Journal of Robotics and Control (JRC)
Volume 6, Issue 4, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i4.26410

A Comparative Study of Metaheuristic Optimization Algorithms in Solving Engineering Designing Problems

Widi Aribowo [1], Hisham A. Shehadeh [2*]
[1] Department of Electrical Engineering, Faculty of Vocational, Universitas Negeri Surabaya, Surabaya, Indonesia
[2] Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Al-Huson, Jordan
Email: [1] widiaribowo@unesa.ac.id, [2] h.shehadeh@bau.edu.jo
*Corresponding Author

***Abstract*—This paper presents a comprehensive comparative study of several metaheuristic optimization algorithms with the aim of identifying the most effective method for solving well-established engineering design problems. The algorithms selected for this study include Sperm Swarm Optimization (SSO), Chernobyl Disaster Optimizer (CDO), Bermuda Triangle Optimizer (BTO), Marine Predators Algorithm (MPA), and Particle Swarm Optimization (PSO). These algorithms are tested and evaluated through both qualitative and quantitative analyses.The first phase of testing involves applying the algorithms to a set of benchmark functions from the Congress on Evolutionary Computation (CEC) 2017 suite. Key performance indicators such as best fitness value, standard deviation, and mean are used to measure solution quality, while convergence curves are analyzed to assess optimization efficiency over iterations. This allows for a robust evaluation of each algorithm's ability to balance exploration and exploitation in the search space. In the second phase, the algorithms are implemented to solve real-world engineering design problems, including Speed Reducer Design, Pressure Vessel Design, Cantilever Beam Design, and Robot Gripper Optimization. These case studies further validate the practical applicability and versatility of the algorithms in handling complex, multidimensional, and constrained optimization tasks. The results indicate varying levels of performance across different problems, highlighting the strengths and limitations of each method. This comparative insight provides valuable guidance for researchers and practitioners in selecting suitable optimization techniques for specific engineering challenges.**

Keywords*—*Swarm Based Optimization Algorithms; Physical Based Optimization Algorithm; Speed Reducer Design; Pressure Vessel Design.

I. INTRODUCTION

The process of creating a large number of potential candidate solutions in order to identify the optimal one that will provide the least amount of value or the most amount of value for the given issue is known as optimization [1]-[6]. One popular method for solving derivative optimization issues is gradient descent [7]-[11]. Derivatives can produce exact optimum solutions, but they can also render NP-hard problems intractable quickly due to their exponential processing cost [12]-[16]. Because of this, scientists are now investigating other techniques that yield almost flawless results in a manageable polynomial period of time [17]-[19].

Metaheuristics have been a major focus of algorithmic and artificial intelligence research in the last few decades [20]-[25]. They provide a strong substitute for traditional gradient-based mathematical techniques in the resolution of challenging optimization issues [26]-[31]. These methods have the benefit of being able to generate almost flawless responses in a reasonable amount of time. Metaheuristics are better than earlier approaches because of their ease of use, scalability, and adaptability [32]-[36]. Their diversity, wide range of applications, and adaptability have also spurred study into the creation and improvement of several optimization problem-solving strategies [37][38][39]. Optimization has developed to tackle a wide range of difficult, high-dimensional problems [40]-[44].

This is especially relevant when problems like non-linearity, discontinuity, or non-convexity in the objective function make standard gradient-based techniques inadequate [45]-[49]. Four main categories of metaheuristic algorithms can be bifurcated to generate an optimal solution to different kinds of problems. These categories can be listed in Table I, which are physical based, swarm based, evolutionary based and manmade optimization algorithms.

Recent, optimization algorithms have a vast range of applications in various domains such as biology, economics, and engineering [61]-[66]. These algorithms are developed to solve complex problems efficiently and quickly [67]-[71]. The exploration and exploitation principles should be applied by the aforementioned categories of algorithms in an effort to reach at the global optimal solution. The capacity of an algorithm to identify every aspect of a problem's dimension is known as the exploration principle. Conversely, exploitation describes an algorithm's capacity to arrive at the best possible answer to a problem.

Hence, these algorithms strive for equilibrium among the previously listed principles [72][73][74]. Bermuda Triangle Optimizer (BTO) [50], Chernobyl Disaster Optimizer (CDO) [52][75], Sperm Swarm Optimization (SSO) [54], Marine Predators Algorithm (MPA) [76], and Particle Swarm Optimization (PSO) [77] are well-known optimization algorithms that have various advantages. These advantages can be listed as follows:

TABLE I. METAHEURISTIC ALGORITHMS AND THEIR ACRONYM AND CITATION

Category	Algorithm	Acronym	Citation
Physics-based	Bermuda Triangle Optimizer	BTO	Shehadeh [50]
	AtomSearch Optimization	ASO	Zhao et al. [51]
	Chernobyl Disaster Optimizer	CDO	Shehadeh [52]
Swarm-based	Sand Cat Swarm Optimization	SCSO	Seyyedabbasi and Kiani [53]
	Sperm Swarm Optimization	SSO	Shehadeh et al. [54]
	Greylag Goose Optimization	GGO	El-Kenawy et al.[55]
	Elk herd optimizer	EHO	Al-Betar et al.[56]
Evolutionary based	Synergistic fibroblast optimization	SFO	Dhivyaprabha et al. [57]
	Differential evolution	DE	Storn and Price [58]
Manmade optimization	Fireworks Algorithm	FA	Tan et al. [59]
	Harmony Search	HS	Geem et al. [60]

- PSO, MPA and SSO are simple to understand and utilize.
- Usually, PSO simply needs to be adjusted for the inertia weight, cognitive coefficient, social coefficient. This simplicity in tuning of parameters helps reduce error appearance in code.
- SSO uses random value in its procedure, such as Ph value, and temperature value, which are an exact values of female reproduction system vital signs. To prevent gaps between these parameters SSO applies the logarithm to normalize these tuning of parameters and help reduce error appearance in code.
- SSO and PSO converge faster in many problems.
- SSO, MPA and PSO can be used to solve a wide variety of optimization problems, such as both continuous and discrete issues, solving multiple objectives optimization problems, and solving problems with and without constraints.
- Existing traditional methods, struggle with high-dimensional constraints; SSO's adaptive penalties offer superior convergence.
- Novel Inspiration of CDO and BTO draws from a unique and impactful of real and historical events, which offer fresh principles in optimization techniques.
- CDO and BTO balance between exploration and exploitation strategies by avoiding local optima, the architectures of them enable efficient exploration of the solution space domain of any problem.
- CDO is powerful, and efficient approache to solving complex optimization problems.
- CDO uses random values of tuning parameters, which are the exact values of human walk speed, gamma particle, beta particle, alpha particle spreading speeds. In addition, it normalizes these values by applying the logarithm on them.
- BTO also apply random values in its procedure to skep local minima.

Based on the aforementioned advantages, we are motivated in this paper to do a comparative study of these algorithms by measuring the efficiency of them with existing CEC 2017 benchmark functions. In addition, to determine the better algorithm between them that could solve engineering designing problems [78]-[82]. This paper is organized as follows: Section 2 presents the review on optimization algorithms and their categories. Section 3 describes the experimental setup and presents the results of SSO, CDO, BTO, MPA, and PSO in solving CEC and engineering problems. We conclude the paper in Section 4.

II. LITERATURE REVIEW

In this paper, we choose Sperm Swarm Optimization (SSO), Chernobyl Disaster Optimizer (CDO), Bermuda Triangle Optimizer (BTO), Particle Swarm Optimization (PSO), and Marine Predators Algorithm (MPA) to review on them and to use them in purpose of optimizing some well-known engineering problems.

A. Sperm Swarm Optimization (SSO)

Shehadeh et al. [54] suggested "Sperm Swarm Optimization (SSO)," a novel optimization technique that simulates the swarm of sperm swimming to reach the egg during fertilization. This process is depicted in Fig. 1.

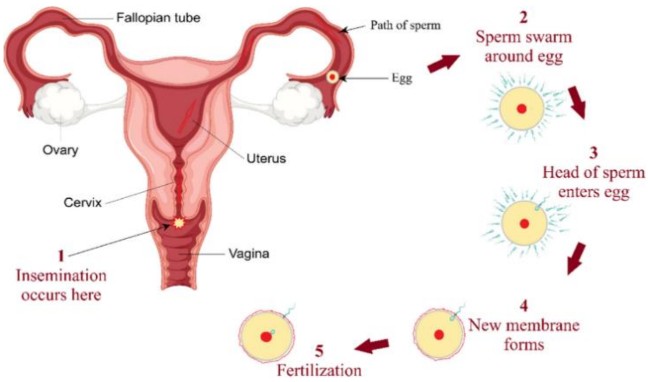


Fig. 1. Fertilization process

The initial velocity, current velocity, and global velocity are the three velocities of the swarm. The winner of the global one is the sperm that is closest to the egg. Fig. 1 shows this velocity [12]. The following mathematical models depict these velocities. The temperature and pH values, which are random values between 35.1 and 38.5 and 7 and 14, respectively, have an impact on the previously indicated velocities.

$$Initial_{Velocyity} = D.V_i.Log_{10}(pH_{Rand_1}) \tag{1}$$

$$\begin{aligned} &Current\_Best\_Solution \\ &= Log\_10(pH - Rand_2) \\ &\cdot Log_{10}(Temp\_Rand).(sb\_solution[] - current[]) \end{aligned} \tag{2}$$

$$\begin{aligned} &Global\_Best\_Solution \\ &= Log\_10(pH - Rand_3) \\ &\cdot Log_{10}(Temp\_Rand_2).(sgb\_solution[] - current[]) \end{aligned} \tag{3}$$

Where D is a factor of velocity damping. This factor takes a value randomly between zero and one. pH_{Rand_1}, pH_{Rand_2}, and pH_{Rand_3} are a random number of pH value, which takes value between seven to fourteen. $Temp\_Rand_1$ and $Temp\_Rand_2$ are temperature values, which take a random value between 35.1 and 38.5. $sb\_solution[]$ is the best solution that has achieved so far. $sgb\_solution[]$ is the global best solution that has obtained by the winner (Fig. 2).

These velocities can be merged in one equation as follows. Where, the $v[]$ is the velocity rule of SSO.

$$v[] = Initial_{Velocity} + Current\_Best\_solution + Global\_Best\_Solition \quad (4)$$

The current best solution can be presented in the following formula.

$$Current[] = Current[] + v[] \quad (5)$$

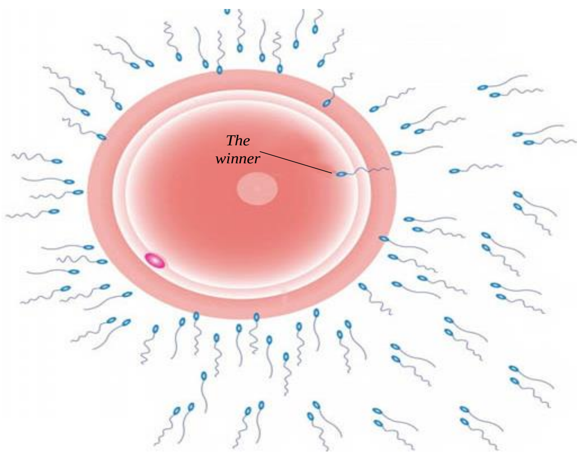


Fig. 2. Swarm of sperm and the winner [54]

B. Chernobyl Disaster Optimizer (CDO)

The 1986 Chernobyl nuclear catastrophe served as the inspiration for the "Chernobyl Disaster Optimizer (CDO)". In order to overcome optimization issues, this method [52], Shehadeh, mimics the impacts and transmission of radiation particles that are excluded from nuclei that target people. The optimizer considers three types of radiation particles: gamma (γ), beta (β), and alpha (α). Fig. 3 [52] shows these particles and the explosion zone.

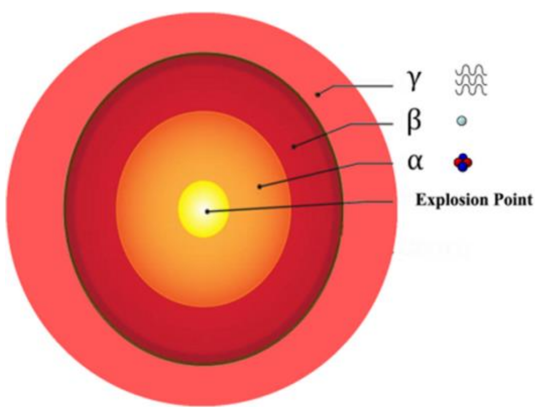


Fig. 3. The radiation particles and explosion point [52]

In CDO, Shehadeh assumes that the present locations of the gamma, beta, and alpha particles are $X_\gamma(t)$, $X_\beta(t)$, and $X_\alpha(t)$ respectively. The following models, in that order, provide the gamma, beta, and alpha particle propagation:

$$\rho_\gamma = \frac{x_h}{S_\gamma} - (WS_h \cdot rand\ ()) \quad (6)$$

$$\rho_\beta = \frac{x_h}{0.5 \cdot S_\beta} - (WS_h \cdot rand\ ()) \quad (7)$$

$$\rho_\alpha = \frac{x_h}{0.25 \cdot S_\alpha} - (WS_h \cdot rand\ ()) \quad (8)$$

$$x_h = r^2 \cdot \pi \quad (9)$$

$$S_\gamma = log\ (rand\ (1{:}\,300{,}000)) \quad (10)$$

$$S_\beta = log\ (rand\ (1{:}\,270{,}000)) \quad (11)$$

$$S_\alpha = log\ (rand\ (1{:}\,160{,}000)) \quad (12)$$

$$WS_h = 3 - 1 * ((3)/\ Maximum\_Iteration\ ) \quad (13)$$

Where x_h is the area of human walking within a circle with a random radius between 0 and 1. S_γ, S_β, and S_α are the normalized and random speeds of the gamma, beta and alpha particles. WS_h is walking speed of human, which is decreased linearly from 3km to 0, defined as Eq (13). The difference between the gamma, beta and alpha particles positions and total position can be calculated using the following formulas, respectively:

$$\Delta_\gamma = |A_\gamma \cdot X_\gamma(t) - X_T(t)| \quad (14)$$

$$\Delta_\beta = |A_\beta \cdot X_\beta(t) - X_T(t)| \quad (15)$$

$$\Delta_\alpha = |A_\alpha \cdot X_\alpha(t) - X_T(t)| \quad (16)$$

Where A_γ, A_β, and A_α are the propagation areas of the gamma, beta, and alpha particles, respectively, represented as the area of a circle with a random radius between 0 and 1as depicted in Fig. 3.

$$A_\gamma = A_\beta = A_\alpha = r^2 \cdot \pi \quad (17)$$

Where X_T is the average speeds of all particles.

$$X_T = \frac{v_\gamma + v_\beta + v_\alpha}{3} \quad (18)$$

Where v_γ, v_β, and v_α are the Gradient Descent Factors of gamma, beta and alpha particles respectively, which are calculated using the following formulas to find the optimal solution [52]:

$$v_\gamma = (X_\gamma(t) - \rho_\gamma \cdot \Delta_\gamma) \quad (19)$$

$$v_\beta = 0.5\,.(X_\beta(t) - \rho_\beta \cdot \Delta_\beta) \quad (20)$$

$$v_\alpha = 0.25\,.(X_\alpha(t) - \rho_\alpha \cdot \Delta_\alpha) \quad (21)$$

C. Bermuda Triangle Optimizer (BTO

This optimizer is a novel metaheuristic optimization algorithm that is proposed in 2025 by Shehadeh [50]. This algorithm simulates the mysterious and phenomena associated with Bermuda triangle in which many aircrafts and ships are pulled into the area of the triangle and disappeared there. In BTO, Shehadeh assumes that there are two areas of forces. The first one has massive attraction force, which is the exploitation area and can be formed as a Bermuda triangle. The second one has less attraction force, which is the area of circle that surrounded by the Bermuda triangle. This area can be used for exploration purpose. These areas are depicted in Fig. 4. In BTO, each attracted object takes a random position on search space domain and a value of probability of force, which are pulled to the center of triangle based on Newton's method of gravity, Eq (22).

$$G_{force} = \frac{CUG.M_1.M_2}{r^2} \tag{22}$$

Where the CUG is $6.67\times10^{-11}\text{Nm}^2\text{Kg}^{-2}$, which is constant of universal gravitation. M_1 represents the mass at the center of the Bermuda Triangle that generates the gravitational field, modeled as a random value. M_2 represents the mass that affect by M_1, modeled as a random value. r is the distance between M_1 and M_2, modeled as a random value.

Fig. 4. Massive attraction force and less attraction force [50]

The movement of these objects will be adjusted based on levy and chaos methods to simulate the exact movement of objects in ocean in which these objects will be affected by ocean flow and tide. Mainly, any an object exhibiting prescience whether inside or outside the Bermuda Triangle, which is randomly determined to be either greater than 0.5 or less than 0.5. When this value exceeds 0.5, the algorithm applies a subtraction operation, signifying a massive gravitational pull. This force is calculated based on the Bermuda Triangle's area as shown in Fig. 4. Objects located within the Bermuda Triangle are already subject to this massive attraction. In Equation (23), the Probability of Force (PoF) is defined as (1 – p-value), which represents the probability that the alternative hypothesis holds true. Conversely, when the prescience value is less than 0.5, the algorithm applies an addition operation, indicating a less attractive force. This force is derived by subtracting the area of the surrounding region (the yellow circular zone) from the Bermuda Triangle's area, as also illustrated in Fig. 4. Based on this computation, the object moves toward the optimal solution as described in Equation (23).

$$X_{i,j}(Iter_i+1) = \begin{Bmatrix} choas \times Triangle_{area} \times Acc \times best(x_j) - PoF \\ \times ((UB - LB \times Zone_{BF} + LB), Random > 0.5 \\ choas \times circle_{area} \times Acc \times best(x_j) - PoF \\ \times ((UB - LB \times Zone_{BF} + LB), Otherwise \end{Bmatrix} \tag{23}$$

Where LB and UB refer to the lower and upper bounds of the problem's search space. $Zone_{BF}$ denotes the Bermuda Force Zone, which can be coefficient computed using Equation (24). PoF stands for the probability ratio of the Bermuda force, which is derived from Equation (25). $best(x_j)$ represents the best value obtained so far. A_{cc} is the acceleration function, which is used to enhance the flow of the ocean current, and it can be calculated by exponential function as in Equation (26). $Triangle_{area}$ is the area of Bermuda triangle. $circle_{area}$ is the subtraction between triangle area and circle area, which refers to the surrounding area of Bermuda.

$$Zone_{BF} = area_{min} + Iter_1{}^{*}\left(\frac{area_{max} - area_{min}}{Iter_T}\right) \tag{24}$$

Where $area_{min}$ is the logarithm of the minimum area of the Bermuda force zone, which is 500,000 square miles. The logarithm is used for normalization. $Iter_1$ is the counter value at the i_{th} iteration. $area_{max}$ is the logarithm of the maximum area of the Bermuda force zone, which is 1,510,000 square miles, also normalized using the logarithm. $Iter_T$is the maximum number of iterations.

The Bermuda force probability ratio can be determined using the following equation. In statistical terms, the p-value is the null hypothesis, which is true, while (1 – p-value) indicates the probability that the alternative hypothesis, which is also true, as shown in Equation (24).

$$PoF = 1 - \left[\frac{iter_i \frac{1}{G_{Force}}}{iter_T \frac{1}{G_{Force}}}\right] \tag{25}$$

$$A_{cc} = r \times e^{\left(-20x\left(\frac{Iter_1}{Iter_T}\right)\right)} \tag{26}$$

Where $iter_i$ is the i_{th} iteration counter. $iter_T$ is the maximum number of iterations. r is random value. G_{Force} is Bermuda triangle force, which is Eq. (22).

D. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a swarm-based metaheuristic optimization algorithm, which is inspired by the social behavior of birds while searching food. It was introduced by Kennedy and Eberhart in 1995. The position of each particle will be adjusted on search space domain based on three steps as in Equation (27), which are initial velocity of particle, best velocity of particle and global best velocity of particle. The global one is the leader of bird swarm. This algorithm has a set of parameters in its velocity rule, which are inertia weight (w), cognitive factor(c_1), and social factor(c_2). The c_1 and c_2 are always have a value of 2 [77].

$$\begin{aligned} fV_{i,m}^{(t+1)} &= w * v_{i,m}^{(t)} + c_1 * \text{rand}_1() * \\ &\left(\text{pbest}_{\ i,m} - x_{i,m}^{(t)}\right) \\ &+ c_2 * \text{rand}_2() * \left(\text{gbest}_{\ m} - x_{i,m}^{(t)}\right) \end{aligned} \tag{27}$$

E. Marine Predators Algorithm (MPA)

MPA is a swarm based metaheuristic optimization algorithm introduced in 2020 by Faramarzi et al. [76] It is inspired by the foraging and hunting strategies of marine predators (e.g., sharks, tunas, sailfish) in the ocean. This algorithm mimics the strategy of marine predators while searching for prey in different ocean zones using Levy and Brownian motion to mimic real-world foraging patterns. It aims to balance between exploration (searching broadly) and exploitation (refining good solutions). The pray catching strategy can be calculated in the following equation.

$$p = \begin{cases} \overrightarrow{Prey_i} + CF[\vec{X}_{min} + \vec{R} \otimes (\vec{X}_{max} - \vec{X}_{min})] \otimes \vec{U} & \text{if } r \leq FADs \\ \overrightarrow{Prey_i} + [FADs(1-r) + r](\overrightarrow{Prey_{r1}} - \overrightarrow{Prey_{r2}}) & \text{if } r > FADs \end{cases} \tag{28}$$

Where $r1$ and $r2$ are random indexes of prey matrix. $\vec{U}$ is the binary vector with arrays including zero and one. $\vec{X}_{min}$ and $\vec{X}_{max}$ are the vectors containing the lower and upper bounds of the dimensions.

F. Optimizing CEC 2017 Benchmark Functions

The aforementioned algorithms are tested in solving the well-known Congress on Evolutionary Computation (CEC) 2017 [54] benchmark functions, which are 23 mathematical models that consist of unimodal, and multimodal benchmark functions. The results of these algorithms are listed in tables from 3 to 5. The qualitative tests are applied in this comparison, which are mean, standard deviation, and best fitness. Based on ranking rows in the following tables, we can notice that BTO algorithm is the best in solving the majority of functions, such as F1, F2, F3, F4, F6, F7, F9, F10, F11, and F12, which has a merit in solving the unimodal benchmark functions, such as functions from F1 to F4. For the ragged and noisy function F5, SSO has been solved it efficiently in which it comes in the first rank. CDO mainly comes in second rank for the majority of benchmark functions, and it solves the multimodal F13 efficiently. MPA solves the composition functions of multimodal efficiently, such as functions from F20 to F23. PSO algorithm mainly comes in the last rank, however it solves F17, F18, F19 in the same rank of MPA and SSO.

III. Experimental Setup and Results

The SSO, CDO, PSO, BTO, and MPA are coded in “MATLAB R2023a” and run on Intel core i5 CPU, 8 GB RAM utilizing Windows 11 to optimize the required engineering problems. Table II hows the parameters of these algorithms.

BTO outperforms the other algorithm overall, clearly dominating the ranking scores, which has the lowest total mean rank as depicted in Fig. 5. SSO and CDO show competitive performance, taking second and third place respectively. MPA and PSO trail significantly behind, with PSO consistently ranking last. For the qualitative test, the algorithm convergence is drawn for each algorithm in solving each benchmark function of CEC 2017. From the Fig. 6, we can notice that BTO has faster convergence, followed by SSO and CDO. The MPA followed them, while PSO comes in the last rank of convergence.

TABLE II. Parameters of SSO, PSO, CDO, BTO, and MPA

SSO	
Damping factor of velocity (D)	Rand (0, 1)
pH	Rand (7, 14)
Temperature	Rand (35.5, 38.5)
Size of population (swarm size)	30
Numbers of iterations	50
PSO	
C_1 and C_2	2
Size of population	30
Numbers of iterations	50
CDO	
S_γ – is the speed of gamma	Rand (1, 300,000) **km/s**
S_β – is the speed of beta	Rand (1, 270,000) **km/s**
S_α – is the speed of alpha	Rand (1, 16,000) **km/s**
the radius of radiations propagation (r)	Rand (0, 1)
Size of population	30
Numbers of iterations	50
BTO	
CUG	$6{,}67\text{x}10^{-11}\text{Nm}^2\text{Kg}^{-2}$
$area_{min}$	log(500000)
$area_{max}$	log(1,510,000)
Size of population	30
Numbers of generations	50
MPA	
r_1 and r_2	Rand(0, 1)
Size of population	30
Numbers of generations	50

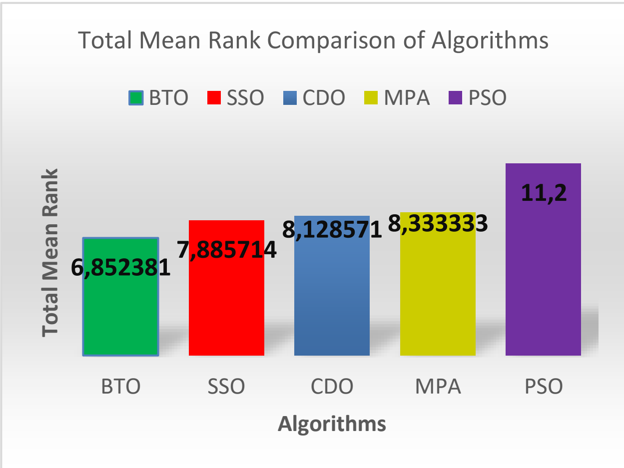


Fig. 5. Sum of total mean rank comparison of algorithms

TABLE III. RESULTS OF BENCHMARK FUNCTIONS FROM F1 TO F13

		CDO	SSO	MPA	PSO	BTO
F1	Best	7.95E-11	2.59E-11	15.6187	60636.99	0.00E+00
	Mean	2.08E-10	0.00014	48.1829	105054.4	1.49E+03
	Worst	4.01E-10	0.00302	158.953	145319.7	6.94E+04
	Std	7.76E-11	0.0007	26.475	15556.29	9.83E+03
	Rank	3	2	4	5	1
F2	Best	3.22E-06	8.67E-08	1.5991	62.2732	0
	Mean	4.81E-06	0.00023	2.9166	6371390	7.93E+10
	Worst	8.52E-06	0.00299	5.2416	1.6E+08	3.81E+12
	Std	1.16E-06	0.00048	0.74176	2509588	5.38E+11
	Rank	3	2	4	5	1
F3	Best	7.21E-08	2.65E-05	742.006	13988.5	0
	Mean	0.034466	81.9606	2104.723	32219.4	21625.28
	Worst	1.6636	2340.86	4299.15	61271	196848.3
	Std	0.23512	338.580	862.339	12018.7	51771.53
	Rank	2	3	4	5	1
F4	Best	1.01E-05	0.00071	2.9285	24.5219	0
	Mean	1.72E-05	0.26101	5.4527	42.3233	2.4463
	Worst	2.64E-05	1.5982	9.355	62.4067	87.2862
	Std	3.71E-06	0.35155	1.3571	7.2709	12.8204
	Rank	2	3	4	5	1
F5	Best	28.5556	28.3847	172.619	170721.	28.7973
	Mean	28.7063	29.1801	939.607	1221120	2165319
	Worst	28.7581	35.9431	2915.25	4797505	3.1E+08
	Std	0.04042	1.3184	577.993	979459.5	74544559
	Rank	2	1	4	5	3
F6	Best	7.5	4.5193	21.2953	1136.246	1.5238
	Mean	7.5	5.5276	51.3867	2631.223	1760.973
	Worst	7.5001	6.1575	95.9805	5229.108	61208.51
	Std	1.13E-05	0.29788	18.5072	972.2749	9362.192
	Rank	3	2	4	5	1
F7	Best	9.02E-05	0.00055	0.00416	0.2663	1.82E-05
	Mean	0.00163	0.00758	0.02039	0.73157	6.6554
	Worst	0.00598	0.02941	0.05734	1.3349	113.0406
	Std	0.00126	0.00697	0.01079	0.28601	24.8793
	Rank	2	3	4	5	1
Sum Rank		17	16	28	35	9
Mean Rank		2.428571	2.285714	4	5	1.285714

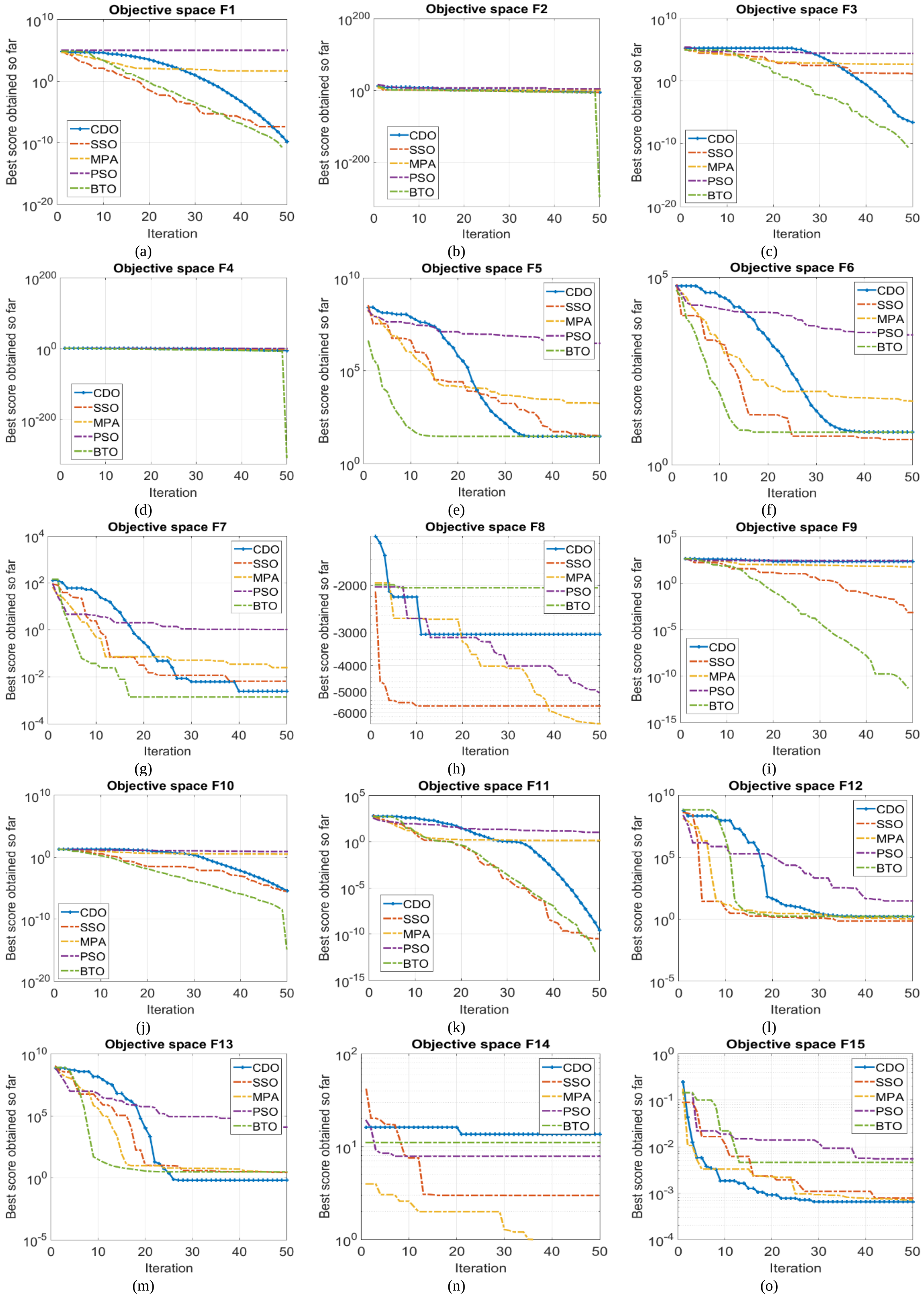
TABLE IV. RESULTS OF BENCHMARK FUNCTIONS FROM F8 TO F23

		CDO	SSO	MPA	PSO	BTO
F8	Best	-4317.92	-5796.57	-7443.59	-6535.82	-4995.54
	Mean	-3116.34	-5631.7	-6100.3	-4741.23	-2666.62
	Worst	-2304.7	-5422.99	-5060.86	-3132.88	-1412.76
	Std	455.7207	72.1446	497.4211	636.6539	699.4465
	Rank	5	3	1	2	4
F9	Best	9.67E-10	6.22E-11	31.6248	153.2982	0.00E+00
	Mean	113.2743	0.003615	78.8014	208.2787	30.7822
	Worst	293.8928	0.079819	137.0074	284.4248	448.1106
	Std	127.6219	0.015228	26.6201	28.4405	107.2275
	Rank	3	2	4	5	1
F10	Best	2.20E-06	1.03E-06	2.0413	6.099	8.88E-16
	Mean	3.52E-06	0.002348	3.072	8.7047	0.97995
	Worst	5.19E-06	0.050361	4.0291	10.5285	20.4984
	Std	6.30E-07	0.007683	0.38387	0.91181	4.033
	Rank	3	2	4	5	1
F11	Best	1.70E-10	2.67E-10	1.2044	6.842	0
	Mean	1.51E-07	0.011695	1.4728	11.8177	12.2338
	Worst	7.54E-06	0.28196	1.833	19.5564	611.6881
	Std	1.07E-06	0.045626	0.13422	2.8481	86.5058
	Rank	2	3	4	5	1
F12	Best	1.669	0.48686	0.54341	10.3876	0.06256
	Mean	1.669	0.72984	1.1347	170.2694	23703337
	Worst	1.669	1.1251	2.0243	4815.722	6.17E+08
	Std	1.69E-06	0.12489	0.39184	707.0014	1.17E+08
	Rank	4	2	3	5	1
F13	Best	0.402	2.4182	2.8697	66.9508	1.6086
	Mean	0.67432	2.8086	5.3869	21368.99	1311517
	Worst	0.9012	3.0976	11.6629	170106.7	65438239
	Std	0.11002	0.1163	1.7066	33284.08	9253988

		CDO	SSO	MPA	PSO	BTO
	Rank	1	3	4	5	2
Sum Rank		18	15	20	27	10
Mean Rank		3	2.5	3.333333	4.5	1.666667

TABLE V. RESULTS OF BENCHMARK FUNCTIONS FROM F14 TO F23

		CDO	SSO	MPA	PSO	BTO
F14	Best	1.0439	1.0007	0.998	0.998	0.99816
	Mean	15.4195	4.9938	1.2761	8.2083	9.3236
	Worst	18.3438	12.6705	2.9821	17.3744	12.6705
	Std	4.4625	3.6623	0.56871	4.6994	3.9032
	Rank	4	3	1	1	2
F15	Best	0.000317	0.000391	0.000311	0.000553	0.000635
	Mean	0.000648	0.00133	0.000684	0.004975	0.025672
	Worst	0.000954	0.02129	0.001223	0.065264	0.14283
	Std	0.000135	0.002917	0.000217	0.010805	0.043797
	Rank	2	3	1	4	5
F16	Best	-1.0311	-1.0316	-1.0316	-1.0316	-1.0306
	Mean	-1.0002	-1.0289	-1.0316	-1.0316	-0.91675
	Worst	-0.99464	-1.0034	-1.0316	-1.0314	1.7767
	Std	0.006122	0.004362	3.36E-11	5.42E-05	0.40009
	Rank	2	1	1	1	3
F17	Best	0.39808	0.39798	0.39789	0.39789	0.40212
	Mean	0.42413	0.42217	0.39789	0.39804	1.0307
	Worst	0.77347	0.56627	0.39789	0.403	4.0299
	Std	0.057782	0.030375	2.64E-09	0.000718	0.66057
	Rank	3	2	1	1	4
F18	Best	3.0599	3	3	3	3.0788
	Mean	74.0409	3.0027	3	6.2423	97.934
	Worst	88.0849	3.0185	3.0002	84.0102	978.6988
	Std	21.3486	0.004288	2.68E-05	16.0352	232.9909
	Rank	2	1	1	1	3
F19	Best	-3.8621	-3.8518	-3.8628	-3.8628	-3.8545
	Mean	-3.8231	-3.5254	-3.8628	-3.8164	-3.3583
	Worst	-3.6396	-2.8282	-3.8622	-3.0897	-2.262
	Std	0.035137	0.26227	0.000105	0.18544	0.37258
	Rank	2	4	1	1	3
F20	Best	-3.2888	-2.8784	-3.322	-3.3218	-2.7277
	Mean	-3.1258	-2.0723	-3.2918	-3.28	-1.6414
	Worst	-2.9004	-1.1699	-3.1909	-3.1962	-0.57082
	Std	0.089368	0.46509	0.050206	0.057656	0.5854
	Rank	3	4	1	2	5
F21	Best	-7.3727	-3.7289	-10.1532	-10.1529	-4.3123
	Mean	-3.3981	-1.6845	-9.2356	-6.3385	-1.3984
	Worst	-1.4638	-0.78537	-5.0551	-2.627	-0.32429
	Std	1.332	0.67619	1.9785	3.5991	0.88441
	Rank	3	5	1	2	4
F22	Best	-6.9876	-3.8699	-10.4029	-10.4023	-3.1585
	Mean	-3.6978	-1.7387	-9.0208	-7.1632	-1.2622
	Worst	-1.7076	-0.79078	-5.0798	-2.749	-0.40566
	Std	1.3351	0.78835	2.3554	3.5734	0.66172
	Rank	3	4	1	2	5
F23	Best	-8.4831	-6.0136	-10.5364	-10.53	-3.7174
	Mean	-3.7424	-2.1887	-9.4548	-6.1071	-1.4092
	Worst	-2.0395	-0.96891	-5.128	-2.4208	-0.50418
	Std	1.4673	0.92407	2.1852	3.8113	0.79923
	Rank	3	4	1	2	5
Sum Rank		27	31	10	17	39
Mean Rank		2.7	3.1	1	1.7	3.9


Objective space F1
Objective space F2
Objective space F3
(a)
(b)
(c)
Objective space F4
Objective space F5
Objective space F6
(d)
(e)
(f)
Objective space F7
Objective space F8
Objective space F9
(g)
(h)
(i)
Objective space F10
Objective space F11
Objective space F12
(j)
(k)
(l)
Objective space F13
Objective space F14
Objective space F15
(m)
(n)
(o)
Best score obtained so far
Iteration
CDO
SSO
MPA
PSO
BTO

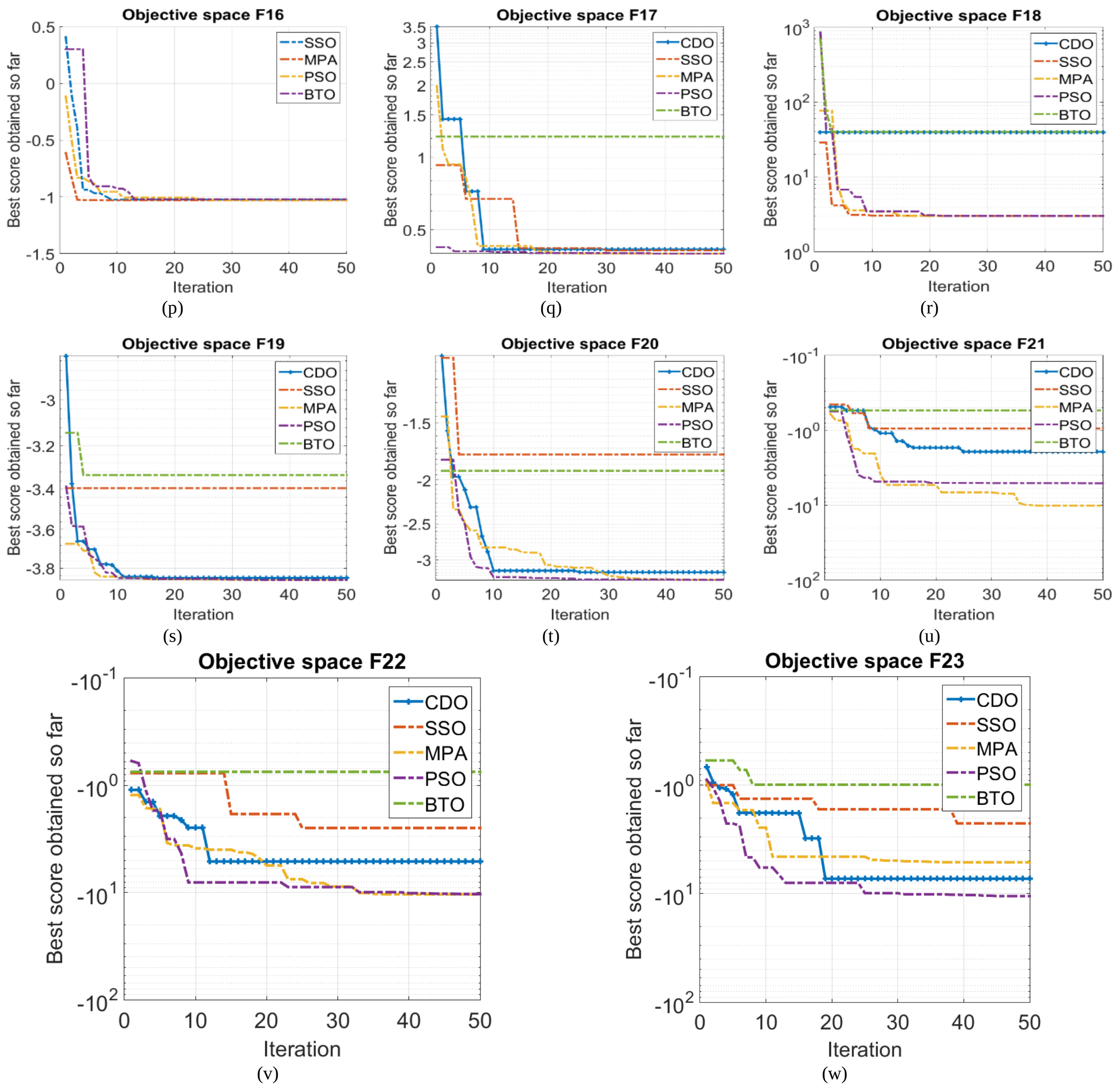


Fig. 6. Convergence rate of algorithms in solving CEC benchmark functions

A. Engineering Design Problems

In this Section, the BTO, MPA, PSO, SSO and CDO have been applied to optimize a set of engineering design problems, such as "Speed Reducer Design, Pressure Vessel Design, Cantilever Beam Design, and Robot Gripper Problem". We can summarize the outcomes as follows:

1) Speed Reducer Design Problem

The mathematical formulation of the Speed Reducer Design problem is shown below, which involves seven variables in its model. The problem's geometric structure is illustrated in Fig. 7. A comparison of BTO, MPA, PSO, SSO and CDO algorithms in optimizing this problem is provided in Table VI. According to the table, all seven variables influence the outcome. The results indicate that CDO achieved the best minimum weight of 3060.7908, which is ranked 1st**.** MPA is Second-best with 3121.451 weight**.** BTO in in third rank with 3143.9640. SSO in fourth rank with 3195.3189 weight. PSO has worst performance with 3242.5112 weight.

Consider:

$$y = [y_1, y_2, y_3, y_4, y_5, y_6, y_7] = [b, m, p, l_1, l_2, d_1, d_2]$$

Minimize:

$$f(y) = 0.7854 y_1 {y_2}^2 (3.3333 {y_3}^2 + 14.9334 y_3 - 43.0934) - 1.508 y_1 ({y_6}^2 + {y_7}^2) + 7.4777 ({y_6}^3 + {y_7}^2) + 0.7854 (y_4 {y_6}^2 + y_5 {y_7}^2).$$

Subject to:

$$g_1(\vec{y}) = \frac{27}{y_1 y_2^2 y_3} - 1 \leq 0, g_2(\vec{y}) = \frac{397.0}{y_1 y_2^2 y_3^2} - 1 \leq 0$$

$$g_3(\vec{y}) = \frac{1.90 y_4^4}{y_2 y_6^4 y_3} - 1 \leq 0, g_4(\vec{y}) = \frac{1.90 y_5}{y_2 y_7^4 y_3} - 1 \leq 0,$$

$$g_5(\vec{y}) = \frac{\sqrt{\left(\frac{745y_5}{y_2y_3}\right)^2 + 16.9\times10^6}}{110y_6^3} - 1 \leq 0$$

$$g_6(\vec{y}) = \frac{\sqrt{\left(\frac{745y_5}{y_2y_3}\right)^2 + 157.5\times10^6}}{85y_7^3} - 1 \leq 0$$

$$g_7(\vec{y}) = \frac{y_2y_3}{40} - 1$$

$$g_8(\vec{y}) = \frac{5y_2}{y_1} - 1 \leq 0$$

$$g_9(\vec{y}) = \frac{y_1}{12y_2} - 1 \leq 0$$

$$g_{10}(\vec{y}) = \frac{1.5y_6 + 1.9}{y_4} - 1 \leq 0$$

$$g_{11}(\vec{y}) = \frac{1.1y_7 + 1.9}{y_5} - 1 \leq 0$$

Where $2.6 \leq y_1 \leq 3.6$, $0.7 \leq y_2 \leq 0.8$, $17 \leq y_3 \leq 28$, $7.3 \leq y_4 \leq 8.3$, $2.9 \leq y_6 \leq 3.9$, $5.0 \leq y_7 \leq 5.5$

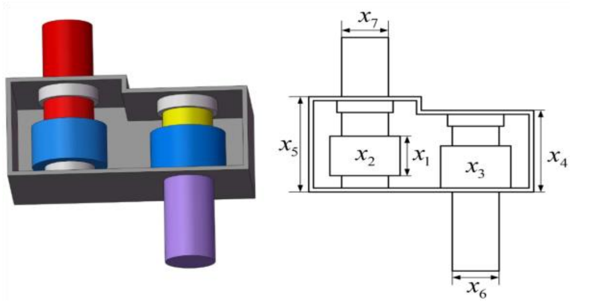


Fig. 7. Speed reducer design problem

2) *Pressure Vessel Design Problem*

The following list of four variables represents the mathematical modelling of the Pressure Vessel Design problem from a different perspective. Fig. 8 shows the geometry of this problem. Table VII displays the findings of the proposed algorithms in optimizing the Pressure Vessel Design issue. According to Table VII, factors 1 through 4 have an impact on the variables of the Pressure Vessel Design problem. The findings demonstrate that the MPA achieved the lowest cost of 4076.5769, which is ranked 1st. CDO comes in second-best with 6091.4687 cost. SSO is the third best with 7350.5363 cost. PSO and BTO have the highest costs, over 7800.

Consider: $y = [y_1, y_2, y_3, y_4] = [T_s, T_h, R, L]$

Minimize:

$$f\left(\vec{y}\right) = 0.6224y_1y_3y_4 + 1.7781y_2y_3^2 + 3.1661y_1^2y_4 + 19.84y_1^2y_3$$

Subject to:

$$g_1(\vec{y}) = -y_1 + 0.0193y_3 \leq 0, g_2(\vec{y}) = -y_3 + 0.00954y_3 \leq 0,$$

$$g_3(\vec{y}) = -\pi y_3^2y_4 - \frac{4}{3}\pi y_3^3 + 1296000 \leq 0, g_4(\vec{y}) = y_4 - 240 \leq 0.$$

Where $0 \leq y_1, y_2 \leq 99$, $10 \leq y_3, y_4 \leq 200$

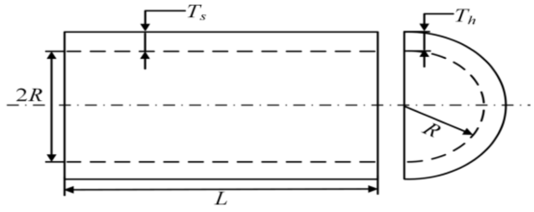


Fig. 8. Pressure vessel design problem

3) *Cantilever Beam Design Problem*

On the other hand, the following equation is mathematical modelling Cantilever Beam Design problem with a list of the five variables. Fig. 9 shows the geometry of this problem. Table VIII displays the outcomes of the Cantilever Beam Design issue, which shows the problem variables from 1 to 5. The CDO algorithm achieves the lowest cost, which is 1.35074, and ranked in 1st rank. SSO comes close behind, which its cost is 1.37208. MPA, BTO, and PSO show much higher values.

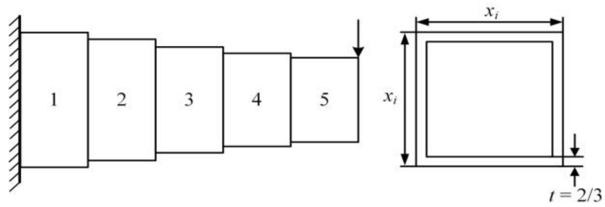


Fig. 9. Cantilever beam design problem

Consider: $y = [y_1, y_2, y_3, y_4, y_5]$

Minimize:

$$f\left(\vec{y}\right) = 0.6224(y_1 + y_2 + y_3 + y_4)$$

Subject to:

$$g\left(\vec{y}\right) = \frac{61}{y_1^3} + \frac{27}{y_0^3} + \frac{19}{y_3^2} + \frac{7}{y_1^3} + \frac{1}{y_5^3} - 1 \leq 0.$$

Where $0.01 \leq y1, y2, y3, y4, y5 \leq 100$.

4) *Robot Gripper Design Problem*

The Robot Gripper Problem is solved using the suggested algorithms, and Table IX displays the results of their performance. The mathematical formulation has seven-variable, which is the model of this problem. Fig. 10 provides a graphic depiction of the geometry of this problem. SSO has the lowest cost of 3.68878, which is ranked 1st. BTO is second-best in which is cost equal to 4.44538. MPA comes in third rank with cost equal to 4.48748. PSO comes in fourth rank at 5.05545. CDO has worst performance here with 5.57659.

Consider: $y = [y_1, y_2, y_3, y_4, y_5, y_6, y_7] = [a, b, c, e, f, l, \delta]$.

Minimize: $f\left(\vec{y}\right) = -min_zF_k(\vec{y}, z) + max_zF_k(\vec{y}, z)$

Subject to:

$$g_1(\vec{y}) = -Y_{min} + h(\vec{y}, Z_{max}) \leq 0, g_2(\vec{y}) = -h(\vec{y}, Z_{max}) \leq 0,$$

$$g_1(\vec{y}) = -Y_{min} + h(\vec{y}, Z_{max}) \leq 0, g_2(\vec{y}) = -h(\vec{y}, Z_{max}) \leq 0,$$
$$g_5(\vec{y}) = l^2 + e^2 - (a+b)^2 \leq 0, g_6(\vec{y}) = b^2 - (a-e)^2 - (l - Z_{max})^2 \leq 0$$
$$g_7(\vec{y}) = Z_{max} - l \leq 0$$

Where $0 \leq e \leq 50, 100 \leq c \leq 200, 10 \leq f, a, b \leq 150, 1 \leq \delta \leq 3.14, 100 \leq l \leq 300$

Other parameter conditions are represented below:

$$F_k = \frac{Pb\sin(\alpha + \beta)}{2c\cos(\alpha)}$$
$$\alpha = \cos^{-1}\left(\frac{a^2 + g^2 - b^2}{2ag}\right) + \phi, g = \sqrt{e^2 + (z-l)^2}$$
$$\beta = \cos^{-1}\left(\frac{b^2 + g^2 - a^2}{2ag}\right) - \phi, \phi = \tan^{-1}\left(\frac{e}{l-z}\right)$$
$$h(\vec{y}, z) = 2(f + e + c\sin(\beta + \delta))$$
$$Y_{min} = 50, Y_{max} = 100, Y_G = 150, Z_{max} = 100, P = 100.$$

IV. CONCLUSION AND FUTURE WORKS

This paper presents the detailed mathematical foundation of the SSO, CDO, MPA, PSO, and BTO. In addition, this paper provides comprehensive study about them in solving CEC 2017 benchmark test suites. These algorithms are also applied in optimizing well known engineering design problem, such as Speed Reducer Design, Pressure Vessel Design, Cantilever Beam Design, and Robot Gripper Problems. The results of CEC 2017 shows that the BTO is the best optimizer for solving unimodal problem. On the other hand, CDO, SSO, and MPA are the best in solving multimodal problems, that is why these algorithms are superior in solving the multi modal engineering design problem. In the future, we will enhance the BTO and PSO algorithms by hybridizing them with other evolutionary or swarm intelligence algorithms to increase their efficiency and performance in solving the multimodal problems.

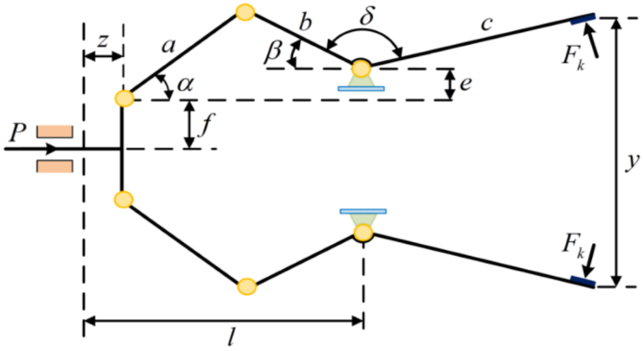


Fig. 10. Robot gripper design problem

TABLE VI. RESULTS OF OPTIMIZING SPEED REDUCER DESIGN PROBLEM BASED ON SSO, CDO, MPA, PSO, AND BTO

Algorithm	y_1	y_2	y_3	y_4	y_5	y_6	y_7	Minimum Weight	Rank
SSO	3.6000	0.7000	17.0000	7.3000	8.3000	3.3775	5.5000	3195.3189	4
CDO	3.5100	0.7000	17.0000	7.3000	8.3000	3.4689	5.3150	**3060.7908**	**1**
MPA	3.0947	0.8000	17.0000	7.3925	7.3000	3.4889	5.0041	3121.4516	2
PSO	3.1594	0.8000	17.0000	7.9999	7.5015	3.2844	5.2239	3242.5112	5
BTO	3.0387	0.8000	17.2919	7.7865	7.5939	3.1295	5.1350	3143.9640	3

TABLE VII. RESULTS OF OPTIMIZING PRESSURE VESSEL DESIGN PROBLEM BASED ON SSO, CDO, MPA, PSO, AND BTO

Algorithm	y_1	y_2	y_3	y_4	Minimum Cost	Rank
SSO	1.0307	0.5141	48.1118	123.3020	7350.5363	3
CDO	0.8195	0.4015	41.4249	185.5132	6091.4687	2
MPA	0.5625	1.6250	15.2920	10	**4076.5769**	**1**
PSO	0.1250	3.8125	25.1443	114.2507	7850.2362	4
BTO	3.9375	1.4375	10.0196	63.8676	8041. 9333	5

TABLE VIII. RESULTS OF CANTILEVER BEAM DESIGN PROBLEM OPTIMIZING BASED ON SSO, CDO, MPA, PSO, AND BTO

Algorithm	y_1	y_2	y_3	y_4	y_5	Minimum weight	Rank
SSO	5.55155	5.04000	5.43018	3.49378	2.47292	1.37208	2
CDO	6.11156	5.57061	4.35571	3.28815	2.32043	**1.35074**	**1**
MPA	0.01000	7.53741	13.2714	5.27661	1.2793	1.70821	3
PSO	6.1094	3.8265	16.6637	12.4117	4.4869	2.71429	5
BTO	7.7526	3.0460	9.3397	9.5052	0.5021	1.8811	4

TABLE IX. RESULTS OF OPTIMIZING PRESSURE ROBOT GRIPPER PROBLEM BASED ON SSO, CDO, MPA, PSO, AND BTO

Algorithm	y_1	y_2	y_3	y_4	y_5	y_6	y_7	Minimum cost	Rank
SSO	149.94360	139.62701	177.71509	8.73136	122.40109	141.14632	2.48982	**3.68878**	1
CDO	150.00000	122.33474	153.97865	17.22967	150.00000	182.04682	3.14000	5.57659	5
MPA	155.81850	121.37676	162.69311	30.71219	70.41388	157.17438	2.38838	4.48748	3
PSO	128.88748	100.57036	149.94457	27.46263	112.54845	116.99098	2.74985	5.05545	4
BTO	150.00000	95.02121	194.69334	50.00000	150.00000	154.46645	3.14000	4.44538	2

Conflicts of Interest: The authors declare no conflicts of interest.

Ethical Approval & Consent for publication:

We give our consent for the publication of identifiable details, which can include photograph(s) and/or videos and/or case history and/or details within the text ("Material") to be published in the above Journal and Article. We confirm that we have seen and been given the opportunity to read both the Material and the Article (as attached) to be published by your journal. In Addition, a sample of data of this paper will be available upon request. The open source code of our algorithms, namely, CDO and SSO are available via the following links:

CDO:
https://www.mathworks.com/matlabcentral/fileexchange/124351-chernobyl-disaster-optimizer-cdo

SSO:
https://www.mathworks.com/matlabcentral/fileexchange/92150-sperm-swarm-optimization-sso

BTO:
https://www.mathworks.com/matlabcentral/fileexchange/180808-bermuda-triangle-optimizer-bto

REFERENCES

[1] X. Cui and S. Ge, "Research on Efficiency Coupling Coordination Feature Model of Digital Economy Based on Multi-Objective Machine Learning Algorithm," *Teh. Vjesn.*, vol. 32, no. 1, pp. 78–87, 2025, doi: 10.17559/TV-20240826001943.

[2] A. N. K. Nasir, M. A. A. Roslan, M. F. M. Jusof, M. R. Ahmad, and A. A. A. Razak, "Mating-Based Manta Ray Foraging Optimization for Fuzzy-Hammerstein Model of an Electric Water Heater," *J. Adv. Res. Appl. Mech.*, vol. 129, no. 1, pp. 32–43, 2025, doi: 10.37934/ARAM.129.1.3243.

[3] N. Pokharna and I. P. Tripathi, "Optimality in interval fractional programming problems using d-invexity," *Iran. J. Fuzzy Syst.*, vol. 22, no. 1, pp. 71–91, 2025, doi: 10.22111/ijfs.2025.48874.8615.

[4] J. D. Rhenals-Julio, H. A. Martínez, J. F. Arango, J. M. M. Fandiño, and M. D. Oviedo, "Economic Assessment of the Potential for Renewable Based Microgrids Generation Systems: An Application in a University Building," *Int. J. Energy Econ. Policy*, vol. 15, no. 1, pp. 206–212, 2025, doi: 10.32479/ijeep.17423.

[5] Y. Xu, K. Liu, T. Zhai, X. Xiong, and Y. Wu, "Towards state-of-the-art semiconductor/dielectric interface in two-dimensional electronics," *J. Mater. Sci. Technol.*, vol. 239, pp. 93–108, 2025, doi: 10.1016/j.jmst.2025.03.049.

[6] W. Aribowo, S. Muslim, Munoto, B. Suprianto, U. T. Kartini, and I. G. P. Asto Buditjahjanto, "Tuning of Power System Stabilizer Using Cascade Forward Backpropagation," in *Proceeding - 2020 3rd International Conference on Vocational Education and Electrical Engineering: Strengthening the framework of Society 5.0 through Innovations in Education, Electrical, Engineering and Informatics Engineering, ICVEE*, pp. 1-5, 2020, doi: 10.1109/ICVEE50212.2020.9243204.

[7] V. R. Nippatla and S. Mandava, "Performance analysis of permanent magnet synchronous motor based on transfer function model using PID controller tuned by Ziegler-Nichols method," *Results Eng.*, vol. 26, 2025, doi: 10.1016/j.rineng.2025.105460.

[8] A. Alhawarat, S. Masmali, I. Masmali, M. Al-Baali, and S. Ismail, "A Modified Conjugate Gradient Method with Taylor Approximation: Applications in Electric Circuits and Image Restoration," *Eur. J. Pure Appl. Math.*, vol. 18, no. 1, 2025, doi: 10.29020/nybg.ejpam.v18i1.5639.

[9] S. Yang, P. Liu, and C. Pehlevan, "Convex Relaxation for Solving Large-Margin Classifiers in Hyperbolic Space," *Trans. Mach. Learn. Res.*, vol. 2025, 2025.

[10] L. Yin, J. Liu, H. Wu, H. Wang, and G. Lai, "EGO-DQN Planner: A Path Planner Integrated with Deep Q-Network," *International Conference on Image, Vision and Intelligent Systems*, pp. 175-189, 2024, doi: 10.1007/978-981-96-2528-4_15.

[11] M. Gan, X.-X. Su, G.-Y. Chen, J. Chen, and C. L. P. Chen, "Online Learning Under a Separable Stochastic Approximation Framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 2, pp. 1317–1330, 2025, doi: 10.1109/TPAMI.2024.3495783.

[12] S. Touati *et al.*, "Performance analysis of steel W18CR4V grinding using RSM, DNN-GA, KNN, LM, DT, SVM models, and optimization via desirability function and MOGWO," *Heliyon*, vol. 11, no. 4, 2025, doi: 10.1016/j.heliyon.2025.e42640.

[13] A. Wadood, H. Albalawi, A. M. Alatwi, H. Anwar, and T. Ali, "Design of a Novel Fractional Whale Optimization-Enhanced Support Vector Regression (FWOA-SVR) Model for Accurate Solar Energy Forecasting," *Fractal Fract.*, vol. 9, no. 1, 2025, doi: 10.3390/fractalfract9010035.

[14] H. Sharifzadeh, "Handling Multiple-Fuel Options in Economic Dispatch of Thermal Power Plants Through a Tight Model Applying Indicator Variables," *Int. Trans. Electr. Energy Syst.*, vol. 2025, no. 1, 2025, doi: 10.1155/etep/1572487.

[15] D. Nataraj and M. Subramanian, "Design and optimal tuning of fractional order PID controller for paper machine headbox using jellyfish search optimizer algorithm," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-85810-9.

[16] E. Cuevas, O. Barba, and H. Escobar, "A novel cheetah optimizer hybrid approach based on opposition-based learning (OBL) and diversity metrics," *Computing*, vol. 107, no. 2, 2025, doi: 10.1007/s00607-024-01397-5.

[17] O. Said Solaiman, R. Sihwail, H. Shehadeh, I. Hashim, and K. Alieyan, "Hybrid Newton–sperm swarm optimization algorithm for nonlinear systems," *Mathematics*, vol. 11, no. 6, p. 1473, 2023, doi: 10.3390/math11061473.

[18] A. Prapanca, Nasreddine Belhaouas, and Imed Mahmoud, "Modified FATA Morgana Algorithm Based on Levy Flight," *Vokasi Unesa Bull. Eng. Technol. Appl. Sci.*, vol. 2, no. 1, pp. 1–11, Mar. 2025, doi: 10.26740/vubeta.v2i1.37066.

[19] W. Aribowo, B. Suprianto, I. G. P. A. Buditjahjanto, M. Widyartono, and M. Rohman, "An improved neural network based on the parasitism – predation algorithm for an automatic voltage regulator," *ECTI Trans. Electr. Eng. Electron. Commun.*, vol. 19, no. 2, pp. 136–144, 2021, doi: 10.37936/ecti-eec.2021192.241628.

[20] A. Yaqoob and N. K. Verma, "Feature Selection in Breast Cancer Gene Expression Data Using KAO and AOA with SVM Classification," *J. Med. Syst.*, vol. 49, no. 1, 2025, doi: 10.1007/s10916-025-02171-6.

[21] S. Lv, J. Zhuang, Z. Li, H. Zhang, H. Jin, and S. Lü, "An enhanced walrus optimization algorithm for flexible job shop scheduling with parallel batch processing operation," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-89527-7.

[22] O. E. Turgut, H. Genceli, M. Asker, M. T. Çoban, and M. Akrami, "Predicting the chemical equilibrium point of reacting components in gaseous mixtures through a novel Hierarchical Manta-Ray Foraging Optimization Algorithm," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-93524-1.

[23] J. Shao, Y. Lu, Y. Sun, and L. Zhao, "An improved multi-objective particle swarm optimization algorithm for the design of foundation pit of rail transit upper cover project," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-87350-8.

[24] F. Kiani, "A multi-objective metaheuristic method for node placement in dynamic IoT environments," *Discov. Internet Things*, vol. 5, no. 1, 2025, doi: 10.1007/s43926-025-00153-1.

[25] B. Tian, "A Crow Search Algorithm integrated with Lévy flight and dynamic awareness probability for optimized numerical control machining parameters," *J. Eng. Appl. Sci.*, vol. 72, no. 1, 2025, doi: 10.1186/s44147-025-00612-0.

[26] N. Sabangban *et al.*, "Comparative Performance of Meta-Heuristic Algorithms for Low-Speed Wind Turbine Blade Structural Optimization," *J. Res. Appl. Mech. Eng.*, vol. 13, no. 1, 2025, doi: 10.14456/jrame.2025.11.

[27] G. A. Rolim and M. S. Nagano, "Designing state-of-the-art metaheuristics: What have we learned from the parallel-machine

scheduling problem with setups?," *Comput. Oper. Res.*, vol. 182, 2025, doi: 10.1016/j.cor.2025.107110.

[28] W. Wang, B. Zhang, P. Zhu, and Z. Liu, "Diversity-enhanced adaptive golden jackal optimization based on multi-strategy and its engineering applications," *Cluster Comput.*, vol. 28, no. 5, 2025, doi: 10.1007/s10586-024-04987-2.

[29] S. Khastar, F. Bashirizadeh, J. Jafari-Asl, and N. Safaeian Hamzehkolaei, "Predicting the cooling and heating loads of energy efficient buildings: a hybrid machine learning approach," *Cluster Comput.*, vol. 28, no. 5, 2025, doi: 10.1007/s10586-024-04993-4.

[30] A. Llanza, A. Nakib, and N. Shvai, "FDS: Fractal decomposition based direct search approach for continuous dynamic optimization," *Inf. Sci. (Ny).*, vol. 715, 2025, doi: 10.1016/j.ins.2025.122237.

[31] R. Zhang, C. Liu, J. Wang, K. Su, H. Ishibuchi, and Y. Jin, "Synergistic integration of metaheuristics and machine learning: latest advances and emerging trends," *Artif. Intell. Rev.*, vol. 58, no. 9, 2025, doi: 10.1007/s10462-025-11266-y.

[32] A. Maqbool, A. U. Rehman, A. Arshad, K. Mahmoud, and M. Lehtonen, "Hybrid metaheuristic optimization based DSM approach towards effective energy recommender system," *Electr. Power Syst. Res.*, vol. 246, 2025, doi: 10.1016/j.epsr.2025.111645.

[33] J. Wang, J. Dong, X. Dong, and H. Zhou, "Population-Based Meta-Heuristic Optimization Algorithm Booster: An Evolutionary and Learning Competition Scheme," *Neurocomputing*, vol. 643, 2025, doi: 10.1016/j.neucom.2025.130405.

[34] S. K. Mogha, S. Deshwal, and P. Kumar, "Current-to-Best Crossover for Modified Jaya Algorithm," *Int. J. Math. Eng. Manag. Sci.*, vol. 10, no. 4, pp. 1055–1079, 2025, doi: 10.33889/IJMEMS.2025.10.4.051.

[35] J.-S. Chou, J.-S. Lien, and C.-Y. Liu, "Integrative AI and UAV-based visual recognition with metaheuristics for automated repair cost analysis of bridge structural deterioration," *Autom. Constr.*, vol. 176, 2025, doi: 10.1016/j.autcon.2025.106273.

[36] R. Nekoueian, T. Servranckx, and M. Vanhoucke, "A dynamic learning-based genetic algorithm for scheduling resource-constrained projects with alternative subgraphs," *Appl. Soft Comput.*, vol. 180, 2025, doi: 10.1016/j.asoc.2025.113316.

[37] P. Sharma and S. Raju, "Metaheuristic optimization algorithms: a comprehensive overview and classification of benchmark test functions," *Soft Comput.*, vol. 28, no. 4, pp. 3123–3186, 2024, doi: 10.1007/s00500-023-09276-5.

[38] A. Sinha, D. Pujara, and H. K. Singh, "Bilevel Optimization-Based Decomposition for Solving Single and Multiobjective Optimization Problems," *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 88-102, 2025, doi: 10.1007/978-981-96-3506-1_7.

[39] G. Tian *et al.*, "A novel water distribution model considering the dynamic coupling of canals and gates," *Comput. Electron. Agric.*, vol. 236, 2025, doi: 10.1016/j.compag.2025.110434.

[40] A. Seyyedabbasi, P. J. Canatalay, G. Hu, H. A. Shehadeh, and X. Wang, "V-shaped and S-shaped binary artificial protozoa optimizer (APO) algorithm for wrapper feature selection on biological data," *Cluster Comput.*, vol. 28, no. 3, 2025, doi: 10.1007/s10586-024-04927-0.

[41] P. Vijai and P. Bagavathi Sivakumar, "A hybrid multi-objective optimization approach with NSGA-II for feature selection," *Decis. Anal. J.*, vol. 14, 2025, doi: 10.1016/j.dajour.2025.100550.

[42] H. Hajimiri and A. Bagheri, "A new R&D-based algorithm for optimization of large-scale problems," *Neural Comput. Appl.*, vol. 37, no. 15, pp. 9063–9094, 2025, doi: 10.1007/s00521-025-11057-0.

[43] Y. Gong, S. Zhong, S. Zhao, F. Xiao, W. Wang, and Y. Jiang, "Optimizing green splits in high-dimensional traffic signal control with trust region Bayesian optimization," *Comput. Aided Civ. Infrastruct. Eng.*, vol. 40, no. 6, pp. 741–763, 2025, doi: 10.1111/mice.13293.

[44] J. Zhang, J. Yang, and F. Yan, "Binary plant rhizome growth-based optimization algorithm: an efficient high-dimensional feature selection approach," *J. Big Data*, vol. 12, no. 1, 2025, doi: 10.1186/s40537-025-01066-0.

[45] K. Joni, "Parameter Estimation Of Photovoltaic based on Chaotic Elite Mountain Gazelle Optimizer," *Vokasi Unesa Bull. Eng. Technol. Appl. Sci.*, pp. 30–37, 2024.

[46] W. Aribowo, B. Suprianto, and A. Prapanca, "A novel modified dandelion optimizer with application in power system stabilizer," *Int J Artif Intell*, vol. 12, no. 4, pp. 2033–2041, 2023.

[47] G. Dei, D. K. Gupta, B. K. Sahu, M. Bajaj, V. Blazek, and L. Prokop, "A novel TID + IDN controller tuned with coatis optimization algorithm under deregulated hybrid power system," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-89237-0.

[48] A. S. Ebrie and Y. J. Kim, "Reinforcement learning-based optimization for power scheduling in a renewable energy connected grid," *Renew. Energy*, vol. 230, 2024, doi: 10.1016/j.renene.2024.120886.

[49] A.-Q. Tian, X.-Y. Wang, H. Xu, H.-X. Lv, J.-S. Pan, and V. Snášel, "Multi-objective optimization model for railway heavy-haul traffic: Addressing carbon emissions reduction and transport efficiency improvement," *Energy*, vol. 294, 2024, doi: 10.1016/j.energy.2024.130927.

[50] H. A. Shehadeh, "Bermuda Triangle Optimizer (BTO): A Novel Metaheuristic Method for Global Optimization," *Int. J. Adv. Soft Compu. Appl*, vol. 17, no. 2, 2025, doi: 10.15849/IJASCA.250730.01.

[51] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem," *Knowledge-Based Syst.*, 2019, doi: 10.1016/j.knosys.2018.08.030.

[52] H. A. Shehadeh, "Chernobyl disaster optimizer (CDO): a novel meta-heuristic method for global optimization," *Neural Comput. Appl.*, vol. 35, no. 15, pp. 10733–10749, 2023.

[53] A. Seyyedabbasi and F. Kiani, "Sand Cat swarm optimization: a nature-inspired algorithm to solve global optimization problems," *Eng. Comput.*, vol. 39, pp. 2627–2651, 2023, doi: 10.1007/s00366-022-01604-x.

[54] H. A. Shehadeh, I. Ahmedy, and M. Y. I. Idris, "Sperm swarm optimization algorithm for optimizing wireless sensor network challenges," in *ACM International Conference Proceeding Series*, 2018, pp. 53–59. doi: 10.1145/3193092.3193100.

[55] E.-S. M. El-kenawy, N. Khodadadi, S. Mirjalili, A. A. Abdelhamid, M. M. Eid, and A. Ibrahim, "Greylag Goose Optimization: Nature-inspired optimization algorithm," *Expert Syst. Appl.*, vol. 238, 2024, doi: 10.1016/j.eswa.2023.122147.

[56] M. A. Al-Betar, M. A. Awadallah, M. S. Braik, S. Makhadmeh, and I. A. Doush, "Elk herd optimizer: a novel nature-inspired metaheuristic algorithm," *Artif. Intell. Rev.*, vol. 57, no. 3, 2024, doi: 10.1007/s10462-023-10680-4.

[57] T. T. Dhivyaprabha, P. Subashini, and M. Krishnaveni, "Synergistic fibroblast optimization: a novel nature-inspired computing algorithm," *Front. Inf. Technol. Electron. Eng.*, vol. 19, pp. 815–833, 2018.

[58] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341-359, 1997, doi: 10.1023/A:1008202821328.

[59] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," *International conference in swarm intelligence*, pp. 355-364, 2010, doi: 10.1007/978-3-642-13495-1_44.

[60] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[61] A. Baihan *et al.*, "A Hybrid Meta-heuristic Algorithm for Optimum Micro-robotic Position Control with PID Controller," *Int. J. Comput. Intell. Syst.*, vol. 18, no. 1, 2025, doi: 10.1007/s44196-025-00799-3.

[62] H. G. Murtza Qamar, X. Guo, E. Seif Ghith, M. Tlija, and A. Siddique, "Assessment of energy management and power quality improvement of hydrogen based microgrid system through novel PSO-MWWO technique," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-024-78153-4.

[63] S. Ekinci *et al.*, "Advanced control parameter optimization in DC motors and liquid level systems," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-85273-y.

[64] M. Braik, H. Al-Hiary, A. Hammouri, M. A. Awadallah, H. Alzoubi, and M. Azmi Al-Betar, "Tornado optimizer with Coriolis force: a novel bio-inspired meta-heuristic algorithm for solving engineering problems," *Artif. Intell. Rev.*, vol. 58, no. 4, 2025, doi: 10.1007/s10462-025-11118-9.

[65] V. Rajput, P. Mulay, and C. M. Mahajan, "Bio-inspired algorithms for

feature engineering: analysis, applications and future research directions," *Inf. Discov. Deliv.*, vol. 53, no. 1, pp. 56–71, 2025, doi: 10.1108/IDD-11-2022-0118.

[66] R. Priyadarshi and R. R. Kumar, "Evolution of Swarm Intelligence: A Systematic Review of Particle Swarm and Ant Colony Optimization Approaches in Modern Research," *Arch. Comput. Methods Eng.*, pp. 1-42, 2025, doi: 10.1007/s11831-025-10247-2.

[67] X. Wang, V. Snášel, S. Mirjalili, J.-S. Pan, L. Kong, and H. A. Shehadeh, "Artificial Protozoa Optimizer (APO): A novel bio-inspired metaheuristic algorithm for engineering optimization," *Knowledge-Based Syst.*, vol. 295, p. 111737, 2024.

[68] B. G. Thengvall, M. P. Deskevich, and S. N. Hall, "Measuring the effectiveness and efficiency of simulation optimization metaheuristic algorithms," *J. Heuristics*, vol. 31, no. 1, 2025, doi: 10.1007/s10732-025-09549-2.

[69] M. T. Hussain *et al.*, "Enhanced MPP Tracking in Partial Shading Conditions for Solar PV Systems: A Metaheuristic Approach Utilizing Projectile Search Algorithm," *IEEE Access*, vol. 13, pp. 50895–50917, 2025, doi: 10.1109/ACCESS.2025.3546351.

[70] N. A. Mansour, M. S. Saraya, and A. I. Saleh, "Groupers and moray eels (GME) optimization: a nature-inspired metaheuristic algorithm for solving complex engineering problems," *Neural Comput. Appl.*, vol. 37, no. 1, pp. 63–90, 2025, doi: 10.1007/s00521-024-10384-y.

[71] Z. Guo, G. Liu, and F. Jiang, "Chinese Pangolin Optimizer: a novel bio-inspired metaheuristic for solving optimization problems," *J. Supercomput.*, vol. 81, no. 4, 2025, doi: 10.1007/s11227-025-07004-4.

[72] W. Aribowo, "Comparison Study On Economic Load Dispatch Using Metaheuristic Algorithm," *Gazi Univ. J. Sci.*, vol. 35, no. 1, pp. 26-40, 2022, doi: 10.35378/gujs.820805.

[73] H. A. Shehadeh and N. M. Shagari, "A hybrid grey wolf optimizer and sperm swarm optimization for global optimization," *Handbook of intelligent computing and optimization for sustainable development*, pp. 487-507, 2022, doi: 10.1002/9781119792642.ch24.

[74] L. Abualigah, D. Oliva, T. Mzili, A. Sabo, and H. A. Shehadeh, "Frilled Lizard Optimization to optimize parameters Proportional Integral Derivative of DC Motor," *Vokasi Unesa Bull. Eng. Technol. Appl. Sci.*, pp. 14–21, 2024, doi: 10.26740/vubeta.v1i1.33973.

[75] W. Aribowo and H. A. Shehadeh, "Novel Modified Chernobyl Disaster Optimizer for Controlling DC Motor," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 35, pp. 1361-1369, 2024, doi: 10.11591/ijeecs.v35.i3.pp1361-1369.

[76] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A nature-inspired metaheuristic," *Expert Syst. Appl.*, vol. 152, p. 113377, 2020, doi: 10.1016/j.eswa.2020.113377.

[77] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, pp. 1942–1948, 1995.

[78] E. H. Haro, D. Oliva, L. A. Beltrán, and A. Casas-Ordaz, "Enhanced differential evolution through chaotic and Euclidean models for solving flexible process planning," *Knowl. Based Syst.*, vol. 314, 2025, doi: 10.1016/j.knosys.2025.113189.

[79] R. Lin, Z. Xu, L. Yu, and T. Wei, "EABC-AS: Elite-driven artificial bee colony algorithm with adaptive population scaling," *Swarm Evol. Comput.*, vol. 94, 2025, doi: 10.1016/j.swevo.2025.101893.

[80] M. Yu *et al.*, "A multi-strategy enhanced Dung Beetle Optimization for real-world engineering problems and UAV path planning," *Alexandria Eng. J.*, vol. 118, pp. 406–434, 2025, doi: 10.1016/j.aej.2025.01.055.

[81] P. Kumar and M. Ali, "Solving the Economic Load Dispatch Problem by Attaining and Refining Knowledge-Based Optimization," *Mathematics*, vol. 13, no. 7, 2025, doi: 10.3390/math13071042.

[82] X. Zhu, J. Zhang, C. Jia, Y. Liu, and M. Fu, "A Hybrid Black-Winged Kite Algorithm with PSO and Differential Mutation for Superior Global Optimization and Engineering Applications," *Biomimetics*, vol. 10, no. 4, 2025, doi: 10.3390/biomimetics10040236.