

Mgr inż. Łukasz WITANOWSKI

DOI: 10.17814/mechanik.2015.7.318

Dr hab. inż. Piotr LAMPART

Instytut Maszyn Przepływowych PAN, Ośrodek Energetyki Ciepłej – Zakład Turbin

WYKORZYSTANIE ALGORYTMÓW PSZCZELICH W PROCESIE POSZUKIWANIA MINIMUM FUNKCJI CELU

Streszczenie: Algorytmy pszczele są jednymi z najczęściej wykorzystywanych algorytmów bazujących na sztucznej inteligencji. ABC oraz BCO symulują poszukiwanie źródła pokarmu przez pszczoły miodne. W poniższej pracy przedstawiono zastosowanie algorytmów pszczelich do wyznaczenia minimum wybranych funkcji testowych. Algorytmy pszczele porównano z metodą prostego poszukiwania oraz z algorytmem genetycznym. Całość przetestowano na trzynastu funkcjach. Zastosowane metody zostaną w przyszłości wykorzystane do rozwiązywania problemów optymalizacyjnych z zakresu sprawności urządzeń energetycznych.

USING OF A BEE ALGORITHMS FOR SEARCHING PROCESS OF MINIMUM OF OBJECTIVE FUNCTION

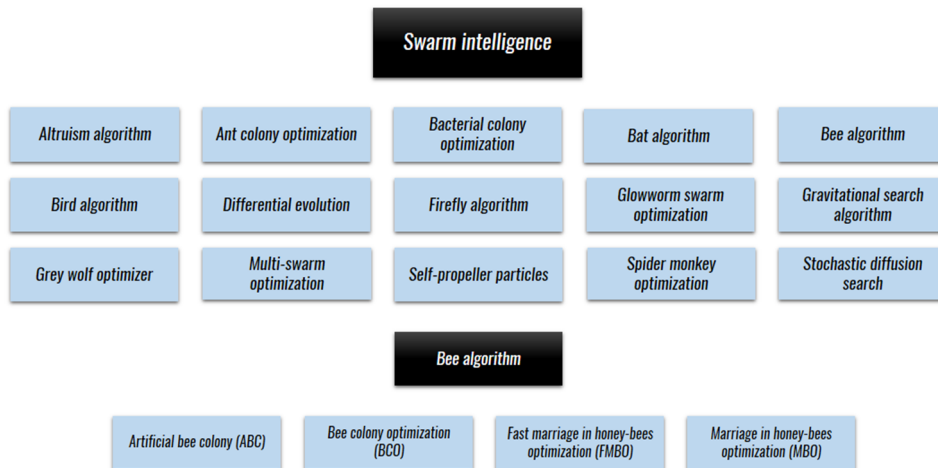
Abstract: Bee Algorithm is one of the most often used swarm-based algorithm. ABC and BCO simulates the foraging behavior of a honeybee swarm. In this article an application of bee algorithms to determine the minimum of a test function is presented. Bee algorithms are compared with simple search method and genetic algorithm. All algorithm are tested on 13 functions. The presented methods will in future be applied for solving optimization problems related to the efficiency of power engineering devices.

Słowa kluczowe: optymalizacja, inteligencja stada, funkcja celu, Artificial Bee Colony, Bee Colony Optimization

Keywords: optimization, swarm intelligence, objective function, Artificial Bee Colony, Bee Colony Optimization

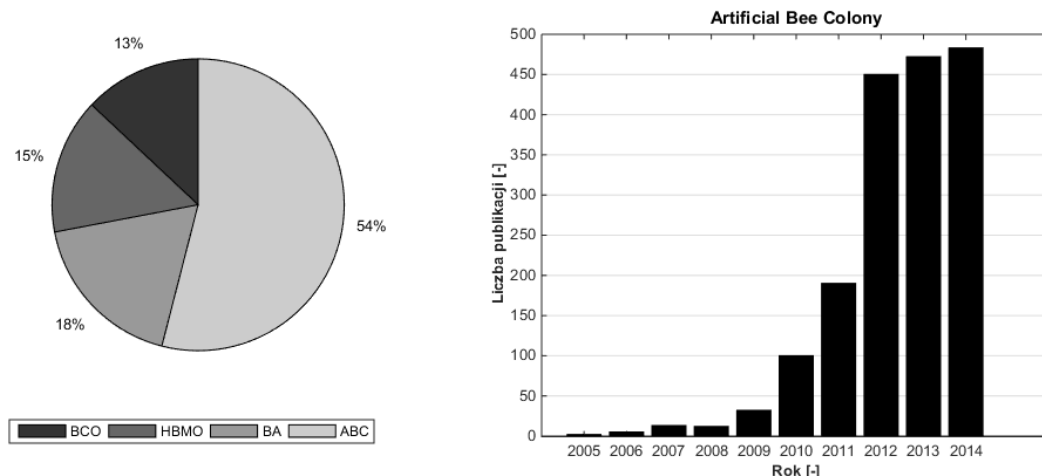
1. WPROWADZENIE

W ostatnich latach znacznie wzrosło zainteresowanie technikami optymalizacyjnymi bazującymi na inteligencji stada. Dynamicznie rozwijane są takie metody jak: Ant Colony Optimization [1, 2], Artificial Bee Colony [3, 4], Bat algorithm [5], Bee Colony Optimization [6], Gravitational Search Algorithm [7], Glowworm swarm optimization [8], Grey Wolf Optimizer [9], Spider Monkey Optimization [10]. Zestawienie algorytmów przedstawiono na rys. 1.



Rys. 1. Metody optymalizacyjne bazujące na zachowaniu stada

W grupie algorytmów pszczelich powstaje wiele nowych metod optymalizacyjnych, np.: Fast Marriage in Honey-Bees Optimization (FMBO) [11], których wyniki potwierdzają wysoką wydajność algorytmów pszczelich. Ponadto algorytmy rojowe są modyfikowane, łączone z dostępnymi algorytmami stochastycznymi i deterministycznymi w celu podwyższenia skuteczności metody. Wśród algorytmów pszczelich najczęściej publikowane są prace opisujące ABC (Artificial Bee Colony) i jego modyfikacje (rys. 2). Wzrost publikacji, w których wykorzystuje się ABC, obserwuje się na rys. 2. W roku 2005 powstały dwie prace dotyczące tego algorytmu, natomiast w roku 2014 liczba publikacji wzrosła aż do blisko 500. W pracy porównano algorytm ABC, BCO z algorytmem genetycznym [12] oraz metodą prostego poszukiwania minimum funkcji celu – metodą bezpośrednią [13].

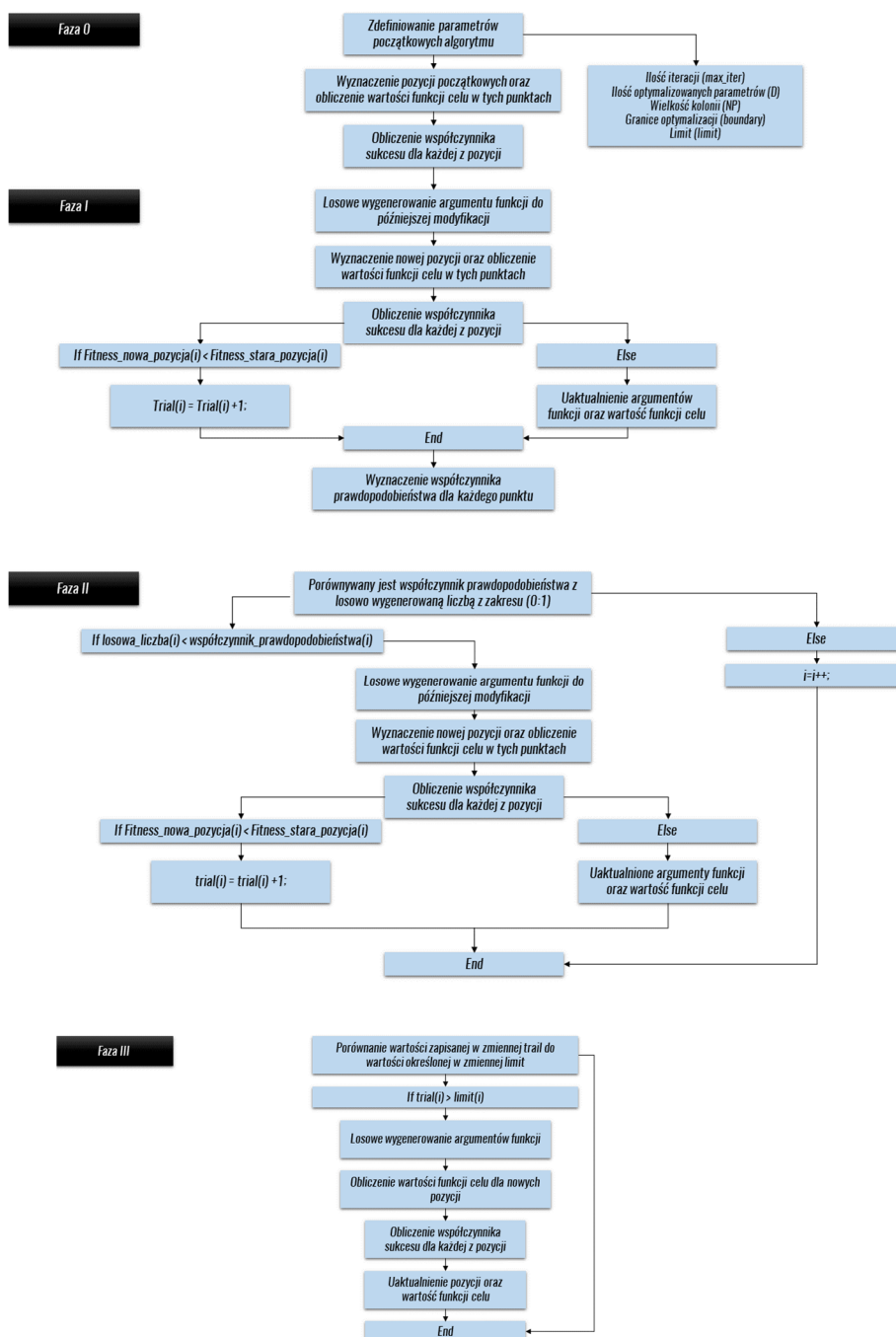


Rys. 2. Podział publikacji dotyczących algorytmów pszczelich (rys. lewy), liczba publikacji dotyczących Artificial Bee Colony na przestrzeni lat

2. OPIS ALGORYTMÓW OPTYMALIZACYJNYCH

2.1. Artificial Bee Colony (ABC)

Metoda optymalizacji bazuje na modelu zaproponowanym przez Tereshko and Loengarov [14], który został zainspirowany zachowaniem pszczół miodnych. Algorytm zaimplementował Dervis Karaboga w 2005 roku [15]. Model składa się z trzech zasadniczych elementów: pracujących i bezrobotnych pszczół oraz źródła pokarmu. Pierwsze dwa elementy, a więc pszczoły, poszukują miejsc bogatych w pożywienie. Na rys. 3 przedstawiono zasadę działania algorytmu.



Rys. 3. Schemat działania algorytmu ABC (Artificial Bee Colony)

W opisywanym algorytmie wartość współczynnika sukcesu zależy od wartości funkcji celu. Jeżeli dla danej pozycji wartość funkcji celu jest większa bądź równa zero ($f(x) \geq 0$), wartość współczynnika sukcesu obliczana jest według wzoru:

$$fit = \frac{1}{f(x)+1}. \quad (1)$$

W innym przypadku, a więc gdy wartość funkcji celu jest mniejsza od zera ($f(x) < 0$), współczynnik sukcesu obliczany jest ze wzoru:

$$fit = 1 + f(x). \quad (2)$$

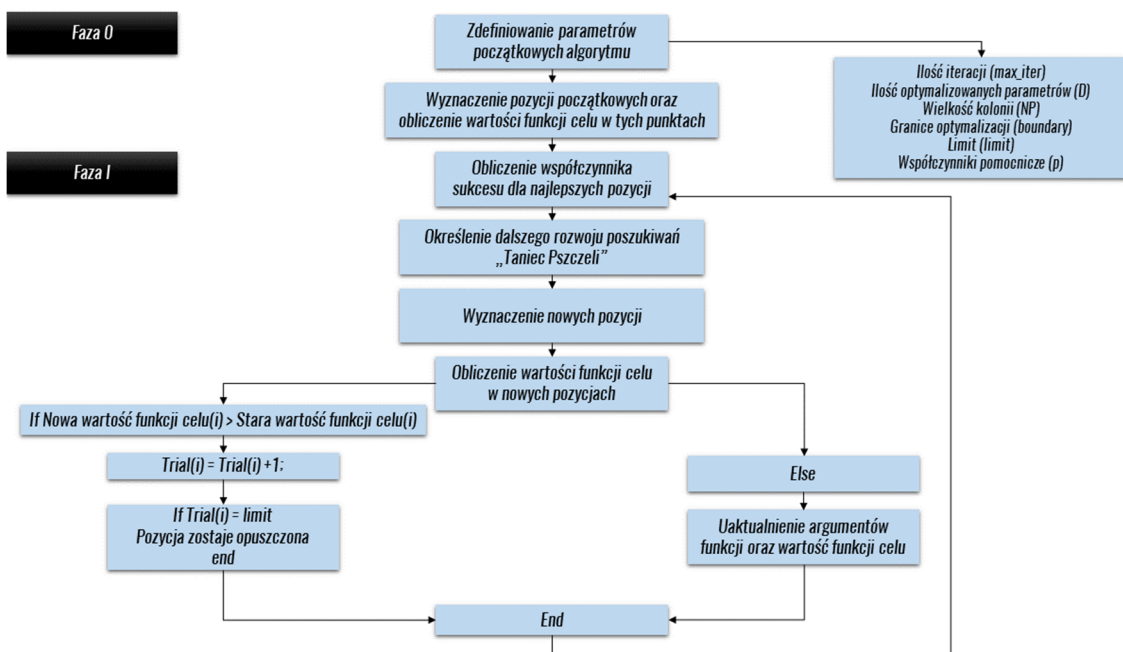
Współczynnik prawdopodobieństwa obliczany jest na podstawie poniższego wzoru:

$$prob = \frac{0.9 * fit(i)}{\max_fit} + 1. \quad (3)$$

Otrzymana wartość zostaje porównana z losowo wygenerowaną liczbą z zakresu 0:1. Jeżeli wygenerowana wartość jest mniejsza od współczynnika prawdopodobieństwa, pozycja zostaje przyjęta i przechodzi do fazy II.

2.2. Bee Colony Optimization (BCO)

Algorytm BCO został zaproponowany przez Teodorovic i Dell'Orco w 2005 roku [6]. Populacja składa się z zadeklarowanej liczby pszczoł NP. Każdej pszczole przyporządkowane jest jedno rozwiązanie. Podczas pojedynczej iteracji wykonywane są dwie fazy (*forward pass and backward pass*), które stanowią jeden krok algorytmu. Schemat działania algorytmu został przedstawiony na rys. 4.



Rys. 4. Schemat działania algorytmu BCO (Bee Colony Optimization)

Istotnym elementem algorytmu jest tzw. pszczeli taniec. W algorytmie BCO po powrocie do ula taniec wykonuje jedynie ta pszczoła, która znalazła lepsze źródło pokarmu niż w poprzednim poszukiwaniu. Podczas tańca pozostałe pszczoły nakłaniane są do przeszukiwania źródła tańczącego. Przed opuszczeniem ula obliczana jest jakość źródła ($Pf(i)$) oraz średnia jakość wszystkich źródeł Pf_{colony} . Na podstawie tabeli 1 określana jest liczba

pszczoł przeszukująca poszczególne źródła. W przypadku $P_{follow} = 0$ żadna pszczoła nie obserwuje tańca, więc do źródła pokarmu wraca jedynie pszczoła tańcząca.

$$Pf(i) = \frac{1}{f(x)} \quad (4)$$

$$Pf_{colony} = \frac{1}{NP} \sum_{i=1}^{NP} Pf(i) \quad (5)$$

Tabela 1. Liczba pszczoł podążających za jednostką rekrutującą

Opis	P_{follow}
$Pf(i) < 0.95 Pf_{colony}$	0.80
$0.95 Pf_{colony} \leq Pf(i) < 0.975 Pf_{colony}$	0.20
$0.975 Pf_{colony} \leq Pf(i) < 0.99 Pf_{colony}$	0.02
$0.99 Pf_{colony} \leq Pf(i)$	0.00

3. FUNKCJE TESTOWE

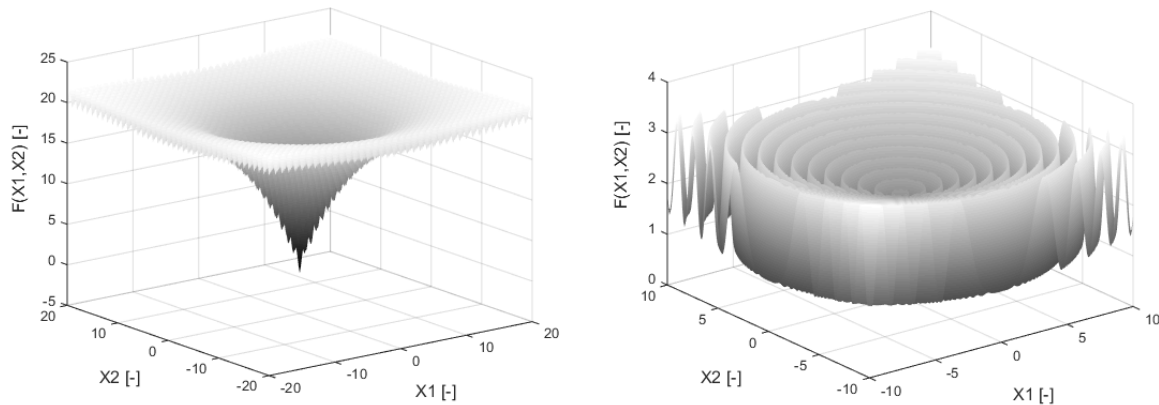
W procesie testowania poprawności pracy algorytmów wykorzystano 13 funkcji [16], których charakterystykę przedstawiono w tabeli 2.

Tabela 2. Zestawienie funkcji testujących

Lp.	Nazwa funkcji	Ilość wymiarów	Parametry funkcji				
			x1	x2	lb	ub	$y(x_1, \dots, x_n)$
1	Beale function	2	3	0.5	-10	10	0
2	Branins function	2	9.4248	2.475	-5	5	0.4063
			π	2.275			
			$-\pi$	12.275			
3	Cosine Mixture function	2	0	0	-10	10	-199.8
4	Damavandi function	2	2	2	0	14	0
5	Easom function	2	0	0	-100	100	0
6	McCormick function	2	-0.5472	-1.5472	-10	10	-10.122
7	Rastrigin function	2	0	0	-10	10	0
8	Rosenbrock function	2	1	1	-30	30	0
9	Salomon function	2	0	0	-100	100	0
10	Schweffe36 function	2	1	1	-10	10	-11200
11	Shubert function	2	-7.0835	4.858	-100	100	-186.724
12	Six Hump Camel function	2	0.0984	-0.7127	-100	100	-1.03163
			-0.0894	0.7127			
13	Zacharov function	2	0	0	-100	100	0

Funkcje te różnią się liczbą minimów lokalnych/globalnych, ich gradientem oraz kształtem w całej dziedzinie. Przejście przez proces testowania pozwala na bliższe poznanie charakterystyki algorytmu, tzn. zdolność do odnalezienia minimum globalnego czy też podatność na wpadnięcie w obszar minimów lokalnych, szybkość pracy algorytmu oraz ograniczenia. Proces testowania pozwala również na konfrontację algorytmu z ogólnie dostępnymi metodami poszukiwania funkcji celu. W procesie testowania algorytmów każdy

z nich został wywołany 100-krotnie. Dla każdego z nich określono współczynnik sukcesu, średnie oraz standardowe odchylenie. Na rysunku 5 przedstawiono funkcję Easom [16] oraz Salomon [16], w których algorytmy osiągnęły najniższy współczynnik sukcesu (SR).



Rys. 5 Wykres funkcji Easom (rys. lewy) oraz funkcji Salomon (rys. prawy)

$$f_{EASOM}(x) = -\cos x_1 \cos x_2 e^{-(x_1-\pi)^2-(x_2-\pi)^2} \quad (6)$$

$$f_{SALOMON}(x) = 1 - \cos(2\pi \sqrt{\sum_{i=1}^D x_i^2}) + 0.1 \left(\sqrt{\sum_{i=1}^D x_i^2} \right) \quad (7)$$

4. WYNIKI

Wszystkie algorytmy zostały przetestowane przy różnych konfiguracjach startowych oraz kryteriach zatrzymania. W wynikach przedstawionych w tabeli 2 jako kryterium zatrzymania funkcji zdefiniowano liczbę wywołań funkcji testowych (1500 wywołań).

Tabela 3. Zestawienie wyników

Lp	Nazwa funkcji	Algorytm Genetyczny		Metoda bezpośrednia		Artificial Bee Colony		Bee Colony Optimization	
		SR [-]	SD [-]	SR [-]	SD [-]	SR [-]	SD [-]	SR [-]	SD [-]
1	Beale function	100.00	0.219	100	0	100	0.017	100	3.246
2	Branins function	99.92	0	99.92	0	99.92	1E-04	99.89	0.872
3	Cosine Mixture function	100.00	0	100	0	100	0	100	0
4	Damavandi function	99.85	0.945	99.77	0	99.9	0.992	99.84	1.054
5	Easom function	70.75	7.205	100	0	95.48	1.174	77.75	6.42
6	McCormick function	99.72	1.574	98.88	0	99.92	0.662	99.6	3.18
7	Rastrigin function	99.65	1.61	100	0	99.98	0.148	99.2	3.759
8	Rosenbrock function	100.00	9.189	100	0	100	0.709	99.99	1.2
9	Salomon function	95.92	0.442	100	0	99.24	0.05	97.48	0.7
10	Schweffe36 function	100.00	0.063	56.52	0	100	0	96.01	2049
11	Shubert function	98.54	21.13	100	0	99.55	1.968	98.47	18.69
12	Six Hump Camel function	100.00	0.308	100	0	100	0.007	99.6	0.92
13	Zacharov function	100.00	0	100	0	100	3.734	100	406.8

Ponadto każdy z algorytmów posiada własne parametry charakterystyczne. W przypadku algorytmów pszczelich ustawiono: liczbę źródeł – 40, liczbę prób poprawy źródła – 4. W algorytmie genetycznym zdefiniowano liczbę populacji na 50. Wszystkie obliczenia wykonano w środowisku Matlab 2014. Na podstawie symulacji stwierdzono:

- ABC jest algorytmem stabilnym, którego wyniki w większości przypadków pokrywają się z tradycyjnymi algorytmami, a w przypadku niektórych funkcji jego rezultaty są znacznie lepsze (np. Easom Function).
- BCO osiągnął niższe wartości współczynnika sukcesu (SR) niż ABC, lecz pomimo tego pozostaje obiecującą metodą optymalizacyjną.
- BCO osiągnął wyższe wartości standardowego odchylenia (SD), co świadczy o dużej rozpiętości wyników algorytmu i jego mniejszej stabilności.

5. WNIOSKI

W artykule przedstawiono zastosowanie dwóch algorytmów pszczelich oraz ich porównanie z algorytmem genetycznym i metodą optymalizacji bezpośredniej. Wszystkie metody optymalizacji przetestowano na trzynastu funkcjach testujących. Na podstawie wyników stwierdzono, że algorytmy pszczele są uniwersalnymi metodami optymalizacji, które w niektórych przypadkach osiągają lepsze wyniki niż tradycyjne algorytmy optymalizacyjne. Badania zostaną rozszerzone o kolejne funkcje testujące oraz realne problemy optymalizacyjne z zakresu sprawności urządzeń energetycznych.

LITERATURA

- [1] Abadeh M.S., Habibi J.: *A Hybridization of Evolutionary Fuzzy Systems and Ant Colony Optimization for Intrusion Detection*, ISC Int. J. Inf. Secur. 2, 2015.
- [2] Lenin K., Reddy R., Kalvathi S.: *Integration of Particle Swarm & Ant Colony Algorithm for Solving Optimal Reactive Power Dispatch Problem*, J. Electr. Control Eng. 4, 2014, 10-15.
- [3] Mirsadeghi E., Shariat Panahi M.: *Hybridizing Artificial Bee Colony with Simulated Annealing*, Int. J. Hybrid Inf. Technol. 5, 2012, 11-18.
- [4] Wu B., Hua Qian C.: *Differential Artificial Bee Colony Algorithm for Global Numerical Optimization*, 2011, 8.
- [5] Yang X.-S.: *A New Metaheuristic Bat-Inspired Algorithm*, in: Nat. Inspired Coop. Strategies Optim. (NISCO 2010), Springer Berlin Heidelberg, Berlin, 2010, 65-74.
- [6] Teodorovic D., Lucic P., Markovic G., Orco M.D.: *Bee Colony Optimization: Principles and Applications*, in: 2006 8th Semin. Neural Netw. Appl. Electr. Eng., IEEE, 2006, 151-156.
- [7] Rashedi E., Nezamabadi-pour H., Saryazdi S.: *GSA: A Gravitational Search Algorithm*, Inf. Sci. (Ny). 179, 2009, 2232-2248.

- [8] Krishnanand K.N., Ghose D.: *Theoretical foundations for rendezvous of glowworm-inspired agent swarms at multiple locations*, Rob. Auton. Syst. 56 (2008), 549-569.
- [9] Mirjalili S., Mirjalili S.M., Lewis A.: *Grey Wolf Optimizer*, Adv. Eng. Softw. 69, 2014, 46-61.
- [10] Kumar S., Kumar Sharma V., Kumari R.: *Self-Adaptive Spider Monkey Optimization Algorithm for Engineering Optimization Problems*, Int. J. Information, Commun. Comput. Technol. II. 2014, 96-107.
- [11] Yang C., Chen J., Tu X.: *Algorithm of Fast Marriage in Honey Bees Optimization and Convergence Analysis*, in: 2007 IEEE Int. Conf. Autom. Logist., IEEE, 2007, 1794-1799.
- [12] Goldberg D.E.: *Genetic Algorithms in Search, Optimization & Machine Learning*, 1989.
- [13] Audet C., Dennis J.E.: *Analysis of generalized pattern searches*, SIAM J. Optim. 13, 2003, 889-903.
- [14] Tereshko V., Loengarov A.: *Collective Decision-Making in Honey Bee Foraging Dynamics*, Sch. Comput. Univ. Paisley, 2015.
- [15] Karaboga D., *An Idea Based On Hoey Bee Swarm For Numerical Optimization*, 2005.
- [16] Jamil M., Yang X.-S.: *A Literature Survey of Benchmark Functions For Global Optimization Problems*, Int. J. Math. Model. Numer. Optim. 4, 2013, 150-194.

PODZIĘKOWANIA

Praca została wykonana z wykorzystaniem Infrastruktury PL Grid oraz na licencji Matlab 2014b otrzymanej z Centrum Informatycznego Trójmiejskiej Akademickiej Sieci Komputerowej.



PROGRAM STRATEGICZNY NCBR

Realizacja zadania badawczego jest dofinansowana przez Narodowe Centrum Badań i Rozwoju w ramach Strategicznego Programu Badań Naukowych i Prac Rozwojowych pt: "Zaawansowane Technologie Pozyskiwania Energii" oraz współfinansowana przez ENERGA SA

Zadanie Badawcze nr 4

OPRACOWANIE ZINTEGROWANYCH TECHNOLOGII WYTWARZANIA PALIW I ENERGII Z BIOMASY, ODPADÓW ROLNICZYCH I INNYCH