do kilku różnych oddalonych od siebie, lecz bliskich co do wartości rozwiązań suboptymalnych: zastosowanie operatora krzyżowania wprowadzać będzie silną eksplorację obszarów pomiędzy obszarami przyciągania rozwiązań suboptymalnych, podwyższając niepotrzebnie liczbę obliczeń funkcji celu.

Najprostszym rozwiązaniem tych problemów byłaby taka modyfikacja operatora krzyżowania, aby stopniowo ograniczać jego działanie do osobników odpowiednio blisko siebie położonych, w szczególności tylko do osobników należących do jednego obszaru przyciągania. Można by to zrealizować przez zadanie jednej wartości, wspólnej dla całej populacji ograniczającej zasięg krzyżowania, wymagałoby to jednak zaproponowania metody zmieniania wartości tego parametru. Można spodziewać się, że wartości te i ich sposób uzmiennienia powinny w pewien sposób zależeć od funkcji celu. Rozwiązaniem alternatywnym, usuwającym tę niedogodność, jest algorytm określający zasięg krzyżowania w czasie działania strategii ewolucyjnej w sposób analogiczny do określania zasięgu mutacji, przedstawiony poniżej.

B. Mechanizm adaptacji zasięgu krzyżowania

Osobnik został wzbogacony o dodatkowy wektor współczynników c , określających zasięg krzyżowania dla każdej zmiennej niezależnej. W celu umożliwienia samoczynnej adaptacji, podczas mutacji wprowadza się dodatkowy etap modyfikacji wartości wektora c . Tak więc operatory genetyczne zdefiniowane są następująco.

B.1. Mutacja

Mutacja przebiega trzyetapowo — mutowane są kolejno wartości wektorów σ , c i x wg wzoru:

$$\begin{aligned}\sigma'_i &= \sigma_i \exp(\tau' \cdot N(0, 1) + \tau N_i(0, 1)) \\ c'_i &= c_i + \sigma'_i N_i(0, 1)\end{aligned}$$

$$x'_i = x_i + \sigma'_i N_i(0, 1)$$

gdzie τ , τ' są parametrami algorytmu (dyskusja ich wartości np. w [5]), $N(0, 1)$, $N_i(0, 1)$ są wartościami zmiennej losowej o standaryzowanym rozkładzie normalnym, przy czym $N(0, 1)$ jest losowana jednorazowo dla całego chromosomu, zaś $N_i(0, 1)$ niezależnie dla każdego i

B.2. Krzyżowanie

Krzyżowanie pary chromosomów jest przeprowadzane z prawdopodobieństwem p_a równym

$$p_a = \left(\frac{c_i^j}{2|x_i^j - x_i^k| + \varepsilon} \right)$$

W schemacie tym wyróżnia się losowo wybrany chromosom rodzicielski (oznaczony we wzorze indeksem górnym j).

W niniejszym artykule przyjęto zmodyfikowany arytmetyczny schemat krzyżowania, polegający na obliczeniu kombinacji liniowej wektorów $\mathbf{x}$ i skopiowania wektorów σ , $\mathbf{c}$. Wynikiem krzyżowania dwojga rodziców jest jeden potomek.

$$\begin{aligned} x'_i &= \frac{1}{2}(x_i^j + x_i^k) \\ \sigma'_i &= \sigma_i^j \\ c'_i &= c_i^j \end{aligned}$$

C. Weryfikacja eksperymentalna wprowadzonej koncepcji

W celu weryfikacji wprowadzonej koncepcji krzyżowania adaptacyjnego (SEzAK), przeprowadzono eksperyment porównawczy z algorytmami z krzyżowaniem arytmetycznym (SEzK) i bez operatora krzyżowania (SEbK). Algorytmem bazowym była strategia ewolucyjna z selekcją $\mu + \lambda$. Porównania przeprowadzono dla funkcji testowych optymalizacji globalnej [6]. Warunkiem zatrzymania było osiągnięcie przez najlepszy osobnik w populacji bazowej $x_{P_t}^*$ zadawalającego poziomu wartości funkcji celu (poziom ten jest osiągany gdy zachodzi $|f(x_{P_t}^*) - f^*| < 0.001$, gdzie f^* jest znaną wartością minimum globalnego).

W Tab. 1 umieszczono wyniki optymalizacji w postaci liczby obliczeń funkcji celu. Dodatkowo w tablicy został umieszczona różnica nakładu obliczeń dla SEzAK i SEbK liczona względem SEzK.

W przypadku funkcji o niewielkiej liczbie minimów lokalnych zastosowanie algorytmu SEzK daje lepsze wyniki niż SEbK. W takich sytuacjach algorytm SEzAK ma właściwości zbliżone do SEzK. Natomiast dla funkcji o dużej liczbie minimów lokalnych, oraz wielu suboptymalnych rozwiązaniach położonych w znacznej odległości od siebie, krzyżowanie prowadzi do niepotrzebnej eksploracji nieobiecujących obszarów. Wówczas zastosowanie algorytmu SEzAK daje znaczną poprawę, zaś wyniki działania algorytmu zbliżone są do SEbK.

W kolejnym eksperymencie przeprowadzono badanie zależności nakładu obliczeń od wzrostu liczby minimów lokalnych. W tym celu wykorzystano funkcję Griewanka, w której liczba minimów lokalnych rośnie wprost proporcjonalnie do pola

Typ optym. funkcji	SEzK wyn.	SEzAK		SEbK	
		wyn.	% popr.	wyn.	% popr.
Branina	1871.0	1838.7	+1.72	2044.8	-9.28
Camel Back	1664.3	1612.5	+3.11	1771.2	-6.42
Camel Back 6	1505.6	1518.1	-0.82	1718.7	-14.15
Griewanka	1778.9	1871.4	-5.19	2469.4	-38.81
Shuberta	4353.9	3599.4	+17.33	3717.8	+14.61

Tab. 1: Porównanie liczby obliczeń wartości funkcji celu (kolumna *wyn.*) dla algorytmów SEbK, SEzAK, SEzK. W kolumnach *% popr.* podano względny stopień poprawy, odniesiony do algorytmu SEzK.

powierzchni obszaru dopuszczalnego. W tablicy 2 przedstawione są wyniki obrazujące zysk ze stosowania SEzAK w zależności od wielkości obszaru dopuszczalnego dla funkcji Griewanka ze współczynnikiem $d = 1000$. Przyjęto obszar dopuszczalny będący kwadratem symetrycznym względem początku układu współrzędnych.

obszar dopuszczalny	SEzK	SEzAK	zysk w %
$(\pm 1, \pm 1)$	572	616	-7
$(\pm 3, \pm 3)$	840	840	-2
$(\pm 6, \pm 6)$	1348	1304	+3
$(\pm 9, \pm 9)$	1828	1456	+20
$(\pm 12, \pm 12)$	1984	1692	+14
$(\pm 15, \pm 15)$	2468	1700	+31
$(\pm 18, \pm 18)$	2720	1832	+32
$(\pm 21, \pm 21)$	2972	2316	+22

Tab. 2: Korzyść z zastosowania SEzAK w stosunku do SEzK w zależności od wielkości obszaru dopuszczalnego dla funkcji Griewanka ze współczynnikiem $d = 1000$.

Zysk ze stosowania SEzAK w stosunku do SEzK wzrasta wraz z liczbą ekstremów optymalizowanej funkcji celu.

W trakcie działania algorytmu została zaobserwowana tendencja zmniejszania się zasięgu krzyżowania, podobnie jak zmniejsza się wartość współczynników mutacji σ .

Stosowanie strategii ewolucyjnej z adaptacyjnym krzyżowaniem nie wymaga radykalnych zmian w istniejących już programach, daje równocześnie potencjalnie możliwość szybszego odnajdywania minimów w przypadku optymalizowania trudnych funkcji wielomodalnych. Z obserwacji zachowania się algorytmu na różnych przykładach funkcji testowych można się spodziewać kilku- do kilkunastoprocentowego wzrostu wydajności działania programów dla większości problemów a tylko w nielicznych przypadkach niewielkich strat.

III. OPERATOR MUTACJI GRADIENTOWEJ

A. Rozważania wstępne

Jako pewien „aksjomat” w algorytmach ewolucyjnych przyjmuje się, że gradient funkcji celu nie jest wykorzystywany na żadnym etapie ewolucji. Zdarza się jed-

nak, że obliczenie gradientu funkcji celu jest możliwe, a nakład obliczeń związany z tą operacją jest porównywalny z obliczeniem wartości funkcji celu. Sytuacja taka może mieć miejsce np. wówczas, gdy optymalizacji podlega model określony analitycznymi zależnościami i nie trzeba przeprowadzać numerycznej estymacji gradientu.

Wiadomo, że różniczkowalność funkcji celu znacznie ułatwia prowadzenie optymalizacji funkcji wypukłych, a metody wykorzystujące informację gradientową mają możliwość szybkiej zbieżności. Gdy funkcja celu nie jest wypukła, wówczas metody gradientowe mogą na ogół być szybciej zbieżne i bardziej dokładne w odnajdowaniu minimum lokalnego, ale mniej odporne od bezgradientowych.

Strategie ewolucyjne wykazują w porównaniu z gradientowymi metodami optymalizacji liczne zalety, do których należą między innymi duża odporność na grzeźnięcie w minimach lokalnych, równoległość obliczeń i możliwość przetwarzania funkcji nieróżniczkowalnych, jednakże główną bolączką jest wciąż niewielka szybkość zbieżności.

Poniżej przedstawiono realizację koncepcji wykorzystania informacji gradientowej w strategii ewolucyjnej. Modyfikacja objęła operator mutacji, który, modyfikując genotyp, korzysta z gradientu optymalizowanej funkcji.

B. Mechanizm gradientowej mutacji

Zasada mutacji gradientowej polega na zsumowaniu losowo wygenerowanego (w sposób właściwy strategiom ewolucyjnym) wektora poprawki i wektora przeciwnego do gradientu. W celu uniknięcia dominacji przez którykolwiek z kierunków, stosuje się odpowiednie przeskalowanie.

Mutacja gradientowa zdefiniowana jest następująco. W pierwszym etapie dokonuje się obliczeń podobnych do schematu standardowego.

$$\sigma'_i = \sigma_i \exp(\tau N_i(0, 1) + \tau' N(0, 1)) \quad (1)$$

$$\Delta x_i = \sigma'_i N_i(0, 1) \quad (2)$$

$$x'_i = x_i + \Delta x_i \quad (3)$$

Następnie oblicza się współczynnik δ .

$$\delta = \gamma \frac{|\Delta \mathbf{x}|}{|\nabla \mathbf{f}|} \quad (4)$$

gdzie γ jest parametrem.

W celu zapobieżenia błędom numerycznym, wprowadzono parametr ε określający minimalną długość wektora $\nabla \mathbf{f}$. Jeżeli $|\nabla \mathbf{f}| < \varepsilon$, wówczas wzór (4) przyjmuje postać

$$\delta = \gamma \frac{|\Delta \mathbf{x}|}{\varepsilon} \quad (5)$$

Następnie wykonuje się zmianę wartości wektora zmiennych niezależnych $\mathbf{x}$:

$$\mathbf{x}'' = \mathbf{x}' - \delta \nabla \mathbf{f} \quad (6)$$

C. Weryfikacja eksperymentalna

W celu weryfikacji eksperymentalnej wykonano optymalizację funkcji testowych. Operator krzyżowania nie był stosowany. Działaniu operatora mutacji poddawano

Funkcja testowa	mutacja standardowa	mutacja gradientowa	γ
Branin	1370	870	1
Camel Back	1580	860	0.9
Camel Back 6	1500	840	0.8
Griewank	1970	1220	0.9
Shubert	3300	1770	0.9

Tab. 3: Liczba obliczeń wartości funkcji celu dla strategii ewolucyjnej z mutacją standardową i gradientową. W przypadku drugiego operatora podano wartość parametru γ dla którego uzyskiwano najlepsze wyniki.

każdy chromosom z populacji potomnej. Dostępna była analityczna postać gradientu. Wyniki optymalizacji są przedstawione w Tablicy 3.

Z powyższych danych wynika, że w większości wypadków, przy poprawnym doborze parametrów, liczba wywołań funkcji celu ulegała zmniejszeniu o ok. 40-60%. Oznacza to, że próbkowano o 40-60% punktów mniej w porównaniu do klasycznego schematu mutacji. Oczywiście, przełożenie tej liczby na czas i nakład obliczeń wymaga znajomości tych wielkości dla konkretnego rozważanego problemu. W przypadku gdyby gradient funkcji celu był przybliżany numerycznie, takie podejście spowodowałoby rzecz jasna zwiększenie nakładu obliczeń. Natomiast w specyficznych zastosowaniach, w których obliczenie wartości funkcji celu i jej gradientu wymaga stworzenia skomplikowanej struktury danych, natomiast proces obliczeniowy jest stosunkowo szybki, wykorzystanie operatora mutacji gradientowej wydaje się być atrakcyjne.

IV. PODSUMOWANIE

W artykule wprowadzono dwa operatory genetyczne dla strategii ewolucyjnej.

Operator krzyżowania adaptacyjnego daje możliwość zaoszczędzenia czasu obliczeń przy zachowaniu (jak się wydaje) dużej uniwersalności. Zastosowanie mutacji gradientowej może dać istotny wzrost efektywności algorytmu tylko przy dość specyficznych funkcjach celu.

Dalsze prace będą się koncentrować na weryfikacji eksperymentalnej dla bogatszego zestawu funkcji testowych i różnych kryteriów zatrzymania. Prowadzone są także prace nad uwzględnieniem gradientu w operacji krzyżowania.

LITERATURA

- [1] J. Arabas, *Algorytmy ewolucyjne ze zmienną licznością populacji i zmiennym zasięgiem krzyżowania*, rozprawa doktorska, Politechnika Warszawska, 1995.
- [2] J. Arabas, J. Mulawka, J. Pokraśniewicz, „New class of crossover operators for numerical optimization”, *Proc. 6th Int. Conf. on Genetic Algorithms*, ICGA97, Pittsburgh, USA, 1995.
- [3] R. Hinterding, Z. Michalewicz, A. Eiben, „Adaptation in evolutionary computation: a survey”, w *Proc. 3rd IEEE Conf. on Evolutionary Computation*, Indianapolis, USA, 1997.
- [4] Z. Michalewicz, *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, WNT, Warszawa, 1996.

- [5] H.-P. Schwefel, *Evolution and Optimum Seeking*, Wiley, Chichester, 1995.
- [6] A. Törn, A. Žilinskas, *Global Optimization*, Springer, 1989.
- [7] R. Wit, *Metody programowania nieliniowego*, WNT, Warszawa, 1989.