Hybridization Technique for Grasshopper Optimization Algorithm (GOA) With Particle Swarm Optimization (PSO)

Sandhya Dahake[1;] Neeraj Kumar Jha[2;]
Gayatri D. Dudhane[3;] Gayatri V. Lambhade[4]

HOD, Dept. Master in Computer Applications, G H Raisoni College of Engineering and Management, Nagpur, Maharashtra, India. [1]
Prof, Dept. Of Master in Computer Applications, G H Raisoni College of Engineering and Management, Nagpur, Maharashtra, India. [2]
Dept. Of Master in Computer Applications, G H Raisoni College of Engineering and Management, Nagpur, Maharashtra, India. [3]
Dept. Of Master in Computer Applications, G H Raisoni College of Engineering and Management, Nagpur, Maharashtra, India. [4]

Abstract: Algorithms for optimization are crucial for solving complex numerical problems in many scientific and technical domains. In this paper local and global search, respectively, metaheuristic techniques Particle Swarm Optimization (PSO) and the Grasshopper Optimization Algorithm (GOA) have demonstrated significant potential. Every technique has drawbacks, though; PSO might not be able to explore high-dimensional search spaces, whereas GOA frequently suffers from premature convergence. In order to improve optimization performance, this study suggests a hybridized strategy that combines GOA's exploratory nature with PSO's exploitative capabilities.

The two techniques are dynamically balanced by the hybrid model, which improves convergence and resilience when dealing with high-dimensional and multimodal optimization issues. Benchmark functions are used for performance evaluation, and the results show notable gains in convergence time, solution accuracy, and stability when compared to solo GOA, PSO, and other traditional optimization methods. For numerical optimization problems in the real world, such as financial modelling, machine learning parameter tweaking, and engineering design, the suggested hybrid technique presents a viable answer.

Keywords: GOA, PSO, Benchmarks Functions, Global Optimization, Nature Based Algorithm, Hybridization.

1.Introduction

Optimization is essential to the resolution of challenging real-world issues in a variety of domains, including bioinformatics, machine learning, engineering, and finance.

Grasshopper Optimization Algorithm (GOA) was inspired by a foraging and swarming behaviour of a grasshopper. In order to identify near-optimal solutions in complex search spaces, a variety of metaheuristic algorithms have been developed in response to the demand for effective optimization strategies. As Grasshopper Optimization Algorithm (GOA) and Particle Swarm Optimization (PSO) are nature inspired algorithm these algorithms imitate natural processes. Particle Swarm Optimization (PSO) and the Grasshopper Optimization Algorithm (GOA) are two of them that have drawn a lot of interest because of how well they solve numerical optimization issues. These algorithms work better in long range and sudden movement.

By striking a balance between local and global searches, GOA, which was inspired by the swarming behaviour of grasshoppers, thrives in exploration. It frequently has early stalling in local optima and sluggish convergence, though. Conversely, PSO, which is modelled after the social behaviour of fish schools or bird flocks, is very effective at exploitation and convergence, but it might not have a diverse range of search agents, which could result in less-than-ideal solutions in intricate, multimodal environments.

DOI: https://doi.org/10.5281/zenodo.18266305

This study offers a hybrid strategy that combines GOA and PSO to take advantage of their complementing advantages in order to overcome these issues. While
using PSO's effective local search mechanism to refine solutions in subsequent iterations, the hybridized GOA-PSO model improves exploration in the early phases of optimization. Improved convergence speed, increased
accuracy of the solution, and resilience to
local optima are all guaranteed by this
adaptive combination.
The performance of the suggested hybrid algorithm is compared to standalone GOA, PSO, and other cutting-edge metaheuristic algorithms using common benchmark functions. The outcomes show that the hybrid model performs better than conventional methods in terms of accuracy, stability, and convergence rate. By providing a more effective method for resolving challenging numerical optimization issues, this study advances the expanding subject of hybrid metaheuristic optimization.
Search Process of Grasshopper Optimization Algorithm (GOA):
Search process of Grasshopper Optimization Algorithm (GOA) is categorized into two types that is Exploration and Exploitation.
Exploration is used to find the fitness value and keep track on sudden movements and jump. Exploitation is used to find the best solution and keep track on local movement of grasshopper obtained during food search.

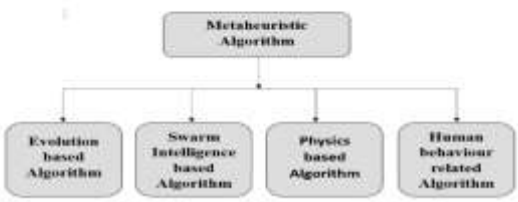


Fig 1. Classification of Metaheuristic Algorithm.

Table1: Literature Review

2.**Literature Review**

In order to address difficult optimization issues, two well-known metaheuristic approaches that derive principles from occurrences in nature are the Grasshopper Optimization Algorithm (GOA) and Particle Swarm Optimization (PSO). PSO was encouraged by the social dynamics of fish schooling and bird flocking, although GOA matches the swarming behaviour of grasshoppers. The goal of integrating these algorithms is to build on their unique advantages, which could result in optimization solutions that are more reliable as well as efficient [7],[14].

AUTHOR TABLE:

Reference No.	**Algorithm Name**	**Author Name**	**Year**
1	**FruitFly Optimization**	**W. Y. Lin**	**2016**
2		**Y. Cheng et al**	**2018**
3	**Hybrid Ant Colony**	**X. Wang et al**	**2018**
4	**Global Optimization**	**I.E. Grossmann**	**1996**
5		**R. V. Rao et al**	**2016**
6	**Grey-Wolf Optimization**	**M.El-Kenawy**	**2020**
7	**Particle-Swarm Optimization**	**M. Nouri et al**	**2018**
8	**Multi-objective Optimization**	**Y. Li et al**	**2018**
9	**HarrisHawks Optimizer**	**D.Yousri et al**	**2020**
10	**Genetic Programming**	**R. Al-Hajj et al**	**2017**
11	**Evolutionary Computing**	**R. Al-Hajj et al**	**2016**
12	**Classical & non-classical**	**R.A. Meyers**	**2000**
13	**Quadratic Programming**	**N. Steffan at al**	**2012**
14	**Grasshopper Optimization**	**M. Mafarja et al**	**2018**
15	**Water Cycle**	**A. A. Heidari et al**	**2017**

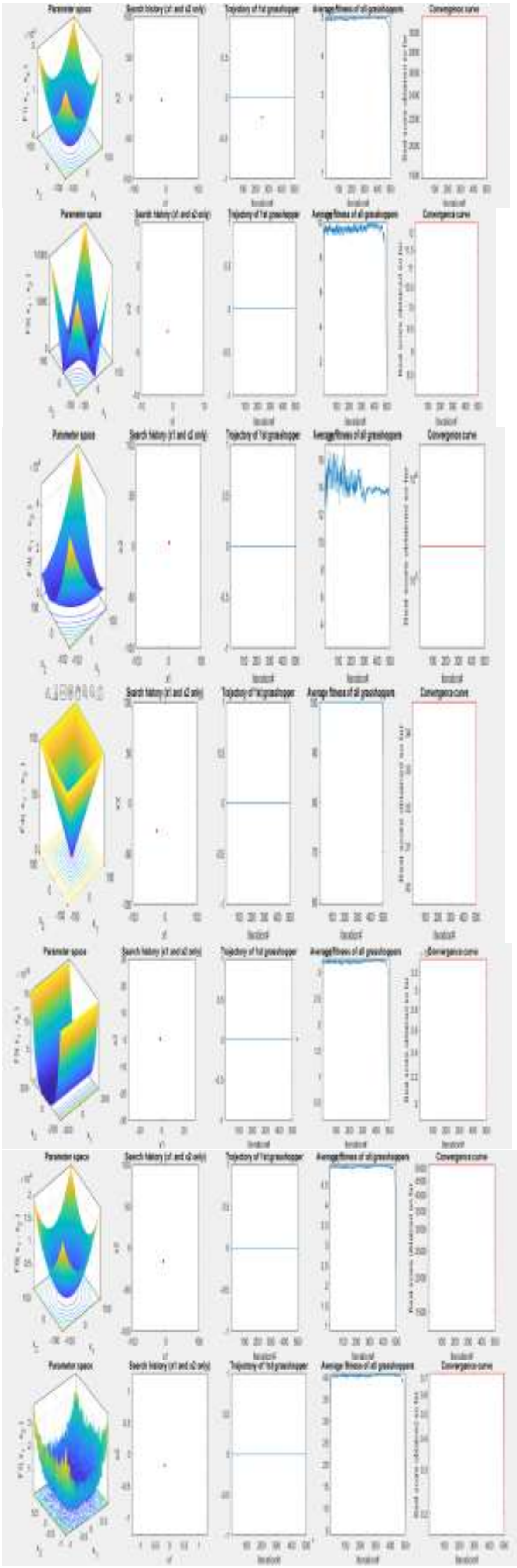
FUNCTION TABLE:

DOI: https://doi.org/10.5281/zenodo.18266305

ISSN NO-2584-2706

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
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
$F_{10}(S)=-20\exp(-0.2\sqrt{(\frac{1}{z}\sum_{m=1}^{z}S_m^2)})-\exp(\frac{1}{z}\sum_{m=1}^{z}\cos(2\pi S_m))+20+e$	(10,30,50,100)	[-32,32]	0
$F_{11}(S)=1+\sum_{m=1}^{z}\frac{S_m^2}{4000}-\prod_{m=1}^{z}\cos\frac{S_m}{\sqrt{m}}$	(10,30,50,100)	[-600, 600]	0
$F_{12}(S)=\frac{\pi}{z}\left[10\sin(\pi r_1)+\sum_{m=1}^{z-1}(r_m-1)^2[1+10\sin^2(\pi r_{m+1})]+(r_z-1)^2\right]+\sum_{m=1}^{z}u(S_m,10,100,4)$; $r_m=1+\frac{S_m+1}{4}$; $u(S_m,b,x,i)=\begin{cases}x(S_m-b)^i & S_m>b\\ 0 & -b<S_m<b\\ x(-S_m-b)^i & S_m<-b\end{cases}$	(10,30,50,100)	[-50,50]	0
$F_{13}(S)=0.1[\sin^2(3\pi S_1)+\sum_{m=1}^{z}(S_m-1)^2[1+\sin^2(3\pi S_m+1)]+(s_z-1)^2[1+\sin^2 2\pi S_z)]$	(10,30,50,100)	[-50,50]	0
Functions	**Dimensions**	**Range**	f_{min}
$F_{14}(S)=[\frac{1}{500}+\sum_{n=1}^{25}\frac{1}{n+\sum_{m=1}^{2}(S_m-b_{mn})^6}]^{-1}$	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
$F_{18}(S)=\left[1+(S_1+S_2+1)^2(19-14S_1+3S_1^2-14S_2+6S_1S_2+3S_2^2)\right]\times\left[30+(2S_1-3S_2)^2(18-32S_1+12S_1^2+48S_2-36S_1S_2+27S_2^2)\right]$	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
$F_{23}(S)=-\sum_{m=1}^{10}[(S-b_m)(S-b_m)^T+d_m]^{-1}$	4	[0, 10]	-10.5363

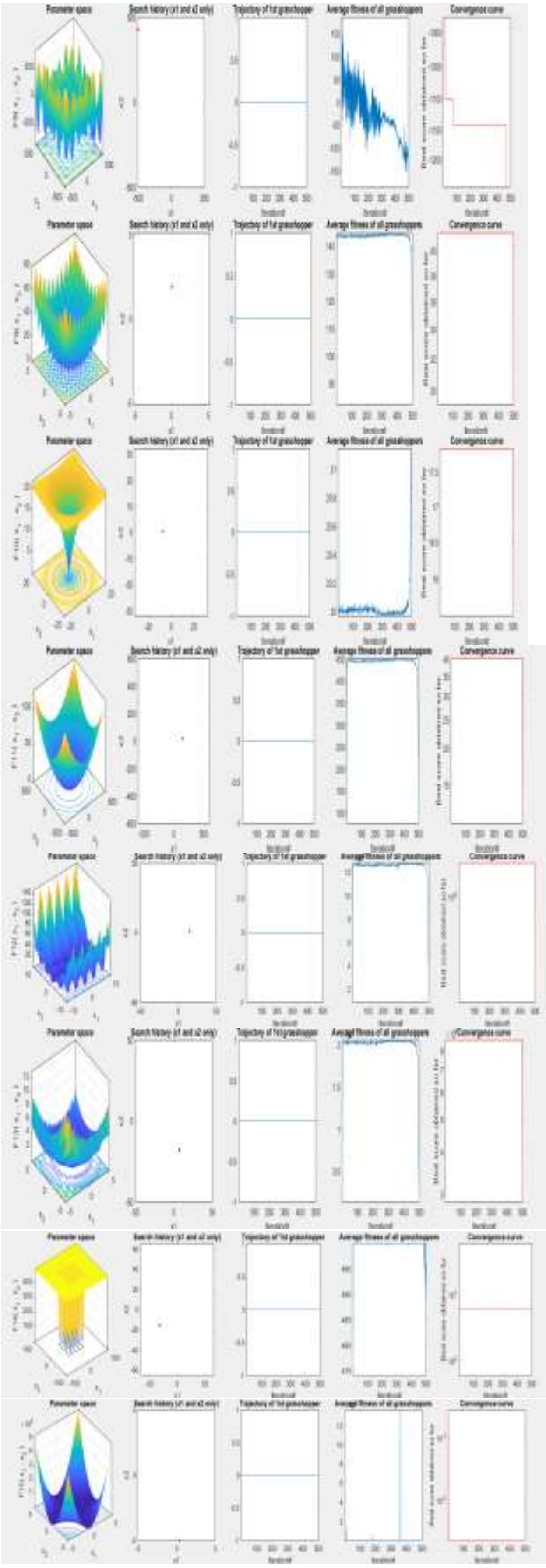
Table 2. Standard UM Benchmark Function



3.Result and Discussion

In this approach firstly we tested original algorithm on 23 benchmark functions and then hybridized GOA algorithm with PSO algorithm and then compared the results of both algorithm from that we find search space and convergence curve which are given below.

DOI: https://doi.org/10.5281/zenodo.18266305



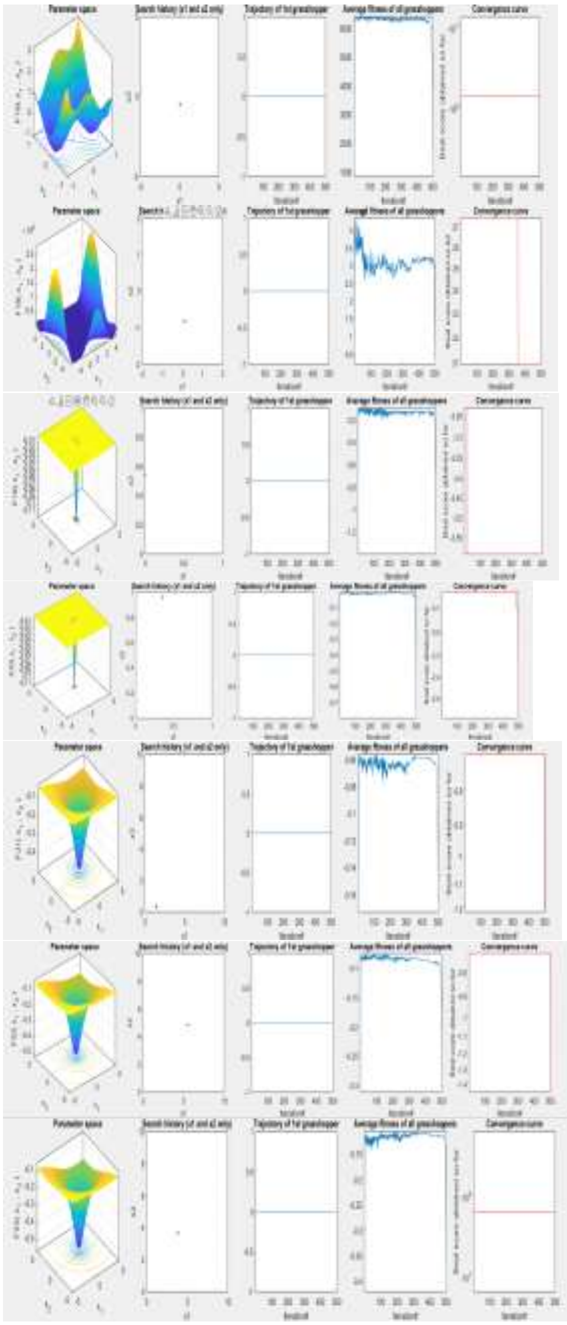


Fig 1: Search Space for Benchmark Functions applied on Hybrid GOA & PSO Algorithm

Result Table : Original GOA Vs Hybrid_GOA_PSO

Function Number	Original Value	Hybrid Value
F1	2.60E+08	0.062468
F2	0.1217	0.091837
F3	1.67E+07	0.91837
F4	0.00014152	0.12599
F5	16.5631	3.0879
F6	2.32E+08	0.030516

DOI: https://doi.org/10.5281/zenodo.18266305

F7	0.0017489	-1976.465
F8	-1476.4761	-1877.766
F9	4.0263	2.9074
F10	1.6462	0.26768
F11	0.098556	0.070532
F12	8.98E+06	0.0036155
F13	3.97E+08	0.0024125
F14	0.998	0.016936
F15	0.0006642	0.0013
F16	-1.0316	0.0008792
F17	0.39789	1.3016
F18	3	3.0032
F19	-3.8628	-3.8547
F20	-3.0867	-3.1269
F21	-5.0552	-10.0065
F22	-10.4029	-10.2339
F23	-10.5364	-10.4744

Table 3. Results for Original GOA vs Hybrid GOA with PSO

4.Conclision

Hybridization of Grasshopper Optimization Algorithm (GOA) with Practical Swarm Optimization (PSO) Algorithm was tested on 23 Benchmark functions (F1-F23) out of which it performs better and provides optimal values in 15 functions which was F1, F2, F3, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F20, F21.

6.References

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

[2] Y. Cheng, S. Zhao, B. Cheng, S. Hou, Y. Shi, and J. Chen, "Modelling and optimization for collaborative business process towards IoT applications," Mob. Inf. Syst., 2018, doi: 10.1155/2018/9174568.

[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 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," Encycl. Anal. Chem., pp. 9678–9689,z 2000, [Online].
Available:
https://pdfs.semanticscholar.org/5c5c/908bb00a54439dcee50ec1ada6b735694a94.pdf

[13] N. Steffan and G. T. Heydt, "Quadratic programming and related techniques for the calculation of locational marginal prices in distribution systems," in 2012 North American

DOI: https://doi.org/10.5281/zenodo.18266305

Power Symposium (NAPS), 2012, pp. 1–6. doi: 10.1109/NAPS.2012.6336310.
[14] M. Mafarja et al., “Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems,” Knowledge-Based Syst., vol. 145, pp. 25–45, 2018, doi: 10.1016/j.knosys.2017.12.037.
[15] A. A. Heidari, R. Ali Abbaspour, and A. Rezaee Jordehi, “An efficient chaotic water cycle algorithm for optimization tasks,” Neural Compute. Appl., vol. 28, no. 1, pp. 57–85, 2017, doi: 10.1007/s00521-015-2037-2.

DOI: https://doi.org/10.5281/zenodo.18266305