

Leonid Rusanov*

Uniwersytet Kazimierza Wielkiego w Bydgoszczy

<https://orcid.org/0009-0000-7172-7053>

ZASTOSOWANIE OPTYMALIZACJI STADEM ŚWIŃ W ROZWIĄZYWANIU PROBLEMÓW O WIELU EKSTREMACH

Streszczenie

W tym artykule autorzy badają skuteczność metody optymalizacji zainspirowanej zachowaniem stada świń Duroc (Artificial Duroc Pigs Optimization). Metoda ADPO jest przykładem udanej implementacji uporządkowanych liczb rozmytych w metodzie optymalizacji roju. Autorzy mieli również za zadanie napisać metodę ADPO w języku programowania Python. Eksperymenty przeprowadzono dla dziesięciu funkcji testowych z wieloma ekstremami. Porównanie przeprowadzono dla tych samych funkcji za pomocą metod optymalizacji rojem cząstek (PSO), algorytmu nietoperzy (BA) i algorytmu genetycznego (GA). W większości przypadków wyniki uzyskane przez ADPO były najlepsze.

Słowa kluczowe: ADPO, PSO, GA, BA, OFN, Logika rozmyta

USE OF ARTIFICIAL DUROC PIG OPTIMIZATION IN SOLVING MULTI-EXTREME PROBLEMS

Abstract

In the article, the authors study the effectiveness of the optimization method inspired directly by the behavior of the Duroc pig herd (Artificial Duroc Pigs Optimization). The ADPO method is an example of the successful implementation of Ordered Fuzzy Numbers into a swarm optimization method. The authors were also challenged to write the ADPO method in the Python programming language. The experiments

* Leonid Rusanov – Katedra Systemów Inteligentnych, Wydziału Informatyki Uniwersytetu Kazimierza Wielkiego w Bydgoszczy, e-mail: leonid.rusanov@ukw.edu.pl

were carried out for ten benchmark functions with many extremes. For comparison, experiments with Particle Swarm Optimization (PSO), Bat Algorithm (BA) and Genetic Algorithm (GA) methods were carried out on the same functions. In most parts of the experiment, the results obtained by ADPO were the best.

Keywords: ADPO, PSO, GA, BA, OFN, Fuzzy Logic

~ • ~

Wstęp

Świnia Duroc to dawna rasa świń domowych ze Stanów Zjednoczonych. Obserwacja stada świń Duroc na podaż żywności stała się podstawą do opracowania algorytmu ADPO¹.

Przeprowadzone ostatnio badania nad metodą ADPO dowiodły, jak użyteczna może być metoda ADPO w porównaniu z innymi algorytmami optymalizacyjnymi. Nowatorskim rozwiązaniem w badaniach autorów jest zastosowanie logiki rozmytej, a konkretnie notacji skierowanych liczb rozmytych, która jest jej nieodłączną częścią, co pozwala na płynny opis przejścia próbki z jednego rozwiązania do drugiego². Dzięki zastosowaniu skierowanych liczb rozmytych populacja osiąga pożądane ekstremum funkcji wcześniej niż inne algorytmy optymalizacyjne. Autorzy przedstawili nową metodę defuzyfikacji regulatora rozmytego, która opiera się na koncepcji złotej proporcji, pochodzącej z szeregu Fibonacciego. Źródłem metody była obserwacja licznych przypadków złotej proporcji w różnych dziedzinach, takich jak biologia, architektura, medycyna i malarstwo. Szczególnym obszarem jej występowania jest genetyka, gdzie znajdujemy złotą proporcję w samej strukturze molekuly DNA (cząsteczki kwasu dezoksyrybonukleinowego mają szerokość 21 angstromów i długość 34 angstromów na każdą pełną długość jednego cyklu podwójnej spirali)³. Fakt ten sprawia, że stosunek w serii Fibonacciego jest w pewnym sensie uniwersalną zasadą projektowania wykorzystywaną zarówno przez człowieka, jak i naturę.

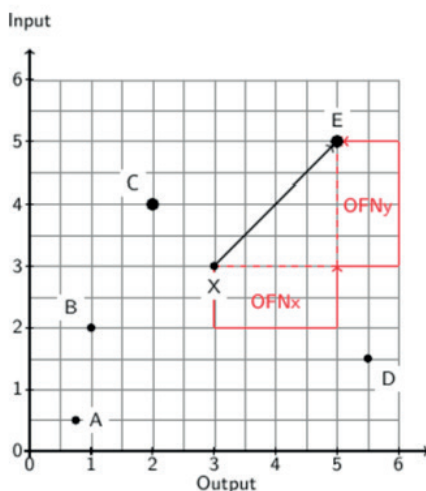
Poniższy rysunek 1 przedstawia schematyczną wizualizację stada Duroc. Punkty symbolizują przemieszczenie świń w przestrzeni. Dwie z nich (C i E) zostały oznaczone jako świnie ryczące. Są one zatem w miejscu, w którym znajduje się żywność. W algorytmie ADPO założono, że kolejnym krokiem w ewolucji jest

¹ Grandin T., Curtis S., Greenough W., *Effects of rearing environment on the behaviour of young pigs*, 2019, s. 57.

² Zob. Czerniak J., Zarzycki H., Ewald D., Augustyn P., *Application of OFN Numbers in the Artificial Duroc Pigs Optimization (ADPO) Method*, 2021.

³ Czerniak J.M., Dobrosielski W.T., Filipowicz I., *Comparing fuzzy numbers using defuzzifiers on ofn shapes*, "Studies in Fuzziness and Soft Computing" 2017, s. 207-222.

poruszanie się świni wzdłuż linii prostej (linii łączącej punkt przebywania z rykiem świni). Na rysunku w punkcie X znajduje się świnia, która może poruszać się w kierunku punktu C lub punktu E. W tym miejscu może zostać ustalony algorytm preferencji wyboru lub priorytetem będzie siła ryku, symbolizująca ilość jedzenia lub bliskość ryczącej świni.



Rysunek 1. Graficzne przedstawienie ruchu w metodzie ADPO

W przypadku przedstawionym na rysunku, wybór następnego węzła zależy od ilości jedzenia, zatem ruch zostanie wykonany z punktu X do punktu E. Kierunek marszu jest znany, ponieważ można go obliczyć z równania prostej przechodzącej przez dwa punkty. Kolejny krok powinien być jednak inny i dostosowany do odległości. Zbyt duży spowoduje zbyt wczesną zbieżność, a zbyt mały spowoduje zbyt wolne wyszukiwanie. W związku z tym wykorzystana zostanie notacja OFN⁴ wraz z metodą Golden Ratio⁵. Oczywiście możemy użyć innych operatorów wyostrzania, co zapewne nastąpi w kolejnych wersjach algorytmu. Ponieważ jednak podstawową cechą operatora Golden Ratio jest fakt, że liczby rozmyte nigdy nie wychodzą poza obszar rozmycia, jest on optymalny do zastosowania w metodzie rozmytej wykorzystywanej w metodzie ADPO. Następny rozdział zawiera pseudokod metody ADPO.

⁴ Chwastyk A., Pisz I., *OFN Capital Budgeting Under Uncertainty and Risk*, Cham 2017, s. 157-169.

⁵ Dobrosielski W., Czerniak J., Szczepanski J., Zarzycki H., *Two new defuzzification methods useful for different fuzzy arithmetics*, IWIFSGN 2016, "Advances in Intelligent Systems and Computing" 2018, nr 559, s. 83-101.

Pseudocode

Pseudokod algorytmu optymalizacji świń duroc:

Dane wejściowe:

Ilość świń – $NP = 30$

Ilość harcerskich świń – $SP = 10$

Ilość przywódców [%] – $LP = 10\% \times NP$

Stado świń – $\{p_1, \dots, p_{NP}\}$

Stado harcerskich świń – $\{s_1, \dots, s_{SP}\}$

Dane wyjściowe:

Rozwiązanie w postaci punktu $R(x, y)$

Funkcji:

- *SendToRandomPlaces()* – wysyłanie świń NP do losowych miejsc,
- *SendToUniquePlaces()* – wysyłanie harcerzy SP do innych niezbadanych miejsc,
- *SortGroup()* – sortowanie rozwiązań od najlepszego do najgorszego,
- *FindLeaderPigs()* – poszukiwanie lidera (liderów), $LP = 10\%$ z poszukiwanej populacji,
- *ReturnOfPigsScouts()* – zwracanie harcerskich świń,
- *NextPigsPositions()* – obliczenie następnej pozycji, na którą przesuwa się świnia dla każdego członka stada, z wyłączeniem przywódców,
- *RefillHerdPigs()* – uzupełnienie próby NP ze zbioru $NP + SP - LP$,
- *AvoidWeakPigs()* – usuwanie rozwiązań, które nie pasowały do NP ,
- *IfNoEnd()* – funkcja sprawdzająca spełnienie warunku końcowego, zadanej dokładności lub granicznej liczby epok spowoduje zakończenie algorytmu.
- BEGIN

krok 1: $\forall p_i \in \langle p_1, \dots, p_{NP} \rangle$ **SendToRandomPlaces**(p_i)

krok 2: $\forall s_i \in \langle s_1, \dots, s_{SP} \rangle$ **SendToUniquePlaces**(s_i)

krok 3: $\forall p_i \in \langle p_1, \dots, p_{NP} \cup s_1, \dots, s_{SP} \rangle$ **SortGroup**(P_i)

krok 4: **FindLeaderPigs**() $\{pl_1, pl_2, \dots, pl_{LP} \in P\}$

krok 5: **RefillHerdPigs**()

krok 6: **AvoidWeakPigs**()

krok 7: **NextPigsPositions**()

$\forall p_i \in \langle p_1, \dots, p_{NP} \rangle P_i \rightarrow P'_i$

niech $P(x_1, y_1)$ oraz $R(x_2, y_2)$

$P(x_1, y_1) \rightarrow P(x'_1, y'_1)$

$$x'_i = GR(OFN_x) = \begin{cases} \min(\text{supp}(OFN_x)) + \frac{|\text{supp}(OFN_x)|}{\phi} & , \text{ dla } (OFN_x) \text{ positive} \\ \max(\text{supp}(OFN_x)) - \frac{|\text{supp}(OFN_x)|}{\phi} & , \text{ dla } (OFN_x) \text{ negative} \end{cases}$$

$$y'_i = GR(OFN_y) = \begin{cases} \min(\text{supp}(OFN_y)) + \frac{|\text{supp}(OFN_y)|}{\phi} & , \text{ dla } (OFN_y) \text{ positive} \\ \max(\text{supp}(OFN_y)) - \frac{|\text{supp}(OFN_y)|}{\phi} & , \text{ dla } (OFN_y) \text{ negative} \end{cases}$$

krok 8: **IfNoEnd()** → **GoTo(Step2)**

END

Gdzie:

- p_i – liczba świni i ,
- s_{si} – liczba świni harcerskiej i , tymczasowy zbiór rozwiązań,
- P_i – liczba świni i z podzbioru nowych osobników.

Proces wykonywania algorytmu odbywa się w siedmiu krokach. Na początku wysyłane są sztuczne przedmioty w $NP = 30$ losowych miejsc. W kolejnym kroku, aby choćby częściowo zniwelować niepożądane działanie generatorów pseudo-losowych, do unikalnych miejsc, w których nie ma świni z pierwszego kroku, wysyłane są kolejne $SP = 10$ harcerzy. Następnie wszystkie rozwiązania $NP + SP$ są sortowane w kolejności malejącej. W czwartym kroku zbierane jest najlepsze 10% NP rozwiązań NP . Zatem kolejni liderzy świni są zapisywani jako stado, nadpisując w ten sposób oryginalną listę osobników NP . Pamięć niepotrzebnych dodatkowych osobników jest zwalniana w szóstym kroku. Krok siódmy został już opisany w poprzednim rozdziale dotyczącym liczb rozmytych (rysunek 1). Do zapisu maksymalnego możliwego przemieszczenia osobnika zastosowano notację OFN. Rzeczywisty ruch jest wynikiem zastosowania operatora rozmycia Złoty podział na współrzędnej, osi rzędnej i osi odciętej. Po sprawdzeniu stanu zakończenia algorytmu, gdy ewolucja jest jeszcze niezbędna, użytkownik przechodzi do drugiego kroku, a po znalezieniu rozwiązania jest ono zwracane i następuje zakończenie algorytmu.

Metodologia badań

Podczas przeprowadzonych testów do porównania wydajności algorytmów wykorzystano metodę ADPO napisaną w języku Python. W przypadku większości algorytmów optymalizacyjnych wykorzystano pakiety PyPI⁶. Jako środowisko programistyczne wykorzystano IDE PyCharm. Wybrano następujące metody poszukujące minimum funkcji w ograniczonym zakresie wartości zmiennych:

⁶ PyPI (Python Package Index) repository of software for the Python programming language, <https://pypi.org/>

- Particle Swarm Optimization (Pyswarms biblioteka⁷, 2021),
- Bat algorithm (BatAlgorithm biblioteka⁸, 2019),
- Genetic Algorithm (PyGAD biblioteka⁹, 2022),
- Artificial Duroc Pigs Optimization (własna implementacja).

Ze względu na ograniczenia dostępnego oprogramowania, eksperyment został ograniczony do 10 funkcji z wieloma ekstremami lokalnymi i jednym ekstremum globalnym. Dla każdej z tych funkcji i dla każdego rozmiaru zbioru poszukującego rozwiązania przeprowadzono po 25 prób optymalizacji dla każdego z algorytmów. Obliczono wartości błędu średniokwadratowego i odchylenia standardowego dla znalezionych ekstremów. Na podstawie analizy prób uzyskano średnie wyniki, które umieszczono w tabelach. Dodano również wykresy procesu poszukiwania rozwiązania dla każdej funkcji oraz wykresy wartości znalezionych minimów względem iteracji. Jak wspomniano powyżej, w trakcie badań zoptymalizowano 10 funkcji wzorcowych z dwiema niewiadomymi¹⁰ poszukując ich globalnego minimum w zadanym zakresie wartości zmiennych¹¹. Tabela 1 przedstawia nazwy użytych funkcji, granice wartości zmiennych, poszukiwane wartości niewiadomych oraz pożądaną wartość globalnego minimum.

Tabela 1. Lista funkcji użytych w eksperymencie

Lp.	Nazwa funkcji	Zakres wartości x1 i x2	Poszukiwane wartości x1 i x2	Poszukiwane wartości f(x1, x2)
1	Eggholder	[-512, 512]	(512, 404)	-959
2	Venter	[-50, 50]	(0, 0)	-400
3	Matyas	[-10, 10]	(0, 0)	0
4	Zirilli	[-10, 10]	(-1.04, 0)	-0,35
5	Easom	[-100, 100]	(3.14, 3.14)	-1
6	Rastrigin	[-5.12, 5.12]	(0, 0)	0
7	Levy N.13	[-10, 10]	(1, 1)	0
8	Drop Wave	[-5.12, 5.12]	(0, 0)	-1
9	Sphere	[-5.12, 5.12]	(0, 0)	0
10	Three-hump camel	[-5, 5]	(0, 0)	0

⁷ Pyswarm biblioteka, <https://pypi.org/project/pyswarm/> [dostęp: 23.01.2023].

⁸ Batalgorithm biblioteka, <https://pypi.org/project/BatAlgorithm/> [dostęp: 23.01.2023].

⁹ Pygad biblioteka, <https://pypi.org/project/pygad/> [dostęp: 23.01.2023].

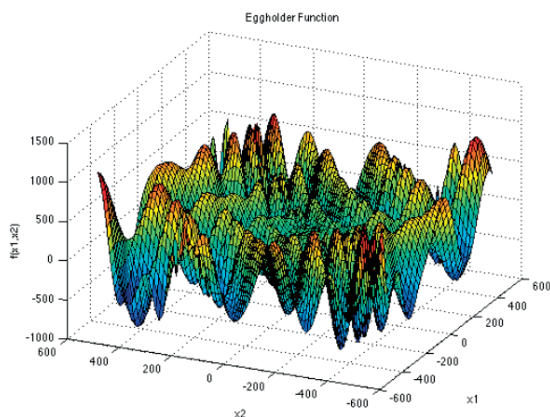
¹⁰ Kacprzyk, J., Szmidt E., *Distances between Intuitionistic Fuzzy Sets*, "Fuzzy Sets and Systems" 2000, nr 114, s. 505-518.

¹¹ Piegat A., Pluciński M., *Fuzzy number addition with the application of horizontal membership functions*, "The Scientific World Journal" 2015.

Zbiory testowe

Równania matematyczne funkcji testowych użytych w badaniu:
Eggholder:

$$f(x) = -(x_2 + 47) \sin\left(\sqrt{\left|x_2 + \frac{x_1}{2} + 47\right|}\right) - x_1 \sin\left(\sqrt{\left|x_1 - (x_2 + 47)\right|}\right)$$



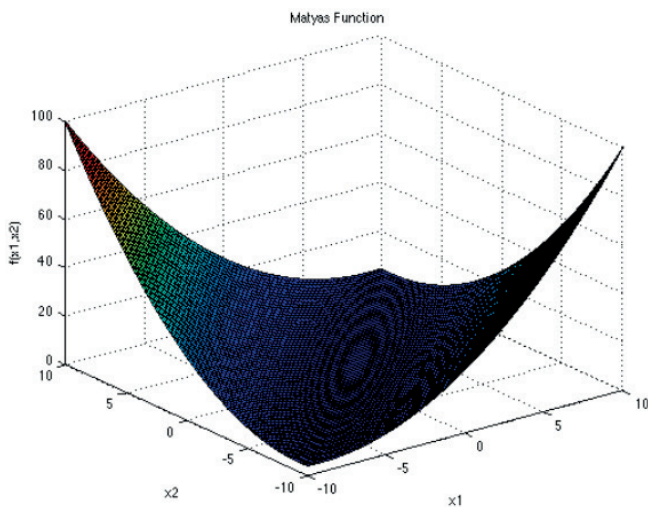
Rysunek 2. Graficzne przedstawienie funkcji Eggholder

Venter:

$$f(x) = x_1^2 - 100 \cos(x_1)^2 - 100 \cos\left(\frac{x_1^2}{30}\right) + x_2^2 - 100 \cos(x_2)^2 - 100 \cos\left(\frac{x_2^2}{30}\right)$$

Matyas:

$$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$$



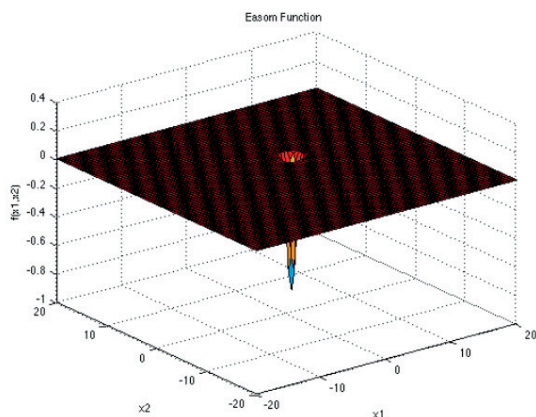
Rysunek 3. Graficzne przedstawienie funkcji Matyas

Zirilli:

$$f(x) = 0.25x_1^4 - 0.5x_1^2 + 0.1x_1 + 0.5x_2^2$$

Easom:

$$f(x) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$$



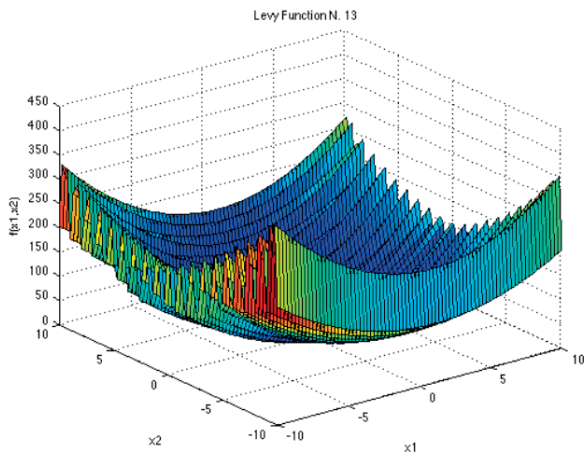
Rysunek 4. Graficzne przedstawienie funkcji Easom

Rastrigin:

$$f(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3\pi x_2) + (x_2 - 1)^2 [1 + \sin^2(2\pi x_2)]]$$

Levy N. 13:

$$f(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3\pi x_2) + (x_2 - 1)^2 [1 + \sin^2(2\pi x_2)]]$$



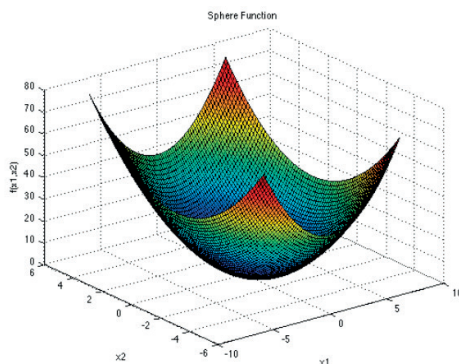
Rysunek 5. Graficzne przedstawienie funkcji Levy N.13

Drop-Wave:

$$f(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$$

Sphere:

$$f(x) = \frac{1}{899} \left(\sum_{i=1}^6 x_i^2 2^i - 1745 \right)$$



Rysunek 6. Graficzne przedstawienie funkcji Sphere

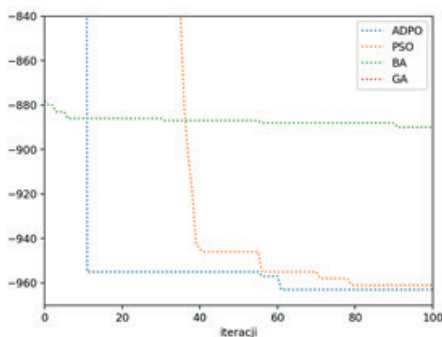
Three-hump camel:

$$f(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$$

Wyniki

Wyniki uzyskane podczas testowania określonych benchmarków metodami algorytmów stadem świń (ADPO), genetycznych (GA), optymalizacji rojem cząstek (PSO) i algorytmów nietoperzy (BA).

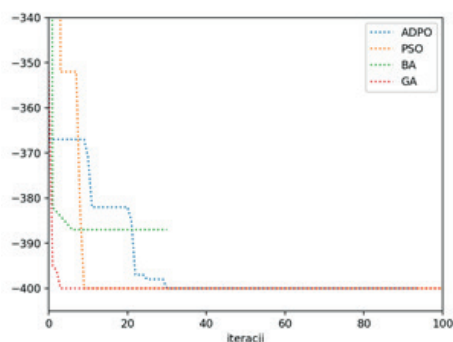
Eggholder:



Rysunek 7. Wartości funkcji Eggholdera znalezione w kolejnych iteracjach

Dobre wyniki zostały osiągnięte przez PSO i APDO ($f(x)$ od -892 do -961 dla PSO i od -930 do -963 dla APDO, wartość poszukiwania wynosi -959). GA wyróżnia się na ich tle dużymi błędami MSE i odchyleniem standardowym wynoszącym kilkadziesiąt jednostek. APDO jest najszybsze. Rysunek 7 pokazuje jak duże są różnice, linia reprezentująca GA ze względu na duży błąd znajduje się poza widocznym obszarem.

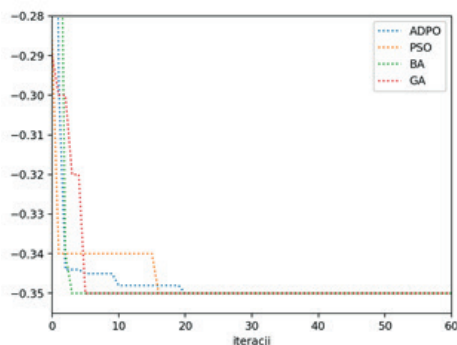
Venter:



Rysunek 8. Wartości funkcji Venter znalezione w kolejnych iteracjach

Idealne wyniki uzyskane za pomocą metody APDO (Rys. 8), nieco gorsze GA i PSO (błędy przy małej populacji), wyraźna niepoprawność wyniku BA (najlepszy to $f(x) = -382$, gdy powinno być -400). Również algorytm GA jest najwolniejszy.

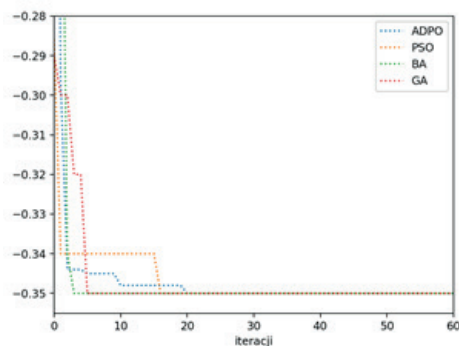
Zirilli:



Rysunek 9. Wartości funkcji Zirilli znalezione w kolejnych iteracjach

Podobnie jak w przypadku Matyas, optymalizacja Zirilli (rys. 9) dała bardzo dobre wyniki jakościowe dla wszystkich metod. Różnica dotyczy czasu wykonania: tym razem GA jest najszybszy.

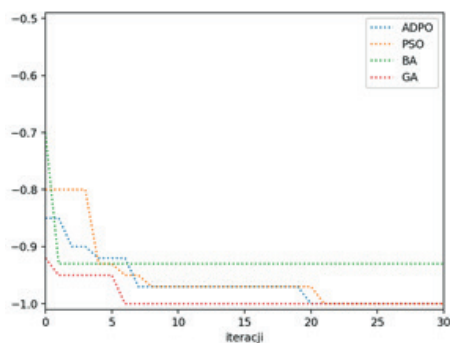
Easom:



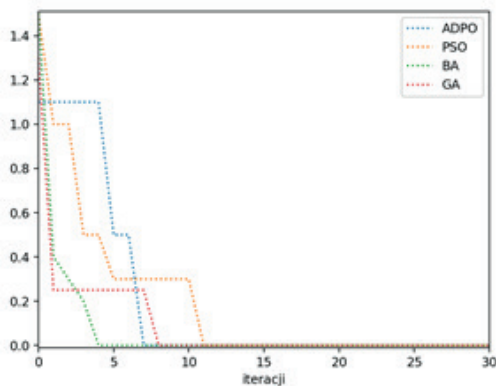
Rysunek 10. Wartości funkcji Easom znalezione w kolejnych iteracjach

Wielkość populacji była w tym przypadku bardzo ważna. Wraz ze wzrostem populacji poprawność optymalizacji znacznie wzrosła. ADPO było najszybsze, podczas gdy GA było najwolniejsze.

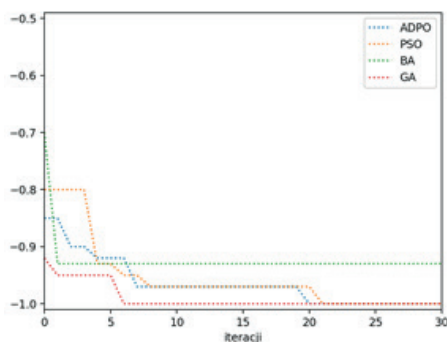
Drop-Wave, Levy N. 13, Rastrigin:



Rysunek 11. Wartości funkcji Drop-Wave znalezione w kolejnych iteracjach



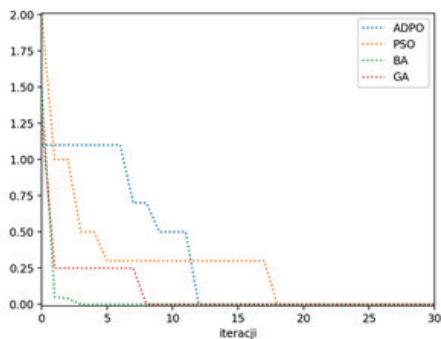
Rysunek 12. Wartości funkcji Levy N. 13 znalezione w kolejnych iteracjach



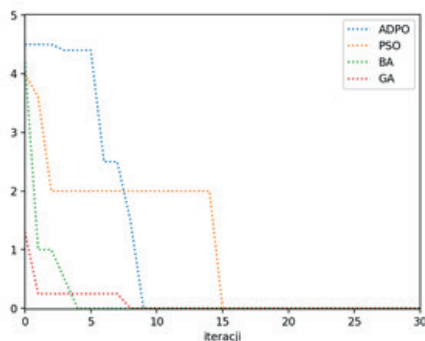
Rysunek 13. Wartości funkcji Rastrigin znalezione w kolejnych iteracjach

Poprawność wyników optymalizacji Drop Wave (Rys. 11), Levy N.13 (Rys. 12) i Rastrigin (Rys. 13) jest podobna we wszystkich metodach, małe błędy i odchylenie standardowe występowały tylko przy niskich rozmiarach populacji. Najszybszym algorytmem w tym przypadku okazała się metoda ADPO i BA. GA ponownie był najwolniejszy.

Sphere, Three-hump camel:



Rysunek 14. Wartości funkcji Sphere znalezione w kolejnych iteracjach



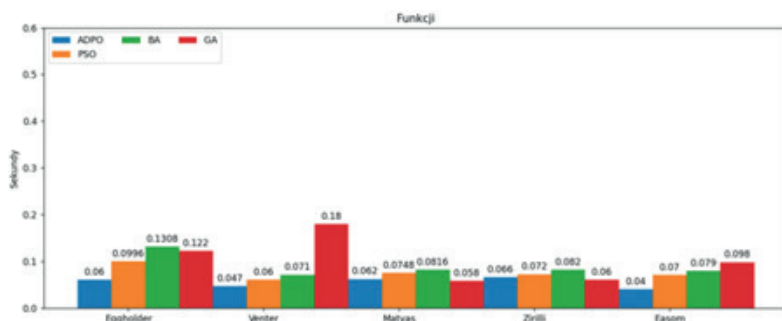
Rysunek 15. Wartości funkcji Three-hump camel znalezione w kolejnych iteracjach

Wyniki funkcji Sphere i Three-Hump Camel były również podobne we wszystkich algorytmach. Różnica polegała na szybkości wykonywania algorytmów. ADPO stał się najszybszy. Najwolniejszy ponownie okazał się GA.

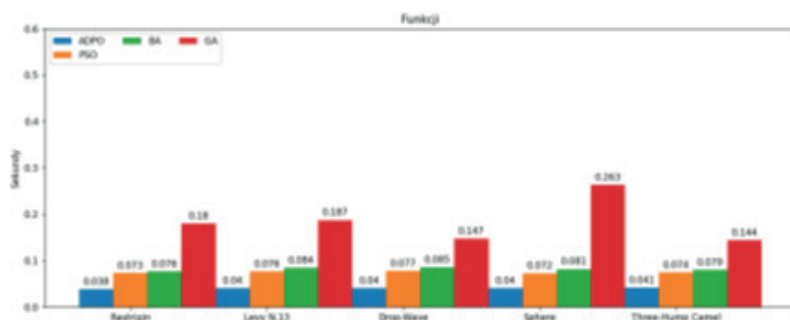
Wnioski

Przedstawione i omówione wyniki badania optymalizacji funkcji matematycznych są bardzo zbliżone do wyników z publikacji wprowadzającej metodę ADPO. Potwierdza to, że wykorzystanie metod sztucznej inteligencji inspirowanych naturą może być bardzo korzystne jako skuteczny sposób rozwiązywania problemów optymalizacyjnych.

Jakość wyników optymalizacji zależała głównie od metody badawczej, funkcji oraz wielkości zbioru poszukującego rozwiązania. Bardzo ważną informacją jest to, że wraz ze wzrostem wielkości populacji zdecydowanie wzrasta poprawność uzyskiwanych rozwiązań. Było to szczególnie widoczne w przypadku metod rojowych operujących na funkcjach Eggholdera czy Ventera. ADPO nie okazała się wolniejsza od pozostałych testowanych metod, a w większości przypadków był najszybszy. Rysunki 16 i 17 pokazują, jak szybkie są metody ADPO w porównaniu z innymi algorytmami optymalizacji.



Rysunek 16. Porównanie szybkości wykonywania algorytmów za pomocą 5 benchmarków (Eggholder, Venter, Matyas, Zirilli, Easom)



Rysunek 17. Porównanie szybkości wykonywania algorytmów za pomocą 5 benchmarków (Rastrigin, Levy N.13, Drop-Wave, Sphere, Three-Hump Camel)

References

Bibliografia

- Czerniak J.M., *Ofn ant method based on tsp ant colony optimization*, in: Prokopowicz, P., Czerniak, J.M., Mikolajewski, D., Apiecionek, L., Slezak, D. (eds.) *Theory and Applications of Ordered Fuzzy Numbers. A Tribute to Professor Witold Kosinski*, chap. 12, "Studies in Fuzziness and Soft Computing" 2017, pp. 207-222.
- Czerniak J.M., Dobrosielski W.T., Filipowicz I., *Comparing fuzzy numbers using defuzzifiers on ofn shapes*, in: Prokopowicz, P., Czerniak, J.M., Mikolajewski, D., Apiecionek, L., Slezak, D. (eds.) *Theory and Applications of Ordered Fuzzy Numbers. A Tribute to Professor Witold Kosinski*, chap. 6, "Studies in Fuzziness and Soft Computing" 2017, pp. 207-222.
- Czerniak J.M., Zarzycki H., *Artificial acari optimization as a new strategy for global optimization of multimodal functions*, "Journal of Computational Science" 2017.
- Czerniak J., Filipowicz I., Ewald D., *The novel shape normalization operator for fuzzy numbers in ofn notation*, in: Kacprzyk, J. e.a. (ed.) *Advances in Fuzzy Logic and Technology 2017. IWIFSGN 2017, EUSFLAT 2017*, "Advances in Intelligent Systems and Computing" 2018, vol. 641, pp. 548-562.
- Dobrosielski W.T., Czerniak J.M., Zarzycki H., Szczepanski J., *Fuzzy numbers applied to a heat furnace control*, in: Prokopowicz P., Czerniak J.M., Mikolajewski D., Apiecionek L., Slezak D. (eds.) *Theory and Applications of Ordered Fuzzy Numbers. A Tribute to Professor Witold Kosinski*, chap. 16, pp. 207-222. "Studies in Fuzziness and Soft Computing" 2017, pp. 207-222.
- Dobrosielski W., Czerniak J., Szczepanski J., Zarzycki H., *Two new defuzzification methods useful for different fuzzy arithmetics*, in: et al., A.K. (ed.) *Uncertainty and Imprecision in Decision Making and Decision Support: CrossFertilization, New Models and Applications. IWIFSGN 2016.*, "Advances in Intelligent Systems and Computing" 2018, vol. 559, pp. 83-101.
- Ewald D., Czerniak, J., Zarzycki H., *Ofnbee method used for solving a set of benchmarks*, in: Kacprzyk, J.e.a. (ed.) *Advances in Fuzzy Logic and Technology 2017. IWIFSGN 2017, EUSFLAT 2017*, "Advances in Intelligent Systems and Computing" 2018, vol. 642, pp. 24-35.
- Ewald D., Czerniak J.M., Paprzycki M., *A new ofnbee method as an example of fuzzy observance applied for abc optimization*, in: Prokopowicz, P., Czerniak, J.M., Mikolajewski, D., Apiecionek, L., Slezak, D. (eds.) *Theory and Applications of Ordered Fuzzy Numbers. A Tribute to Professor Witold Kosinski*, chap. 12, "Studies in Fuzziness and Soft Computing" 2017, pp. 207-222.
- Grandin T., Curtis S., *Toy preferences in young pigs* 59, 1984.
- Grandin T., Curtis S., Greenough W., *Effects of rearing environment on the behaviour of young pigs* 57, 1983.
- Harris A., Patience J., Lonergan S., Dekkers J., Gabler N., *Improved nutrient digestibility and retention partially explains feed efficiency gains in pigs selected for low residual feed intake* 90, 164-166, 2013.
- Held S., Mason G., Mendl M., *Using the piglet scream test to enhance piglet survival on farms: Data from outdoor sows* 16, 2007.
- Kacprzyk J., Wilbik A., *Using fuzzy linguistic summaries for the comparison of time series: an application to the analysis of investment fund quotations*, in: "IFSA/EUSFLAT Conf." 2009, pp. 1321-1326.
- Kosinski W., *On fuzzy number calculus*. "International Journal of Applied Mathematics and Computer Science" 2006, 16(1), 51-57.
- Kosinski W., Frischmuth K., Wilczyńska-Sztyma, D., *A New Fuzzy Approach to Ordinary Differential Equations*, in:
- Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) *Proceedings of ICAISC 2010, Part I*. "Lecture Notes in Computer Science" 2010, vol. 6113, pp. 120-127.

- Kovac D., Beres M., Kovacova I., Vince T., Molnar J., Dziak J., Jacko P., Bucko R., Tomcikova I., Schweiner D.,
Circuit elements influence on optimal number of phases of DC/DC buck converter, "Electronics Letters" 2018, 54(7), 435–436. <https://doi.org/10.1049/el.2018.0043>
- Kuhlmeier V., Boysen S., *Animal cognition*, 2006
- Małolepsza O., *Methods of adaptation of knowledge systems based on fuzzy sets*, "Studies and Materials in Applied Computer Science" 2023, 15(1), 11-19.
- Marszałek A., Burczynski T., *Modeling and forecasting financial time series with ordered fuzzy candlesticks*, "Information Science" 2014, 273 (144-155).
- McGlone J., E. Curtis S., *Behavior and performance of weanling pigs in pens equipped with hide areas* 60, 20-4, 1985.
- Patel B., Chen H., Ahuja A., F. Krieger J., Noblet J., Chambers S., S. Kassab G., *Constitutive modeling of the passive inflation-extension behavior of the swine colon* 77, 2017.
- Pettigrew J.E., *Essential role for simulation models in animal research and application*, "Animal Production Science" 2018, 58(4), 704-708.
- Piegat A., Pluciński M., *Computing with words with the use of inverse rdm models of membership functions*, "International Journal of Applied Mathematics and Computer Science" 2015, 25(3), 675-688.
- Stachowiak A., Dyczkowski K., *A similarity measure with uncertainty for incompletely known fuzzy sets*. Proceedings Of The 2013 Joint Ifsa World Congress And Nafips Annual Meeting (Ifsa/ Nafips), 2013, pp. 390-394.
- Szmidt E., Kacprzyk J., *Distances between intuitionistic fuzzy sets*, "Fuzzy Sets and Systems" 2000, 114, 505-518.