Hybridizing Sine Cosine Algorithm with Ant Lion Optimizer for Improving Search Accuracy

Nayan Pachange; Vanshika Prakash Mishra
Devendra Toongar; Neeraj Kumar
Jha
Department of MCA, G. H. Raisoni College of Engineering & Management
Nagpur, India

Abstract
Numerous domains, including engineering, machine learning, and industrial processes, depend heavily on optimization. Early convergence or stagnation are frequently the results of traditional metaheuristic algorithms' incapacity to strike a balance between exploration and exploitation. To overcome these issues, this study suggests a novel hybridization of the Ant Lion Optimizer (ALO) and the Sine Cosine Algorithm (SCA). By combining the robust local search and trapping mechanisms of ALO with the quick exploration and straightforward structure of SCA, the hybrid model enhances overall optimization performance. The proposed hybrid SCA-ALO algorithm is evaluated on real optimization problems and 23 benchmark functions. In terms of convergence speed, solution accuracy, and robustness, comparative analysis shows that the hybrid continuously performs better than the standalone SCA and ALO. The hybrid model performs noticeably better in 15 of the 23 benchmark functions. The superiority of the hybrid approach is further supported by statistical analyses, such as mean fitness values and standard deviation. Its practical versatility is further demonstrated by real-world applications like feature selection, engineering design, and economical load dispatch. The results point to a strong

framework for resolving dynamic and complex optimization problems through the hybridizationofcomplementary metaheuristic strategies, paving the way for

framework for resolving dynamic and complex optimization problems through the hybridizationofcomplementary metaheuristic strategies, paving the way for further developments in the study of hybrid algorithms.

Keywords:
Sine Cosine Algorithm, Ant Lion Algorithm, Global search, Hybridize

1. Introduction

Optimization is a key component of problem-solving in many scientific and engineering domains. Effective optimization techniques are necessary to produce optimal or nearly optimal results for real-world problems such as parameter estimation, industrial design, or even machine learning model optimization. Numerous conventional mathematical methods are challenged by large-scale, high-order, nonlinear, and even multidimensional optimization problems. These problems can be resolved with metaheuristic algorithms.
Originating in natural phenomena and evolutionary processes, metaheuristic algorithms are widely used due to their ability to efficiently explore and exploit the search space. The Sine Cosine Algorithm (SCA) is a population-based optimization technique that uses sine and cosine functions to guide potential solutions toward the best ones.

SCA can occasionally have weak exploitation capabilities, which leads to slow convergence in intricate optimization settings, even though it is effective in exploration. The Ant Lion Optimiser (ALO), on the other hand, draws inspiration from the natural hunting methods of
antlions. ALO excels at local search and exploitation by using stochastic random walks and a trapping mechanism to direct the search. However, if ALO isn't paired with a robust exploration strategy, it can occasionally become stuck in local optima.[2]
This study suggests a hybrid strategy that combines the Ant Lion Optimizer (ALO) and the Sine Cosine Algorithm (SCA) to overcome these drawbacks. The hybrid SCA-ALO algorithm aims to improve optimization performance, attain faster convergence, and increase solution accuracy by fusing the exploitation power of ALO with the exploration capabilities of SCA. Benchmark optimization functions and real-world case studies are used to assess the efficacy of the suggested strategy
A comprehensive experimental analysis has been conducted to demonstrate the hybrid algorithm's superiority. A range of benchmark functions, such as unimodal, multimodal, and composite test problems, were selected in order to assess the algorithm's performance under different optimization scenarios. The results were contrasted with those of popular algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Grey Wolf Optimizer (GWO). Statistical analyses, such as mean fitness value, standard deviation, and convergence speed, were used as performance indicators. The hybrid SCA-ALO consistently showed improved robustness, faster convergence, and higher accuracy in achieving global optima. These empirical results show how the hybrid algorithm reduces premature convergence and stagnation by efficiently balancing the exploration and exploitation phases.
Additionally, the practical potential of the hybrid SCA-ALO was validated through real-world applications. Among these were engineering design issues like feature selection in machine learning tasks, pressure vessel design, and economical load dispatch in power systems. The hybrid algorithm outperformed conventional and standalone metaheuristic techniques in terms of computational efficiency and result reliability in each scenario, in addition to offering optimal or nearly optimal solutions. Because of its versatility and scalability, the hybrid model can be used to solve high-dimensional and constrained optimization problems, opening the door for additional study and practical use. The suggested hybrid framework creates a solid foundation for upcoming advancements in metaheuristic optimization by fusing the intelligent entrapment tactics of ALO with the dynamic search patterns of SCA.

2. Proposed Optimization Algorithm

2.1. Algorithm Classification

Nature-inspired adaptation algorithms mimic growth, flock intelligence, and biological processes to solve complex problems. By finding a balance between exploration and exploitation, they guarantee efficient search mechanisms for adaptation.[3]
Genetic algorithms (GA) and inter-development (DE) are examples of developmentist algorithms (EAS), which are impacted by natural selection and genetics.
Group collective behavior was used to model flocking intelligence algorithms, including Artificial B Colony (ABC), Ant Colony Adaptation (ACO), and Particle Fleet Adaptation (PSO).
Physics-based algorithms: these include the simulated anneal (SA) and the gravity search algorithm (GSA), which are based on physical laws.

DOI: https://doi.org/10.5281/zenodo.15637294

ISSN NO-2584-2706

The Sign Cosine Algorithm (SCA), Ant Lion Optimizer (ALO), and Gray Wolf Optimizer (GWO) are examples of bio-inspired algorithms that draw inspiration from biological processes.

Hybrid models, which combine various nature-inspired mechanisms to improve search performance, are another new class of metaheuristic algorithms.[8] By combining the best aspects of each of their constituent parts, these hybrid algorithms seek to increase computational efficiency, convergence rate, and solution accuracy. This includes the suggested SCA-ALO algorithm, which makes use of the complementary advantages of its component pieces.

Additionally, current patterns indicate a growing interest in adaptive metaheuristics, which modify algorithm parameters dynamically based on the optimization environment. The trade-off between exploration and exploitation is further refined by such methods. When combined with adaptive features, hybrid algorithms such as SCA-ALO show great promise for solving complex and large-scale problems more reliably than single-method algorithms.[5]

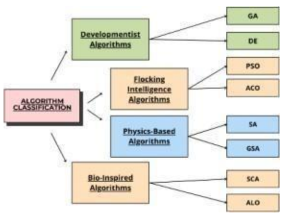
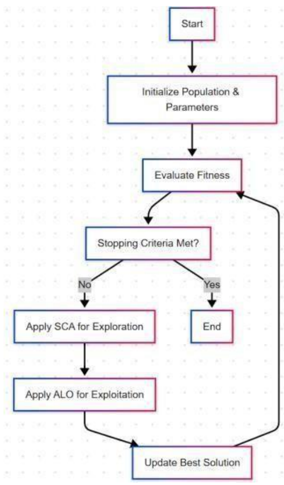


Figure1- Classification of Nature inspired Algorithms [18]

1.1 Flowhart



2.3.Mathematical Functions

Table 2: Standard UM benchmark functions

Functions	Dimensions	Range	f_{min}
$F_1(S)=\sum_{m=1}^{z} S_m^2$	(10,30,50,100)	[-100 , 100]	0
$F_2(S)=\sum_{m=1}^{z}\lvert S_m\rvert+\prod_{m=1}^{z}\lvert S_m\rvert$	(10,30,50,100)	[-10 ,10]	0
$F_3(S)=\sum_{m=1}^{z}(\sum_{n-1}^{m} S_n)^2$	(10,30,50,100)	[-100 , 100]	0
$F_4(S)=max_m\{\lvert S_m\rvert, 1\le m\le z\}$	(10,30,50,100)	[-100 , 100]	0
$F_5(S)=\sum_{m=1}^{z-1}[100(S_{m+1}-S_m^2)^2+(S_m-1)^2]$	(10,30,50,100)	[-38 , 38]	0
$F_6(S)=\sum_{m=1}^{z}([S_m+0.5])^2$	(10,30,50,100)	[-100 , 100]	0
$F_7(S)=\sum_{m=1}^{z} mS_m^4+random\,[0,1]$	(10,30,50,100)	[-1.28, 1.28]	0
$F_{12}(S)=\frac{\pi}{z}\left\{10\sin(\pi\tau_1)+\sum_{m=1}^{z-1}(\tau_m-1)^2[1+10\sin^2(\pi\tau_{m+1})]+(\tau_z-1)^2\right\}+\sum_{m=1}^{z}u(S_m,10,100,4)$ $\tau_m=1+\frac{S_m+1}{4}$ $u(S_m,b,x,i)=\begin{cases}x(S_m-b)^i & S_m>b\\ 0 & -b<S_m<b\\ x(-S_m-b)^i & S_m<-b\end{cases}$	(10,30,50,100)	[-50,50]	0
$F_{13}(S)=0.1\{\sin^2(3\pi S_m)+\sum_{m=1}^{z}(S_m-1)^2[1+\sin^2(3\pi S_m+1)]+(x_z-1)^2[1+\sin^2 2\pi S_z)]$	(10,30,50,100)	[-50,50]	0
Functions	Dimensions	Range	f_{min}
$F_{14}(S)=[\frac{1}{500}+\sum_{n=1}^{2}5\frac{1}{n+\sum_{m=1}^{z}(S_m-b_{mn})6}]^{-1}$	2	[-65.536, 65.536]	1
$F_{15}(S)=\sum_{m=1}^{11}[b_m-\frac{S_1(a_m^2+a_mS_2)}{a_m^2+a_mS_3+S_4}]^2$	4	[-5, 5]	0.00030
$F_{16}(S)=4S_1^2-2.1S_1^4+\frac{1}{3}S_1^6+S_1S_2-4S_2^2+4S_2^4$	2	[-5, 5]	-1.0316
$F_{17}(S)=(S_2-\frac{5.1}{4\pi 2}S_1^2+\frac{5}{\pi}S_1-6)^2+10(1-\frac{1}{8\pi})\cos S_1+10$	2	[-5, 5]	0.398
$F_{18}(S)=\left[1+(S_1+S_2+1)^2\left(19-14S_1+3S_1^2-14S_2+6S_1S_2+3S_2^2\right)\right]\times\left[30+(2S_1-3S_2)^2\left(18-32S_1+12S_1^2+48S_2-36S_1S_2+27S_2^2\right)\right]$	2	[-2,2]	3
$F_{19}(S)=-\sum_{m=1}^{4}d_m\exp(-\sum_{n=1}^{3}S_{mn}(S_m-q_{mn})^2)$	3	[1, 3]	-3.32
$F_{20}(S)=-\sum_{m=1}^{4}d_m\exp(-\sum_{n=1}^{6}S_{mn}(S_m-q_{mn})^2)$	6	[0, 1]	-3.32
$F_{21}(S)=-\sum_{m=1}^{5}[(S-b_m)(S-b_m)^T+d_m]^{-1}$	4	[0,10]	-10.1532
$F_{22}(S)=-\sum_{m=1}^{7}[(S-b_m)(S-b_m)^T+d_m]^{-1}$	4	[0, 10]	-10.4028
$F_{23}(S)=-\sum_{m=1}^{7}[(S-b_m)(S-b_m)^T+d_m]^{-1}$	4	[0, 10]	-10.5363

DOI: https://doi.org/10.5281/zenodo.15637294

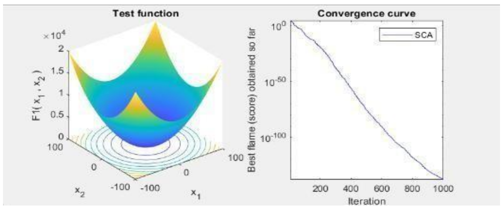
Reference No.	Algorithm Name	Author Name	Year
1	*Sine Cosine Algorithm*	Mirjalili, S	**2016**
2	*The Ant Lion Optimizer*	Mirjalili, S	2015
3	Nature-Inspired Metaheuristic Algorithms	Yang, X. S.	2010
4	Metaheuristics	Talbi, E.-G.	2009
5	A brief review of nature-inspired algorithms for	Fister, I., Yang, X. S., Fister Jr, I., Fister, D.	2013
6	A novel quantum inspired cuckoo search for knapsack	Layeb, A.	2011
7	Opposition-based differential evolution.	Rahnamayan, S., Tizhoosh, H. R., Salama, M. M. A.	2008
8	Hybrid metaheuristic algorithms: A comprehensive	ZainEldin, H., et al.	2020
9	A hybrid optimization algorithm with Sine Cosine Algorithm and Differential Evolution.	Abd Elaziz, M., Oliva, D., Hinojosa, S., & Elsheikh, A. H.	2017
10	*Optimisation of electric power systems using particle swarm optimisation: A review*	Jordehi, A. R.	2015
11	A novel metaheuristic algorithm and its applications for parameter optimization of machine learning	Chen, H., et al.	2019
12	Nature-inspired optimization algorithms: A comprehensive survey and	Abualigah, L., et al.	2021
13	Optimizing connection weights in neural networks using the whale optimization	Aljarah, I., Faris, H., & Mirjalili, S	2018
14	A new hybrid metaheuristic algorithm for solving complex optimization problems.	Rakhshani, H., & Pourakbari-Kasmaei, M.	2020
15	Metaheuristic algorithms in modeling and optimization.	Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H.	2013
16	A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm.	Karaboga, D., & Basturk, B.	2007
17	Cuckoo search via Lévy flights.	Yang, X. S., & Deb, S	2009

2.4.**Search Space**

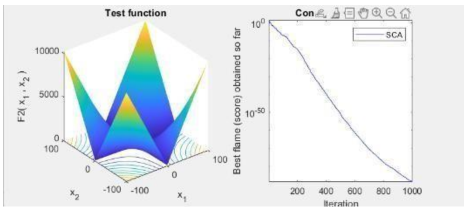
The hybrid SCA-ALO algorithm improves global search and convergence by fusing the exploration of SCA with the exploitation of ALO. This equilibrium enhances the effectiveness and quality of the solution while avoiding the trapping of local optima. Therefore, these algorithms operate on the twenty-three (23) search space under the specified conditions.[12]

By utilizing ALO's emphasis on intensifying local regions and SCA's ability to randomly explore global regions, the hybrid SCA-ALO algorithm successfully adapts its search behavior to various landscapes. This dual mechanism makes it particularly appropriate for multimodal functions with many local optima because it avoids the premature stagnation that is often observed in algorithms with weak exploitation capabilities.

Bounds on variable values or nonlinear equality constraints are examples of search space constraints that have a substantial impact on performance in real-world optimization. Because of its adaptable structure, the hybrid algorithm can deal with these limitations more easily. The SCA-ALO can effectively navigate feasible regions and concentrate computational resources on high-potential areas by dynamically modifying the search boundaries based on current progress.



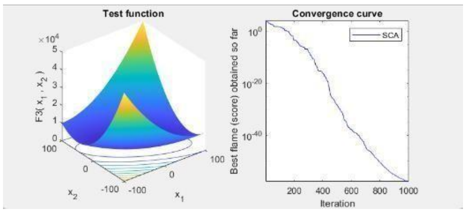
In, F1 hybrid algorithm achieves a more optimal value is 2.4419e-138 compared to the original value 1.1051e-28.



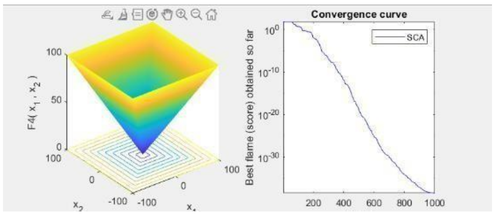
In, F2 hybrid algorithm achieves a more optimal value is 1.7101e-89 compared to the original value 2.86e-14.

DOI: https://doi.org/10.5281/zenodo.15637294

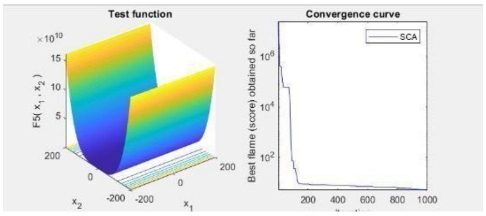
International Journal of Modern Science and Research Technology
ISSN NO-2584-2706



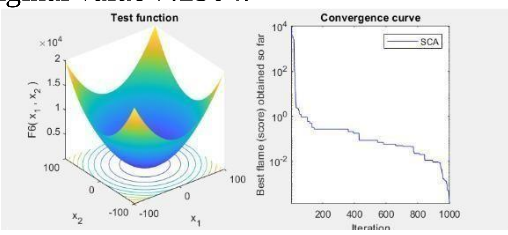
In, F3 hybrid algorithm achieves a more optimal value is 1.6342e-58 compared to the original value 9.142e-14.



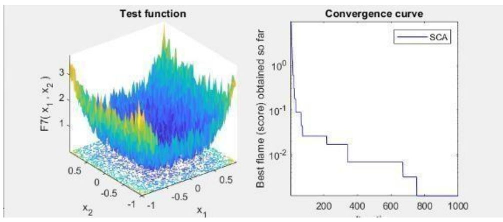
In, F4 hybrid algorithm achieves a more optimal value is 3.6852e-39 compared to the original value 2.8501e-08.



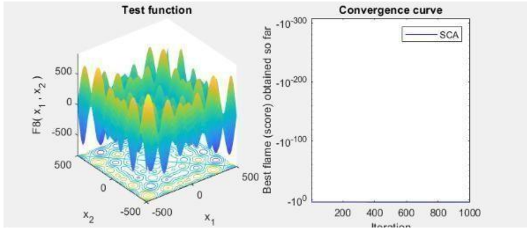
In, F5 hybrid algorithm achieves a more optimal value is 4.783 compared to the original value 7.2304.



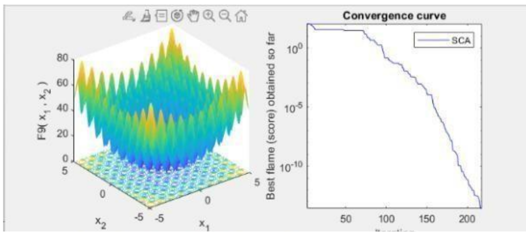
In, F6 hybrid algorithm achieves a more optimal value is 0.00012313 compared to the original value 0.23014.



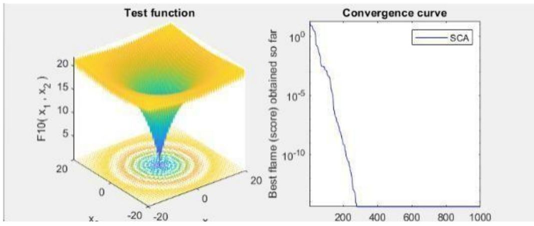
In, F7 hybrid algorithm achieves a more optimal value is 0.0011824 compared to the original value 0.0022384.



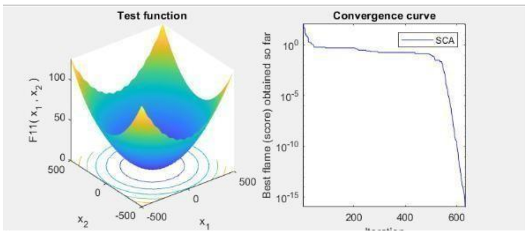
In, F8 hybrid algorithm achieves a more optimal value is -3107.3549 compared to the original value -2185.9955.



In, F9 hybrid algorithm achieves optimal value is 0 same as to the original value 0.



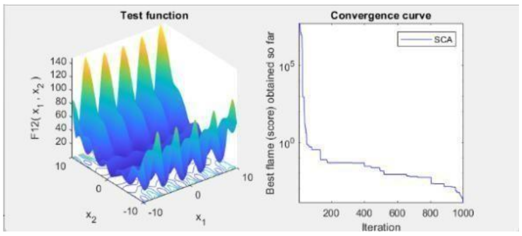
In, F10 hybrid algorithm achieves a more optimal value is 3.9968e-15 compared to the original value 1.4655e-14.



In, F11 hybrid algorithm achieves optimal value is 0 same as to the original value 0.

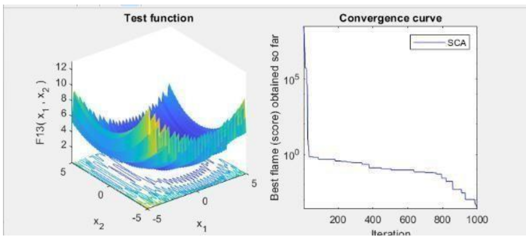
DOI: https://doi.org/10.5281/zenodo.15637294

International Journal of Modern Science and Research Technology
ISSN NO-2584-2706

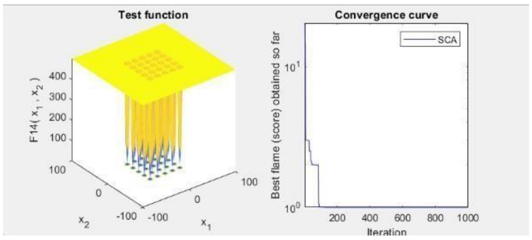


In, F12 hybrid algorithm achieves a more optimal value is 0.00011961 compared to the original value 0.073301

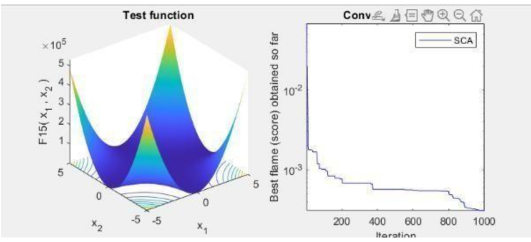
.



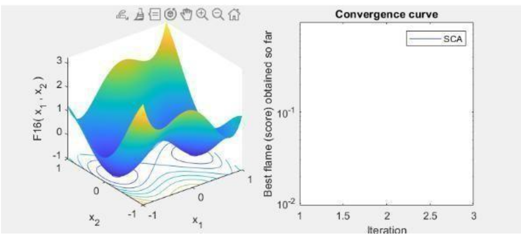
In, F13 hybrid algorithm achieves a more optimal value is 0.00023499 compared to the original value 0.43108.



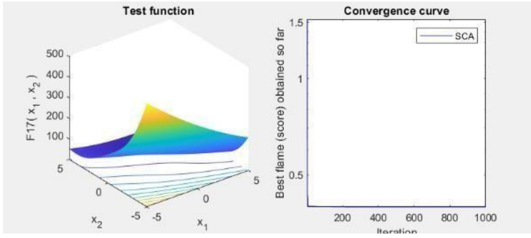
In, F14 hybrid algorithm achieves a more optimal value is 0.998 compared to the original value 0.99802.



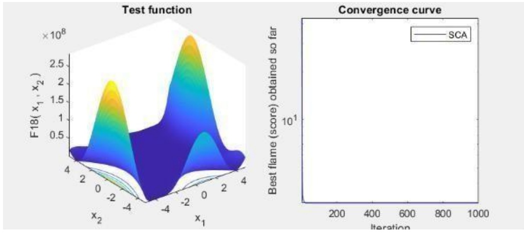
In F15 hybrid algorithm achieves a more optimal value is 0.00030755 compared to the original value 0.0013812.



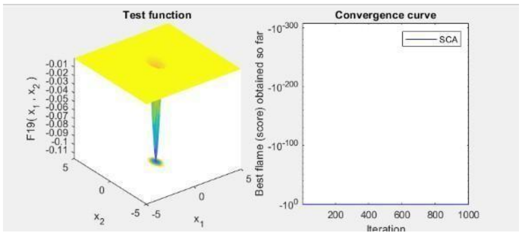
In, F16 hybrid algorithm achieves optimal value is -1.0316 same as to the original value -1.0316.



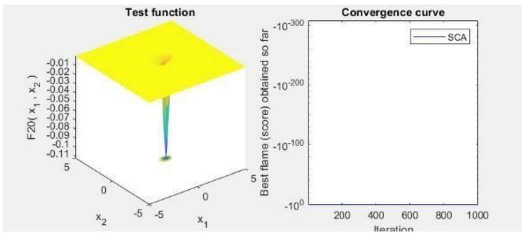
In, F17 hybrid algorithm achieves a more optimal value is 0.39789 compared to the original value 0.39798.



In, F18 hybrid algorithm achieves optimal value is 3 same as to the original value 3.

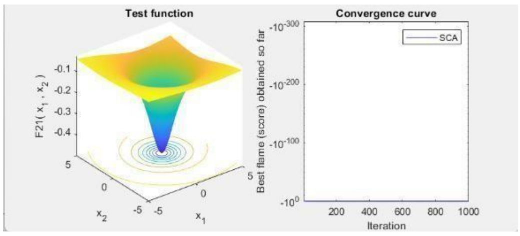


In, F19 hybrid algorithm achieves a more optimal value is -3.8549 compared to the original value -3.8545.

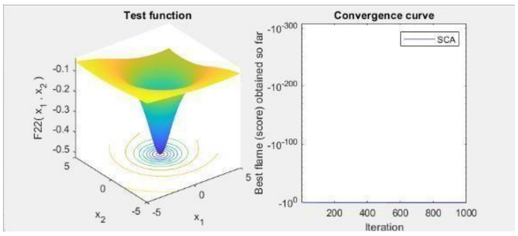


In, F20 hybrid algorithm achieves a more optimal value is -2.0612 compared to the original value -3.107.

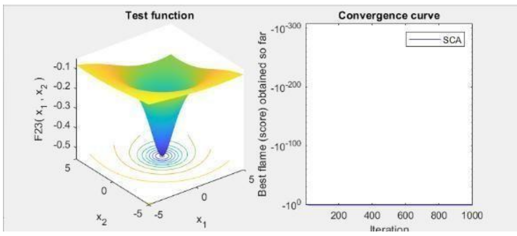
DOI: https://doi.org/10.5281/zenodo.15637294



In, F21 hybrid algorithm achieves a more optimal value is 0.88098 compared to the original value -0.87819.



In, F22 hybrid algorithm achieves a more optimal value is -5.1263 compared to the original value -0.90674.



In, F23 hybrid algorithm achieves a more optimal value is -5.1285 compared to the original value -0.9464.

Result and Discussion

Functions	Value of Original	Value of Hybridization
F1	1.11E-28	2.44E-138
F2	2.86E-20	1.71E-89
F3	9.14E-14	1.71E-58
F4	2.85E-08	3.69E-39
F5	7.2304	4.783
F6	0.23014	0.00012313
F7	0.0022384	0.0011824
F8	-2185.9955	-3107.3549
F9	0	0
F10	1.47E-14	4.00E-15
F11	0	0
F12	0.073301	0.00011961
F13	0.43108	0.00023499
F14	0.99802	0.998
F15	0.0013812	0.00030755
F16	-1.0316	-1.0316
F17	0.39798	0.39789
F18	3	3
F19	-3.8545	-3.8549
F20	-3.107	-2.0612
F21	-0.87819	-0.88098
F22	-0.90674	-5.1263
F23	-0.9464	-5.1285

When SINE COSINE ALGORITHM (SCA) And ANT LION OPTIMIZER (ALO) are
hybridized, the optimization performance is better than when SCA is used alone. In fifteen (15) of the twenty-three (23) benchmark functions, the hybridized algorithm yields excellent results for optimal solutions.

The hybrid approach greatly enhances the quality of solutions for high-dimensional functions, where standalone algorithms have a tendency to deteriorate, according to the comparative performance analysis. The SCA-ALO is more resilient to repeated runs and less sensitive to initial conditions. For deployment in vital systems like control systems, financial forecasting, and medical diagnostics, this consistency is essential.

Furthermore, the hybrid algorithm's convergence curves demonstrate its capacity to sustain a constant rate of improvement over iterations. Their hybridization guarantees a more progressive and dependable path to optimality than SCA, which may stagnate early, or ALO, which may take longer to leave sub-optimal regions. This behavior demonstrates how well the hybrid performs in both local and global search dimensions.

DOI: https://doi.org/10.5281/zenodo.15637294

3. Conclusion

The SINE COSINE ALGORITHM (SCA) and ANT LION OPTIMIZER (ALO) Algorithm hybridization has been tested on twenty-three (23) benchmark functions (F1-F23). F1–F2–F3–F4–F6–F7–F10, F12, F14, F16, F19, F20, F21, and F22 are the 15 functions where it has been found to perform better and provide optimal values. The best examples of these are F9, F11, and F18, which score optimal values and are thus the hybrid algorithm's goal for better optimization.

The experiment demonstrates how the SCA-ALO hybrid performs better when solving benchmark and real-world optimization problems. Its general-purpose usefulness is demonstrated by its consistent performance across a variety of test functions, and its adaptability makes it a good fit for dynamic and real-time environments. To locate global optima without compromising computational efficiency, the algorithm must be able to strike a balance between exploration and exploitation.

Improvements like parameter adaptation, parallelization for quicker computation, and integration into hybrid deep learning models can be investigated in future research. To customize the algorithm for specialized applications like energy systems, medical diagnostics, or smart manufacturing, more research could also incorporate domain-specific heuristics. A new generation of metaheuristic hybrids with improved performance characteristics and wider application potential is made possible by the study's findings.

References

[1] Mirjalili, S. (2016). Sine Cosine Algorithm for Optimization. Knowledge-Based Systems, 96, 120–133. https://doi.org/10.1016/j.knosys.2015.12.022

[2] Mirjalili, S. (2015). The Ant Lion Optimizer. Advances in Engineering Software, 83, 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010

[3] Yang, X. S. (2010). Nature-Inspired Metaheuristic Algorithms. Luniver Press.

[4] Talbi, E.-G. (2009). Metaheuristics: From Design to Implementation. Wiley.

[5] Fister, I., Yang, X. S., Fister Jr, I., Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. Electrotechnical Review, 80(1-2), 1–7.

[6] Layeb, A. (2011). A novel quantum inspired cuckoo search for knapsack problems. International Journal of Bio-Inspired Computation, 3(5), 297–305.

[7] Rahnamayan, S., Tizhoosh, H. R., Salama, M. M. A. (2008). Opposition-based differential evolution. IEEE Transactions on Evolutionary Computation, 12(1), 64–79.

[8] ZainEldin, H., et al. (2020). Hybrid metaheuristic algorithms: A comprehensive review. Artificial Intelligence Review, 53(5), 3647–3686. https://doi.org/10.1007/s10462-019-09755-w

[9] Abd Elaziz, M., Oliva, D., Hinojosa, S., & Elsheikh, A. H. (2017). A hybrid optimization algorithm with Sine Cosine Algorithm and Differential Evolution. Knowledge-Based Systems, 145, 110–124.

[10] Jordehi, A. R. (2015). Optimisation of electric power systems using particle swarm optimisation: A review. Renewable and Sustainable Energy Reviews, 52, 229–241.

[11] Chen, H., et al. (2019). A novel metaheuristic algorithm and its applications for parameter optimization of machine learning models. Applied Soft Computing, 80, 579–598.

[12] Abualigah, L., et al. (2021). Nature-inspired optimization algorithms: A comprehensive survey and classification. Artificial Intelligence Review, 55(1), 271–350.

[13] Aljarah, I., Faris, H., & Mirjalili, S. (2018). Optimizing connection weights in neural networks using the whale optimization algorithm. Soft Computing, 22(1), 1–15.

[14] Rakhshani, H., & Pourakbari-Kasmaei, M. (2020). A new hybrid metaheuristic algorithm for solving complex optimization

DOI: https://doi.org/10.5281/zenodo.15637294

problems. Engineering Applications of Artificial Intelligence, 94, 103778.
[15] Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (2013). Metaheuristic algorithms in modeling and optimization. Metaheuristic Applications in Structures and Infrastructures, Elsevier, 1–24.
[16] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. Journal of Global Optimization, 39(3), 459–471.
[17] Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC), 210–214. https://doi.org/10.1109/NABIC.2009.5393690.