Zeszyty Naukowe WWSI, No 13, Vol. 9, 2015, s. 7-19

Zastosowanie algorytmu optymalizacji rojem cząstek do znajdowania ekstremów globalnych wybranych funkcji testowych

Mateusz Wiatrak, Ewa Figielska*

Warszawska Wyższa Szkoła Informatyki

Streszczenie

Praca dotyczy zastosowania algorytmu optymalizacji rojem cząstek do znajdowania ekstremów globalnych dla wybranych funkcji jedno i wielomodalnych. Na podstawie wyników eksperymentu obliczeniowego wyłoniono warianty ustawień parametrów algorytmu zapewniające jego największą skuteczność.

Słowa kluczowe – optymalizacja rojem cząstek, ekstremum globalne, funkcje testowe.

1 Wprowadzenie

Algorytmy, które powstały na bazie obserwacji natury, zyskały w ostatnich latach wielką popularność. Pozwalają one efektywnie rozwiązywać złożone problemy optymalizacyjne występujące w praktyce inżynieryjnej.

Zwierzęta żyjące w stadach wybierają korzystne dla siebie czynniki środowiska, bazując na własnych doświadczeniach oraz informacjach przekazywanych przez

* E-mail: efigielska@poczta.wwsi.edu.pl

Zgłoszono do druku 22 października 2015 r.

inne osobniki stada. Dzięki analogii do takich i innych zachowań istot żywych powstało wiele algorytmów. Najbardziej znane z nich to: algorytm optymalizacji rojem cząstek (ang. *particle swarm optimization*, PSO), algorytm mrówkowy (ang. *ant colony optimization*), algorytm pszczeli (ang. *bee algorithm*), algorytm ławicy ryb (ang. *fish school search*). Algorytmy te dobrze sprawdzają się w praktyce, ponieważ można je łatwo przystosować do rozwiązywania niemal każdego problemu, niezależnie od postaci funkcji celu, nałożonych ograniczeń, rozmiaru przestrzeni rozwiązań itp. Wynika to z faktu, że nie przetwarzają one bezpośrednio parametrów zadania, lecz ich zakodowaną postać. Algorytmy te stosują probabilistyczne reguły wyboru, co sprawia, że są zdolne do wychodzenia z ekstremów lokalnych.

W artykule rozpatrywane jest działanie algorytmu optymalizacji rojem cząstek zastosowanego do znajdowania ekstremów globalnych dla wybranych funkcji jedno i wielomodalnych. Celem jest znalezienie takich wartości parametrów algorytmu, które zapewniają jego największą skuteczność.

2 Algorytm optymalizacji rojem cząstek

Algorytm optymalizacji rojem cząstek (PSO), zaproponowany roku w 1995 przez przez Kennediego i Eberharta [6], jest stochastycznym algorytmem obliczeniowym naśladującym zachowanie przemieszczającego się stada ptaków. W celu znalezienia pożywienia każdy osobnik stada przemieszcza się z prędkością określoną na podstawie swoich własnych najlepszych poprzednich doświadczeń oraz najlepszych doświadczeń pozostałych osobników stada. Jednocześnie prędkości poruszania się dobierane są tak, żeby zachowany był odpowiedni dystans między poszczególnymi osobnikami [1,2,3].

Algorytm PSO pracuje ze zbiorem (nazywanym rojem) potencjalnych rozwiązań problemu (nazywanych cząstkami). Każda cząstka i jest opisana przez swoją pozycję $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$ oraz prędkość $V_i = (v_{i1}, v_{i2}, \dots, v_{im})$, przy czym rozwiązanie problemu (w m wymiarowej przestrzeni) jest reprezentowane przez wektor X_i. Rój składa się z *n* cząstek. Proces poszukiwania najlepszego rozwiązania odbywa się iteracyjnie. W każdej iteracji, każda cząstka roju przesuwa się do nowej pozycji uwzględniając swoją poprzednią pozycję i poprzednią prędkość, a także swoją najlepszą do tej pory znalezioną pozycję $P_i = (p_{i1}, p_{i2}, \dots, p_{im})$ oraz najlepszą dotychczasową pozycję w całym roju, $G = (g_1, g_2, \dots, g_m)$..

Niech t oznacza numer bieżącej iteracji algorytmu. Prędkość cząstki i w iteracji t, obliczana jest według następującego wzoru:

$$v_{ij}^{t} = wv_{ij}^{(t-1)} + c_1 r_1 (p_{ij} - x_{ij}^{t-1}) + c_2 r_2 (G_j - x_{ij}^{t-1}) \quad \text{dla } j = 1, \ldots, m \tag{1}$$

gdzie w jest współczynnikiem nazywanym wagą inercji, c_1, c_2 są współczynnikami uczenia (nazywanymi, odpowiednio, kognitywnym i socjalnym), r_1 , r_2 – liczbami losowymi z przedziału [0,1].

Nowa pozycja cząstki i, X_i, wyznaczana jest ze wzoru:

$$x_{ij}^{t} = x_{ij}^{t-1} + v_{ij}^{t} \quad \text{dla } j = 1, \ldots m \tag{2}$$

Algorytm optymalizacji rojem cząstek działa według poniższego schematu.

1. Ustaw wartości początkowe parametrów: w, c_1 , c_2.
2. Ustaw $t = 0$. Wygeneruj początkowy rój $R(t)$.
3. Dla każdej cząstki i w $R(t)$ wyznacz wartości funkcji celu $f(X_i)$.
4. Zapamiętaj najlepszą cząstkę G w $R(t)$. Dla każdej cząstki i w $R(t)$ zapamiętaj jej aktualną pozycję jako najlepszą, P_i.
5. Dopóki kryterium stopu nie jest spełnione, wykonuj:
6. Ustaw $t = 1 + 1$.
7. Wygeneruj nowy rój $R(t)$ na podstawie $R(t-1)$, stosując wzory (1) i (2).
8. Dla każdej cząstki i w $R(t)$ wyznacz wartość funkcji celu $f(X_i)$.
9. Dla każdej cząstki i w $R(t)$ zaktualizuj jej najlepszą pozycję P_i.
10. Zaktualizuj najlepszą pozycję G w całym roju $R(t)$.
11. Zwróć najlepsze znalezione rozwiązanie, G.

W naszej implementacji początkowa pozycja każdej cząstki w roju była generowana losowo, a prędkość początkowa ustawiona była na 1. Algorytm kończył działanie po wykonaniu 1000000 iteracji.

3 Opis badanych funkcji

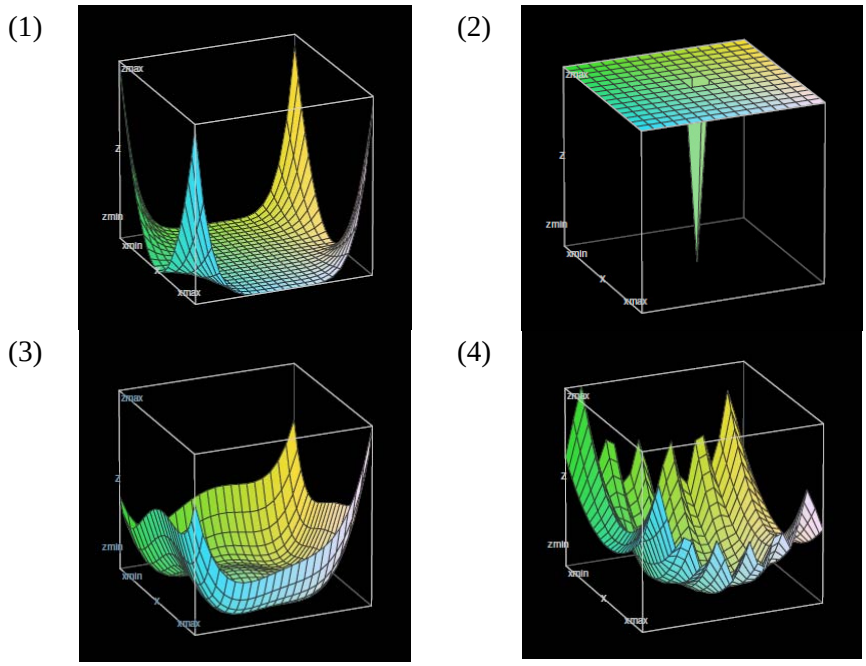
Do wykonania badań algorytmu PSO wybrane zostały 4 typowe funkcje testowe: funkcja Beale'a, Easoma, Himmemblau i Levy'ego [4,5,7,8,9]. Opisy tych funkcji

zawarte są w tabeli 1. Rysunek 1 przedstawia ich wizualizacje wykonane za pomocą narzędzia 3D Function Grapher z parametrem grid=22 [10].

Tabela 1. Funkcje testowe

Nr	Funkcja	Opis
1	Funkcja Beale'a $f(x,y) = (1{,}5 - x + xy)^2 + (2{,}25 - x + xy^2)^2 + (2{,}625 - x + xy^3)^2$ dla $-4{,}5 \leq x, y \leq 4{,}5$	Funkcja jednomodalna. Ma jedno minimum globalne w punkcie $p^* = (3; 0{,}5),\ f(p^*) = 0.$
2	Funkcja Easoma $f(x,y) = -\cos(x)\cos(y)\,\mathrm{e}^{-(x-\pi)^2-(y-\pi)^2}$ dla $-100 \leq x, y \leq 100$	Funkcja jednomodalna. Ma jedno minimum globalne w punkcie $p^* = (\pi, \pi),\ f(p^*) = -1.$ Obszar zawierający minimum globalne jest stosunkowo mały względem przestrzeni przeszukiwań.
3	Funkcja Himmemblau $f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$ dla $-6 \leq x, y \leq 6$	Funkcja wielomodalna. Ma cztery minima globalne w punktach: $p_1^* = (3{,}0; 2{,}0),$ $p_2^* = (-2{,}805118; 3{,}131312),$ $p_3^* = (-3{,}779310; -3{,}283186),$ $p_4^* = (3{,}584428; -1{,}848126).$ $f(p_1^*) = f(p_2^*) = f(p_3^*) = f(p_4^*) = 0.$
4	Funkcja Levy'ego nr 13 $f(x,y) = \sin^2(3\pi x) + (x-1)^2(1 + \sin^2(3\pi y)) + (y-1)^2(1 + \sin^2(2\pi y))$ dla $-10 \leq x, y \leq 10$	Funkcja wielomodalna. Ma jedno minimum globalne w punkcie $p^* = (1{,}1),\ f(p^*) = 0.$ Posiada dużą liczbę minimów lokalnych.

Rysunek 1 zamieszczony na następnej stronie przedstawia ich wizualizacje wykonane za pomocą narzędzia 3D Function Grapher z parametrem grid=22 [10].



Rysunek 1. Wizualizacje funkcji (1) Beale'a (2) Easoma, (3) Himmemblau , (4) Levy'ego.

4 Eksperyment obliczeniowy

Niniejszy rozdział zawiera wyniki badania działania algorytmu PSO znajdującego minima globalne 4 opisanych wcześniej funkcji w zależności od wartości jego parametrów: wagi inercji (w), współczynników uczenia (c_1 i c_2) oraz rozmiaru populacji (n). Dla każdej funkcji użyto 36 różnych wariantów ustawień parametrów algorytmu. Przedstawione są one w tabeli 2. Dla każdego wariantu parametrów algorytm uruchamiany był 10 razy. Tak więc, w całym eksperymencie wykonanych zostało1440 uruchomień algorytmu (prób znalezienia ekstremum globalnego).

Jako miarę jakości algorytmu przyjęto jego skuteczność, która jest zdefiniowana jako procentowy stosunek liczby udanych prób znalezienia ekstremum globalnego do liczby wszystkich prób, przy czym próbę uznaje się za nieudaną, jeżeli liczba iteracji przekroczy 1000000, a ekstremum globalne nie zostanie znalezione. Dla każdego wariantu wyznaczona została także średnia liczba iteracji dla prób skutecznych.

Tabela 2. Wykaz wariantów ustawień parametrów algorytmu

Nr	c_1, c_2	w	n	Nr	c_1, c_2	w	n	Nr	c_1, c_2	w	n
1	2	0.1	10	13	2	0.1	25	25	2	0.1	50
2	2	0.5	10	14	2	0.5	25	26	2	0.5	50
3	2	1	10	15	2	1	25	27	2	1	50
4	2	Rand [0;1]	10	16	2	Rand [0;1]	25	28	2	Rand [0;1]	50
5	1.5	0.1	10	17	1.5	0.1	25	29	1.5	0.1	50
6	1.5	0.5	10	18	1.5	0.5	25	30	1.5	0.5	50
7	1.5	1	10	19	1.5	1	25	31	1.5	1	50
8	1.5	Rand [0;1]	10	20	1.5	Rand [0;1]	25	32	1.5	Rand [0;1]	50
9	1	0.1	10	21	1	0.1	25	33	1	0.1	50
10	1	0.5	10	22	1	0.5	25	34	1	0.5	50
11	1	1	10	23	1	1	25	35	1	1	50
12	1	Rand [0;1]	10	24	1	Rand [0;1]	25	36	1	Rand [0;1]	50

4.1 Badanie dla funkcji Beale’a

Wyniki poszukiwania minimum globalnego dla funckji Beale’a przedstawione są w tabeli 3. Średnia skuteczność algorytmu po wszystkich wariantach ustawień parametrów wyniosła 71,4%, natomiast średnia liczba iteracji we wszystkich udanych próbach – 18,8.

Skuteczność równa 100% osiągnięta została dla 8 następujących wariantów ustawień parametrów algorytmu: 1, 13, 16, 19, 25, 26, 27 i 35, przy czym najmniej iteracji, średnio 8.2, wykonanych zostało dla wariantu 25.

Przy zastosowaniu ustawień parametrów o numerach 9 i 21, algorytm nie znalazł minimum globalnego (przy założonym kryteriu stopu, tzn. w 1000000 iteracji). Dla obu tych wariantów $c_1 = c_2 = 1$, $w = 0.1$, a wielkość populacji nie przekracza 25.

Tabela 3. Wyniki obliczeń dla funkcji Beale'a

Nr	Skuteczność [%]	Liczba iteracji	Nr	Skuteczność [%]	Liczba iteracji	Nr	Skuteczność [%]	Liczba iteracji
1	100	80,2	13	100	10,1	25	100	8,2
2	80	23,5	14	90	20, 6	26	100	17,1
3	60	34,2	15	90	31,7	27	100	24,1
4	80	30,1	16	100	17,0	28	90	12, 6
5	20	11,0	17	80	9,3	29	80	7,1
6	70	16,7	18	70	12,6	30	90	11,4
7	70	28,3	19	100	18,9	31	80	17,4
8	70	28,7	20	80	13,3	32	80	10,5
9	0	–	21	0	–	33	10	7,0
10	10	16	22	60	12,0	34	90	11,3
11	90	23,3	23	80	17,0	35	100	13,7
12	40	15,8	24	60	11,8	36	50	9,6

4.2 Badanie funkcji Easoma

Wyniki obliczeń dla funkcji Easoma znajdują się w tabeli 4.

Algorytm PSO dobrze sprawdził się w poszukiwaniu minimum globalnego tej funkcji osiągając 100% skuteczność dla 28 badanych wariantów parametrów. Średnia skuteczność dla wszystkich wariantów wyniosła 87,2%, a średnia liczba iteracji dla udanych prób – 27,0.

Najmniej iteracji przy 100% skuteczności – średnio 11,3 – algorytm potrzebował przy ustawieniu parametrów zgodnie z wariantem 29.

Dla dwóch wariantów 5 i 9, minimum globalne nie zostało znalezione.

Tabela 4. Wyniki obliczeń dla funkcji Easoma

Nr	Skuteczność [%]	Liczba iteracji	Nr	Skuteczność [%]	Liczba iteracji	Nr	Skuteczność [%]	Liczba iteracji
1	80	46,6	13	100	18,4	25	100	14,1
2	100	44,3	14	100	27,2	26	100	22,4
3	100	56,1	15	100	39,4	27	100	35,2
4	100	38,2	16	100	26,5	28	100	20,1
5	0	–	17	100	15,5	29	100	11,3
6	90	31,4	18	100	21,2	30	100	17,3
7	100	39,9	19	100	29,8	31	100	25,6
8	80	31,9	20	100	18,5	32	100	25,6
9	0	–	21	30	21,0	33	100	16,6
10	30	30,0	22	100	19,5	34	100	15,5
11	100	35,3	23	100	27,2	35	100	20,7
12	30	39,0	24	100	19,3	36	100	15,6

4.3 Badanie funkcji Himmemblau

Wyniki obliczeń dla funkcji Himmemblau przedstawione są w tabeli 5.

Algorytm PSO dobrze sprawdził się w wyszukiwaniu minimów globalnych tej funkcji. Średnia skuteczność po wszystkich wariantach wartości parametrów jest równa 89,4%, przy czym skuteczność 100% została osiągnięta dla 27 wariantów. Najmniejsza liczba iteracji, średnio 8,6, wymagana była dla wariantu 29. Średnia liczba iteracji dla prób skutecznych wyniosła 17,6.

Minimum globalne nie zostało znalezione tylko dla jednego wariantu – o numerze 9.

Tabela 5. Wyniki obliczeń dla funkcji Himmelblau

Nr	Skuteczność [%]	Liczba iteracji	Nr	Skuteczność [%]	Liczba iteracji	Nr	Skuteczność [%]	Liczba iteracji
1	100	30,4	13	100	11,4	25	100	8,9
2	100	21,4	14	100	18,2	26	100	16,7
3	90	41,6	15	100	27,0	27	100	20,1
4	100	24,1	16	100	16,7	28	100	13,8
5	40	12,8	17	100	9,9	29	100	8,6
6	90	21,3	18	100	15,2	30	100	13,4
7	100	27,3	19	100	21,1	31	100	18,0
8	100	19,2	20	100	13,2	32	100	12,0
9	0	–	21	60	14,16	33	70	8,9
10	60	13,2	22	100	13,6	34	100	11,7
11	90	24,3	23	100	19,0	35	100	16,0
12	20	25,5	24	100	14,0	36	100	11,6

4.4 Badanie funkcji Levy'ego nr 13

W tabeli 6 przedstawiono wyniki badania minimum funkcji Levy'ego.

Skuteczność równa 100% osiągnięta została dla ponad połowy (19) badanych wariantów. Średnia skuteczność po wszystkich próbach wyniosła 80,3%. Średnia liczba iteracji dla udanych prób wyniosła 2948,3. Wartość ta wynika z wykonania bardzo dużej liczby iteracji dla 3 wariantów ustawień parametrów (1, 6 i 24). Dla pozostałych wariantów liczba iteracji była znacznie mniejsza i nie przekraczała 40. Średnia liczba iteracji dla udanych prób z pominięciem wariantów 1, 6 i 24 wynosi 20,4. Najmniej iteracji – średnio 8,2 – wykonał algorytm z ustawieniami parametrów o numerze 29.

Najgorzej wypadły warianty, gdzie $c_1 = c_2 = 1$ i $w = 0{,}1$. Przy takich wartościach parametrów i populacji składającej się z 10 i 25 osobników nie udało się znaleźć minimum globalnego. Dla $n = 50$, skuteczność algorymu wyniosła 30%, co w porównaniu do pozostałych wariantów parametrów jest słabym wynikiem.

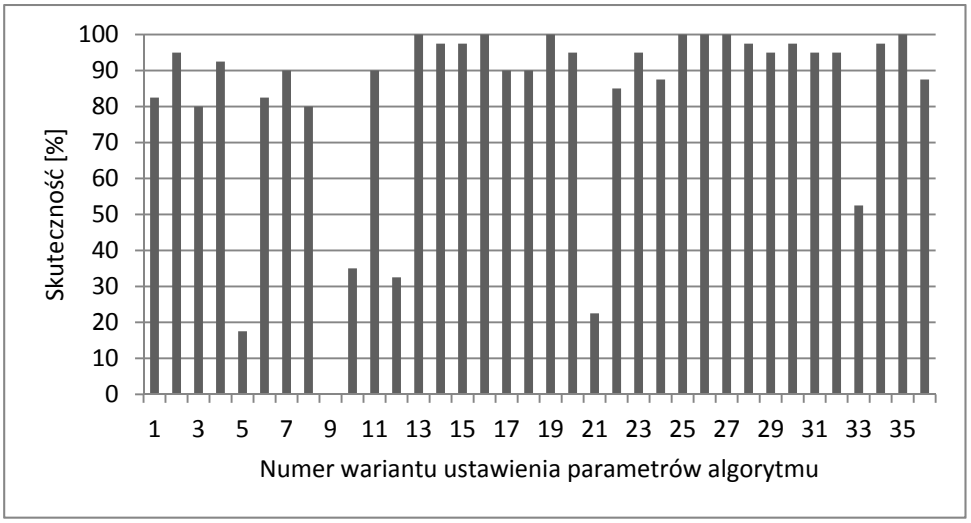
Tabela 6. Wyniki obliczeń dla funkcji Levy'ego

Nr	Skuteczność [%]	Liczba iteracji	Nr	Skuteczność [%]	Liczba iteracji	Nr	Skuteczność [%]	Liczba iteracji
1	50	59687,4	13	100	15,0	25	100	10,3
2	100	31,1	14	100	20,4	26	100	17,0
3	70	47,9	15	100	31,5	27	100	25,4
4	90	29,7	16	100	22,4	28	100	15,4
5	10	16,0	17	80	15,3	29	100	8,2
6	80	14164,9	18	90	16,6	30	100	13,7
7	90	32,7	19	100	23,5	31	100	21,0
8	70	28,6	20	100	16,4	32	100	13,1
9	0	–	21	0	–	33	30	10,3
10	40	17,75	22	80	16,6	34	100	13,0
11	80	37,6	23	100	20,3	35	100	18,6
12	40	15,8	24	90	25757,1	36	100	12,3

4.5 Podsumowanie uzyskanych wyników

Minimum funkcji zostało znalezione w 82% wszystkich 1440 uruchomień algorytmu, przy czym dobór parametrów był kluczowy w odniesieniu sukcesu. Zastosowanie pewnych wariantów parametrów dawało skuteczność równą 100% dla każdej z badanych funkcji, a w przypadku niektórych innych wariantów dla każdej z funckji obserwowano niską skuteczność algorytmu.

W niniejszym podrozdziale przeanalizujemy działanie algorytmu PSO na podstawie wartości średniej skuteczności liczonej po wszystkich badanych funkcjach testowych. Rysunek 2 przedstawia wartości średniej skuteczności dla każdego z 36 wariantów ustawień parametrów.



Rysunek 2. Średnia skuteczność po wszystkich badanych funkcjach

Na rysunku tym widzimy, że rozmiar populacji ma znaczny wpływ na jakość wyników. Przy $n = 10$ (warianty 1-12) średnia skuteczność nigdy nie osiągnęła 100%, podczas gdy dla $n = 25$ i 50 (warianty 13-24 i 25-36), stuprocentowa skuteczność dla wszystkich 4 funkcji testowych została uzyskana w 7 przypadkach. Liczba wariantów, dla których średnia skuteczność jest nie mniejsza niż 90% jest równa 4, 9 i 10, gdy rozmiar populacji wynosi, odpowiednio, 10, 25 i 50.

Dla wariantu o numerze 9, algorytm PSO nie znalazł ekstremum globalnego dla żadnej z testowanych funkcji. W wariancie tym $c_1 = c_2 = 1$, $w = 0.1$ i $n = 10$. Przy takich ustawieniach współczynników uczenia i wagi inercji również dla większych populacji średnia skuteczność algorytmu jest niewielka. Wynosi ona 22.5% i 52.5%, odpowiednio, dla $n = 25$ i 50 (warianty 21 i 33).

W tabeli 7 zebrane zostały dwa rodzaje wariantów ustawień parametrów algorytmu, a mianowicie warianty, przy których średnia skuteczność wynosi 100% oraz warianty ze średnią skutecznością nie przekraczającą 75%.

W tabeli tej widzimy, że w 5 spośród 7 najlepszych wariantów ustawień parametrów współczynniki uczenia mają wartość 2 i, jak już było wspomniane wcześniej, we wszystkich 7 najlepszych wariantach $n \geq 25$.

Jeżeli chodzi o najgorsze warianty, to w większości z nich $c_1 = c_2 = 1, w = 0{,}1$ oraz $n = 10$.

Tabela 7. Najlepsze i najgorsze ustawienia parametrów algorytmu

	Skuteczność [%]	Nr	c_1, c_2	w	n
Najlepsze ustawienia parametrów:	100	13	2	0.1	25
	100	16	2	Rand [0;1]	25
	100	19	1.5	1	25
	100	25	2	0.1	50
	100	26	2	0.5	50
	100	27	2	1	50
	100	35	1	1	50
Najgorsze ustawienia parametrów:	0%	9	1	0.1	10
	17.5%	5	1.5	0.1	10
	22.5%	21	1	0.1	25
	32.5%	12	1	Rand [0;1]	10
	35%	10	1	0.5	10
	52.5%	33	1	0.1	50

5 Wnioski i uwagi końcowe

W pracy zaimlementowano i zbadano działanie algorytmu optymalizacji rojem cząstek zastosowanego do znajdowania ekstremów globalnych funkcji. Wykonany został eksperyment obliczeniowy z wykorzystaniem 4 funckji testowych.

W eksperymencie tym zbadano 36 różnych wariantów ustawień parametrów algorytmu. Na podstawie uzyskanych wyników określone zostały takie wartości parametrów algorytmu, które dają wysoką skuteczność znajdowania ekstremum globalnego oraz wartości, przy których działanie algorytmu nie jest satysfakcjonujące.

Literatura

[1] Bautu A., *Generalizations of Particle Swarm Optimization*, LAP LAMBERT Academic Publishing, 2012
[2] Chan F.T.S, M. K. Tiwari, *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, I-Tech Education and Publishing, 2007
[3] Elbeltagi E., T. Hegazy, D. Grierson, Comparison among five evolutionary-based optimization algorithms, *Advanced Engineering Informatics* 19, 43-53, 2005
[4] Himmelblau's function, https://en.wikipedia.org/wiki/Himmelblau's_function (dostęp: 20.05.2015)
[5] Jamil M., X.-S. Yang, *A literature survey of benchmark functions for global optimization problems*, „International Journal of Mathematical Modelling and Numerical Optimisation", Vol. 4, No. 2, 2013
[6] Kennedy, J., R.C. Eberhart, *Particle Swarm Optimization*, *Proceedings of IEEE International Conference on Neural Networks* IV, 1942–1948, 1995
[7] Molga M., C. Smutnicki, *Test functions for optimization needs*, http://www.robertmarks.org/ Classes/ENGR5358/Papers/functions.pdf (dostęp: 20.05.2015)
[8] Surjanovic, S., D. Bingham, *Virtual Library of Simulation Experiments: Test Functions and Datasets*, http://www.sfu.ca/~ssurjano (dostęp: 20.05.2015)
[9] *Test functions for optimization*, https://en.wikipedia.org/wiki/Test_functions_for_optimization (dostęp: 9.10.2015)
[10] *3D Function Grapher v.1.3*, http://www.math.uri.edu/~bkaskosz/flashmo/graph3d2 (dostęp: 9.10.2015)

A particle swarm optimization algorithm for finding global extrema of some benchmark functions

Abstract

In this paper, we present the particle swarm optimization algorithm for finding the global extrema of several single and multimodal functions. The values of the algorithm parameters which ensure its best performance are determined on the basis of the computational results.

Keywords – Particle swarm optimization, global extremum, benchmark functions