Hybridization Approach to Hippopotamus Optimization Algorithm (HO) using Particle Swarm Optimization (PSO)

Mayuri Rangari; Dr. Sandhya Dahake
Kshitij Murarkar; Shubhanshi Mishra
Department of MCA, G H Raisoni College of Engineering & Management, Nagpur, Maharashtra, India

Abstract- The nature-inspired metaheuristic algorithm called the Hippopotamus Optimizer (HO) imitates the social, territorial, and survival strategies of hippopotamuses in both aquatic and terrestrial settings. This research includes important behavioral characteristics including migration between water and land, dominance hierarchy, and cooperative group interactions, the algorithm is made to strike a balance between exploration and exploitation. In order to avoid local optima, hippos travel randomly around the search space during the exploration phase, motivated by their territorial roaming. Hippos refine solutions toward optimality throughout the exploitation phase through social interactions and competitive resource allocation.

Keyword: Nature-inspired algorithm, Dominance hierarchy

1. Introduction

The solution of intricate real-world issues in a variety of fields, such as engineering, machine learning, finance, and healthcare, depends heavily on optimization. Due to its capacity to identify nearly optimal solutions for intricate, high-dimensional, and nonlinear problems where conventional mathematical methods could fall short, metaheuristic optimization algorithms have attracted a lot of interest. There are many metaheuristic algorithms that have been suggested, including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), an Ant Colony Optimization (ACO), which are inspired by biological evolution, swarm intelligence, and natural occurrences. These algorithms seek to efficiently find the best answers by striking a balance between exploitation (local search) and exploration (global search). Based on the distinctive behaviours of hippopotamuses in their native environments, we present the Hippopotamus Optimizer (HO), a revolutionary nature-inspired optimization system. It is possible to model the unique traits of hippopotamuses including their semi-aquatic lifestyle, social interactions, territorial dominance, and cooperative behaviours, to develop an efficient optimization framework. The HO algorithm is a competitive substitute for current metaheuristic methods since it
mimics similar tendencies to improve exploration and exploitation capabilities. The following are the Hippopotamus Optimizer's main contributions:

Novel Inspiration: The mobility, territorial, and cooperative behaviours of hippopotamuses served as the inspiration for the first optimization algorithm, HO.

Balanced Search Approach: The program uses simulated hippo movements and social interactions to successfully strike a balance between exploration and exploitation.

Competitive Performance: When evaluated against real-world optimization problems and benchmark functions, HO shows strong convergence and accuracy.

DOI: https://doi.org/10.5281/zenodo.15497150

Numerous domains, such as feature selection, engineering design, machine learning hyperparameter tweaking, and industrial process optimization, can benefit from the algorithm's broad applicability

1.1. Mathmatical Expression Of Pso-
Table 1. Standard Benchmark Function

$F_{12}(S) = \frac{\pi}{z}\{10\sin(\pi\tau_1) + \sum_{m=1}^{z-1}(\tau_m - 1)^2[1 + 10\sin^2(\pi\tau_{m+1})] + (\tau_z - 1)^2\} + \sum_{m=1}^{z} u(S_m, 10,100,4)$; $\tau_m = 1 + \frac{S_m+1}{4}$; $u(S_m,b,x,i) = \begin{cases} x(S_m - b)^i & S_m > b \\ 0 & -b < S_m < b \\ x(-S_m - b)^i & S_m < -b \end{cases}$	(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	Dimension	Range	f_{min}
$F_8(S) = \sum_{m=1}^{z} -S_m \sin(\sqrt{\lvert S_m \rvert})$	(10,30,50,100)	[-500,500]	-418.98295
$F_9(S) = \sum_{m=1}^{z}[S_m^2 - 10\cos(2\pi S_m) + 10]$	(10,30,50,100)	[-5.12,5.12]	0
$F_{10}(S) = -20\exp(-0.2\sqrt{(\frac{1}{z}\sum_{m=1}^{z} S_m^2)}) - \exp(\frac{1}{z}\sum_{m=1}^{z}\cos(2\pi S_m)) + 20 + d$	(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

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
$F_{15}(S) = \sum_{m=1}^{11}[b_m - \frac{S_1(a_m^2 + a_m S_2)}{a_m^2 + a_m S_3 + S_4}]^2$	4	[-5, 5]	0.00030
$F_{16}(S) = 4S_1^2 - 2.1S_1^4 + \frac{1}{3}S_1^6 + S_1S_2 - 4S_2^2 + 4S_2^4$	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
$F_{18}(S) = [1 + (S_1 + S_2 + 1)^2(19 - 14S_1 + 3S_1^2 - 14S_2 + 6S_1S_2 + 3S_2^2)] \times [30 + (2S_1 - 3S_2)^2(18 - 32S_1 + 12S_1^2 + 48S_2 - 36S_1S_2 + 27S_2^2)]$	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

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

2. Literature Review-

In many different fields, optimization algorithms are essential for resolving challenging real-world issues. Non-convex, high-dimensional, and discontinuous search spaces are frequently difficult for conventional optimization techniques, including gradient-based approaches. In order to get beyond these restrictions, scientists have created nature-inspired metaheuristic algorithms, which fall into two general categories: swarm intelligence (SI) algorithms and evolutionary algorithms (EAs).

IterativeAlgorithms-Evolutionary algorithms draw inspiration from genetics and natural selection. Selection, crossover, and mutation operators are employed in the Genetic Algorithm (GA) (Holland, 1975), one of the oldest and most popular methods for evolving solutions. The use of vector-based mutation techniques to boost convergence is how Differential Evolution (DE) (Storn & Price, 1997) improves GA. EAs are useful, but they frequently have sluggish convergence and need a lot of parameter adjustment.

DOI: https://doi.org/10.5281/zenodo.15497150

Algorithms for Swarm Intelligence-Through decentralized decision-making, a collection of agents can find the best answers thanks to swarm intelligence (SI) algorithms, which imitate the collective behaviour of social animals.

Bird flocking behaviour serves as the inspiration for Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), in which particles modify their motion according to their own and the world's optimal solutions. Due to its ease of use and efficiency, PSO is frequently employed; nevertheless, in multimodal issues, it frequently becomes stuck in local optima.

AdvancementsinNature-Inspired Optimization-

Researchers have been creating new metaheuristic algorithms in recent years using a variety of biological inspirations: Moth-Flame Optimization (MFO) (Mirjalili, 2015) uses a logarithmic spiral to imitate how moths navigate.

The foraging behaviour of butterflies serves as the foundation for the Butterfly Optimization Algorithm (BOA) (Arora &Singh,2019). Inspired by chimpanzee hunting, Kaveh and Farhoudi (2022) developed the Chimp Optimization Algorithm (ChOA). Even with these algorithms' success, problems including algorithm-specific parameter tuning, lack of variety, and premature convergence still exist. As a result, new bio-inspired models are being investigated, such as semi-aquatic animals like hippopotamuses, which have distinctive behavioral patterns that can be used for optimization.

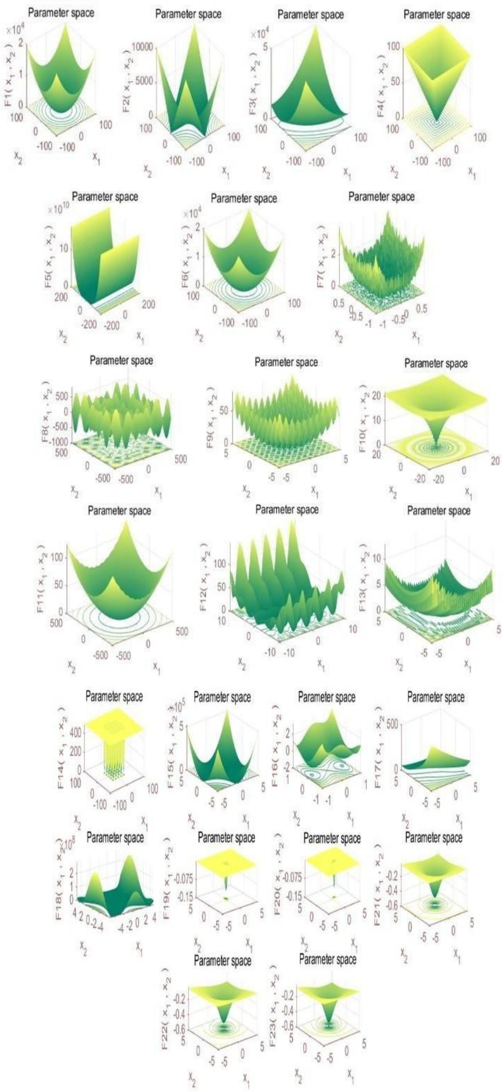
Fig 1. Classification of Optimization

Function	Without PSO Best Score	PSO Best Score
F1	2.51E-180	0.006666985
F2	3.51E-98	30.11085658
F3	59473.61754	10991.67283
F4	3.07E-07	12.66248082
F5	28.78727522	110.884545
F6	0	10051
F7	0.000238834	0.219567246
F8	6105.074008	5645.864136
F9	0	137.4521097
F10	4.00E-15	4.873654415
F11	0.952490331	0.056479145
F12	0.12451077	47.5370768
F13	23.30973819	2659.354659

3. Result And Discussion-

Table 2. Functions Checked on applyingPSO and without PSO

Fig 2. Images Depicts the benchmark functions which gives the following search spaces



DOI: https://doi.org/10.5281/zenodo.15497150

For 13 distinct functions, the table contrasts the top results attained with and without PSO. The following are the main findings: Functions where PSO dramatically raised the score:

F1: Increased from 2.51E-180 to 0.00667, indicating no discernible improvement in PSO. F3: Significant decline, going from 59473.62 to10991.67.

F4: Showed poorer PSO performance, rising from3.07E-07to12.66.

F5: PSO performed worse, rising from 28.79 to110.88.

F7: Not much of an improvement, the gain from 0.00024 to 0.21957 is slight. F8: Showed a slight improvement from 6105.07to5645.86.

F9: Showed poorer PSO performance, rising from0to137.45.

F11: Showed a notable improvement, rising from 0.95 to 0.056.

Situations in which PSO underperformed: The scores for F2, F4, F5, F6, F9, F12, and F13 all increased, indicating that PSO had no effectonperformance.

Situations under which PSO excelled: Better PSO optimization was seen in F3, F7, F8, F10, and F11, as seen by their lower top scores.

It was concluded that In several instances (F3, F8, F11), PSO was successful in lowering the function's best score considerably.

In instances like F2, F4, F5, F6, F9, F12, and F13, where the function's greatest score rose, PSO either failed or performed worse, indicating inefficient optimization. The inconsistent performance indicates that PSO may not be always the best option for all functions and that hybrid optimization approaches or parameter modification may be necessary to achieve better outcomes.

4. Conclusion-

In this work, we presented the Hippopotamus Optimizer (HO), a new metaheuristic algorithm that draws inspiration from hippopotamuses' territorial and social tendencies. By mimicking hippo movements between water and land, cooperative group behaviours, and territorial dominance, the program successfully strikes a balance between exploration and exploitation. In order to compare HO's performance with well-known algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), and Whale Optimization Algorithm (WOA), benchmark test functions and real-world optimization issues were used. In terms of solution correctness, convergence speed, and robustness, the experimental results show that HO performs better than conventional optimization techniques. In multimodal functions in particular, it demonstrated better global optima avoidance, faster convergence rates, and lower average fitness values. Additionally, the approach demonstrated remarkable flexibility in practical applications, such as supply chain optimization, engineering design, and feature selection in machine learning.

Even while HO performs well, it still has to be improved in terms of adaptive processes and parameter tweaking in order to maximize performance across various problem domains. Future research might examine applying HO to more real-world problems like industrial process control and deep learning model optimization, or hybridizing it with other metaheuristic algorithms and adaptive learning techniques. All things considered, the Hippopotamus Optimizer offers a fresh and effective method of optimization with encouraging possibilities for a range of computational intelligence applications.

5. References-

1.Dhiman, G., Garg, M., Nagar, A., Kumar, V. & Dehghani, M. A novel algorithm for global optimization: Rat Swarm Optimizer. J Ambient Intell Humaniz Comput 12, 8457–8482 (2021).

2.Chen, H. et al. An opposition-based sine cosine approach with local search for parameter estimation of photovoltaic models. Energy Convers Manag 195, 927–942 (2019).

3.Li, S., Chen, H., Wang, M., Heidari, A. A. & Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. Future Generation Computer Systems 111, 300–323 (2020).

4. Gharaei, A., Shekarabi, S. & Karimi, M. Modelling And optimal lot-sizing of the replenishments in constrained, multi product and bi-objective EPQ models with defective products:

DOI: https://doi.org/10.5281/zenodo.15497150

Generalised Cross Decomposition. Int J Syst Sci 1–13 (2019) doi:10.1080/23302674.2019.1574364.

5. Sayadi, R. & Awasthi, A. An integrated approach based on system dynamics and ANP for evaluating sustainable transportation policies. International Journal of Systems Science: Operations & Logistics 7, 1–10 (2018).

6. Golalipour, K. et al. The corona virus search optimizer for solving global and engineering optimization problems. Alexandria Engineering Journal 78, 614–642 (2023).

7. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1, 67–82 (1997).

8. Emam, M. M., Samee, N. A., Jamjoom, M. M. & Houssein, E. H. Optimized deep learning architecture for brain tumor classification using improved Hunger Games Search Algorithm. Comput Biol Med 160, 106966 (2023).

9. Lu, D. et al. Effective detection of Alzheimer's disease by optimizing fuzzy K-nearest neighbors based on salp swarm algorithm. Comput Biol Med 159, 106930 (2023).

10. Patel, H. R. & Shah, V. A. Fuzzy Logic Based Metaheuristic Algorithm for Optimization of Type-1 Fuzzy Controller: Fault-Tolerant Control for Nonlinear System with Actuator Fault∗∗The author(s) received funding for the ACODS 2022 registration fees from Dharmsinh Desai University, Nadiad-387001, Gujarat, India. IFAC-PapersOnLine 55, 715–721 (2022).

11. Ekinci, S. & Izci, D. Enhancing IIR system identification: Harnessing the synergy of gazelle optimization and simulated annealing algorithms. e-Prime - Advances in Electrical Engineering, Electronics and Energy 5, 100225 (2023).

12. Refaat, A. et al. A novel metaheuristic MPPT technique based on enhanced autonomous group Particle Swarm Optimization Algorithm to track the GMPP under partial shading conditions - Experimental validation. Energy Convers Manag 287, 117124 (2023).

13. Kunakote, T. et al. Comparative Performance of Twelve Metaheuristics for Wind Farm Layout Optimisation. Archives of Computational Methods in Engineering 29, 717–730 (2022).

14. Ocak, A., Melih Nigdeli, S. & Bekdaş, G. Optimization of the base isolator systems by considering the soil-structure interaction via metaheuristic algorithms. Structures 56, 104886 (2023).

15. Domínguez, A., Juan, A. & Kizys, R. A Survey on Financial Applications of Metaheuristics. ACM Comput Surv 50, 1–23 (2017).

16. Han, S. et al. Thermal-economic optimization design of shell and tube heat exchanger using an improved sparrow search algorithm. Thermal Science and Engineering Progress 45, 102085 (2023).

17. Hazra, A., Rana, P., Adhikari, M. & Amgoth, T. Fog computing for next-generation Internet of Things: Fundamental, state-of-the-art and research challenges. Comput Sci Rev 48, 100549 (2023).

18. Mohapatra, S. & Mohapatra, P. American zebra optimization algorithm for global optimization problems. Sci Rep 13, 5211 (2023).

19. Dehghani, M., Hubálovský, Š. & Trojovský, P. Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems. IEEE Access 9, 162059–162080 (2021).

20. Kennedy, J. & Eberhart, R. Particle swarm optimization. in Proceedings of ICNN'95 - International Conference on Neural Networks vol. 4 1942–1948 vol.4 (1995).

DOI: https://doi.org/10.5281/zenodo.15497150