

# Sprawozdanie

## Algorytmy genetyczne w problemie optymalizacyjnym

Amadeusz Filipek

Laboratorium komputerowe WFiS AGH

### 1. Wstęp

Celem ćwiczenia jest wykorzystanie algorytmu genetycznego w problemie optymalizacyjnym i zbadanie jak poszczególne parametry algorytmu wpływają na jakość wyniku.

Algorytm genetyczny to heurystyczna (czasem zwana także metaheurystyczną) metoda optymalizacyjna. Algorytm genetyczny nie gwarantuje znalezienia wyniku optymalnego a jedynie wynik dobry. Jednakże, algorytm ten zastosować można w problemach, których złożoność wykracza poza ramy stosowalności metod dokładnych czy to ze względu na wymaganą moc obliczeniową lub na czas obliczeń. Algorytmy genetyczne wzorują się na mechanizmach funkcjonujących w przyrodzie mianowicie na ewolucji, mutacji oraz dopasowaniu do środowiska. Algorytm genetyczny wykorzystuje populację osobników, w której każdy reprezentowany jest poprzez chromosom złożony z genów. Geny każdego z osobnika bezpośrednio przekładają się na wynik optymalizowanej funkcji, zwanej funkcją celu gdy szukana jest wartość maksymalna lub też funkcją kosztu gdy szukane jest minimum. Algorytm genetyczny składa się z trzech podstawowych etapów: selekcji, krzyżowania oraz mutacji. Po każdej pętli algorytm sprawdza czy uzyskane rozwiązanie jest akceptowalne (jeśli taki warunek istnieje) lub czy osiągnięto inny warunek stopu. Wynikiem algorytmu jest wartość osiągnięta przez najlepszą jednostkę z populacji.

Selekcja ma na celu wyłonienie z populacji jednostki które są najlepiej dopasowane do otoczenia (uzyskują największą funkcję celu). Proces selekcji można wykonać na wiele sposobów np. przy użyciu metody ruletki. Polega ona na zbudowaniu wirtualnego koła, którego wycinki odpowiadają poszczególnym osobnikom. Im osobnik jest lepiej przystosowany (uzyskuje większą wartość funkcji celu) tym obszar jego tarczy jest większy. Następnie na podstawie tarczy losowane są osobniki, które zostają rodzicami. Reszta niewylosowanej populacji zostaje odrzucona. Dopuszczalne jest wylosowanie tego samego osobnika wiele razy. Alternatywną metodą selekcyjną jest metoda rankingowa, która polega na uszeregowaniu osobników od najlepszego do najgorszego i wybraniu na rodziców pierwszych z listy i odrzuceniu reszty. Liczba osobników w populacji po procesie selekcji pozostaje bez zmian.

Krzyżowanie to proces polegający na wygenerowaniu potomków, kolejnej generacji populacji. Pary z populacji wybierane są w sposób losowy. Proces krzyżowania także można wykonać na wiele sposobów. Podstawową metodą jest krzyżowanie jednopunktowe. Jeżeli geny osobników zakodowane są w sposób binarnego ciągu liczb to dla każdego potomka losowana jest liczba zwana locus, która określa punkt odcięcia chromosomów. Chromosom potomka tworzony jest poprzez sklejenie lewej części jednego rodzica z prawą częścią drugiego rodzica, drugi potomek tworzony jest z pozostałych części. Dwoje rodziców zastępowane jest zatem zawsze przez dwóch potomków. Metodą alternatywną jest krzyżowanie dwupunktowe, w którym losowane są dwie liczby locusa chromosomy potomków tworzone są jako kombinacje odpowiednich części pociętych za pomocą locusów chromosomów rodziców.

Mutacja to proces polegający na wprowadzeniu losowych zmian do chromosomów losowo wybranych osobników. Mutacja zachodzi z pewnym przyjętym prawdopodobieństwem określającym ilość zdarzeń mutacji dla każdej generacji populacji. Dla chromosomów binarnych proces mutowania polega na losowym wybraniu jednej z wartości binarnej i zamienieniu ją na wartość przeciwną. Proces mutacji ma za zadanie lekko zmieniać populację wprowadzając różnorodność i zapobiegać przedwczesnej zbieżności algorytmu. Mutacja pełni ważną rolę w algorytmie ponieważ zapobiega zatrzymywaniu się algorytmu na minimach lokalnych. Prawdopodobieństwo mutacji jest zazwyczaj niskie gdyż zbyt silna mutacja może mieć efekt odwrotny do zamierzonego, niszcząc dobre rozwiązania i zaburzając zbieżność algorytmu.

## 2. Realizacja ćwiczenia

W ramach ćwiczenia zaimplementowano algorytm genetyczny w celu rozwiązania problemu znalezienia maksimum globalnego danej funkcji:

$$f(x) = x \sin(10\pi x) + 1$$

w przedziale  $x \in [-1, 2]$ . Powyższa funkcja stanowi zatem funkcję celu, gdyż celem jest znalezienie argumentu dla którego funkcja przyjmuje wartość największą. Problem zakodowany został przy użyciu 22-bitowej rozdzielczości. Chromosom każdego osobnika ma zatem długość 22 liczb binarnych. Każdy chromosom odpowiada argumentowi  $x$  funkcji przeliczając ciąg binarny chromosomu z systemu binarnego na dziesiętny w sposób następujący:

$$x = -1 + x_{10} \frac{3}{2^{22} - 1}$$

Wygenerowano populację liczącą  $P = 20$  osobników. Populacja została zainicjalizowana w sposób losowy. Ustalona liczba iteracji algorytmu wynosiła 100 epok i jest jedynym warunkiem stopu. Selekcję wykonano za pomocą metody ruletki oraz zastosowano operator krzyżowania jednopunktowego. W ćwiczeniu zbadano wyniki algorytmu w zależności od ustalonych parametrów prawdopodobieństwa krzyżowania  $p_{cross}$  oraz prawdopodobieństwa mutacji  $p_{mut}$ . Prawdopodobieństwo krzyżowania określa ile osobników z całej populacji uczestniczy w krzyżowaniu. Ilość wybranych do krzyżowania osobników wynosi  $p_{cross} \cdot P$ . Sprawdzone także, jak zmieniają się wyniki algorytmu przy zmianie kodowania chromosomu z kodu binarnego na kod Greya. Zmiana ta dotyczy procesu mutacji, ponieważ kod Greya inaczej numeruje kolejne wartości ciągu znaków, zatem mutacja jednej liczby wprowadza mniejszą zmianę argumentu  $x$  odpowiadającego danemu chromosomowi. Sprawdzone następujące wartości parametrów:  $p_{cross} = 0.7, 0.8, 0.9$  oraz  $p_{mut} = 0.1, 0.2, 0.3$ . Dla każdej kombinacji pracę algorytmu przetestowano 20 razy ze względu na statystyczny charakter wyników. Wyniki przedstawiono w postaci wartości średniej z uzyskanych optymalnych wyników oraz ich odchylenia standardowego. Uzyskane wyniki przedstawiono dla kodu binarnego oraz kodu Greya przedstawiono na poniższych tabelach.

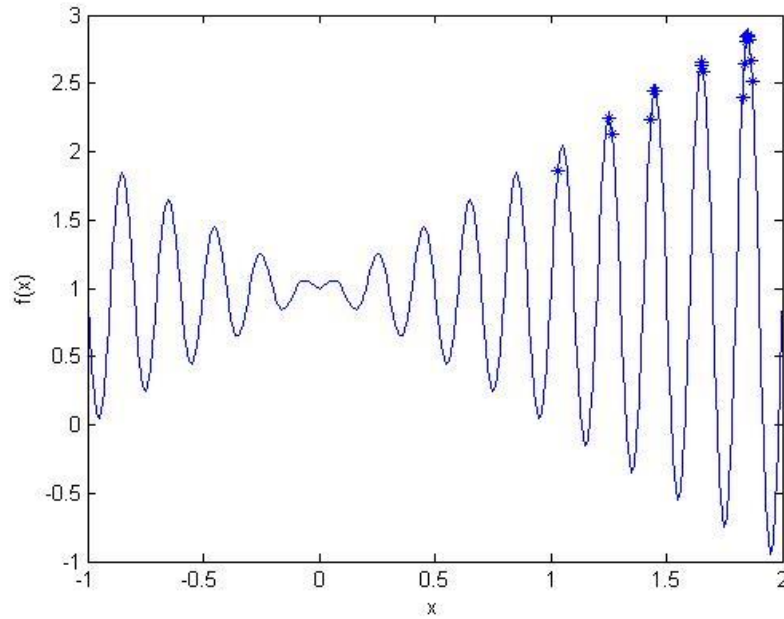
Tabela 1. Uzyskane wyniki algorytmu genetycznego przy użyciu kodu binarnego

		mean	std			mean	std			mean	std
p_cross	0.7	2.671	0.1741	p_cross	0.7	2.3036	0.3309	p_cross	0.7	2.2565	0.4134
p_mut	0.1			p_mut	0.2			p_mut	0.3		
p_cross	0.8	2.4979	0.3241	p_cross	0.8	2.4078	0.3993	p_cross	0.8	2.2558	0.2962
p_mut	0.1			p_mut	0.2			p_mut	0.3		
p_cross	0.9	2.5504	0.2812	p_cross	0.9	2.4953	0.3323	p_cross	0.9	2.2987	0.395
p_mut	0.1			p_mut	0.2			p_mut	0.3		

Tabela 2. Uzyskane wyniki algorytmu genetycznego przy użyciu kodu Greya

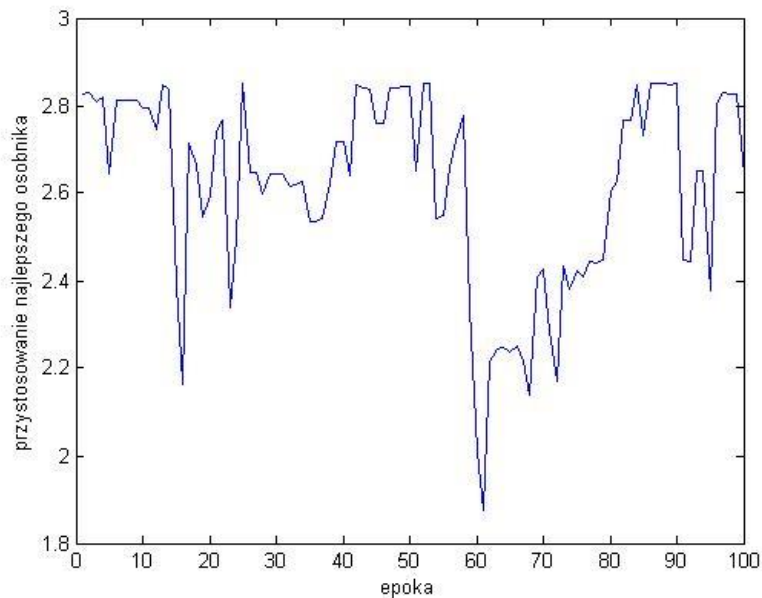
		mean	std			mean	std			mean	std
p_cross	0.7	2.5755	0.2281	p_cross	0.7	2.2332	0.389	p_cross	0.7	2.4891	0.2813
p_mut	0.1			p_mut	0.2			p_mut	0.3		
p_cross	0.8	2.5535	0.2587	p_cross	0.8	2.4329	0.4019	p_cross	0.8	2.2957	0.3222
p_mut	0.1			p_mut	0.2			p_mut	0.3		
p_cross	0.9	2.573	0.2873	p_cross	0.9	2.4176	0.287	p_cross	0.9	2.3562	0.3659
p_mut	0.1			p_mut	0.2			p_mut	0.3		

Na podstawie powyższych tabel stwierdzić można, że wraz ze wzrostem prawdopodobieństwa mutacji, średni wynik algorytmu spada co świadczy o tym, że wyższe wartości  $p_{mut}$  zaburzają bieżność algorytmu niszcząc dobre rozwiązania. Zjawisko to osłabione jest przy użyciu kodu Greya, gdzie dla  $p_{mut} = 0.3$  uzyskane wyniki są lepsze niż odpowiadające im wyniki dla kodu binarnego. Dla parametru  $p_{cross}$  natomiast nie widać jednoznacznej tendencji w zakresie badanych wartości. Niezależnie od kodu, algorytm uzyskał najlepsze wyniki dla  $p_{mut} = 0.1$ . Najlepszy wynik algorytmu zaznaczony został w tabeli na szaro i był uzyskany dla parametrów  $p_{mut} = 0.1$  i  $p_{cross} = 0.7$  przy użyciu kodu binarnego. Dla kombinacji parametrów, dla których uzyskano najlepszy wynik, wykonano wykres 1. przedstawiający uzyskane 20 wyników algorytmu na tle optymalizowanej funkcji.



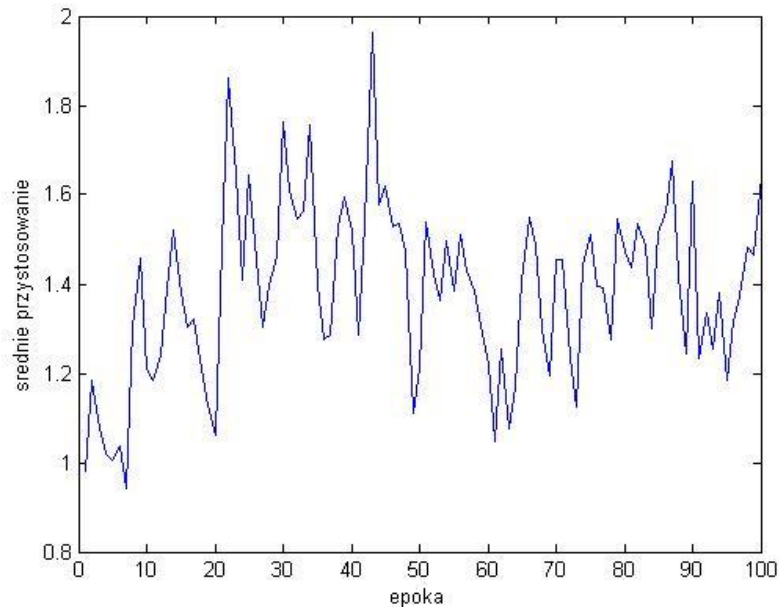
Wykres 1. Wykres optymalizowanej funkcji wraz z 20 wynikami algorytmu dla  $p_{\text{mut}} = 0.1$  i  $p_{\text{cross}} = 0.7$

Na wykresie widać, że algorytm w większości przypadków znajduje maksimum globalne, jednakże w paru przypadkach algorytm utknął w maksimach lokalnych. Jednakże, pomimo tego, że optymalizowana funkcja jest symetryczna, żaden wynik nie utknął na ujemnych wartościach osi x. Dla tej najlepszej kombinacji parametrów zbadano także jak zmienia się najlepsza wartość z populacji, wartość średnia wyniku populacji, odchylenie standardowe populacji. Uzyskane wyniki przedstawiono na poniższych wykresach.



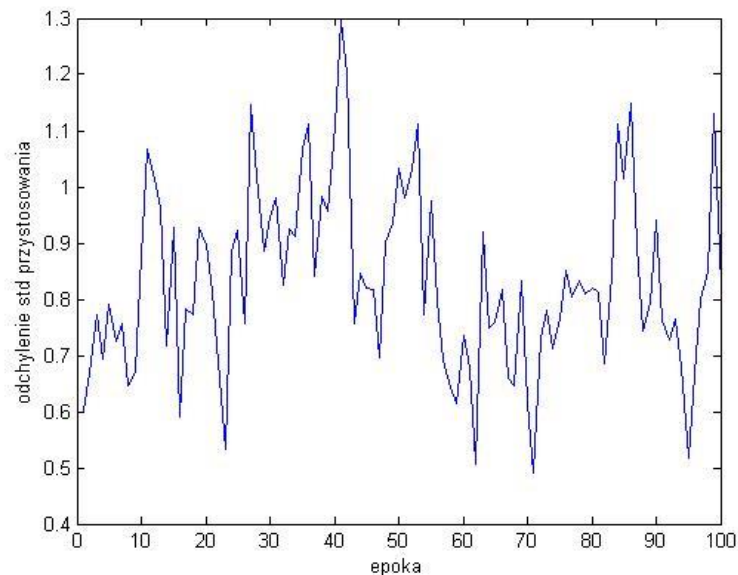
Wykres 2. Uzyskane wyniki algorytmu w kolejnych iteracjach dla  $p_{\text{mut}} = 0.1$  i  $p_{\text{cross}} = 0.7$

Na powyższym wykresie widać, że dla wybranych okresów algorytm znajduje się w maksimum globalnym, jednakże nie zostaje w nim na długo i po pewnym czasie z niego wychodzi, co pokazują ostre pikowe zejścia w dół. Po takim zejściu algorytm na nowo zmierza w kierunku ekstremum globalnego.



Wykres 3. Średni wynik populacji w kolejnych iteracjach algorytmu dla  $p_{mut} = 0.1$  i  $p_{cross} = 0.7$

Średnie przystosowanie populacji wykazuje silne oscylacje. Na starcie algorytmu wartość ta wykazuje trend wzrostowy a następnie utrzymuje się na stałym poziomie cały czas oscylując.



Wykres 4. Odchylenie standardowe średniego wyniku populacji w kolejnych iteracjach dla  $p_{mut} = 0.1$  i  $p_{cross} = 0.7$

Odchylenie standardowe średniego przystosowania na początku działania algorytmu wykazuje tendencję wzrostową a następnie w miarę rozwoju algorytmu spada. Podobnie jak wartość średnia przystosowania, odchylenie standardowe wykazuje silne oscylacje.

## 1. Podsumowanie

W ramach ćwiczenia zaimplementowano algorytm genetyczny. Zbadano wyniki pracy algorytmu dla różnych parametrów  $p_{mut}$  i  $p_{cross}$ . Zaobserwowano, że wraz ze wzrostem prawdopodobieństwa mutacji wyniki algorytmu stają się gorsze. Nie zaobserwowano jednoznacznej tendencji wyników algorytmu w zależności od prawdopodobieństwa krzyżowania. Sprawdzono także, jak wyniki te zmieniają się gdy chromosomy zamiast systemu binarnego wykorzystują kod Greya. Przy wykorzystaniu kodu Greya, czułość algorytmu na prawdopodobieństwo mutacji maleje, zatem wyniki przy wykorzystaniu kodu Greya są lepsze dla wysokich wartości  $p_{mut}$ . Zmiana systemu kodowania z binarnego na Greya nie wnosi jednak zmiany co do najlepszych uzyskanych wyników a zwiększa czas pracy algorytmu ze względu na rachunki związane z przeliczeniem kodu Greya na system dziesiętny. Zbadany przebieg pracy algorytmu w tym wartości maksymalnej, średniej oraz odchylenia standardowego z populacji w trakcie iteracji pokazuje statystyczny charakter tej metody i związane z tym oscylacje. W szczególności algorytm w trakcie pracy uzyskuje parę razy maksimum lokalne wychodząc z niego i wracając z powrotem po pewnej ilości iteracji.