

Andrzej GRZYB, Tomasz KUCZEKPOLITECHNIKA KRAKOWSKA, INSTYTUT POJAZDÓW SZYNOWYCH,
al. Jana Pawła II 37, 31-864 Kraków**Porównanie skuteczności wybranych dwu metod optymalizacji****Prof. nzw. dr hab. inż. Andrzej GRZYB**

Kierownik Zakładu Dynamiki i Konstrukcji Pojazdów Szynowych, Wiceprzewodniczący Polskiego Towarzystwa Symulacji Komputerowej. W ramach działalności naukowej zajmuje się zagadnieniami dynamiki pojazdów oraz opracowywaniem i praktycznym zastosowaniem algorytmów optymalizacyjnych – głównie ewolucyjnych – do mechanizmów stosowanych w pojazdach szynowych i samochodowych.



e-mail: agrzyb@mech.pk.edu.pl

Dr inż. Tomasz KUCZEK

Ukończył studia wyższe na Politechnice Krakowskiej w 2004 roku, po których rozpoczął pracę naukowo-badawczą na Wydziale Mechanicznym PK. W 2009 roku uzyskał stopień doktora nauk technicznych. Obecnie pracuje jako adiunkt w Instytucie Pojazdów Szynowych Politechniki Krakowskiej. Jego obszarem zainteresowań naukowych jest dynamika pojazdów i symulacja komputerowa, jest autorem kilkunastu prac naukowych i raportów badawczych.



e-mail: kuczek@m8.mech.pk.edu.pl

Streszczenie

W artykule zaprezentowano dwa stosunkowo nowe algorytmy stosowane do optymalizacji bez ograniczeń funkcji jednej lub wielu zmiennych. Są to algorytmy: ewolucji różnicowej oraz roju cząstek. Przedstawiono w skrócie cechy charakterystyczne algorytmów, najważniejsze informacje dotyczące zasad ich działania. Ponadto opisano sposób ich badania, mający na celu ocenę skuteczności tych algorytmów. Zamieszczono wyniki badań dotyczące kilku wybranych funkcji testowych oraz sformułowano uwagi dotyczące porównania skuteczności badanych metod.

Słowa kluczowe: optymalizacja, algorytmy genetyczne, algorytm roju cząstek, algorytm różnicowy.

Comparison of effectiveness of two selected optimisation methods**Abstract**

The paper presents two relatively new algorithms used for optimisation without limitations of single- or multi-variable functions. They are algorithms of differential evolution and particle swarm optimisation. The paper describes characteristic features of the two algorithms and provides vital information about their functioning. Moreover, the paper presents methods used to estimate the algorithm effectiveness. The comparison of efficiency was conducted on the basis of several specially selected test functions. The functions can be found in [5]. The optimum point is known for these functions. For each of the functions, numerous optimisations using various sequences of pseudorandom numbers were conducted [1]. The examination results for a few test function are given and the effectiveness of the tested methods is discussed. The algorithm of the differential evolution method is more reliable than that of the particle swarm method because the latter is often ineffective with multi-variable functions.

Keywords: optimisation, genetic algorithms, particle swarm optimisation, differential evolution.

1. Wstęp

Celem pracy jest porównywanie skuteczności wybranych dwu metod optymalizacyjnych: ewolucji różnicowej i roju cząstek. Metody te wykorzystują liczby pseudolosowe. Porównywanie metod przeprowadzono na podstawie odpowiednio dobranych kilku funkcji testowych wybranych spośród przebadanych kilkunastu. Funkcje te dostępne są w literaturze, przykładowo w [5]. Są to funkcje wielu zmiennych przeznaczane do optymalizacji bez ograniczeń. Dla funkcji tych znany jest punkt optymalny. Dla każdej z nich przeprowadzono wiele optymalizacji wykorzystując różne ciągi liczb pseudolosowych.

2. Omówienie algorytmów

Ewolucja różnicowa (Differential Evolution – DE) jest prostą, a jednocześnie bardzo skuteczną, ewolucyjną metodą optymalizacji. Została zaproponowana w 1995 r. przez Storn'a i Price'a [5].

Algorytm tej metody jest jednym z najlepszych algorytmów do optymalizacji funkcji, podobny do algorytmów genetycznych. Pozwala na optymalizację nieróżniczkowalnych i nieliniowych funkcji. Populacja potencjalnych rozwiązań znajdujących się w n -wymiarowej przestrzeni poszukiwań, składa się ze skończonej liczby uprzednio wylosowanych wektorów. Wektory te ewoluują w przestrzeni poszukiwań podczas optymalizacji w wyniku specyficznych mutacji i warunkowych krzyżowań przeprowadzanych z zadanym, ustalonym prawdopodobieństwem. Mutacje przeprowadzane są jedną, wybraną z kilku zaproponowanych strategii. Metoda umożliwia wyznaczenie – z dużym prawdopodobieństwem – ekstremum globalnego badanej funkcji.

Na podstawie wyników testowań kilkunastu problemów z różnymi funkcjami celu, autorzy [5] stwierdzili, że ich metoda jest znacznie skuteczniejsza – zwłaszcza w przypadkach trudniejszych – w porównaniu z metodami sympleksu Neldera i Meada oraz symulowanego wyzarzania (Adaptive Simulated Annealing). Optymalizacja rojem cząstek (Particle Swarm Optimization – PSO) to metoda inspirowana biologicznie na podstawie zachowania się stad ptaków i ławic ryb. Została zaproponowana w 1995 r. przez Eberhart'a i Kennedy'ego [2]. Przedstawia możliwe rozwiązania jako zbiór cząstek podobnych do stada ptaków lecących poprzez rozwiązania. Rój cząstek podąża za bieżącym, najlepszym rozwiązaniem, zmieniając kierunek i prędkość, gdy tylko zostanie wyznaczone lepsze rozwiązanie. Cząstki przystosowują się do środowiska poprzez zdolność zapamiętywania i powracania do wcześniej poznanych obszarów o najlepszych właściwościach. Zdaniem autorów metody, badania dotyczące zastosowania metody roju cząstek pozwalają stwierdzić, że taka optymalizacja może umożliwiać wyznaczanie w krótszym czasie lepszych rozwiązań złożonych problemów.

3. Badania i funkcje testowe

W celu oceny algorytmów wykonano pewną liczbę optymalizacji, tak aby w każdej z nich średnia liczba obliczeń funkcji celu była w przybliżeniu niezmienna. Użyto kilku znanych z literatury funkcji testowych bez ograniczeń, dla których punkt optymalny jest znany. Szczególną uwagę zwrócono na funkcje wykorzystywane do badania algorytmów ewolucyjnych, dla których klasyczne metody optymalizacji są zwykle nieskuteczne. Wykorzystano także następujące parametry:

$$e_m = \frac{1}{l_{opt}} \sum_{j=1}^{l_{opt}} \Delta f_j, \quad \sigma_e = \sqrt{\frac{1}{l_{opt}} \sum_{j=1}^{l_{opt}} \Delta f_j^2 - e_m^2}, \quad \Delta f_j = f_{opt j} - f_{min} \quad (1)$$

gdzie: e_m – średni błąd obliczania wartości f_{min} w l_{opt} optymalizacjach, f_{min} – wartość funkcji celu $f(\mathbf{x})$ w punkcie \mathbf{x}_{min} , $f_{opt j}$ – najlepsza wartość funkcji $f(\mathbf{x})$ obliczona podczas minimalizacji numer j , l_{opt} – liczba optymalizacji wykonywanych z różnymi ciągami

gami stosowanych liczb pseudolosowych, σ_e – odchylenie standardowe błędów obliczania wartości f_{min} w l_{opt} optymalizacjach.

Do badań wykorzystano poniższe funkcje oraz przyjęte dla nich wektory \mathbf{x}_s i \mathbf{u} wyznaczające wstępne obszary poszukiwania rozwiązań. Wektory: \mathbf{x}_s oraz \mathbf{u} są zadawane. Pierwszy z nich wyznacza środek obszaru, w którym tworzona jest populacja pierwotna, a drugi służy do wyznaczenia ograniczeń dolnych \mathbf{x}_{low} i górnych \mathbf{x}_{upp} przedziałów obszaru tej populacji.

$$\mathbf{x}_{low} = \mathbf{x}_s - \mathbf{u}, \quad \mathbf{x}_{upp} = \mathbf{x}_s + \mathbf{u}. \quad (2)$$

1. Funkcja stanowiąca formę kwadratową n zmiennych [3]:

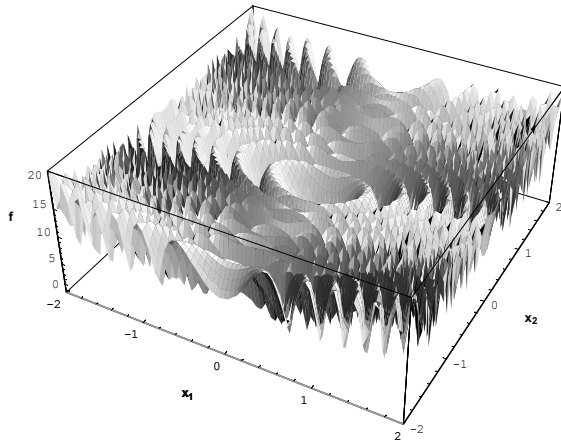
$$f(\mathbf{x}) = \sum_{j=1}^n (j x_j^2), \quad (3)$$

której minimum globalne równe 0 jest w punkcie $\mathbf{x}_{min} = (0, 0, \dots)$. Do analizy przyjęto: $n = 3$, $\mathbf{x}_s = (1, 1, 1)$ i $\mathbf{u} = (2, 2, 2, 2, 2)$.

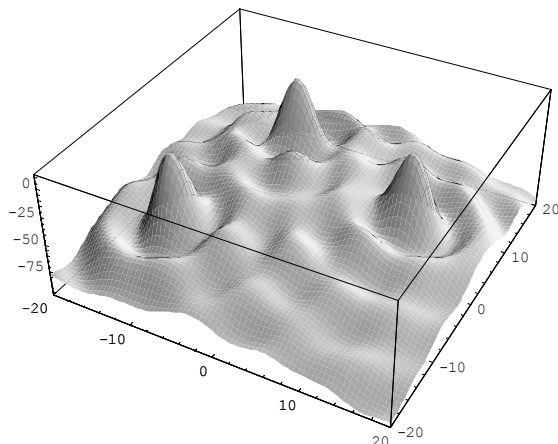
2. Funkcja pochodząca z pracy [4] i zaczerpnięta z programu GlobalOptimization.nb dostępnego na stronie internetowej firmy WOLFRAM RESEARCH:

$$f(\mathbf{x}) = x_1^2 + x_2^2 + \ln[1 + 20|\sin(x_1^2 + x_2^2)|] + 10|\sin(5x_1^2 + 15x_2^2)| \quad (4)$$

Jej minimum globalne jest równe 0 w punkcie $\mathbf{x}_{min} = (0, 0)$. Przyjęto: $\mathbf{x}_s = (0, 0)$ i $\mathbf{u} = (2, 2)$.



Rys. 1. Wykres funkcji (4)
Fig. 1. Plot of function (4)



Rys. 2. Wykres funkcji (7)
Fig. 2. Plot of function (7)

3. Uogólniona funkcja Rosenbrocka [6]:

$$f(\mathbf{x}) = \sum_{j=1}^{n-1} [(1-x_j)^2 + 100(x_{j+1} - x_j^2)^2], \quad (5)$$

której minimum globalne jest równe 0 w punkcie o współrzędnych:

$$x_j = 1, \quad j = 1, 2, \dots, n.$$

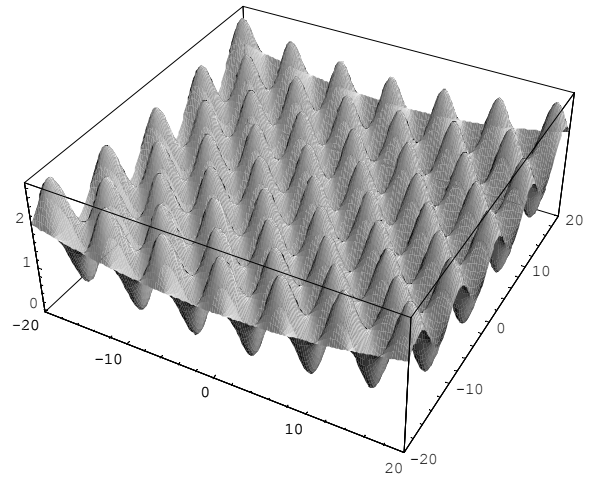
Przyjęto [3]:

$$\mathbf{x}_s = (-1, 2, 1, -1, 2, 1, \dots), \\ \mathbf{u} = (2, 5, 2, 5, \dots, 2, 5)$$

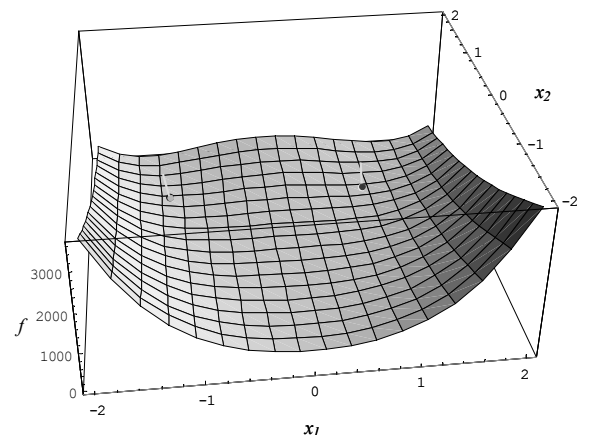
oraz $n = 2, 3, 7$ i 15 .

4. Funkcja Griewangka pochodząca z prac [3, 5]:

$$f(\mathbf{x}) = \frac{1}{400n} + \sum_{j=1}^n x_j^2 - \prod_{j=1}^n \cos\left(\frac{x_j}{\sqrt{j}}\right). \quad (6)$$



Rys. 3. Wykres funkcji (6)
Fig. 3. Plot of function (6)



Rys. 4. Wykres funkcji (3)
Fig. 4. Plot of function (3)

Do optymalizacji przyjęto $n = 2$, $\mathbf{x}_s = (0, 0)$, $\mathbf{u} = (600, 600)$ oraz następnie $n = 10$, $\mathbf{x}_s = (0, 0, \dots, 0)$, $\mathbf{u} = (400, 400, \dots, 400)$. Minimum globalne tej funkcji jest równe 0 w punkcie $\mathbf{x}_{min} = \mathbf{x}_s$.

5. Funkcja określona wzorem:

$$f(x,y) = f_p(x+6,y-9) + f_p(x-11,y-3) + f_p(x+11,y+9)$$

$$f_p(x,y) = \frac{50 \sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}} - \sqrt{x^2+y^2}, \quad (7)$$

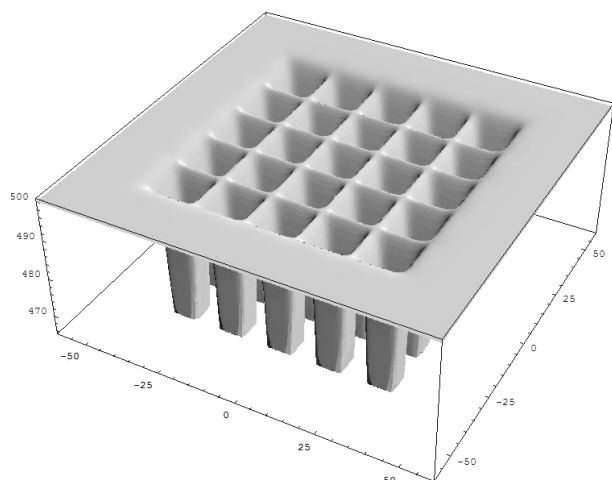
Do optymalizacji przyjęto $\mathbf{x}_s = (0, 0)$, $\mathbf{u} = (20, 20)$ Maksimum globalne tej funkcji jest równe 10,84317778119569 w punkcie $\mathbf{x}_{max} = (-6,017165863509005, 9,06021919972498)$.

6. Funkcja pochodząca z zestawu funkcji testowych De Jonga [4]:

$$f(\mathbf{x}) = \frac{1}{0,002 + \sum_{j=1}^{j=25} \frac{1}{(x_1 - a_{1,j})^6 + (x_2 - a_{2,j})^6 + j}} \quad (8)$$

Po licznych optymalizacjach można przyjąć, że minimum globalne tej funkcji $f_{min} = 0,9980038377944...$ jest w punkcie: $\mathbf{x}_{min} = (-31,97833483..., -31,97833483...)$. Do analizy przyjęto: $\mathbf{x}_s = (0, 0)$ oraz $\mathbf{u} = (65,536, 65,536)$. Współczynniki a_{ij} przyjmują wartości: -32, -16, 0, 16 i 32, stanowiące elementy następującej macierzy o wymiarach 2×25 :

$$\mathbf{a} = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & -32 & -16 & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 & 32 & 32 \end{bmatrix}$$



Rys. 5. Wykres funkcji (8)
Fig. 5. Plot of function (8)

Parametry optymalizacji dobierane były metodą próbnych testowań, w celu otrzymania najlepszych wyników. Założono, że liczba l_f obliczeń funkcji celu w przypadku algorytmu roju cząstek będzie podobna do liczby obliczeń algorytmu ewolucji różnicowej.

Wyniki zamieszczone w tabelicy 2 były uzyskane przy dodatkowym warunku – stosowanym również w pracy [5] – powodującym przerwanie działania odpowiednio skorygowanych procedur z badanymi metodami optymalizacyjnymi, gdy w najlepszym aktualnie wyznaczonym w nich punkcie \mathbf{x}_{opt} spełniony zostaje warunek:

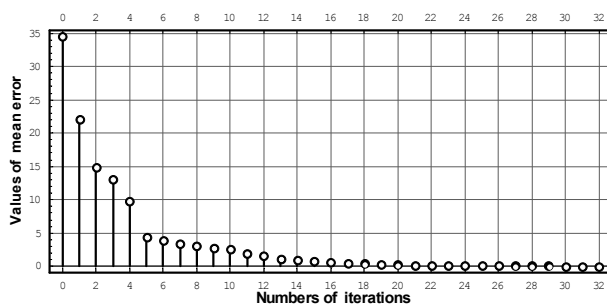
$$f(\mathbf{x}_{opt}) \leq f_{min} + 10^{-m}, \quad (9)$$

gdzie $m = 1, 2, \dots, 5$, dobierano odpowiednio do optymalizowanej funkcji. Jak można zauważyć, w przypadku funkcji $n_f = 3$ oraz $n_f = 4$ nie udało się uzyskać wymaganej dokładności wyników

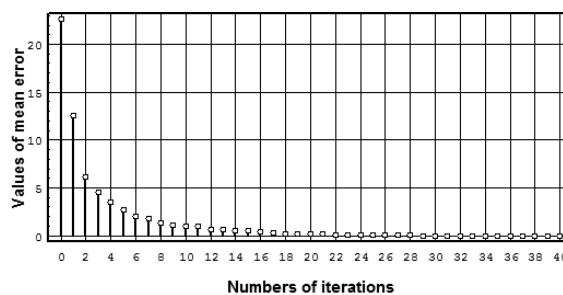
przy stosowaniu algorytmu roju cząstek, co świadczy o wyraźnym spadku jego skuteczności w przypadku wzrostu liczby zmiennych optymalizowanych funkcji. Brak wyników w przypadku metody roju cząstek zastosowanych do niektórych funkcji uwzględnianych w tabelicy 2 wynikał z braku możliwości uzyskania sensownych wyników. Oznacza to, że metoda ta była nieefektywna.

Tab. 1. Porównawcze wyniki optymalizacji
Tab. 1. Comparative results of optimisation

n_f	Metoda	l_{opt}	l_f	l_p	e_m	σ_m
1	PSO	64	2310	30	0.00002253	0.00001768
	DE		2293	20	$7.329 \cdot 10^{-32}$	$9.625 \cdot 10^{-32}$
2	PSO	64	3040	20	0,0001188	0,0001522
	DE		3044	40	$3.508 \cdot 10^{-24}$	$1,159 \cdot 10^{-23}$
3 $n = 2$	PSO	128	1620	20	0,001622	0,002612
	DE		1612	20	$2.691 \cdot 10^{-6}$	$2,916 \cdot 10^{-5}$
3 $n = 3$	PSO	64	5400	25	0.05148	0,018839
	DE		5389	30	$4,467 \cdot 10^{-9}$	$3,178 \cdot 10^{-8}$
3 $n = 7$	PSO	4	95450	25	0,319406	0,071
	DE		95463	40	$1.423 \cdot 10^{-24}$	$2,459 \cdot 10^{-24}$
3 $n = 15$	PSO	8	275300	140	12,28	0,9536
	DE		275301	100	0,4983	1,318
4 $n = 2$	PSO	32	8580	130	0,01889	0,01652
	DE		8556	140	$1,591 \cdot 10^{-8}$	$8,437 \cdot 10^{-8}$
4 $n = 10$	PSO	16	40080	120	0.5824	0.1215
	DE		40004	40	0,03148	0,01227
5	PSO	64	4040	40	0,002272	0,002991
	DE		4035	40	0,00001325	0,00009784
6	PSO	32	2970	90	0,0111	0,024861
	DE		2979	90	$2.708 \cdot 10^{-9}$	$9.142 \cdot 10^{-9}$



Rys. 6. Wykres zmian błędu średniego dla $n_f = 6$ i metody roju cząstek
Fig. 6. Changes of the average error for $n_f = 6$ and the PSO method



Rys. 7. Wykres zmian błędu średniego dla $n_f = 6$ i metody algorytmu różnicowego
Fig. 7. Changes of the average error for $n_f = 6$ and the DE method

Wykresy zamieszczone na rysunkach 6 i 7 przedstawiają przebiegi zmian wartości błędu średniego e_m w kolejnych iteracjach dla funkcji $n_f = 6$ pochodzącej z zestawu funkcji testowych De Jonga, dla każdej z rozpatrywanych metod. Wyniki dotyczą itera-

cji wstępnej, oznaczonej numerem 0 i iteracji zasadniczych o numerach zaczynających się od 1.

Tab. 2. Porównawcze wyniki minimalizacji
Tab. 2. Comparative results of minimisation

n_f	Metoda	l_{opt}	l_p	m	l_f
1	PSO	64	25	4	1643
	DE		20	5	387
2	PSO	64	20	3	1428
	DE		40	5	965
3 $n=2$	PSO	128	20	4	4546
	DE			5	736
3 $n=3$	PSO	64	360	2	47458
	DE		30	5	2774
3 $n=7$	PSO	4	25	1	50025
	DE		40	5	30077
3 $n=15$	PSO	8	25	-	-
	DE		100	5	91909
4 $n=2$	PSO	32	130	3	46466
	DE		140	5	5850
4 $n=10$	PSO	16	25	-	-
	DE		100	5	77572
5	PSO	64	40	1	1238
				2	2301
	DE		60	1	1606
				2	2098
6	PSO	32	90	3	2731
	DE				1244
	PSO	32	90	1	1654
	DE				949

Końcowe wartości błędów widocznych na wykresach zamieszczone są w tabeli 1. W metodzie algorytmu ewolucji różnicowej krzyżowania wykonywane są z zadaniem prawdopodobieństwem w tym przypadku równym 0,3. Przy podobnej liczbie obliczeń

wartości funkcji celu i w związku z tym podobnym czasie optymalizacji, w metodzie ewolucji różnicowej może być wykonanych więcej iteracji niż w przypadku metody roju cząstek, co widoczne jest na rysunkach 6 i 7.

4. Uwagi końcowe

Na podstawie przedstawionych w pracy porównań wybranych metod znacznie lepszą metodą optymalizacji jest metoda ewolucji różnicowej. Uzasadnieniem podanego stwierdzenia są przedstawione wyniki analiz. W przypadku metody roju cząstek, wartości funkcji celu bardzo wolno zbliżają się do ekstremum globalnego, przez co uzyskanie dużej dokładności jest bardzo trudne przy porównywalnej liczbie jej obliczeń. Podsumowując, algorytm ewolucji różnicowej jest bardziej niezawodny niż algorytm roju cząstek, gdyż ten ostatni jest często nieskuteczny w przypadku funkcji wielu zmiennych, co można zauważyć zapoznając się z wynikami zamieszczonymi w tablicach.

5. Literatura

- [1] Grzyb A.: Ekstrapolacyjny algorytm ewolucyjny. Polioptymalizacja i komputerowe wspomaganie projektowania, Wydawnictwo Uczelniane Politechniki Koszalińskiej, Koszalin 2005.
- [2] Kennedy J. and Eberhart R.C.: Particle swarm optimization. International Conference on Neural Networks, Piscataway, NJ, 1995.
- [3] Koulocheris D., Vrazopoulos H., Dertimanis V.: Optimization Methodology for Tuning Fuzzy Logic Controllers. 15th Triennial World Congress, Barcelona 2002.
- [4] Pintér J.D.: Global Optimization in Action. Kluwer Academic Publishers. 1996.
- [5] Storn R., Price K.: Differential Evolution – a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95 012, ICSI, 1995.
- [6] Wit R.: Metody programowania nieliniowego. WNT Warszawa 1986.

otrzymano / received: 25.05.2011

przyjęto do druku / accepted: 03.10.2011

artykuł recenzowany

INFORMACJE

Newsletter PAK

Wydawnictwo PAK wysyła drogą e-mailową do osób zainteresowanych Newsletter PAK, w którym są zamieszczane:

- spis treści aktualnego numeru miesięcznika PAK,
- kalendarz imprez branżowych,
- ważniejsze informacje o działalności Wydawnictwa PAK.

Newsletter jest wysyłany co miesiąc do osób, które w jakikolwiek sposób współpracują z Wydawnictwem PAK (autorzy prac opublikowanych w miesięczniku PAK, recenzenci, członkowie Rady Programowej, osoby które zgłosiły chęć otrzymywania Newslettera).

Celem inicjatywy jest umocnienie w środowisku pozycji miesięcznika PAK jako ważnego i aktualnego źródła informacji naukowo-technicznej.

Do newslettera można zapisać się za pośrednictwem:

- strony internetowej: www.pak.info.pl, po dodaniu swojego adresu mailowego do subskrypcji,
- adresu mailowego: wydawnictwo@pak.info.pl, wysyłając swoje zgłoszenie.

Otrzymywanie Newslettera nie powoduje żadnych zobowiązań ze strony adresatów. W każdej chwili można zrezygnować z otrzymywania Newslettera.