

Studium przypadku skuteczności nowych metod optymalizacji roju w porównaniu do metod znanych.

Mgr inż Jan Baumgart

Mgr inż Belco Sangho

¹ Uniwersytet Kazimierza Wielkiego, Instytut Informatyki, jan.baumgart@ukw.edu.pl

² Uniwersytet Kazimierza Wielkiego, Instytut Informatyki, belco.sangho@ukw.edu.pl

Streszczenie: Porównanie skuteczności nowych metod optymalizacji roju w porównaniu z metodami znanymi w dziedzinie. Inspirowane naturą algorytmy metaheurystyczne stają się coraz bardziej popularne w rozwiązywaniu problemów optymalizacyjnych. Dzięki ich popularności niemal codziennie możemy zobaczyć nowe podejścia i proponowane rozwiązania. W tym artykule przedstawię porównanie, które pokaże kilka najnowszych prac z tej dziedziny w porównaniu z niektórymi algorytmami traktowanymi jako podstawa dziedziny. Głównym celem było porównanie ostatnio wprowadzonych algorytmów roju i określenie, kiedy nowe rozwiązania są faktycznie szybsze i bardziej precyzyjne. Podsumowując, czy przetestowane nowe podejścia są lepsze niż obecne, dobrze znane i ugruntowane w terenie algorytmy. Algorytmy brane pod uwagę w tej pracy to: Particle Swarm Optimization [5], Artificial Bee Colony [3], Elephant Herding Optimization [7], Whale Optimization [4] i Grasshopper Optimization [6]. Algorytmy uznawane za nowe w tej dziedzinie porównano z dwoma popularnymi i dobrze znanymi algorytmami metaheurystycznymi pod względem dokładności znalezionych rozwiązań i szybkości. Zgodnie z wynikami eksperymentów większość porównywanych nowych algorytmów dawała zadowalające wyniki w użytkowaniu.

Słowa kluczowe: algorytm roju; wzorce; inspirowane naturą; metaheurystyka; pojedyncze obiektywne problemy optymalizacji; optymalizacja funkcji, algorytmy optymalizacji

Abstract: Comparing the effectiveness of new methods of swarm optimization in comparison with known methods. Nature-inspired metaheuristic algorithms are becoming more and more popular in solving optimization problems. Thanks to their popularity, we can see new approaches and proposed solutions almost every day. In this article, I will present a comparison that will show some of the most recent works in this field compared to some algorithms considered as the basis of the field. The main goal was to compare the recently introduced swarm algorithms and determine when new solutions are actually faster and more precise. In conclusion, are the new approaches tested better than the current, well-known and field-grounded algorithms? The algorithms considered in this paper are Particle Swarm Optimization, Artificial Bee Colony, Elephant Herding Optimization, Whale Optimization, and Grasshopper Optimization. Algorithms considered new in this field were compared with two popular and well-known metaheuristic algorithms in terms of accuracy of solutions found and speed. According to the experimental results, most of the compared new algorithms gave satisfactory results in use.

Keywords: swarm algorithm; patterns; inspired by nature; metaheuristics; single objective optimization problems; function optimization, optimization algorithms.

1. Wprowadzenie

Istnieje wiele opcji do wyboru spośród algorytmów optymalizacji roju, które zostały wprowadzone od „początku”. Wszystkie te algorytmy wykazały swój potencjał w rozwiązywaniu wielu problemów optymalizacyjnych i czasami są implementowane nawet w rzeczywistych problemach. Artykuł ten zawiera przegląd znanych algorytmów optymalizacyjnych oraz nowych, ostatnio proponowanych rozwiązań. Wybrane algorytmy są ze sobą wszechstronnie porównywane poprzez eksperymenty przeprowadzane z wykorzystaniem pięciu dobrze znanych funkcji porównawczych w celu ustalenia mocnych i słabych stron każdego algorytmu. Przeprowadzono szereg testów w celu określenia znaczących korzyści w zakresie wydajności każdego z nich. Dalsze części artykułu są uporządkowane w następujący sposób. W następnej sekcji znajdują się informacje o tym, które algorytmy zostały wybrane i dlaczego, następnie przedstawiono krótki opis każdego z algorytmów, szczególnie dotyczące funkcji do testowania optymalizacji. Na zakończenie prezentacja wyników eksperymentalnych przeprowadzonych w celu oceny zaproponowanego algorytmu oraz wnioski i tematy do dalszych badań.

2. Wybór algorytmów

Przedstawione w artykule algorytmy zostały podzielone na dwie grupy oraz wybrane na podstawie kilku określonych ściśle zasad. Grupy jakie zostały wydzielone to algorytmy uznane w dziedzinie za podstawę oraz nowo zaproponowane algorytmy w ciągu ostatnich lat.

W grupie uznanych algorytmów zasady wyboru były następujące. Algorytm musiał być zaimplementowany co najmniej kilka razy, kod źródłowy musi być dostępny w formie Open Source. Ponadto wybrane algorytmy muszą być traktowane w dziedzinie Swarm Intelligence jako baza i zostały zaprezentowane ponad 10 lat temu.

Druga grupa algorytmów zawiera metody opracowane relatywnie niedawno i ponadto nie zostały jeszcze powszechnie uznane. W tym przypadku kod źródłowy również musiał być dostępny w formie open-source.

Na podstawie opisanych powyżej zasad wybrane zostały algorytmy zaprezentowane w poniższej tabeli.

Każdy z powyższych algorytmów przyjmuje swój zakres różnych parametrów. W związku z tym aby rozwiązać problem niespójności badań wszystkie algorytmy zostały uruchomione z wartościami zaproponowanymi w ich oryginalnej implementacji. Żadne wartości nie były poprawiane w trakcie badań.

Particle Swarm Optimization	1995	Social behavior of bird flocking or fish schooling.	Eberhart, R. C., & Kennedy, J.
Artificial Bee Colony Optimization	2007	Natural Foraging behavior of real honey bees	Dervis Karaboga
Elephant Herding Optimization	2015	Herding behavior of the elephant groups	Nand K. Meena
Whale Optimization	2016	Social Behavior of humpback whales.	Ali R. Kashani
Grasshopper Optimization	2017	Behavior of grasshopper swarms in nature	Szymon Łukasik

Rysunek 1: Tabela przedstawiająca wybrane algorytmy celem przeprowadzenia badań.

Decyzja ta została podjęta po tym jak algorytmy podczas badań uruchamiane były kilkakrotnie z różnymi wartościami parametrów ale tylko te z oryginalnej implementacji dawały najlepsze wyniki dla wszystkich testowanych funkcji, nie faworyzując jednej kosztem drugiem.

3. Wybór funkcji testowych

Funkcje zostały wybrane do testów na podstawie trudności ich rozwiązań. Aby zapewnić sobie różnorodność funkcji zostały one wybrane na podstawie indeksu funkcji testowych Infinity77 [1].

Funkcje również były wybierane na podstawie przynajmniej jednego pomyselnego uruchomienia dla poszczególnych algorytmów, ograniczenie to nadal pozostawiło ogromną listę do wyboru. Aby ograniczyć listę i wybrać ostateczne funkcje zastosowano zasadę różności stylu funkcji np. Bowl-Shped, Valley-Shaped etc. to pozwoliło na wyłonienie ostatecznej listy wybranych funkcji.

FuncName	Dimensions	„Hardness” % of success	Global Minimum	X	X2
SixHumpCamel	2	82.17	-1.032	0.0898	-0.7126
StyblinskiTang	2	70.50	-78.33198	-2.90353	-2.90353
GoldsteinPrice	2	65.50	3	0	-1
Rastrigin	2	39.50	0	0	0
DropWave	2	21.25	-1	0	0

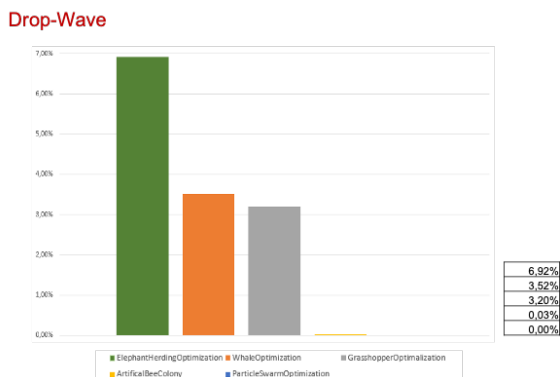
Rysunek 2: Tabela przedstawiająca wybrane funkcje celem przeprowadzenia badań.

Zestaw testów zawiera różne problemy Globalnej Optymalizacji, niektóre z nich są trudniejsze do

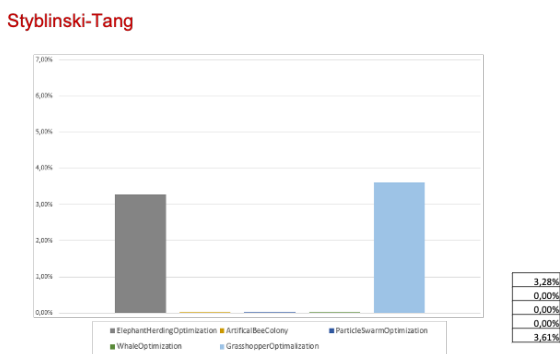
rozwiązania niż inne, niezależnie od wybranego algorytmu minimalizacji funkcji testu. Poniższą tabelę uzyskano, uruchamiając wszystkie dostępne globalne optymalizatory względem wszystkich funkcji testowych N-D dla zbioru 100 losowych punktów początkowych, a następnie uśredniając udane minimalizacje we wszystkich optymalizatorach.

4. Wyniki badań

Poniżej przedstawiono uśrednione wyniki osiągnięte dla każdego z algorytmów. Wyniki dla funkcji były bardzo zadowalające ponieważ każda z funkcji przynajmniej raz odnalazła optimum globalne. Zgodnie z załączonymi wykresami poniżej widzimy że najslabiej poradził sobie algorytm Elephant Herding Optimization, pozostałe dwa nowe rozwiązania były bardzo zbliżone pod względem błędu osiadania w optimach lokalnych. [2]



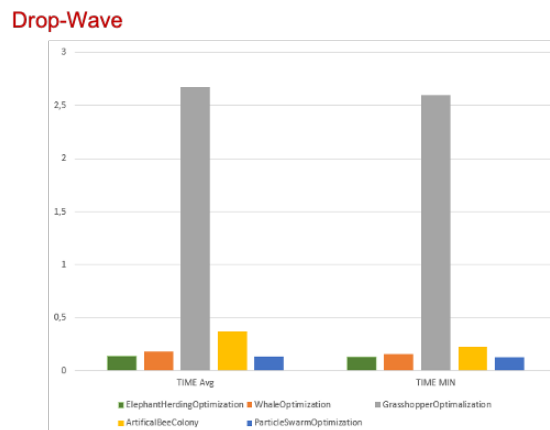
Rysunek 3: Przedstawienie odchylenia średniego błędu dla 20 uruchomień algorytmów dla funkcji Drop-Wave



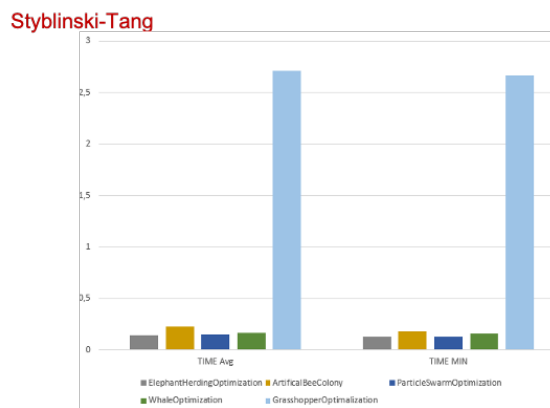
Rysunek 4: Przedstawienie odchylenia średniego błędu dla 20 uruchomień algorytmów dla funkcji Styblinski-Tang

Algorytmy również były zmierzone pod kątem prędkości działania, jak widać nowo zaproponowa-

ne algorytmy dobrze konkurują z metodami uznanymi. Jedynie algorytm GrassHopper Optimization wykazywał się wolniejszym działaniem, jest to jednak spowodowane uruchomieniem z parametrami nieoptymalizowanymi. W dalszych testach bez restrykcji parametryzacji o której pisałem wcześniej algorytm działał znacznie lepiej.



Rysunek 5: Przedstawienie średniego czasu pracy dla 20 uruchomień algorytmów dla funkcji Drop-Wave



Rysunek 6: Przedstawienie średniego czasu pracy dla 20 uruchomień algorytmów dla funkcji Styblinski-Tang

Algorytmy zachowywały się bardzo podobnie dla każdej z kolejnych funkcji co widać na załączonej poniżej tabeli. Tylko w poszczególnych przypadkach znacznie trudniejszych funkcji algorytm Elephant Optimization nie podolał odnalezieniu optimum globalnego pozostałe algorytmy zawsze dawały satysfakcjonujące wyniki.

Nazwa Funkcji	Algorytm	Wartość	Wynik
DropWave	ABC	-0,9997	-1
DropWave	EHO	-0,9308	-1
DropWave	GOA	-0,968	-1
DropWave	PSO	-1	-1
DropWave	WOA	-0,9648	-1
Goldstein-Price	ABC	3,0014	3
Goldstein-Price	EHO	13,53265	3
Goldstein-Price	GOA	3	3
Goldstein-Price	PSO	3	3
Goldstein-Price	WOA	3,0003	3
StyblinskiTang	ABC	-78,332	-78,3319
StyblinskiTang	EHO	-75,7593	-78,3319
StyblinskiTang	GOA	-75,5048	-78,3319
StyblinskiTang	PSO	-78,332	-78,3319
StyblinskiTang	WOA	-78,332	-78,3319
SixHumpCamel	ABC	-1,032	-1,032
SixHumpCamel	EHO	-0,8544	-1,032
SixHumpCamel	GOA	-1,032	-1,032
SixHumpCamel	PSO	-1,032	-1,032
SixHumpCamel	WOA	-1,032	-1,032
Rastrigin	ABC	0	0
Rastrigin	EHO	1,1053	0
Rastrigin	GOA	0,199	0
Rastrigin	PSO	0	0
Rastrigin	WOA	0	0

5. Wnioski

Wyraźnie można zauważyć, że metody „klasyczne” pod pewnymi względami przewyższają nowe proponowane rozwiązania w testach, jednak jest wiele scenariuszy, w których nowe rozwiązania mogą działać lepiej. Na przykład pod względem szybkości różnice między algorytmami ABC i Whale są znikome. Również algorytm Grasshopper może działać znacznie szybciej, gdy jest ograniczony do mniejszej grupy owadów. Jednak wtedy poświęcamy stabilność wyników. Jak również algorytm stada słońi można porównać do roju cząstek pod względem szybkości, rozbieżności w wynikach są zbyt duże, aby traktować to rozwiązanie jako „gotowe do użycia” przynajmniej przy domyślnych parametrach. Również optymalizacja wielorybów jest świetna do prostszych problemów, co możemy ocenić po osiągniętych wynikach. Szybkość i precyzja we wszystkich problemach poza Drop Wave jest na bardzo dobrym poziomie. Jak widać, wszystkie nowe proponowane rozwiązania można wykorzystać do prostszych problemów, ale brakuje im dojrzałości w zakresie trudniejszych problemów i optymalizacji parametrów funkcji. Wiadomo, że dobrze znane algorytmy, takie jak PSO czy ABC, są dobrze zoptymalizowane, zwłaszcza w przypadku funkcji testowych, a najlepsze parametry zostały już odkryte, co nie ma miejsca w przypadku nowych, jesz-

cze w pełni nieodkrytych właściwie algorytmów. Chociaż może się wydawać, że proponowane nowe algorytmy są znacznie gorsze niż te uznane w dziedzinie. W rzeczywistości w większości przypadków są one tylko nieznacznie odbiegające, co może być spowodowane nieoptymalizowanymi jeszcze parametrami testowanych funkcji. Co prowadzi do tego, że algorytmy roju stają się z czasem coraz lepsze, gdy zostaną w pełni odkryte a przede wszystkim przetestowane.

Literatura

- [1] “Test functions index - ampgo 0.1.0 documentation.” [Online]. Available: <http://infinity77.net>
- [2] J. Baumgart, *Application of OFN notation in Swarm Algorithms -basic implementation problems in the R environment*, 2019.
- [3] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” *In:Journal of global optimization*39, vol. 3, pp. 459–471, 2007.
- [4] A. R. Kashani, C. V. Camp, M. Armanfar, and A. Slowik, “Whale optimization algorithm,” in *Swarm Intelligence Algorithms*. CRC Press, 2020, pp. 323–333.
- [5] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *In:Proceedings of ICNN95-international conference on neural networks*, vol. 4, pp. 1942–1948, 1995.
- [6] S. Łukasik, “Grasshopper optimization algorithm,” in *Swarm Intelligence Algorithms*. CRC Press, 2020, pp. 193–205.
- [7] N. K. Meena, J. Yang, and A. Slowik, “Elephant herding optimization,” in *Swarm Intelligence Algorithms*. CRC Press, 2020, pp. 149–162.