Hybridized Parrot Optimizer with Particle Swarm Optimization for Enhanced Optimization Performance

Arti Mule; Aryan Meshram; Sandhya Dahake
Department of Master in Computer Application
G H Raisoni College of Engineering & Management
Nagpur, India

Abstract
The paper proposes a novel hybrid optimization algorithm, PO-PSO which includes Metaheuristic adaptation algorithms play an important role in solving complicated problems with the real world. In this study, a new and improved version of Partot Optimizer (PO) by hybridization to improve convergence speed, solution accuracy and general performance. The parrot optimizer with PSO integrates POS exploration skills with the utilization efficiency of the Optimizer PSO, leading to a more balanced exploration system. The proposed algorithm is evaluated on benchmark -adaptation functions and real world problems, which perform better performance than standard pos, psos and other traditional adaptation methods. Experimental results suggest that HPO-PSO gets rapid convergence and high solution quality, making it a promising approach to complex adaptation functions. This article is a new adaptation method inspired by the specific behavior of the trained pyirhura Molina, which presents Paper Parrot Optimizer (PO)[3].

Keywords:
Hybrid Optimization, Parrot Optimizer (PO), Particle Swarm Optimization (PSO), Metaheuristic Algorithms, Swarm Intelligence, Exploration-Exploitation Balance, Optimization Performance.

1. Introduction

PO algorithm is a modern adaptation technique inspired by the specific behaviour of Pyirhura Molina Parrot. It mimics the four important symptoms seen in a trained parrot - that is, to achieve the primary adaptation goals with forging, comfort, communication and strangers' exploration (diversification) and utilization (intensity). This behaviour is prepared mathematically to be effective to repeat the adaptation process. Evaluation of the proposed algorithm on benchmark functions and real adaptation problems to demonstrate its efficiency. With continuous progress in artificial intelligence (AI), the field of adaptation faces many challenges in both educational research and real-world technical applications [1][2].

Design challenges include natural adaptation problems that require appropriate adaptation techniques and algorithms. As the problems of modern design are more complex, traditional mathematical adaptation methods have become inadequate to provide sharp and effective solutions. Such a traditional approach is a gradient -based algorithm, which solves adaptation problems by using the lens function. In recent years, a significant focus has been focused on overcoming the boundaries of standard adjustment methods and developing more effective alternatives. Technological progress has inspired the emergence of advanced adaptation algorithm that improves efficiency,

DOI: https://doi.org/10.5281/zenodo.15589753

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

accuracy and speed in handling complex adaptation functions. The most important contributions to this work include, develop a new hybrid algorithm that connects PO and PSO for better adjustment performance. Increasing balance between exploration and utilization and reduces premature convergence [1][3].

On the other hand, Particles are self -confidence (PSO) A well -established swarm -based adaptation algorithm known for its effective information sharing mechanism and strong utilization capacity. PSO updates the candidate solutions based on individual experiences and global best solutions, enabling rapid convergence. However, it sometimes suffers from premature convergence and lacks sufficient variation in high -dimensional search spaces. To address these limits, this paper is a hybridized parrot optimizes with flock adaptation (HPO-PSO), which is a new and better version of Parrot Optimizer. By integrating PSO's information sharing mechanism with POS search power, the proposed hybrid approach increases convergence speed, maintains the solution variety and improves the general adaptation performance [2].

2. Literature Review

2.1 Foundation of Development And Algorithm

Adaptation techniques are important for solving complex calculation problems on different domains. Traditional methods often struggle with efficiency and accuracy, causing bio -inspired and flock -based algorithms such as Particle Herd Optimization (PSO) and Parrot Optimizer (PO). Needs hybrid adaptation approach, while individual algorithms such as PO and PSO show promising results, they also have underlying limits, PSO is Fast convergence, but exposed to local optima. PO is Strong investigative skills, but slow convergence. When hybridizing PO with PSO, the proposed method aims to balance exploration and utilization to increase adaptation. Search improves efficiency, PSO's convergence combines POS diverse search behaviour at speed. Addresses the boundaries of the algorithm. Stagnation and recording to Local Optima.

2.2 Evolutionary and flock-based adaptation algorithms

Evolutionary algorithm (EAS) and self - intelligence (SI) techniques have been used to solve adaptation problems. C-based methods inspired by natural group behaviour have shown considerable ability to address non-focused, multi-models and high-dimensional adaptation challenges. Of these, Particle Herd Optimization (PSO) has emerged as a popular method because of its simplicity and ability to effectively detect the search spaces. PSO developed by Kennedy and Aberhart (1995), the PSO mimics the social behaviour of birds and fish by adjusting particle positions based on individual and collective experiences.

2.3 Parrot Optimizer is fresh bio -inspired approach

Parrot Optimizer (PO) is a relatively new customization algorithm inspired by the intelligent behaviour of Pyorrhoea Molina, the parrot. It integrates four primary behaviour - to communicate the most important adjustment mechanisms for dialogue, forging, rest and strangers - exploration (diversification) and utilization (intensity). PO has performed competitive performance compared to traditional adaptation techniques and performed strong search functions and adaptability.

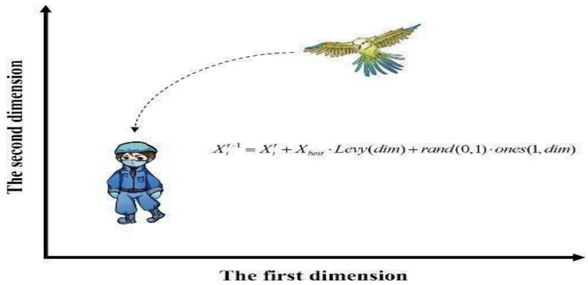


Fig. 1. The Communicating behavior

DOI: https://doi.org/10.5281/zenodo.15589753

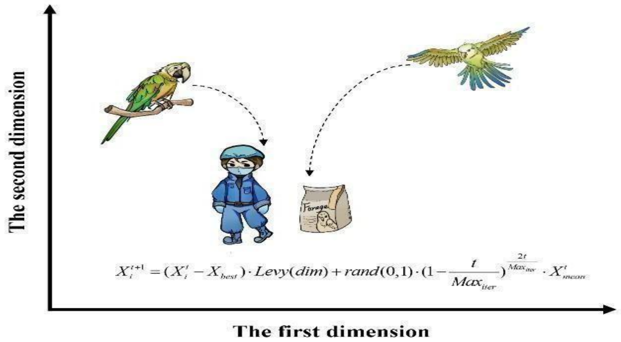


Fig 2: The Forgaging Behaviour

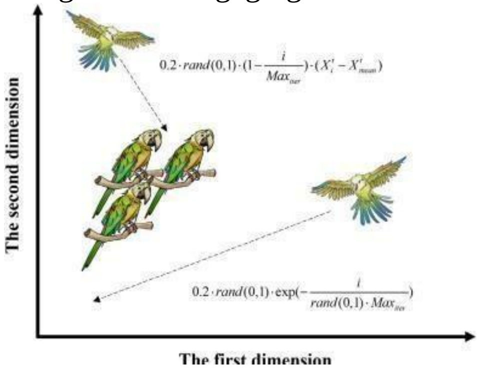


Fig 3: The communicating Behavior

A) Algorithms and Authors

This section mentions the algorithm's name, the principal authors of the research paper and the general purpose.

Table 1: Algorithm, Authors & Year of publishing

Sr. No	Algorithm name	Author name	Year
1	Sine Cosine Algorithm	Seyedali Mirjalili	2016
2	Equilibrium Optimizer	Abdollah Asghari Varzaneh et al	2020
3	Differential Equation	Rainer Stom et al	1997
4	Backtracking Search Algorithm	P Civicioglu	2013
5	Particle Swarm Optimization	James Kennedy et al	1995
6	Slime Moul Algorithm	Mohammed H Saremi	2020
7	Sunflower Evolutionary Optimization Algorithm	Osman K Erol	2021
, 8	Teaching Learning Based Optimization	Rao et al	2011

B) Pseudo Code

In pseudo code, the PO optimization process begins with a random creation of an initial population of candidate solutions. PO's search strategy advances through a sequence of activities, guiding the search towards areas close to the optimal solution or where the best solution is already identified. During the optimization, every solution changes its position dynamically based on the best one found so far within the PO algorithm. The search proceeds until the termination condition is reached. PO integrates exploration

and exploitation approaches, enabling it to effectively explore the search space while slowly converging towards optimal solutions.

1. Initialize Parameters:
   Set population size NN, max iterations Maxterm_iter, and bounds.
2. Initialize Population:
   Randomly place NN parrots (solutions) in the search space.

3. Evaluate Fitness:
   Compute the objective function for each solution.

4. Find Key Positions:
   Identify the best and worst solutions.

5. Iterate Until Convergence:
   - For i=1i = 1 to Maxterm_iter:
   - For each parrot jj:
   - Choose a behaviour randomly (St=Randi ([1,4]) St = Randi ([1,4])):
   - Foraging (St == 1): Move toward the best solution with some randomness.
   - Staying (St == 2): Adjust position slightly toward the best-known position.
   - Communication (St == 3): Update position based on both best and worst positions.
   - Fear of Strangers (St == 4): Move away from the worst solution to avoid bad areas.

6. **The Pseudocode for the Parrot Optimizer Algorithm:**

1: Initialize the PO parameters
2: Initialize the solutions' positions randomly
3: For i = 1: Max_iter do
4: Calculate the fitness function
5: Find the best position and worst position
6: For j = 1: N do
7: St = Randi ([1, 4])
8: Behaviour 1: The foraging behaviour
9: If St == 1 Then

DOI: https://doi.org/10.5281/zenodo.15589753

10: Update position by Eq. (2)
11: Behaviour 2: The staying behaviour
12: Elseif St == 2 Then
13: Update position by Eq. (5)
14: Behaviour: The communicating behaviour
15: Elseif St == 3 Then
16: Update position by Eq. (6)
17: Behaviour 4: The fear of strangers' behaviour
18: Elseif St == 4 Then
19: Update position by Eq. (7)
20: End
21: Return the best solution
22: End change it little bit and extend it with explanation.

C) Standard UM benchmark functions

This experiment compares the proposed algorithm performance to the standard PO for benchmark functions. Details of these functions, which vary from unimodal to multimodal types, are given in Table 2.

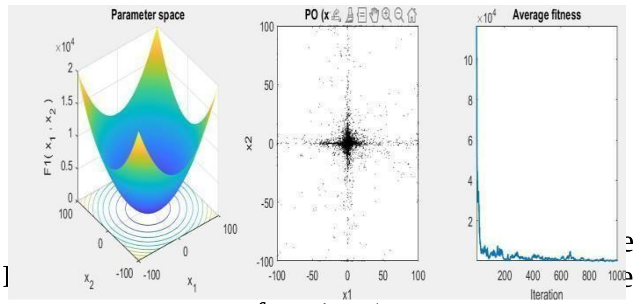
Table 2: Standard UM benchmark functions

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
$F_8(S) = \sum_{m=1}^{z} - S_m sin(\sqrt{\lvert S_m \rvert})$	(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
$F_{16}(S) = 4S_1^2 - 2.1S_1^4 + \frac{1}{3}S_1^6 + S_1S_2 - 4S_2^2 + 4S_2^4$	2	[-5, 5]	-1.0316
$F_{17}(S) = (S_2 - \frac{5.1}{4\pi 2}S_1^2 + \frac{5}{\pi}S_1 - 6)^2 + 10(1-\frac{1}{8\pi})cosS_1 + 10$	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
$F_{22}(S) = -\sum_{m=1}^{7}[(S - b_m)(S - b_m)^T + d_m]^{-1}$	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

3. Results and Discussions

These are the outcomes of the parrot optimizer algorithm from F1 to F23 showing different parameter spaces and the conversion curve.

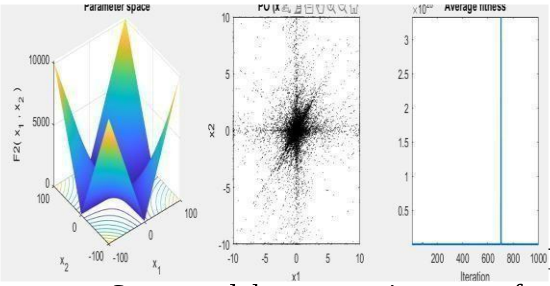
- Function No. 1:



function 1

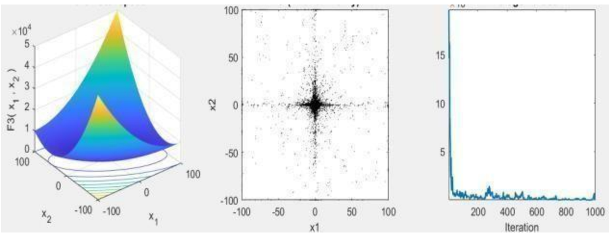
- Function No. 2:

DOI: https://doi.org/10.5281/zenodo.15589753



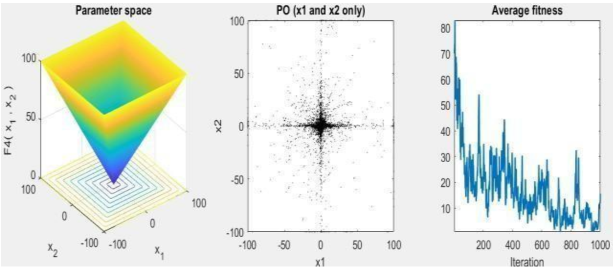
he Parameter Space and the conversion curve for the function 2.

- Function No. 3:



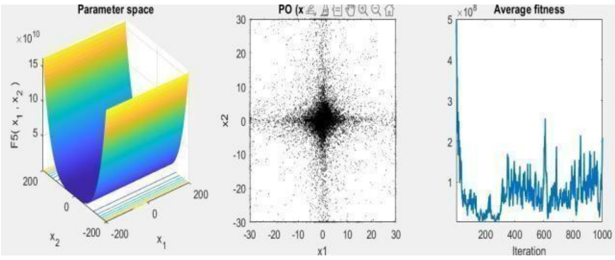
The Parameter Space and the conversion curve for the function 3.

- Function No. 4:



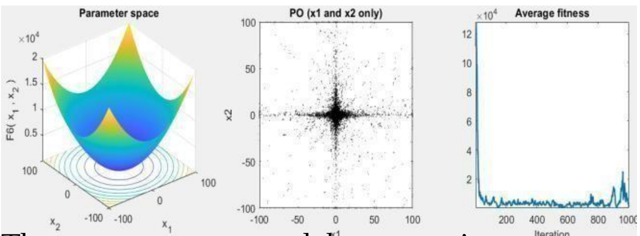
The parameter space and the conversion curve for the function 4.

- Function No. 5:



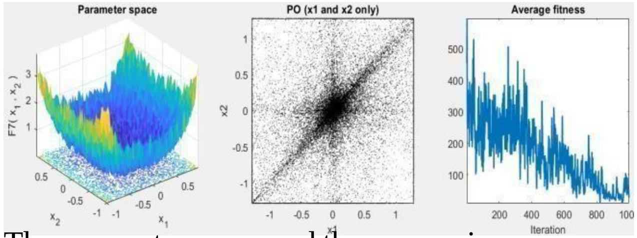
The parameter space and the conversion curve for the function 5.

- Function No. 6:



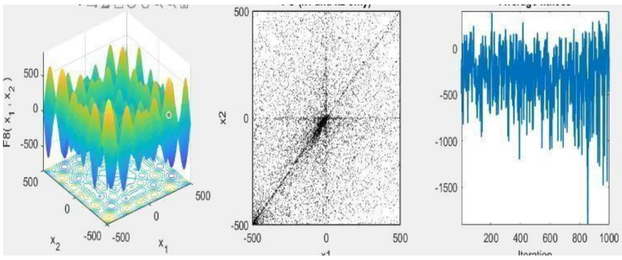
The parameter space and the conversion curve for the function 6.

- Function No. 7:



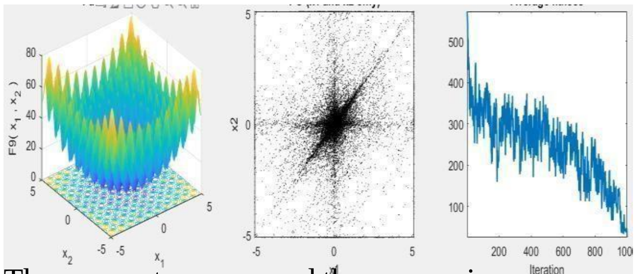
The parameter space and the conversion curve for the function 7.

- Function No. 8:



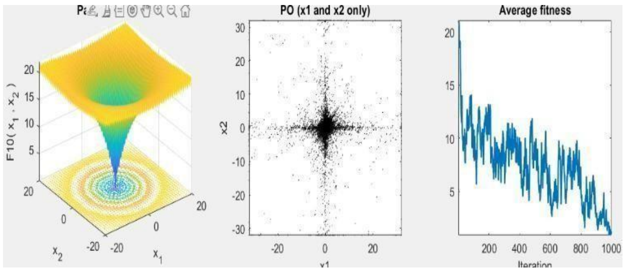
The parameter space and the conversion curve for the function 8.

- Function No. 9:



The parameter space and the conversion curve for the function 9.

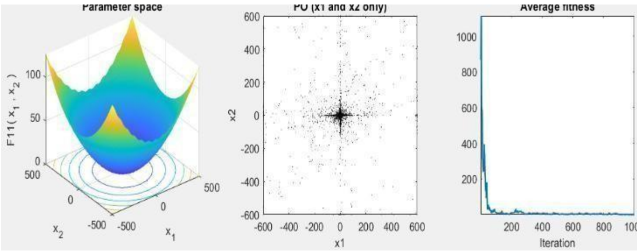
- Function No. 10:



The parameter space and the conversion curve for the function 10.

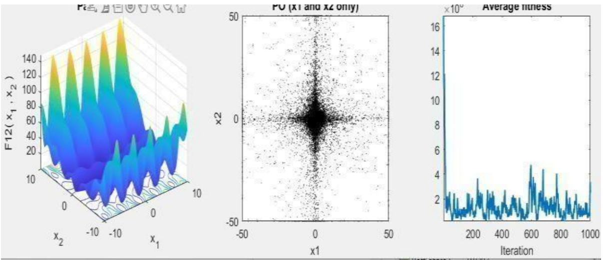
- Function No. 11:

DOI: https://doi.org/10.5281/zenodo.15589753



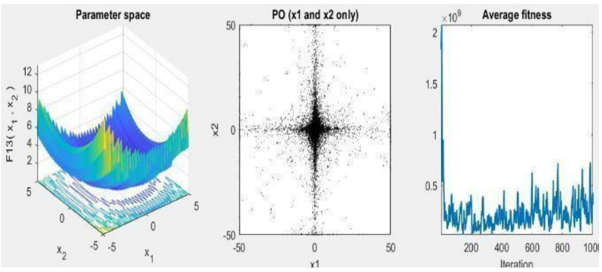
The parameter space and the conversion curve for the function 11.

- Function No. 12:



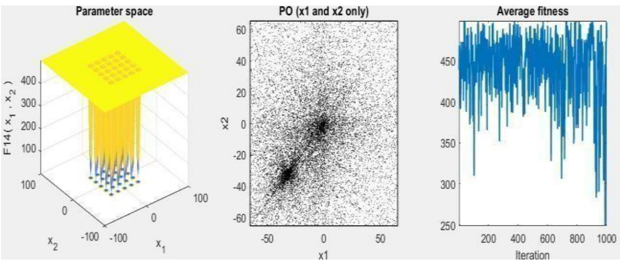
The parameter space and the conversion curve for the function 12.

- Function No. 13:



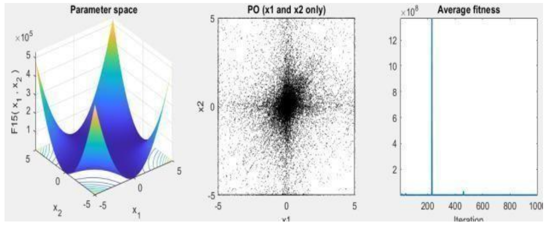
The parameter space and the conversion curve for the function 13.

- Function No. 14:



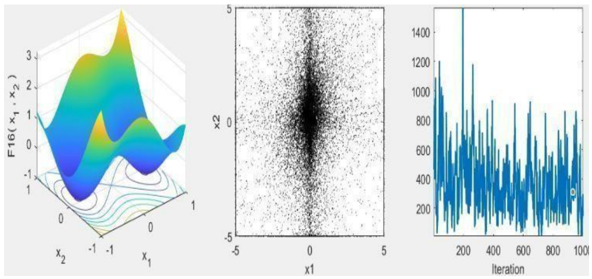
The parameter space and the conversion curve for the function 14.

- Function No. 15:



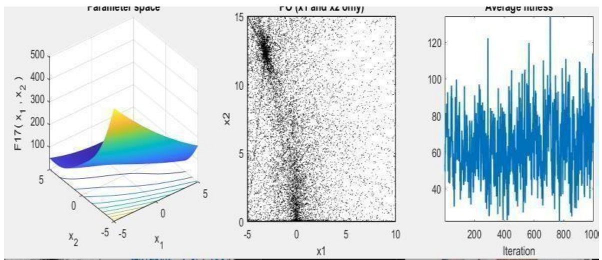
The parameter space and the conversion curve for the function 15.

- Function No. 16:



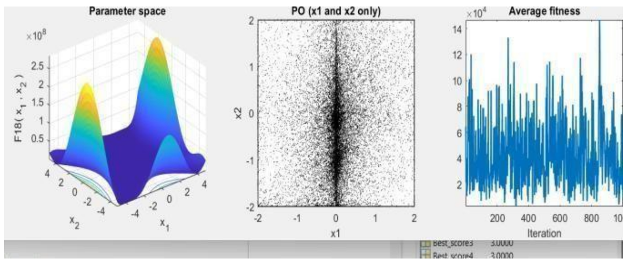
The parameter space and the conversion curve for the function 16

- Function No. 17:



The parameter space and the conversion curve for the function 17.

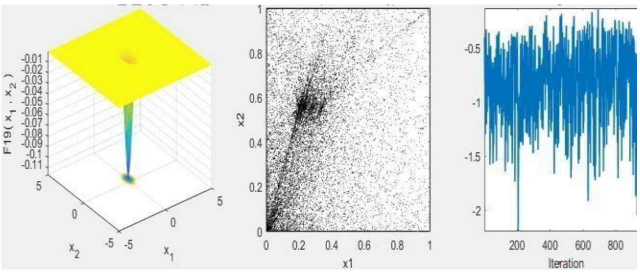
- Function No. 18:



The parameter space and the conversion curve for the function 18.

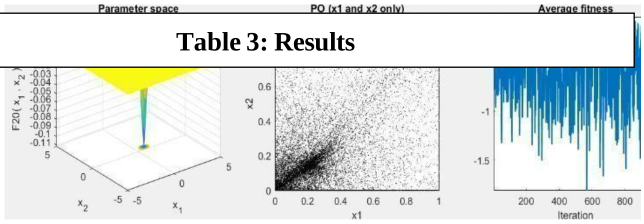
DOI: https://doi.org/10.5281/zenodo.15589753

- Function No. 19:



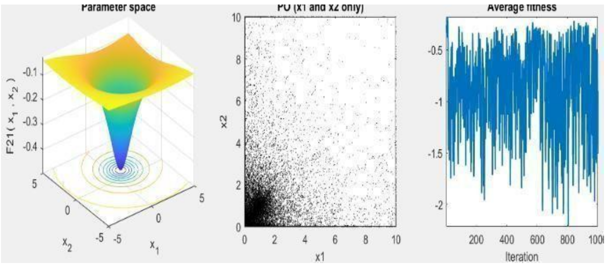
The parameter space and the conversion curve for the function 19.

- Function No. 20:



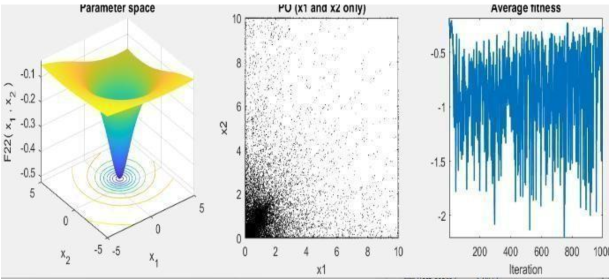
The parameter space and the conversion curve for the function 20

- Function No. 21:



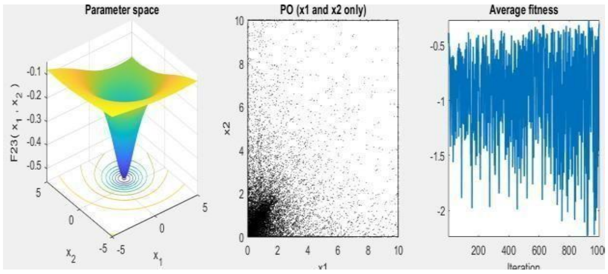
The parameter space and the conversion curve for the function 21

- Function No. 22:



The parameter space and the conversion curve for the function 22.

- Function No. 23:



The parameter space and the conversion curve for the function 23.

The Original value denotes the true solution, and the Hybrid value is the outcome generated by the Particle Swarm Optimizer that executing the algorithm to determine an optimal solution.

Table 3: Results

Function	Actual Value	Hybrid Value
F1	1.65e-306	8888.561
F2	7.79e-163	117.412
F3	2.15e-296	32323.9
F4	1.36e-135	49.58036
F5	25.3364	3422657
F6	0	17515.74
F7	2.74e-06	12.06899
F8	-12146.1	-7190.55
F9	0	249.6254
F10	4.44e-16	17.63578
F11	0	90.17439
F12	5.52e-08	2503892
F13	1.1958	20258913
F14	0.998	3.96825
F15	0.000307	0.019539
F16	-1.0316	-1.03163
F17	0.39789	0.397887
F18	3	3
F19	-3.8628	-3.86278

DOI: https://doi.org/10.5281/zenodo.15589753

F20	-3.322	-3.322
F21	**-10.1532**	**-10.1532**
F22	**-10.4029**	**-3.7243**
F23	**-10.5364**	**-2.42173**

4. Conclusion

The conclusion on the hybridized parrot optimizes with particle flock adaptation (HPO-PSO) emphasizes the effectiveness of the hybrid algorithm proposed to increase adaptation performance. The values of hybridized from f1 to f23 are extracted from these hybridized algorithms. By integrating the search functions of parrot optimizes (PO) with the strength of the utilization of particle flock adaptation (PSO), the hybrid approach achieves better convergence speed, accuracy and strength into different benchmark functions and real adaptation problems. Experimental results show that HPO-PSO exceeds traditional pos, PSOs and other metaheuristic algorithms beyond strength in terms of quality and calculation efficiency of solution.

5. References

[1] "Parrot optimizer: Algorithm and applications to medical problems" Junbo Lian [1,2,3], Guohua Hui [1,2,3], Ling Ma [1,2,3], Ting Zhu [1,2,3], Xincan Wu [1,2,3], Ali Asghar Heidari [4], Yi Chen [5], Huiling Chen [5*]

[2] An efficient multi-objective parrot optimizer for global and engineering optimization problems Mohammed R. Saad1, Marwa M. Emam2 & Essam H. Houssein2,3.

[3] 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.

[4] I. E. Grossmann, Global Optimization in Engineering Design (Nonconvex Optimization and Its Applications), vol. 9. 1996.

[5] R. V. Rao and G. G. Waghmare, "A new optimization algorithm for solving complex constrained design optimization problems," vol. 0273, no. April 2016, doi: 10.1080/0305215X.2016.1164855.

[6] E.-S. M. El-Kenawy, M. M. Eid, M. Saber, and A. Ibrahim, "MbGWO-SFS: Modified Binary Grey Wolf Optimizer Based on Stochastic Fractal Search for Feature Selection," IEEE Access, 2020, doi: 10.1109/access.2020.3001151.

[7] M. Nouiri, A. Bekrar, A. Jemai, S. Niar, and A. C. Ammari, "An effective and distributed particle swarm optimization algorithm for flexible job- shop scheduling problem," J. Intell. Manuf., 2018, doi: 10.1007/s10845-015-1039-3.

[8] Y. Li, J. Wang, D. Zhao, G. Li, and C. Chen, "A two-stage approach for combined heat and power economic emission dispatch: Combining multi-objective optimization with integrated decision making," Energy, 2018, doi: 10.1016/j.energy.2018.07.200.

[9] D. Yousri, T. S. Babu, and A. Fathy, "Recent methodology-based Harris hawks optimizer for designing load frequency control incorporated in multi-interconnected renewable energy plants," Sustain. Energy, Grids Networks, 2020, doi: 10.1016/j.segan.2020.100352.

[10] R. Al-Hajj and A. Assi, "Estimating solar irradiance using genetic programming technique and meteorological records," AIMS Energy, 2017, doi: 10.3934/energy.2017.5.798.

[11] R. Al-Hajj, A. Assi, and F. Batch, "An evolutionary computing approach for estimating global solar radiation," in 2016 IEEE International Conference on Renewable Energy Research and Applications, ICRERA 2016, 2017. doi: 10.1109/ICRERA.2016.7884553.

[12] R. A. Meyers, "Classical and Nonclassical Optimization Methods Classical and Nonclassical Optimization Methods 1 Introduction 1 1.1 Local

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

and Global Optimality 2 1.2 Problem Types 2 1.3 Example Problem: Fitting Laser-induced Fluorescence Spectra 3 1.4 Criteria for Optimization 4 1.5 Multicriteria Optimization 4,”.

DOI: https://doi.org/10.5281/zenodo.15589753