FOLIA 158

Annales Universitatis Paedagogicae Cracoviensis

Studia Technica VII (2014)

Wiktor Hudy

Porównanie algorytmów genetycznych o stałej i zmiennej liczbie osobników

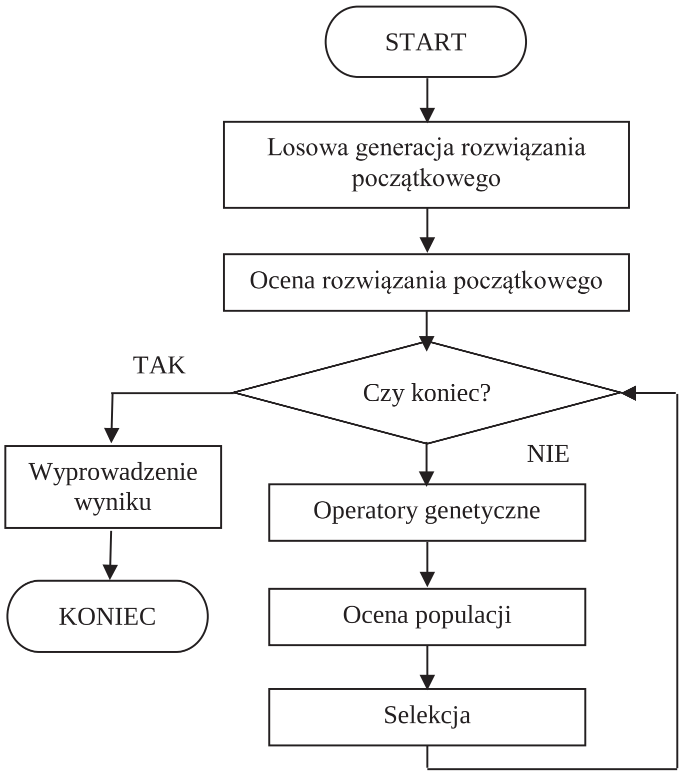
Wstęp

Algorytmy genetyczne są nowoczesną metodą poszukiwania najlepszego rozwiązania spośród dostępnych rozwiązań [1, 2, 3, 4]. Znajdują zastosowanie tam gdzie przestrzeń poszukiwań jest zbyt szeroka dla klasycznych algorytmów, gdzie przestrzeń poszukiwań jest nieciągła lub zawiera znaczną liczbę ekstremów lokalnych. Otrzymane wyniki działania tych algorytmów nie uznaje się za optymalne z uwagi np. na konieczność próbkowania przestrzeni rozwiązań oraz na losowy charakter tego typu algorytmów. Zasada działania oparta jest na ewolucji, gdzie w pewnej przestrzeni „żyją" osobniki, które podlegają krzyżowaniu oraz mutacji. Każdy osobnik posiada wartość funkcji przystosowania pełniącą rolę środowiska. Za jakość populacji odpowiada operator selekcji, który dba o to, by populacja ewoluowała w odpowiednim kierunku. W niniejszej pracy porównano dwa algorytmy genetyczne. Jeden z nich miał z góry ustaloną liczbę osobników (stała liczba populacji). Drugi z nich był algorytmem o zmiennej liczbie osobników w populacji. Pozostałe wartości parametrów obu algorytmów były takie same i ustalone *a priori*.

Po starcie algorytmu generowana jest populacja początkowa, która składa się z losowo wygenerowanych osobników. Osobniki te są następnie oceniane, czyli przypisywana jest im wartość funkcji oceny. Wartość tej funkcji określa, jak dobry jest dany osobnik. W przypadku minimalizacji wskaźnika jakości im mniejszą wartość funkcji oceny ma osobnik, tym jest lepszy (lub inaczej, jest lepiej dostosowany). Proces ewolucji trwa wewnątrz głównej pętli sprzężenia zwrotnego. Operatory genetyczne tworzą nowe osobniki na podstawie osobników z aktualnej populacji. Osobniki te są następnie oceniane. Operator selekcji wybiera osobniki z aktualnej populacji poszerzonej o osobniki wygenerowane przez operatory genetyczne do nowej populacji. Nowa populacja staje się aktualną populacją w kolejnym obiegu pętli programu. Po spełnieniu warunku kończącego proces ewolucji wyprowadzany jest wynik obliczeń.

Algorytm genetyczny

Ogólną zasadę działania algorytmów genetycznych/ewolucyjnych przedstawiono graficznie na rys. 1.



Rys. 1. Schemat blokowy algorytmu genetycznego/ewolucyjnego [1, 2, 3, 4]

Reprezentacja zadania

W niniejszym opracowaniu analizowano algorytmy genetyczne, które miały binarną reprezentację zadania. W tej reprezentacji współrzędne przestrzeni poszukiwań są zakodowane ciągiem zero-jedynkowym. W literaturze znanych jest wiele metod zapisu binarnego, np. kodem naturalnym dwójkowym, kodem Graya lub innymi. W niniejszej pracy zastosowano binarną reprezentację zadania zakodowaną kodem naturalnym dwójkowym.

Operatory genetyczne

Niezbędnym elementem procesu ewolucji są operatory genetyczne. Operatory te tworzą na podstawie wylosowanych osobników (lub osobnika) nowe osobniki, które po przypisaniu im wartości funkcji oceny są dołączane do aktualnej populacji. Najczęściej wykorzystywane operatory to:

- operator mutacji – tworzy osobnika potomnego poprzez niewielkie zmiany w kodzie osobnika rodzicielskiego. Najczęściej odbywa się to poprzez zmianę wartości pojedynczego losowo wybranego genu (z wartości 0 na 1 oraz odwrotnie),
- operator krzyżowania – tworzy dwa osobniki potomne na podstawie dwóch osobników rodzicielskich. Najczęściej odbywa się to poprzez wymianę części kodu binarnego pomiędzy osobnikami.

Metody selekcji

Głównym zadaniem operatora selekcji jest wybieranie do nowej populacji osobników lepszych (o mniejszych wartościach funkcji oceny w przypadku minimalizacji wskaźnika jakości) w celu ochrony ich genotypu. Jeżeli presja tego operatora jest zbyt duża, to algorytm może utknąć w ekstremum lokalnym wskaźnika jakości. Jeśli jest zbyt mała, to algorytm może nie być zbieżny. Istnieje wiele różnych metod selekcji:

- metoda ruletki – w metodzie tej na podstawie wartości funkcji oceny przypisuje się osobnikom wycinki koła. Im lepiej dopasowany osobnik, tym posiada większy wycinek koła. Po „ułożeniu” wszystkich osobników na jednym kole, „kręci się” nim. W wyniku otrzymuje się jednego osobnika, którego przepisuje się do nowej populacji. Proces „kręcenia kołem” powtarza się ustaloną liczbę razy.
- metoda deterministyczna – w metodzie tej ustala się listę rankingową osobników na podstawie ich wartości funkcji oceny. Do nowej populacji przepisuje się wszystkie osobniki najlepsze w zadanej z góry ilości.
- metoda turnieju – w metodzie tej wybiera się przykładowo dwa osobniki z populacji. Porównuje się ich wartość funkcji oceny. Do nowej populacji przepisuje się osobnika o lepszej wartości tej funkcji. Proces powtarza się zadaną liczbę razy.

Algorytmy o zmiennej liczbie populacji

Wybór odpowiedniej liczności populacji jest wyborem nietrywialnym. Jeśli wybierze się zbyt małą liczbę osobników w populacji, algorytm może zbiegać się do lokalnych ekstremów wskaźnika jakości. Jeśli wybierze się zbyt dużą liczność populacji, obliczenia będą trwały niepotrzebnie dłużej. Rozwiązaniem tego problemu może być zastosowanie algorytmu genetycznego ze zmienną liczbą osobników w populacji.

Rozwiązanie zaproponował Michalewicz [4], polega ono na zastosowaniu algorytmu, którego:

- proces selekcji odbywa się na zasadzie losowego wyboru bez nadawania osobnikom jakichkolwiek rang,
- każdy osobnik ma jednakową szansę na wybór do nowej populacji,
- osobnikom przypisano wartość, która określa, przez ile pokoleń ma istnieć dany osobnik w populacji. Parametr ten zależy od wartości funkcji oceny danego osobnika i jest zwiększany, gdy osobnik jest lepiej przystosowany do populacji, a zmniejszany, gdy jest gorzej przystosowany,

- osobnikom odejmuje się jednostkowy czas życia, gdy algorytm przechodzi do kolejnej pętli sprzężenia zwrotnego,
- osobniki usuwa się z populacji, gdy ich czas życia osiągnie wartość 0.

Testowane funkcje

Funkcja De Jonga

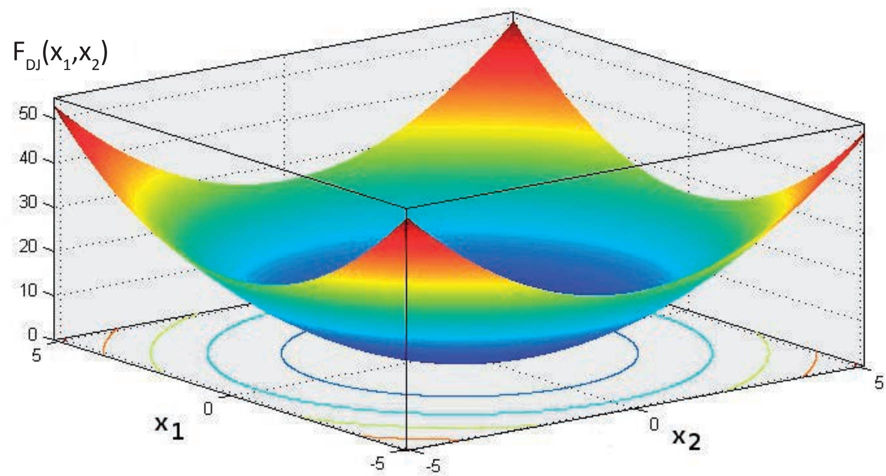
Funkcja De Jonga jest funkcją wypukłą o jednym minimum globalnym znajdującym się w punkcie $F_{DJ}(0,0) = 0$ daną wzorem (1).

$$F_{DJ} = \sum_{i=1}^{2} x_i^2 \tag{1}$$

Przedział zmienności funkcji określono na:

$$\begin{cases} x_1 = \langle -5.12, 5.12 \rangle \\ x_2 = \langle -5.12, 5.12 \rangle \end{cases}$$

Dla tych założeń wykres funkcji wykreślono na rys. 2.



Rys. 2. Wykres funkcji De Jonga dla przyjętych założeń

Funkcja Rosenbrocka

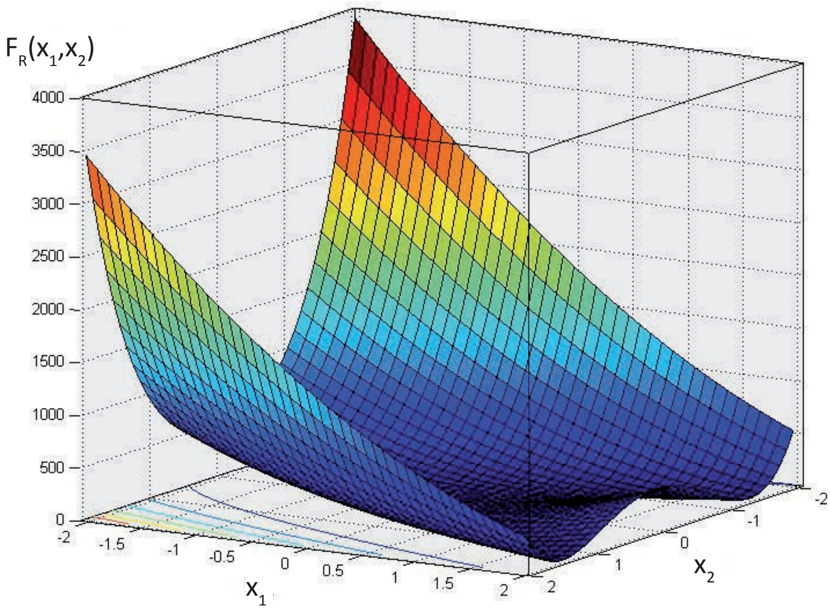
Funkcja Rosenbrocka jest funkcją o jednym minimum globalnym znajdującym się w punkcie $F_R(0,0) = 0$ daną wzorem (2).

$$F_R = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2) + (1 - x_i)^2] \tag{2}$$

Przedział zmienności funkcji określono na:

$$\begin{cases} x_1 = \langle -2.048, 2.048 \rangle \\ x_2 = \langle -2.048, 2.048 \rangle \end{cases}$$

Dla tych założeń wykres funkcji wykreślono na rys. 3.



Rys. 3. Wykres funkcji Rosenbrocka dla przyjętych założeń

Funkcja Rastrigina

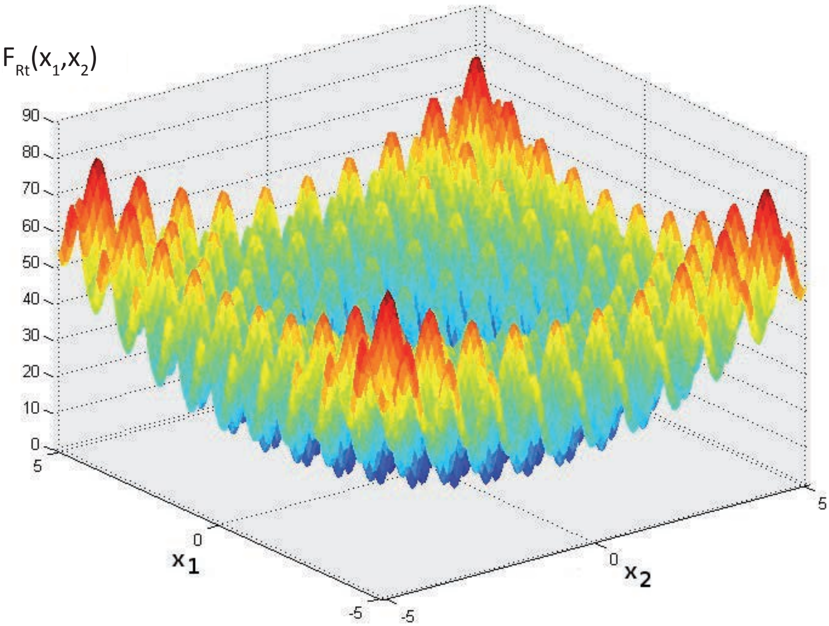
Funkcja Rastrigina jest funkcją nieliniową posiadającą wiele ekstremów lokalnych i tylko jedno minimum globalne znajdujące się w punkcie $F_{Rt}(0,0)=0$ daną wzorem (3).

$$F_{Rt} = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i)) \tag{3}$$

Przedział zmienności funkcji określono na:

$$\begin{cases} x_1 = \langle -5.12, 5.12 \rangle \\ x_2 = \langle -5.12, 5.12 \rangle \end{cases}$$

Dla tych założeń wykres funkcji wykreślono na rys. 4.



Rys. 4. Wykres funkcji Rastrigina dla przyjętych założeń

Funkcja Goldsteina–Price'a

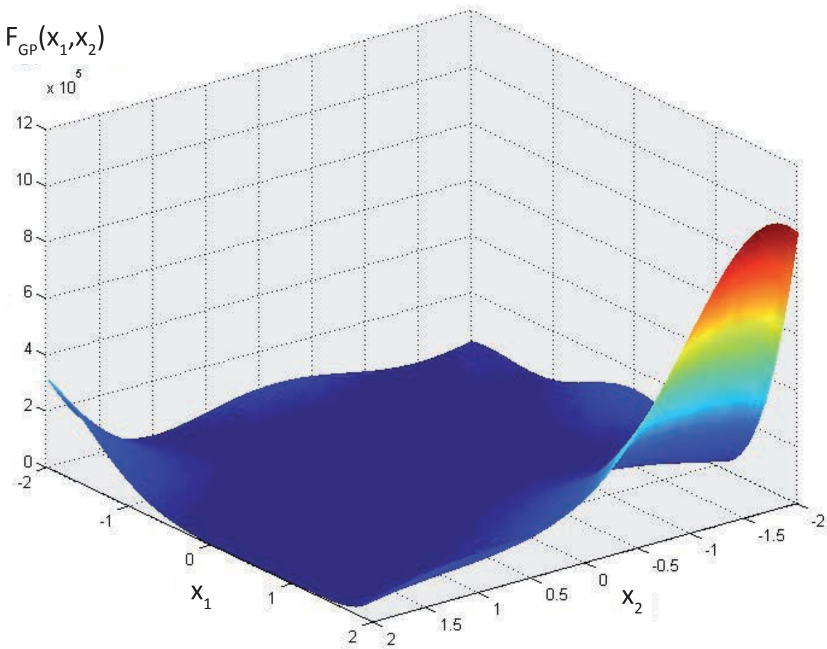
Funkcja Goldsteina–Price'a posiada wiele ekstremów lokalnych i tylko jedno minimum globalne w punkcie $F_{GP}(0,-1)=3$ daną wzorem (4).

$$F_{GP}=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]$$
$$[30+(2x_1-3x_2)^2(18-32x_1+12x_2^2+48x_2-36x_1x_2+27x_2^2)] \tag{4}$$

Przedział zmienności funkcji określono na:

$$\begin{cases} x_1=\langle -2,2\rangle \\ x_2=\langle -2,2\rangle \end{cases}$$

Dla tych założeń wykres funkcji wykreślono na rys. 5.



Rys. 5. Wykres funkcji Goldsteina–Price'a dla przyjętych założeń

4. Porównanie algorytmów genetycznych

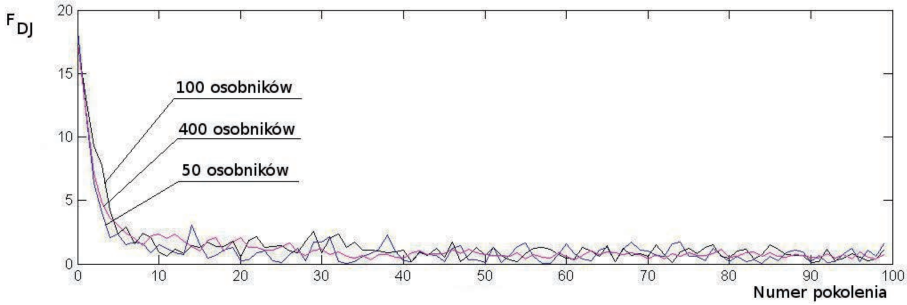
Zbadano dwa algorytmy genetyczne. Pierwszy z nich był algorytmem o stałej liczbie osobników, drugi o zmiennej liczbie osobników. Pozostałe parametry obu algorytmów były takie same i przedstawiono je w tabeli 1.

Tab. 1. Wspólne parametry algorytmów genetycznych

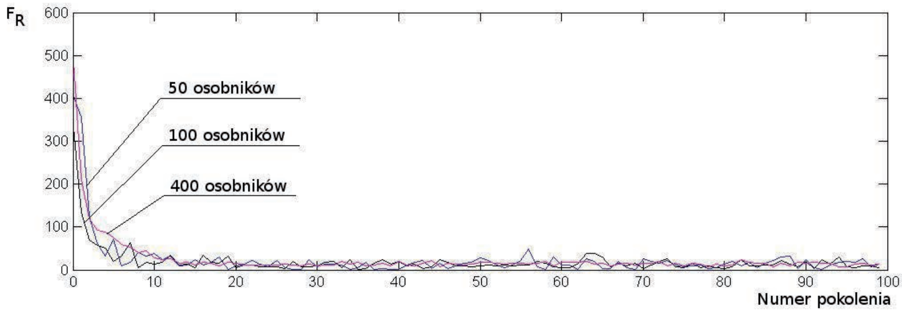
Prawdopodobieństwo krzyżowania	1,0
Prawdopodobieństwo mutacji	0,01
Długość chromosomu (ciągu bitów w reprezentacji binarnej)	30
Liczba pokoleń	100

Algorytm genetyczny ze stałą liczbą osobników w populacji

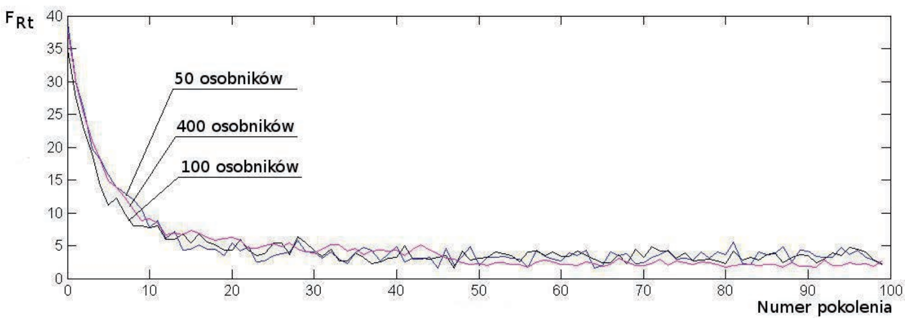
Badania przeprowadzono dla poszczególnych funkcji testowych z różną liczbą osobników w populacji (dla algorytmu genetycznego ze stałą licznością osobników) i z różną liczbą osobników w populacji początkowej (dla algorytmu genetycznego ze zmienną licznością osobników). Wyniki przedstawiono na rys. 6–9, gdzie na osi rzędnych zaznaczono średnią wartość funkcji oceny dla całej populacji dla danego pokolenia. Natomiast na osi odciętych zaznaczono numer pokolenia.



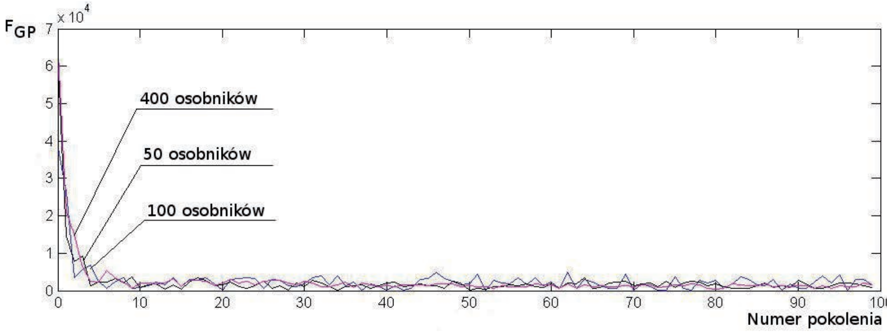
Rys. 6. Wyniki dla funkcji De Jonga



Rys. 7. Wyniki dla funkcji Rosenbrocka



Rys. 8. Wyniki dla funkcji Rastrigina



Rys. 9. Wyniki dla funkcji Goldsteina–Price'a

Tab. 2. Czas działania algorytmu genetycznego ze stałą licznością populacji oraz najlepsze osobniki znalezione przez program

Funkcja De Jonga				
Wartości minimum globalnego		$F_{DJ}(x_1, x_2)$	x_1	x_2
0		0	0	
Liczba osobników w populacji	Czas działania algorytmu	$F_{DJ}(x_1, x_2)$	x_1	x_2
30	125 [ms]	$4{,}883 * 10^{-8}$	–0,0001563	0,0001563
50	141 [ms]	$4{,}883 * 10^{-8}$	–0,0001563	0,0001563
76	219 [ms]	$4{,}883 * 10^{-8}$	0,0001563	0,0001563
100	265 [ms]	$4{,}883 * 10^{-8}$	–0,0001563	0,0001563
200	530 [ms]	$4{,}883 * 10^{-8}$	–0,0001563	0,0001563
400	1014 [ms]	$4{,}883 * 10^{-8}$	–0,0001563	0,0001563
Funkcja Rosenbrocka				
Wartości minimum globalnego		$F_R(x_1, x_2)$	x_1	x_2
0		0	0	
Liczba osobników w populacji	Czas działania algorytmu	$F_R(x_1, x_2)$	x_1	x_2
30	93 [ms]	1,000	$-6{,}104*10^{-6}$	$6{,}104*10^{-5}$
50	187 [ms]	0,1492	0,6148	0,3749
76	203 [ms]	0,01198	1,1094	1,2309
100	266 [ms]	0,08816	0,7031	0,4941
200	515 [ms]	0,1406	0,6250	0,3906
400	983 [ms]	0,1493	0,6148	0,3750
Funkcja Rastrigina				
Wartości minimum globalnego		$F_{Rt}(x_1, x_2)$	x_1	x_2
0		0	0	
Liczba osobników w populacji	Czas działania algorytmu	$F_{Rt}(x_1, x_2)$	x_1	x_2
30	109 [ms]	2,4750	–0,9599	–0,9599
50	187 [ms]	1,2375	$1{,}563*10^{-4}$	0,9599
76	219 [ms]	$9{,}687*10^{-6}$	$-1{,}562*10^{-4}$	$-1{,}562*10^{-4}$
100	281 [ms]	$4{,}659*10^{-3}$	$-1{,}562*10^{-4}$	$-4{,}844*10^{-3}$
200	515 [ms]	$9{,}687*10^{-6}$	$1{,}562*10^{-4}$	$1{,}562*10^{-4}$
400	967 [ms]	$9{,}687*10^{-6}$	$1{,}562*10^{-4}$	$1{,}562*10^{-4}$
Funkcja Goldsteina–Price'a				
Wartości minimum globalnego		$F_{GP}(x_1, x_2)$	x_1	x_2
3		0	-1	
Liczba osobników w populacji	Czas działania algorytmu	$F_{GP}(x_1, x_2)$	x_1	x_2
30	110 [ms]	3,000	$-6{,}103*10^{-5}$	–0,9999
50	171 [ms]	3,000	$-6{,}103*10^{-5}$	–0,9999
76	218 [ms]	3,000	–0,5998	–0,4001
100	265 [ms]	3,000	$6{,}103*10^{-5}$	–0,9999
200	514 [ms]	3,000	$6{,}103*10^{-5}$	–1,000
400	983 [ms]	3,000	$6{,}103*10^{-5}$	–0,9999

Jak wynika z rys. 6–9, algorytm genetyczny ze stałą liczbą osobników był zbieżny, o czym świadczy kształt wykreślonych funkcji (wszystkie zmierzają do minimum. Dodatkowo zmierzono czas działania algorytmu, a wyniki przedstawiono w tabeli 2.

Jak wynika z tabeli 2, najlepszą wartością liczby osobników w populacji jest wartość 100. Wynika to z dokładności otrzymanych wyników oraz czasu działania algorytmu. Zwiększanie liczności populacji zwiększało czas obliczeń przy niezauważalnej poprawie wyników ewolucji.

Algorytm genetyczny ze zmienną liczbą osobników w populacji

Algorytm genetyczny ze zmienną liczbą osobników w populacji z uwagi na charakter działania posiada inne dostępne do ustawienia parametry. Dodatkowe jego parametry zestawiono w tabeli 3.

Tab. 3. Dodatkowe parametry algorytmu genetycznego ze zmienną liczbą osobników w populacji

Populacja początkowa	100
Minimalny czas życia osobnika	1
Maksymalny czas życia osobnika	15

W celach porównawczych ustalono, że populacja początkowa będzie liczyła 100 osobników. Należy jednak pamiętać, że liczba ta od tego momentu będzie zmienna, zależna od dopasowania poszczególnych osobników. W tabeli 4 zebrano wyniki doświadczeń.

Tab. 4. Wyniki działania algorytmu genetycznego ze zmienną liczbą osobników w populacji dla czterech funkcji testowych

Funkcja De Jonga			
Wartości minimum globalnego	$F_{DJ}(x_1, x_2)$	x_1	x_2
	0	0	0
Czas działania algorytmu	$F_{DJ}(x_1, x_2)$	x_1	x_2
453 [ms]	$4{,}948 * 10^{-4}$	0,01172	–0,01890
Funkcja Rosenbrocka			
Wartości minimum globalnego	$F_R(x_1, x_2)$	x_1	x_2
	0	0	0
Czas działania algorytmu	$F_R(x_1, x_2)$	x_1	x_2
359 [ms]	0,01533	0,8766	0,7674
Funkcja Rastrigina			
Wartości minimum globalnego	$F_{Rt}(x_1, x_2)$	x_1	x_2
	0	0	0
Czas działania algorytmu	$F_{Rt}(x_1, x_2)$	x_1	x_2
390 [ms]	$5{,}511 * 10^{-3}$	$5{,}156 * 10^{-3}$	$1{,}094 * 10^{-3}$

Funkcja Goldsteina–Price'a			
Wartości minimum globalnego	$F_{GP}(x_1, x_2)$	x_1	x_2
	3	0	–1
Czas działania algorytmu	$F_{GP}(x_1, x_2)$	x_1	x_2
390 [ms]	13,91	–0,1687	–0,9839

Podsumowanie

Po przeprowadzonej analizie wyników zawartych w tabelach 2 i 4 stwierdza się, że obydwa algorytmy były zbieżne. Otrzymane wyniki znacznie różniły się od siebie. Algorytm ze stałą liczbą osobników w populacji obliczał minimum globalne z większą dokładnością niż algorytm genetyczny ze zmienną liczbą osobników. Ponadto pierwszy z algorytmów cechował się krótszym czasem obliczeń dla populacji początkowej równej 100 osobnikom. Dla badanych funkcji testowych algorytm ze stałą liczbą osobników okazał się algorytmem dokładniejszym, lepszym. Co nie oznacza, że dla innych funkcji oceny wyciągnięte wnioski będą takie same.

Literatura

[1] Goldberg David E., *Algorytmy genetyczne i ich zastosowania*, WNT, Warszawa 1998.
[2] Gwiazda T.D., *Algorytmy genetyczne. Kompendium,* t. 1–2, PWN, Warszawa 2007.
[3] Hudy W., Jaracz K., *Influence of evolutionary algorithm's parameters on optimization quality, on a basis of travelling salesman problem (Wpływ parametrów algorytmu ewolucyjnego na jakość optymalizacji na przykładzie problemu komiwojażera)*. Annales Academiae Paedagogicae Cracoviensis. Studia Technica IV, Wydawnictwo Naukowe UP, Kraków 2011, s. 48–53 (in Polish).
[4] Michalewicz Z., *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, WNT, Warszawa 1999.

The Comparison of Genetic Algorithms with a Fixed and Variable Number of Individuals

Abstract

In the preset article two genetic algorithms are compared: with a fixed number of individuals and with a variable number of individuals. As fitness functions the following function are used: De Jong', Rosenbrock', Rastrigina' and Goldstein–Price'. The rest of genetic algorithms' parameters are the same for both algorithms.

Key words: genetic algorithm, fixed number of individuals, variable number of individuals

Wiktor Hudy
Instytut Techniki
Uniwersytet Pedagogiczny im. KEN
ul. Podchorążych 2
30-084 Kraków, Polska