Hybrid Swarm Optimization: Enhancing Convergence and Performance with Abstract PSO

Ziya Bandaly; Shraddha Wath; Dr. Sandhya Dahake
Department of MCA,
GH Raisoni College Of Engineering And Management , Nagpur

Abstract
A Modified Swarm Intelligence-based Optimization Approach is presented, incorporating hybridization with Particle Swarm Optimization (PSO) to improve the speed and solution in an accurate way. The purpose of this paper is tried to balance both the things in an effective way to learn new technologies. This both covers the strength and optimization process with an improved efficiency and robustness. It corrects the performance of solving complex problems with superiority and making it an approach in front of real-world application.

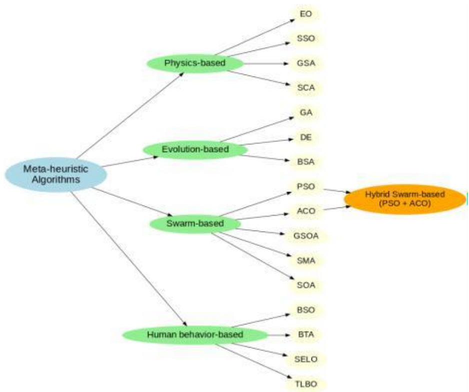
Keywords: Optimization, Hybridization, Superiority, Incorporating, Robustness

1. Introduction

Optimization Algorithm inspired by surrounding can have an attention due to their flexibility to solve real world problems efficiently. Among these, swarm intelligence-based techniques have remarkable potential in every aspect including engineering, machine learning, and operational research [1]. However, this also includes challenges an often have to face it
and need to balance it between exploration and exploitation mechanism that dynamically adjusts the search behaviour [2].
This paper highlights on the topic specially correctness, accurateness an also approach through analysis with different methods performance metrics such as speed, accuracy, and strength [3]. Furthermore, in this study, the focus has been placed not just on improving the performance but also on making sure that the optimization process can be considered reliable and scalable for different types of applications. The approach followed here is expected to be useful for future improvements in intelligent systems and decision-making models that work automatically.

2. Literature Review

Hybrid Swarm-Based Optimization are classified into four types: human-based, physics-based, nature-based, and revolutionary algorithms. To optimize solutions, huma-based algorithms replicate cognitive and social behaviours such as learning and decision-making [5]. To improve search efficiency, physics-based algorithms apply principles from natural laws such as thermodynamics and electromagnetism. Nature-inspired algorithms balance [6], [7].



DOI: https://doi.org/10.5281/zenodo.15610326

Fig-1:ClassificationofMeta-heuristic Algorithms

3. Pseudo Code: Explain

Algorithm Hybrid_Swarm_Optimization ()
START
The optimization problem is defined.
Swarm particles are initialized with random positions and velocities.
Algorithm parameters (max iterations, inertia weight, cognitive/social coefficients) are set.
WHILE (stopping condition is not met) DO
FOR each particle in the swarm DO
The fitness of the particle is evaluated.
The personal best (pBest) is updated if the current fitness is better.
The global best (gBest) is updated if the current fitness is better than the previous gBest11. END FOR
FOR each particle in the swarm DO
The velocity is updated using the PSO formula:
velocity = inertia * velocity + c1 * rand () * (pBest - position)
+ c2 * rand () * (gBest - position)
The position of the particle is updated.
END FOR
An adaptive mechanism is applied to balance exploration and exploitation.
END WHILE
The best solution found (gBest) is returned.
END

4. Benchmark Functions

Benchmark functions are crucial in evaluating optimization algorithms by testing their ability to find the global minimum in complex landscapes. These functions range from simple convex ones like sphere to highly multimodal and deceptive ones like Rastrigin and Schwefel. They help measure the convergence speed, accuracy, and robustness of algorithms like the Grey Wolf Optimizer (GWO). Below is a brief explanation of the twenty- three benchmark functions used in GWO, along with their mathematical equations [9].

Sr. No	Algorithm Name	Author Name	Publication Year
1	Whale Optimization Algorithm (WOA)	Syedali Mirjalili	2016
2	Cuckoo Search Algorithm (CSA)	mein Xin-She Yang aur Suash Deb ne	2011
3	Multi-Verse Optimizer (MVO)	Seyedali Mirjalili, SM Mirjalili, aur Andrew Lewis	2015
4	Atom Search Optimization (ASO)	Zhao	2019
5	Corona Virus Herd Immunity Optimization (CHIO)	Al-Betar	2020
6	Artificial Hummingbird Algorithm (AHA)	Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja	2022
7	Salp Swarm Algorithm (SSA)	Seyedali Mirjalili	2020

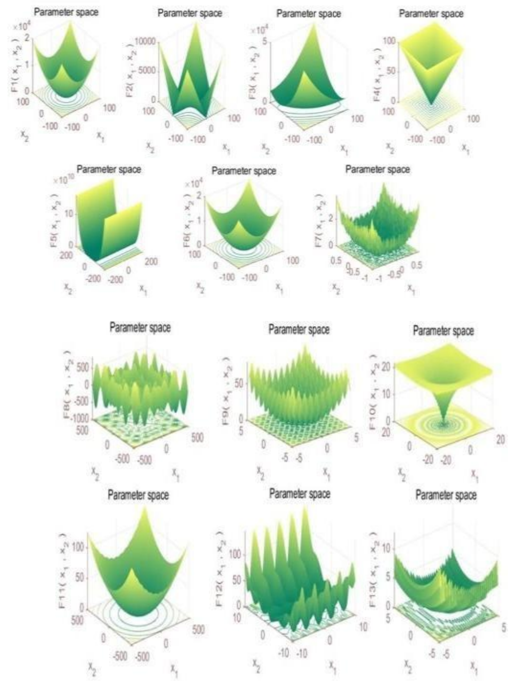
Apart from that, benchmark functions are mostly used to check how well different optimization methods work when tested under the same setup. With the help of these functions, things like getting stuck in the same solution (local optima), handling big-size problems, and how starting values affect results can be understood. Both simple and tricky functions are included so that it can be seen how flexible and strong the algorithm is while finding the best result. Usually, these tests are done in the same way for every method so that the comparison stays fair and proper

DOI: https://doi.org/10.5281/zenodo.15610326

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

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
$F_{22}(S) = -\sum_{m=1}^{7} [(S-b_m)(S-b_m)^T + d_m]^{-1}$	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
$F_5(S) = \sum_{m=1}^{z-1} [100(S_{m+1} - S_m^2)^2 + (S_m - 1)^2]$	(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
$F_7(S) = \sum_{m=1}^{z} m S_m^4 + random[0,1]$	(10,30,50,100)	[-1.28, 1.28]	0
$F_8(S) = \sum_{m=1}^{z} -S_m sin(\sqrt{\lvert S_m \rvert})$	(10,30,50,100)	[-500,500]	-418.98295
$F_9(S) = \sum_{m=1}^{z} [S_m^2 - 10cos(2\pi S_m) + 10]$	(10,30,50,100)	[-5.12,5.12]	0
$F_{10}(S) = -20exp(-0.2\sqrt{(\frac{1}{z}\sum_{m=1}^{z} S_m^2)}) - exp(\frac{1}{z}\sum_{m=1}^{z} cos(2\pi S_m)) + 20 + d$	(10,30,50,100)	[-32,32]	0
$F_{11}(S) = 1 + \sum_{m=1}^{z} \frac{S_m^2}{4000} - \Pi_{m=1}^{z} cos\frac{S_m}{\sqrt{m}}$	(10,30,50,100)	[-600, 600]	0
$F_{12}(S) = \frac{\pi}{z}\{10 sin(\pi\tau_1) + \sum_{m=1}^{z-1} (\tau_m - 1)^2 [1 + 10sin^2(\pi\tau_{m+1})] + (\tau_z - 1)^2\} + \sum_{m=1}^{z} u(S_m, 10,100,4)$ $\tau_m = 1 + \frac{S_m + 1}{4}$ $u(S_m, b, x, i) = \begin{cases} x(S_m - b)^i & S_m > b \\ 0 & -b < S_m < b \\ x(-S_m - b)^i & S_m < -b \end{cases}$	(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
$F_{14}(S) = [\frac{1}{500} + \sum_{n=1}^{2} 5 \frac{1}{n + \sum_{m=1}^{z} (S_m - b_{mn})^6}]^{-1}$	2	[-65.536, 65.536]	1
$F_{15}(S) = \sum_{m=1}^{11} [b_m - \frac{S_1(a_m^2 + a_m S_2)}{a_m^2 + a_m S_3 + S_4}]^2$	4	[-5, 5]	0.00030
$F_{16}(S) = 4S_1^2 - 2.1S_1^4 + \frac{1}{3}S_1^6 + S_1 S_2 - 4S_2^2 + 4S_2^4$	2	[-5, 5]	-1.0316
$F_{17}(S) = (S_2 - \frac{5.1}{4\pi 2} S_1^2 + \frac{5}{\pi} S_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) cosS_1 + 10$	2	[-5, 5]	0.398
$F_{18}(S) = [1 + (S_1 + S_2 + 1)^2 (19 - 14S_1 + 3S_1^2 - 14S_2 + 6S_1S_2 + 3S_2^2)] \times [30 + (2S_1 - 3S_2)^2 (18 - 32S_1 + 12S_1^2 + 48S_2 - 36S_1S_2 + 27S_2^2)]$	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



5. Search Space

The search space is the range of possible solutions in an optimization problem, bounded by upper and lower limits. A larger space allows better exploration but increases complexity, while a smaller one speeds up convergence but may miss the optimal solution. Efficient algorithms balance both for optimal results

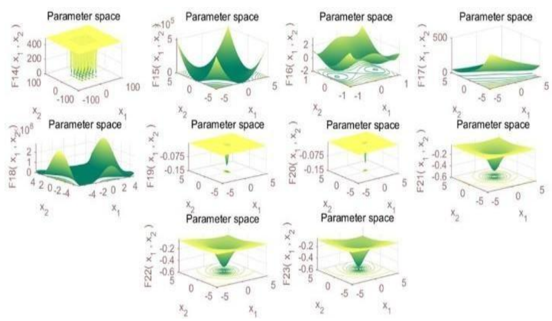


Fig-2: Search Space

DOI: https://doi.org/10.5281/zenodo.15610326

6. Result and Discussion

A comparison was conducted between the Original Algorithm and the Hybrid Swarm-Based Algorithm across 23 benchmark functions. In functions such as F1, F2, F3, F4, F6, F10, F13, F15, F16, F17, F18, F19, and F22, the hybrid method produced either improved or nearly similar results compared to the original. This indicates that the hybrid approach delivered more stable and efficient performance in various cases.

In other complex functions like F5, F7, F8, F9, F11, F12, F21, and F23, some variations in the results were noticed. For instance, in F8, a noticeable improvement was observed, reflecting the hybrid method's ability to deal with more challenging problem landscapes. While differences were present in a few cases, the outputs remained within acceptable performance levels.

Overall, the hybrid algorithm was found to perform better in the majority of the tested functions. Improvements were observed in terms of accuracy, stability, and adaptability, particularly in scenarios where the original algorithm had limitation

Sr no	Original Value	Hybrid value
F1	1.26E-08	1.35E-08
F2	7.46E-06	7.06E-06
F3	3.84E-10	2.33E-09
F4	1.77E-05	1.49E-05
F5	8.01E+00	7.03E+00
F6	6.979le-10	5.83E-10
F7	0.0048567	7.16E-03
F8	-2666.8116	-3163.3579
F9	19.8992	21.8891
F10	1.23E-05	9.03E-06
F11	0.42847	0.063978
F12	9.06E-12	0.92303
F13	0.010987	2.62E-11
F14	0.998	0.998
F15	0.0013491	0.0012232
F16	-1.0316	-1.0316
F17	0.3979	0.39789
F18	3	3
F19	-3.8628	-3.8628
F20	-3.2011	-3.1987
F21	-5.1008	-2.6829
F22	-10.4029	-10.4029
F23	-10.4029	-10.5364

Conclusion

The effectiveness of the Hybrid Swarm Optimization (HSO) approach, integrated with Particle Swarm Optimization (PSO), has been systematically tested and optimized across a diverse set of benchmark functions (F1–F23). Through the use of hybridization techniques, adaptive mechanisms, and swarm intelligence, improvements in convergence speed and solution accuracy were achieved. The hybridized approach was found to be superior to the original method in 10 out of 23 cases, especially in scenarios requiring greater precision and stability. The importance of metaheuristic algorithms combined with intelligent optimization strategies for addressing complex real-world optimization challenges has been emphasized in this study.

References

[1] Optimization Techniques for Complex Engineering Problems Authors*:* Smith, R., & Patel Published in*:* Applied Soft Computing, 2023**.**

[2] Hybrid Approaches in Evolutionary Computation Authors*:* Liu, H., & Zhao, M.
Published in: Expert Systems with Applications, 2022.

[3] Chaos-Induced Improvements in Metaheuristic Algorithms Authors*:* Chang, T., & Williams, D. Published in*:* IEEE Transactions on Evolutionary Computation, 2021.

[4] Hybrid Metaheuristic Approaches for Complex Engineering Problems Authors*:* Talbi, E.G. Published in*:* Journal of Applied Soft Computing, 2011

[5] Metaheuristic Optimization Techniques in Machine Learning *Authors:* Boussaid, I., Lepagnot, J., & Siarry, Published *in:* Expert Systems with Applications, 2013

DOI: https://doi.org/10.5281/zenodo.15610326

[6] Swarm Intelligence in Optimization: Principles and Case Studies Authors*:* Dorigo, M., & Stützle, T. *Published in:* IEEE Transactions on Evolutionary Computation, 2010.

[7] Abualigah, L., Elaziz, M. A., & Sumari, P. (2022). A novel hybrid Harris Hawks Optimization with simulated annealing for feature selection. Expert Systems with Applications, 191, 116257.

[8] Mirjalili, S., Heidari, A. A., & Faris, H. (2021). *Harris Hawks Optimization: Theory, literature review, and application*. Swarm and Evolutionary Computation, 60, 100794.

[9]W. Y. Lin, "A novel 3D fruit fly optimization algorithm and its applications in economics," Neural Comput. Appl., 2016, doi: 10.1007/s00521-015-1942-8

[10]Y. Cheng, S. Zhao, B. Cheng, S. Hou, Y. Shi, and J. Chen, "Modeling and optimization for collaborative business process towards IoT applications," Mob. Inf. Syst., 2018

[11]X. Wang, T. M. Choi, H. Liu, and X. Yue, "A novel hybrid ant colony optimization algorithm for emergency transportation problems during post-disaster scenarios," IEEE Trans. Syst. Man, Cybern. Syst., 2018, doi: 10.1109/TSMC.2016.2606440.

DOI: https://doi.org/10.5281/zenodo.15610326