

Enhanced Dung Beetle Optimization Algorithm Combined Harris Hawks Optimizer and Nelder Mead Simplex Method for Solar-Connected Smart-grid Application

Huynh Tuyet Vy¹, Ho Pham Huy Anh²

¹Faculty of Electronic Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City

²Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, Dien Hong Ward, Ho Chi Minh City, Viet Nam

²Vietnam National University Ho Chi Minh City (VNU-HCM), Dong Hoa Ward, Ho Chi Minh City, Viet Nam

Article Info

Article history:

Received Oct 4, 2024

Revised Dec 3, 2025

Accepted Dec 18, 2025

Keywords:

Harris Hawks Optimizer
Dung Beetle Optimization
Nelder-Mead simplex algorithm
Solar-connected home power
smart-grid

ABSTRACT

The Dung Beetle Optimization algorithm (DBO) is a swarm-based intelligence algorithm with competitive performance against other popular optimization algorithms. However, its process is often fallen in local optimum, insufficient accuracy, and slow convergence speed due to a lack of combination or collaboration between search agents. This research proposes an advanced DBO approach combining the Harris Hawks Optimizer (HHO) and Nelder Mead method to improve slow convergence speed, insufficient accuracy, and premature convergence. The Nelder Mead method is used in the sub-population of ball-rolling to reduce the probability of falling in local optimum, along with “seven kills” strategy of HHO method that is combined in the former iterations of the DBO algorithm to enhance its global search capacity and convergence speed. The performance of the proposed enhanced dung beetle optimization (EDBO) algorithm is evaluated via 30 CEC-2017 benchmark functions and compared with several representative meta-heuristic algorithms, including the original DBO and HHO, as well as three recently proposed methods: RUN, SMA, and AO. The result shows that EDBO consistently achieves superior performance over most of the C-test functions in terms of solution quality and robustness. Additionally, when applied to the optimization of the operating cost of a solar-connected residential power system, the proposed EDBO attains the best or highly competitive global optimum compared with the competing algorithms.

Copyright © 2025 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Ho Pham Huy Anh

Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, Dien Hong Ward, Ho Chi Minh City, Viet Nam

Vietnam National University Ho Chi Minh City (VNU-HCM), Dong Hoa Ward, Ho Chi Minh City, Viet Nam

Email: hphanh@hcmut.edu.vn

1. INTRODUCTION

Population-based meta-heuristic (MH) algorithms are the most popular algorithms used to solve many complex optimization problems in life. Based on the general view that MH algorithms can be classified according to their underlying source of inspiration [1], they are commonly organized into major categories: swarm-based, evolutionary, human-based, physics-inspired, system-based, and biology-based. The number of swarm-based algorithms, which use the swarming compartment of particles in a swarm to search for the global optimum result in a searching region, takes up a large proportion of population-based MH algorithms. The popular swarm-intelligent approaches compose of PSO, Grey Wolf Optimizer (GWO), Harris Hawks Optimizer (HHO) [2], Dung Beetle Optimization (DBO) [3], etc. The new MH algorithms have been proposed continuously every month, evenly every week [4]. Compared with the recently proposed optimization techniques regarding precision and stability, the DBO obtains highly super performance. Therefore, it is applied to deal with the optimization problems in some specified applications. It has been applied to smart residential

energy management, where the algorithm was assessed for improving the performance of energy management system (EMS) control strategies [5]. An improved version of the algorithm has also been proposed for the optimal scheduling and capacity sizing of off-grid hybrid renewable energy systems, demonstrating enhanced convergence and solution quality [6]. DBO has been employed to optimize the hyperparameters of hybrid deep-learning-based models for ultra-short-term photovoltaic power forecasting [7]. In addition, the algorithm has been utilized to enhance scheduling procedures in microgrid operational planning [8]. Although these advanced DBO algorithms' optimization performance is better than most traditional optimization algorithms, there are still some drawbacks, related to sluggish convergent velocity along with premature convergence to the local optimum solution.

Since its introduction in late 2022, several studies have proposed enhanced variants of the DBO algorithm to overcome its inherent limitations. One improvement incorporates the non-dominated sorting mechanism into the original framework to form the NSDBO, enabling better handling of multi-objective scheduling problems in microgrids by strengthening both search diversity and convergence stability [9]. Another study introduces adaptive control strategies and refined position-update mechanisms to improve global exploration and prevent stagnation when solving large-scale optimization tasks [10]. A further enhancement has combined mean fitness–distance balancing with multi-strategy fusion techniques, which significantly improve the balance between exploration and exploitation and yield more reliable performance in complex engineering applications [11]. Additional research has explored multi-strategy fusion with improved parameter adaptation to accelerate convergence while avoiding premature entrapment in local optima [12]. A distribution-based improved version of DBO has also been applied to energy-related optimization problems, demonstrating greater robustness and higher solution quality [13]. More recently, a hybrid adaptive DBO integrating advanced learning and hybridization mechanisms has been shown to yield higher optimization efficiency and enhanced stability when solving highly nonlinear real-world problems [14].

Although recent studies have introduced several improved variants of DBO, certain limitations in its search behavior remain. Several works highlight that the original DBO struggles to maintain an appropriate balance between global exploration and local exploitation, which may reduce its robustness in complex optimization environment [9],[12],[13]. In addition, some studies report that DBO exhibits limited effectiveness during its initial search phase, where the algorithm tends to rely heavily on stochastic search steps, resulting in weak guidance toward promising regions and an increased likelihood of premature stagnation [10], [11], [14]. Furthermore, existing hybrid or multi-strategy versions of DBO often employ static or partially adaptive components, which may constrain their responsiveness when handling highly nonlinear, multimodal, or real-world engineering problems [12], [13]. These observations suggest that there is still potential to develop more adaptive and better-balanced DBO scheme.

To overcome the limitations outlined above, this paper proposes an enhanced Dung Beetle Optimizer that integrates three complementary mechanisms. First, the fast convergence characteristics of HHO algorithm are incorporated to reinforce exploitation and accelerate the convergence process. Second, the global search capability of the Nelder–Mead (NM) simplex method is embedded to improve the algorithm's ability to escape local optima. Finally, the evolutionary direction of the original DBO is preserved during the early search phase to provide more effective guidance and reduce stochastic wandering at the beginning of the optimization process. Through this hybrid design, the proposed method aims to simultaneously strengthen exploration and exploitation, increase convergence speed, and mitigate premature convergence—two of the primary limitations identified in the canonical DBO.

The key contributions of this study are as follows:

- A hybrid DBO scheme (EDBO) is developed by enhancing both exploration and exploitation through diversified random initialization and an improved guidance mechanism during the early search stage.
- The Nelder–Mead simplex method is incorporated for the first time into the DBO algorithm to reinforce local refinement, accelerate convergence, and reduce the likelihood of premature stagnation in both low- and high-dimensional optimization problems.
- Selected search agents are further strengthened using the fast convergence strategies of HHO algorithm, enabling adaptive exploitation and improved convergence consistency across complex landscapes.
- The effectiveness of the proposed EDBO is validated on 30 CEC-2017 benchmark functions and a practical energy-management case study in a solar-connected residential power system, demonstrating superior accuracy, stability, and convergence behavior.
- The results has statistically analyzed by overall ranking and Friedman–Nemenyi statistical analyses along with a comprehensive comparison with the canonical DBO, HHO, and several state-of-the-art MH algorithms that confirm the superior robustness and performance of the proposed EDBO across diverse optimization scenarios.

The remainder of the paper includes the following parts: A concise introduction of the original DBO, HHO algorithm and Nelder Mead simplex algorithm are presented in Section 2, 3 and 4, respectively. The EDBO is fully presented in Section 5. The results executed by 30 CEC-2017 benchmark functions are comparatively presented in Section 6. Section 7 describes the results of applying this proposed algorithm into a practical application. The study is concluded in the final section.

2. THE ORIGINAL DBO ALGORITHM

The DBO technique [3], a novel SI-based optimization technique, aims to improve search performance by modeling dung beetle behavior and improving search performance. The DBO algorithm models dung beetles' behavior, composing ball-roll, dance, forage, steal, and reproduce actions. Dung beetles create dung balls, lay eggs, raise future generations, and provide essential food efficiently. Thus, DBO contains four critical dung beetle swarms. Each swarm has N agents, with each representing a candidate solution. The distribution ratio of each swarm in population shows not specific but settled in terms of true-world tasks.

The four sub-populations of the original DBO algorithm perform five behaviors as follows:

2.1. Ball-rolling Behavior

Dung beetles use a light source such as the sun, the moon to move and roll the dung ball straightly. Its behavior is formulated as follows:

$$\begin{aligned} x_i(t+1) &= x_i(t) + \alpha \cdot k \cdot x_i(t-1) + b \cdot \Delta x \\ \Delta x &= |x_i(t) - X^w| \end{aligned} \quad (1)$$

with t denotes the number of iterations; $x_i(t)$ denotes the location of the i^{th} beetle at the t^{th} iteration; k and b are constant values; α represents a parameter between 1 and -1; Δx measures changes related to light density and X^w indicates the worst location.

2.2. Dancing Behavior

While rolling the dung ball, beetles encounter hurdles and they are not able to continuously navigate. Then it requires determining another movement via dance behavior. The position of new orientation is adjusted and described as:

$$x_i(t+1) = x_i(t) + \tan(\theta) |x_i(t) - x_i(t-1)| \quad (2)$$

with θ denotes the deflecting angular value within $[0, \pi]$.

2.3. Reproductive Behavior

The brooding region of female beetles is described by:

$$\begin{aligned} Lb^* &= \max(X^* \cdot (1-R), Lb), \\ Ub^* &= \min(X^* \cdot (1+R), Ub) \end{aligned} \quad (3)$$

The location of brooding ball is updated in (4):

$$B_i(t+1) = X^* + b_1(B_i(t) - Lb^*) + b_2(B_i(t) - Ub^*) \quad (4)$$

with X^* represents the present optimum location solution; Lb , Ub denotes the lower/upper bound, respectively, of the optimized solution; Lb^* , Ub^* denotes the lower/upper of brooding area; $R = 1 - \frac{t}{T_{\max}}$ with $T_{\max}$ denotes the maximal iterations; $B_i(t)$ denotes the position of the i^{th} brood ball at the t^{th} iteration; b_1 and b_2 are two random vectors.

2.4. Foraging Behavior

The optimum foraging region of an adult beetle is determined as:

$$\begin{aligned} Lb^b &= \max(X^b \cdot (1-R), Lb), \\ Ub^b &= \min(X^b \cdot (1+R), Ub), \\ x_i(t+1) &= x_i(t) + C_1 \cdot (x_i(t) - Lb^b) + C_2 \cdot (x_i(t) - Ub^b) \end{aligned} \quad (5)$$

with X^b denotes the global best location; C_1 be a casual value and C_2 be a casual vector.

2.5. Stealing Behavior

“Thieves” dung beetles steal dung balls from other ones. The location of the thieving beetles is updated as:

$$x_i(t+1) = X^b + S.g. \left(|x_i(t) - X^*| + |x_i(t) - X^b| \right) \quad (6)$$

with S be an unchanged value and g be a casual vector.

3. THE ORIGINAL HHO ALGORITHM

Harris Hawks Optimizer (HHO) [2] represents a swarm-based, nature-inspired optimization paradigm inspired by surprising action in Harris' hawks. The approach mimics dynamic patterns and behaviors, providing promising and competitive results in comparison with popular meta-heuristic approaches. Harris' hawks use the surprise pounce or "seven kills" strategy to capture prey. This cooperative attack involves multiple hawks attacking from different directions, converging on a detected escaping rabbit, and sometimes multiple dives. Harris' hawks can display various chasing styles relying on the prey's behavior and circumstances. Switching tactics occur when the best hawk gets lost, causing the chase to continue. These cooperative tactics increase the rabbit's vulnerability and prevent it from recovering its defensive capabilities. Harris' hawks' main behaviors can be observed in nature.

The HHO algorithm includes two phases:

- Exploration phase: Harris' hawk can detect prey using its keen eyes, but sometimes it is unavailable. The fact is that they observe, and watch desert sites to detect prey. They perch randomly at locations and wait for prey using two strategies: perching on family members and rabbits and perching on tall trees within their home range.
- Exploitation phase: Harris' hawks perform surprise pounces by attacking prey, but prey often escape, requiring different chasing styles. Four strategies available are suggested in HHO to estimate this attack period. Hawks attempt to catch prey by performing a soft or hard besiege, relying on the prey's energy. They encircle the prey from different directions to enhance their chances of cooperatively slaughtering the prey. As the prey loses energy, they enforce the besiege procedure to grasp the exhausted prey.

4. THE NELDER MEAD SIMPLEX ALGORITHM

This algorithm represents a best-known pattern search optimization algorithm for solving multi-dimensional unconstrained optimization problems without derivatives [15], [16]. This algorithm maintains a simplex to approximate an optimal point. The vertices are sorted by the objective function value. Depending on the worst point and the center of the best vertices, the worst vertex is replaced by a new point via four operations: reflection, expansion, contraction, and shrink. The algorithm is used as a local or global search procedure, and it is also very sensitive to the initial simplex (initial guess). Initially, when the simplex is large, the algorithm may jump over many local optima, but as it contracts, it may get stuck in one of the local optima.

Unfortunately, there is no general rule to determine the best initial simplex of the Nelder-Mead optimization because this requires at least a vague knowledge of the response surface. Therefore, various approaches are proposed, such as the axis-by-axis simplex, the regular simplex by Spendley, the randomized bounds, and Pfeffer's simplex [17]. All approaches mentioned are based on the starting point. In common practice, the most frequent choice is Pfeffer's simplex initialization method. This approach scales the “guess” vertex with respect to the characteristic length of the problem via the axis-by-axis method:

$$x_i = \begin{cases} ((x_1)_1, \dots, (x_1)_{i-2}, 0.05(x_1)_{i-1}, (x_1)_i, \dots, (x_1)_{d+1}), & \text{if } (x_1)_{i-1} \neq 0 \\ ((x_1)_1, \dots, (x_1)_{i-2}, 0.00025, (x_1)_i, \dots, (x_1)_{d+1}), & \text{else} \end{cases} \quad (7)$$

A simplex is a convex hull of $d + 1$ vertices in a d -dimensions space: $x_1, x_2, \dots, x_{d+1}$. At each iteration, the vertices are sorted by the objective function values $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{d+1})$ with x_1 being the best vertex and x_{d+1} being the worst vertex. Four operations are associated with scalar parameters including α reflection, β expansion, γ contraction and δ shrink and are run based on the center vertex of d best vertices ($\underline{x}$):

$$\underline{x} = \frac{1}{d} \sum_{i=1}^d x_i \quad (8)$$

4.1. Refection

Evaluate and sort vertices based on their object function values before executing the refection operation. Calculate the refection point x_r and its object function value $f_r = f(x_r)$. Replace the worst point x_{d+1} by x_r if $f(x_1) \leq f_r < f(x_d)$. x_r is defined as follows:

$$x_r = \underline{x} + \alpha(\underline{x} - x_{d+1}) \quad (9)$$

4.2. Expansion

If $f_r < f(x_1)$ then the expansion point is calculated as follows:

$$x_e = \underline{x} + \beta(x_r - \underline{x}) \quad (10)$$

Replace the worst point x_{d+1} by x_e if $f(x_e) < f_r$. Otherwise, replace x_{d+1} with x_r .

4.3. Inside and Outside Contraction

If $f(x_d) \leq f_r < f(x_{d+1})$ then the outside contraction point is calculated as follows:

$$x_{oc} = \underline{x} + \gamma(x_r - \underline{x}) \quad (11)$$

Replace the worst point x_{d+1} by x_{oc} if $f(x_{oc}) \leq f_r$. Otherwise, execute shrink operation.

If $f_r \geq f(x_{d+1})$ then the inside contraction point is calculated as follows:

$$x_{ic} = \underline{x} - \gamma(x_r - \underline{x}) \quad (12)$$

Replace the worst point x_{d+1} by x_{ic} if $f(x_{ic}) < f_{d+1}$. Otherwise, execute shrink operation.

4.4. Shrink

Replace all points except the best point x_1 as follows:

$$x_i = x_1 + \delta(x_i - x_1) \quad (13)$$

5. PROPOSED ENHANCED DUNG BEETLE OPTIMIZATION (EDBO) ALGORITHM

5.1. HHO and DBO's Exploration and Exploitation

The search mechanism of DBO, which relies on direction-based transitions among four sub-populations, still faces challenges in sustaining both adequate exploration and rapid convergence throughout the search process. A fundamental limitation arises from the fact that DBO's four sub-populations operate mostly as independent search units, without the cooperative information-sharing mechanisms observed in algorithms such as HHO. HHO employs explicitly cooperative behaviors, where agents adapt based on the positions of other members or the average state of the entire population. By comparison, DBO agents update their positions primarily through their previous locations, the local best, the global best, and direction-based movement rules. While this mechanism preserves diversity, it also limits the algorithm's capacity to adapt collectively and to exploit shared knowledge, which may ultimately slow convergence and hinder effective navigation of complex landscapes.

DBO's search behavior is distributed among four sub-populations:

- Ball-rolling beetles: This group is mainly responsible for global exploration. Each agent updates its position using its previous location and a direction pointing toward the current worst individual, encouraging outward movement into unexplored regions. This mechanism helps broaden the search space and reduces the likelihood of early convergence.
- Brood-ball beetles: These agents focus on local exploitation by updating their positions around the current local best within a small-radius neighborhood. Their movement direction guides them to refine solutions in a confined region. This behavior strengthens the algorithm's ability to improve promising areas with precision.
- Small dung beetles: This sub-population provides a balance between exploration and exploitation. Agents move within a larger boundary centered on the global best, combining moderate global search with local refinement. This helps maintain diversity while ensuring attention to the algorithm's most promising region.
- Thieves: Thief beetles intensify local refinement around the best available food source. Their position update uses a direction derived from both global and local best information, enabling precise searches near the current optimum. This mechanism helps extract incremental improvements in solution quality.

Among these groups, the ball-rolling sub-population, which is approximately 20% of the total population, is primarily responsible for exploration. As illustrated in Fig. 1 for benchmark function C25, DBO's exploration percentage decreases over time but consistently remains above 20%, indicating a persistent exploratory tendency. In contrast, HHO performs exploration mainly during the early phase, while the exploitation phase contains no exploration actions. This difference motivates the hybrid strategy proposed in this study.

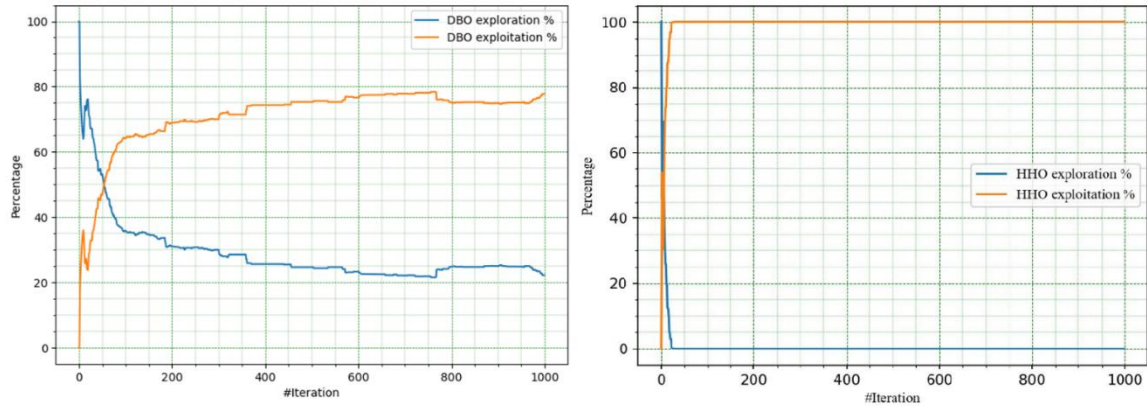


Figure 1. DBO and HHO's exploration vs. exploitation chart (%) for benchmark function C25 (with 30 search agents in a 30-dimensional search space).

5.2. Integration of the Nelder–Mead Local Search

Although the combination of HHO and DBO improves global exploration, the framework still lacks a reliable mechanism for local refinement when the search enters promising regions. To strengthen exploitation, the Nelder–Mead (NM) simplex method is incorporated as a deterministic local optimizer. NM is widely recognized for its ability to refine solutions in non-smooth and derivative-free problems, making it a suitable complement to the stochastic nature of HHO and DBO.

In the proposed design, the initial point of each simplex is selected from the best positions produced by HHO. This choice is essential because the NM algorithm is highly sensitive to the initial simplex: a simplex placed far from a promising region may lead to slow or ineffective convergence [18], [19]. By starting from HHO-generated high-quality positions, the NM search is immediately guided toward regions with greater potential, increasing the likelihood of successful refinement and reducing wasted computational effort.

To construct the simplex itself, Pfeffer's method is used to ensure that the simplex is well-shaped and non-degenerate. Traditional simplex initialization methods often generate ill-conditioned or distorted simplexes, which may affect the stability of NM's reflection, expansion, and contraction operations. Pfeffer's method avoids these issues by producing a balanced geometric structure around the initial point, enabling NM to operate more reliably and consistently across different problem dimensions.

Through this integration, the NM method acts as a precise local search layer: HHO identifies promising regions, Pfeffer's method guarantees a stable simplex structure, and NM further refines the solution. This combination enhances exploitation without compromising exploration, thereby strengthening the overall effectiveness of the proposed EDBO algorithm.

5.3. Proposed Enhanced Dung Beetle Optimization (EDBO) Algorithm

The Proposed EDBO algorithm is developed to mitigate the risk of premature convergence caused by excessive exploitation in the original DBO, where three sub-populations simultaneously perform exploitation. The design of EDBO integrates the fast convergence characteristics of the HHO algorithm with the local search efficiency of the NM simplex method.

To incorporate these mechanisms effectively into the DBO framework, the hybridization is performed sequentially across the algorithm's two behavioral sub-populations. In the exploration-oriented ball-rolling subgroup, HHO's exploration strategy is activated during the first 10% of iterations to accelerate the identification of promising regions. At this stage, an NM simplex is initialized for each agent using Pfeffer's method, and during the subsequent phase (10–50% of iterations), agents randomly alternate between HHO-based exploration and NM simplex operations to balance diversification and local refinement. Beyond 50% of iterations, the subgroup returns to DBO's original ball-rolling and dancing behaviors, preserving population diversity while maintaining computational simplicity.

In the exploitation-oriented small dung beetle subgroup, HHO's exploitation strategy is employed during the first 10% of iterations to quickly intensify search around high-quality areas. After this phase, agents

revert to the original DBO forging behavior, ensuring stable and consistent local exploitation without overwhelming the global search dynamics.

The overall procedure of EDBO is shown in Fig. 2. The algorithm begins by initializing the population, dividing individuals into the four standard DBO sub-populations, evaluating their fitness, and determining the global-best solution. During each iteration, only the ball-rolling and small dung beetle sub-populations are hybridized: the ball-rolling subgroup adopts HHO-oriented exploration in the early stage and incorporates a NM simplex phase before mid-run, while the small dung beetle subgroup employs an HHO-based exploitation phase before returning to its native behavior. The remaining two sub-populations operate exactly as in the original DBO. After four sub-populations update their positions according to their respective mechanisms, every agent is re-evaluated, and the global-best solution is updated. This iterative process continues until the maximum number of iterations is reached.

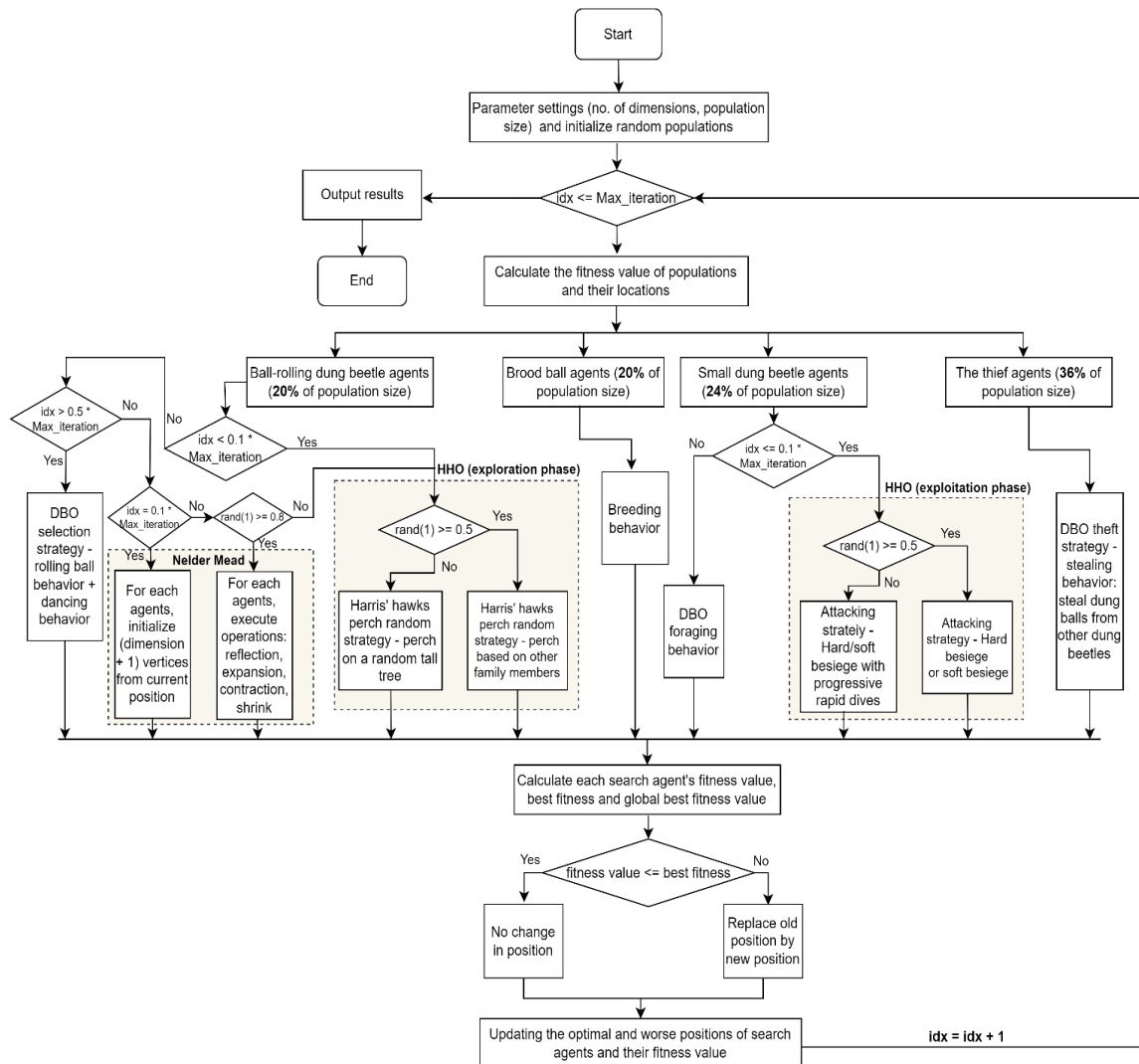


Figure 2. Flowchart of the proposed EDBO algorithm.

6. SIMULATION RESULTS WITH BENCHMARK FUNCTIONS

To evaluate the effectiveness of the proposed EDBO algorithm, a comprehensive set of 30 benchmark test functions from the CEC-BC-2017 suite [20] was employed. These functions span a variety of unimodal, multimodal, hybrid, and composite landscapes, providing a rigorous and diverse testing environment. The global optimum values of these functions serve as the primary metric for assessing the search accuracy and overall optimization capability of the algorithms.

The proposed EDBO algorithm was benchmarked against its original counterparts (DBO and HHO) and three recent MH algorithms: Runge Kutta Optimization (RUN), Slime Mould Algorithm (SMA), and

Aquila Optimizer (AO). RUN leverages Runge–Kutta-based strategies for precise search space exploitation, SMA mimics slime mould dynamics to balance exploration and exploitation, and AO employs eagle-inspired hunting tactics to enhance global search and escape local optima. Control parameters for all algorithms are summarized in Table 1. For a fair comparison, each algorithm was independently executed 20 times on every benchmark function, with a maximum of 1000 iterations per run.

To analyze the performance across different search space complexities, nine combinations of dimensionality and population size were considered. Specifically, each algorithm was tested on dimensions of 30, 50, and 100, with corresponding population sizes of 30, 50, and 100 agents. For every configuration, the mean optimum values obtained over 20 runs are documented in Appendix 1. A two-stage evaluation protocol was employed to compare all algorithms across the benchmark suite. An overall ranking analysis was first used to assess their relative performance on each function. Subsequently, statistical significance was examined using the Friedman test with the post-hoc Nemenyi procedure to identify pairwise differences and verify whether the performance gains achieved by the proposed EDBO algorithm are statistically significant.

Table 1. The applied algorithms' control parameters.

Algorithms	Parameters
DBO	$k = 0.1; b = 0.3; S = 0.5$.
HHO	$q = 0.5$ (perching strategy), $J \in [0, 2]$, $E_0 \in [-1, 1]$, $E \in [0, 2]$ (energy of rabbit).
Nelder-Mead	$\alpha = 1$ (reflection), $\beta = 2$ (expansion), $\gamma = 0.5$ (contraction), $\delta = 0.5$ (shrink).
RUN	$h = 0.5$ (step size).
SMA, AO	No requirements

6.1. Overall Ranking Analysis

To enable a unified comparison across all benchmark functions and dimensional settings, an overall ranking scheme [21] was employed. The ranking score for each algorithm is computed as follows:

$$Rank = 0.3 \sum_{i=1}^{30} rank_{30D} + 0.3 \sum_{i=1}^{30} rank_{50D} + 0.4 \sum_{i=1}^{30} rank_{100D} \quad (14)$$

To compute the ranking scores, each benchmark function is evaluated independently. For a given test function, the algorithms are ranked based on their mean optimum value over 20 runs, where rank = 1 corresponds to the best performance and rank = m corresponds to the worst among m competing algorithms. This procedure is repeated for all 30 functions in each dimensional setting, producing three sets of ranking sums for the 30D, 50D, and 100D cases. Equation (14) then aggregates these scores using dimension-dependent weights, giving greater emphasis to high-dimensional problems where scalability becomes more critical. A lower final score indicates better optimization capability. The aggregated ranking results for population sizes of 30, 50, and 100 agents are provided in Table 2. EDBO consistently achieves the best performance with the lowest scores (2.143, 1.997, and 1.947), confirming its robustness under varying configurations. RUN and AO follow as intermediate performers, while HHO ranks slightly lower. SMA and DBO show the weakest results, with scores above 4.9 and 5.4 in all cases. Although ranking values fluctuate slightly with population size, the overall hierarchy remains unchanged. These results highlight the success of the proposed hybridization strategy. By effectively mitigating the premature convergence issues observed in DBO, EDBO achieves more accurate global optima and faster convergence across both low- and high-dimensional test landscapes. The superior performance on high-dimensional functions is particularly notable, indicating a stronger balance between exploration and exploitation.

Table 2. Average ranking score of algorithms following Eq. (14).

POPULATION	HHO	DBO	RUN	SMA	AO	EDBO
30 agents	3.000	5.547	2.610	4.980	2.720	2.143
50 agents	2.883	5.493	2.753	5.010	2.863	1.997
100 agents	2.840	5.447	2.800	4.987	2.980	1.947

It is important to note that the performance trends reported here differ from those in [3], where DBO outperformed HHO on 20 benchmark functions. This discrepancy arises primarily from differences in test dimensionality: the study in [3] used a relatively low dimension of 10, whereas the present study evaluates algorithms at dimensions of 30, 50, and 100. The results here indicate that HHO becomes more competitive—and often superior—when dimensionality increases, while the performance of DBO degrades more noticeably. Consequently, DBO appears advantageous only for low-dimensional problems.

6.2. Friedman Test and Post-hoc Nemenyi Analysis

To further substantiate the comparative results obtained from the ranking analysis, a non-parametric statistical evaluation was carried out. The Friedman test [22] with confidence level of 95% ($\alpha = 0.05$) was selected to assess overall performance differences across all algorithms, while the post-hoc Nemenyi test [23] was adopted to determine statistically significant pairwise distinctions. The Iman–Davenport test was then applied to refine the Friedman analysis and provide a more powerful assessment of overall differences. As shown in Table 3, the computed F-statistics were extremely high for all configurations: 114.322 for 30 agents, 115.403 for 50 agents, and 108.919 for 100 agents, with degrees of freedom $df_1 = 5$ and $df_2 = 445$. All corresponding p-values were far below 0.0001, confirming that the observed performance differences among algorithms are statistically significant at the 95% confidence level. These results strongly support the rejection of the null hypothesis and justify the subsequent application of the Nemenyi post-hoc test to identify pairwise differences.

Table 3. Iman–Davenport Statistic.

POPULATION	N	k	F-distribution	df1	df2	p-value
30 agents	90	6	114.322	5	445	1.571E-77
50 agents	90	6	115.403	5	445	4.861E-78
100 agents	90	6	108.919	5	445	6.097E-75

Table 4. Critical difference (CD) and average ranks (Nemenyi, $\alpha = 0.05$).

POPULATION	CD ($\alpha=0.05$)	HHO	DBO	RUN	SMA	AO	EDBO
30 agents	0.7947	3.022	5.533	2.600	4.978	2.756	2.111
50 agents	0.7947	2.922	5.489	2.767	4.989	2.911	1.922
100 agents	0.7947	2.867	5.444	2.833	4.956	3.033	1.867

Table 5. Pairwise average-rank differences and Nemenyi decisions ($\alpha = 0.05$).

Algorithm Pair	30 agents	50 agents	100 agents
HHO vs DBO	2.511 (Sig.)	2.567 (Sig.)	2.577 (Sig.)
HHO vs RUN	0.422 (NS)	0.155 (NS)	0.034 (NS)
HHO vs SMA	1.956 (Sig.)	2.067 (Sig.)	2.089 (Sig.)
HHO vs AO	0.266 (NS)	0.011 (NS)	0.166 (NS)
HHO vs EDBO	0.911 (Sig.)	1.000 (Sig.)	1.000 (Sig.)
DBO vs RUN	2.933 (Sig.)	2.722 (Sig.)	2.611 (Sig.)
DBO vs SMA	0.555 (NS)	0.500 (NS)	0.488 (NS)
DBO vs AO	2.777 (Sig.)	2.578 (Sig.)	2.411 (Sig.)
DBO vs EDBO	3.422 (Sig.)	3.567 (Sig.)	3.577 (Sig.)
RUN vs SMA	2.378 (Sig.)	2.222 (Sig.)	2.123 (Sig.)
RUN vs AO	0.156 (NS)	0.144 (NS)	0.200 (NS)
RUN vs EDBO	0.489 (NS)	0.845 (Sig.)	0.966 (Sig.)
SMA vs AO	2.222 (Sig.)	2.078 (Sig.)	1.923 (Sig.)
SMA vs EDBO	2.867 (Sig.)	3.067 (Sig.)	3.089 (Sig.)
AO vs EDBO	0.645 (NS)	0.989 (Sig.)	1.166 (Sig.)

Note: "Sig." denotes $|\Delta R| > CD$; "NS" denotes $|\Delta R| \leq CD$.

Table 6. Significant wins per algorithm (Nemenyi, $\alpha = 0.05$).

POPULATION	HHO	DBO	RUN	SMA	AO	EDBO
30 agents	2	0	2	0	2	3
50 agents	2	0	2	0	2	5
100 agents	2	0	2	0	2	5

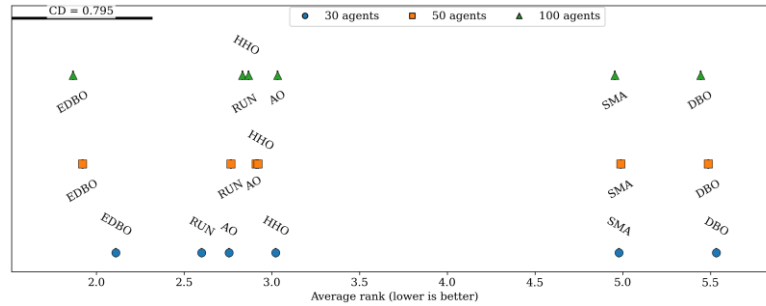


Figure 3. Combined CD diagram for 30, 50, and 100 agents (CD = 0.795).

Following the confirmation of overall significance by the Iman–Davenport test, the Nemenyi post-hoc procedure was employed to identify pairwise differences among algorithms. **Table 4** and **Fig. 3** report the critical difference (CD) at $\alpha = 0.05$ and the average ranks of six algorithms across three population sizes (30, 50, and 100 agents). The CD value was computed as 0.7947 for all configurations ($k = 6$, $N = 90$). EDBO consistently achieved the lowest average ranks (2.111, 1.922, and 1.867), followed by RUN and AO, while SMA and DBO exhibited the highest ranks, confirming the hierarchy observed in the overall ranking analysis.

Table 5 presents the pairwise average-rank differences and Nemenyi decisions for each population setting. For the 30-agent configuration, EDBO was significantly better than HHO, SMA, and DBO, as the rank differences exceeded the CD threshold, whereas its differences with RUN and AO were not significant. In contrast, for 50-agent and 100-agent settings, EDBO demonstrated statistically significant superiority over all other algorithms, with every pairwise difference surpassing the CD value. These results indicate that EDBO's advantage becomes more pronounced as population size increases.

To provide a concise summary, **Table 6** aggregates the number of significant wins for each algorithm. EDBO achieved three significant wins in the 30-agent setting and five wins in both 50-agent and 100-agent configurations, outperforming all competitors. Other algorithms, such as RUN and AO, recorded only two wins each, while DBO and SMA consistently failed to achieve any significant advantage. This reinforces the robustness and dominance of EDBO across diverse experimental conditions.

7. SIMULATION RESULTS WITH REAL APPLICATION

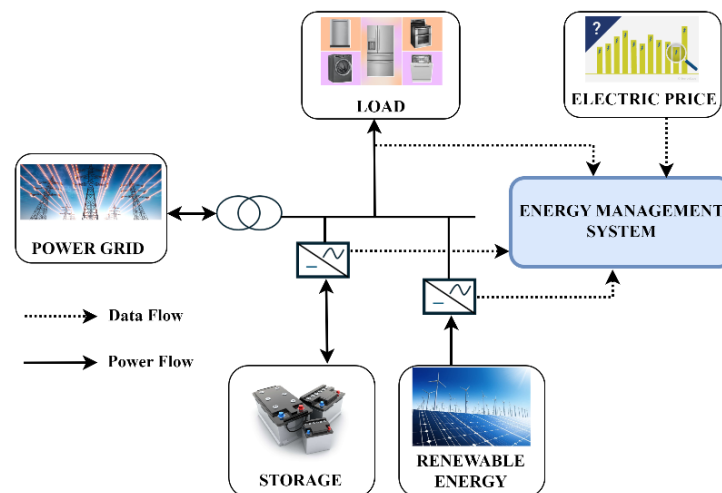


Figure 4. Home power grid with integrated renewable energy and storage.

In this section, we assess and contrast the effectiveness of the proposed EDBO algorithm in solving the optimization problem of operating costs in a home power grid with integrated solar energy and battery. A home power grid analyzed in this application is composed of four parts: main grid, renewable energy, energy storage and load. Its structure is shown in Fig. 4. A similar home energy management structure integrating solar generation and battery storage has been adopted in previous studies, such as those in [5], [24]. An energy management system (EMS) manages the power source from the utility grid and solar panels to meet load demand. To make use of the free renewable energy, a battery is invested with its capacity that can fully satisfy the load demand at times when solar energy is low or absent, for instance, at night or on rainy days. The battery

will store the surplus of solar energy and discharge to supply power to loads, which depends on the EMS's control data to ensure the operating cost at least. Solar power is a priority to use more than grid or battery power. The surplus of solar energy can be used to charge batteries or sold through the grid. The battery can be charged with solar power or grid power. The loads can be powered by batteries, solar, or the grid. Charging and discharging of batteries are not permitted to occur at the same time. Based on the selling price, buying price, load demand, power obtained from the solar panel, and stored power capacity, EMS defines the operations of the components in the system.

7.1. Battery Model

In this study, Lithium-ion is utilized as battery storage, which has charging and discharging principles described as follows [25]:

$$S_{stor}^t = S_{stor}^{t-1} (1 - \delta) + \Delta TP_{ch}^t \eta_{ch} / E_{tstor} - \Delta TP_{dis}^t / (\eta_{dis} E_{tstor}) \quad (15)$$

where battery capacity is represented by S_{stor}^t ; δ represents the battery's self-discharge efficiency; P_{ch}^t and P_{dis}^t represent the battery's charging and discharging energies, respectively; η_{ch} and η_{dis} represent the battery's charging and discharging efficiency, respectively; E_{tstor} represents the battery's maximum capacity.

7.2. Control Objectives And Constraints

The study aims to achieve the lowest weekly system operating cost by minimizing electricity transactions and battery storage depreciation costs. Following is the objective function:

$$\min C = \sum_{t=1}^{168} \left(\frac{e_{buy}^t + e_{sell}^t}{2} P_G^t + \frac{e_{buy}^t - e_{sell}^t}{2} |P_G^t| + C_{stor} P_{stor}^t \right) \quad (16)$$

where consider e_{buy}^t and e_{sell}^t to be, respectively, the price of buying and selling electricity; P_G^t represents the power transaction between the system and the utility grid (where the value of the electricity purchased is positive and the value of the electricity sold is negative; P_{stor}^t represents the battery storage's electrical power with charging power being negative and discharging power being positive; let C_{stor} represent the price of battery storage depreciation.

Constraint condition of load balance:

$$P_G^t + P_{stor}^t + P_{PV}^t = P_{load}^t \quad (17)$$

where $P_G^t, P_{stor}^t, P_{PV}^t$, and P_{load}^t represent the grid energy, battery energy, solar energy, and load energy, respectively at the t current time.

Constraint condition of the lithium-ion battery storage:

System state $x = (x_1, x_2, \dots, x_i, \dots, x_{168})$, x_i indicates the charging or discharging energy of the battery at the i^{th} hour. Let $power_init$ represent the battery's initial energy while $C_{ratedbat}$, bat_max , bat_min represent its rated power, maximum energy, and minimum energy, respectively:

$$\begin{aligned} bat_min \leq S_i = power_init + \sum_{t=1}^i x_t \leq bat_max, 1 \leq i \leq n \\ -C_{ratedbat} \leq x_i \leq C_{ratedbat} \end{aligned} \quad (18)$$

Violation of constraint condition results in penalty, calculated as power difference exceeding battery's maximum or minimum capacity. The constraint condition is considered violated when charging over the battery's maximum capacity and discharging below its minimum capacity.

7.3. Basic Data For Simulation

We simulate the proposed EDBO solution when it comes to solving the operating cost optimization problem in a smart home power grid with the data and parameters described as follows:

- The Lithium-ion battery has a maximum energy capacity of 12 kWh, with an initial state of charge of 6 kWh. Its rated power is 3 kW, while the maximum allowable charging capacity is 10 kWh and the maximum discharging capacity is 2 kWh. The charging and discharging processes operate with an efficiency of 0.95. In addition, the battery incurs a depreciation cost of 0.15 €/kWh per cycle [26].
- The load power [27] and solar power [28], which are referred to in the National Renewable Energy Laboratory (NREL), are presented in Figs. 5 and. 6. We collected data on load power and solar power in 168 hours during the first week of July 2019 in San Francisco. Then, they are taken on average over a 24-hour period. Furthermore, solar energy was scaled up by a factor of three to ensure conditions to fully check the different scenarios, including instances where solar energy exceeds the total load demand and surplus can be charged for batteries or sold to the main grid, or situations where solar energy is not enough to satisfy load demand and the system needs to buy power from the main grid.
- The buying price, as illustrated in Fig. 7, is sourced from the U.S. Bureau of Labor Statistics [29], while the selling price is set at 90% of the buying price.
- The simulation parameters for algorithms are shown in Table 1.

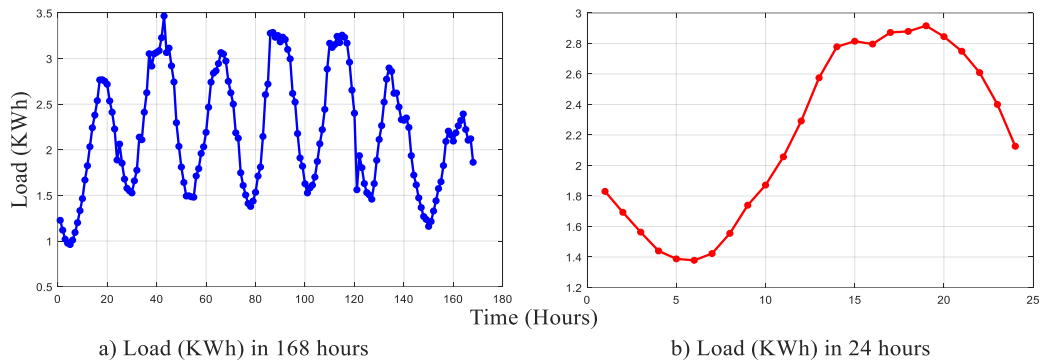


Figure 5. Load demand in 168 hours (a) and average load demand in 24 hours (b).

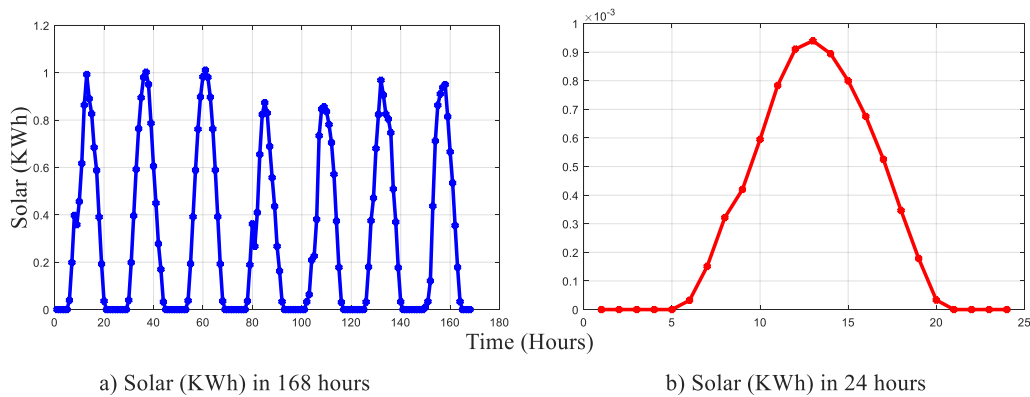


Figure 6. Solar power in 168 hours (a) and average solar power in 24 hours (b).

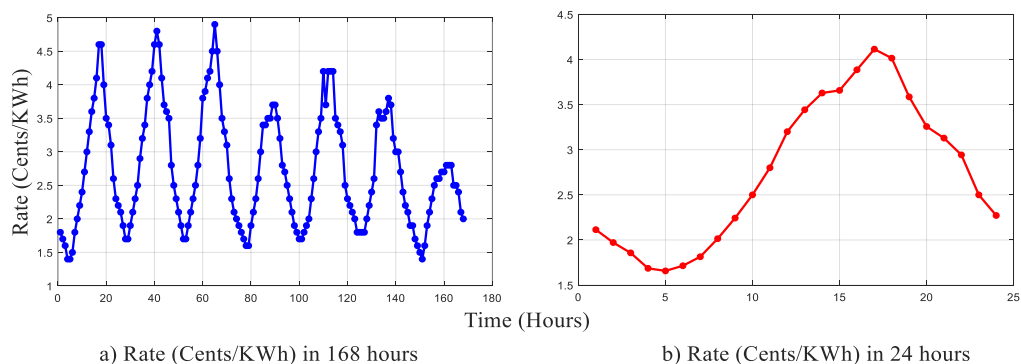


Figure 7. The electricity rate for buying power from the grid.

7.4. Results And Discussion

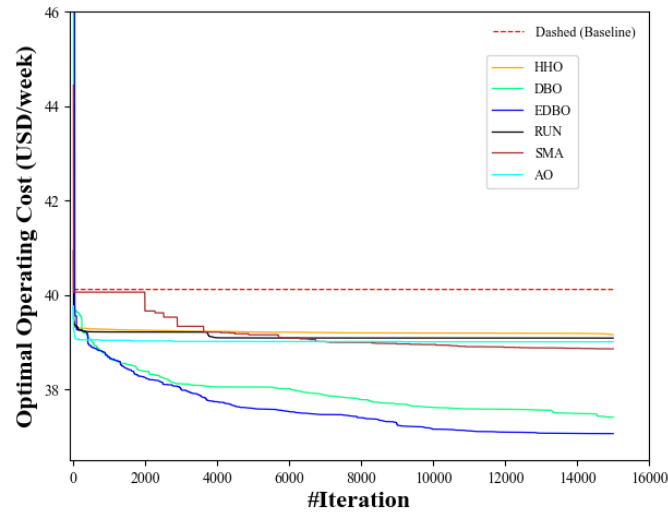


Figure 8. Operating costs compared with other algorithms.

Table 7. The algorithms’ efficiency in minimizing the smart home operating cost.

Algorithms	HHO		DBO		RUN		SMA		AO		Proposed EDBO	
	Cost (USD/ week)	Rate (%)	Cost (USD/ week)	Rate (%)	Cost (USD/ week)	Rate (%)	Cost (USD/ week)	Rate (%)	Cost (USD/ week)	Rate (%)	Cost (USD/ week)	Rate (%)
Baseline – without storage.	40.012 USD/week											
With storage and optimization algorithm.	39.162	2.12	37.413	6.50	39.090	2.30	38.861	2.88	39.011	2.50	36.981	7.58

This study will also investigate the effectiveness of integrating storage in the solar-connected home power grid. If storage remains unused, the system will buy power from the main grid when solar power falls short of meeting the load demand, and surplus solar energy will be sold when it exceeds the load demand. Using the above-mentioned load demand and solar power data, the total cost under this scenario, referred to as the baseline for subsequent analysis, is approximately 40.012 USD per week. With storage integrated into the system, the objective is to ensure that the total operating cost remains below the baseline value.

The simulation results are shown in Fig. 8 and Table 7. The applied algorithms, including the proposed EDBO, DBO, HHO, RUN, SMA and AO, are executed with a population size of 35, maximum number of iterations of 15000, and random-generated initial population. Each algorithm was run 10 times for every experiment and the values reported in Table 7 are the averages across these runs to reduce the effect of random initialization. Their efficiency will be evaluated and compared with the baseline case in terms of the operating cost. The proposed EDBO attains the best performance with an average operating cost of 36.981 USD/week, corresponding to a 7.58% reduction (≈ 3.03 USD/week savings) relative to the baseline. These results demonstrate that the EDBO not only finds the best (lowest) average cost among the applied algorithms but also provides the largest economic benefit from integrating battery storage under the simulated conditions.

8. CONCLUSIONS

In this study, the Nelder-Mead simplex algorithm and “seven kills” strategy of the HHO are merged into the DBO’s process to improve its convergence speed and insufficient accuracy. The proposed EDBO was evaluated on 30 CEC-2017 benchmark functions, and compared with DBO, HHO, and three recent meta-heuristic algorithms (RUN, SMA, and AO). The results, supported by overall ranking and Friedman–Nemenyi statistical analyses, show that EDBO achieves strong performance in terms of solution quality, robustness, and convergence behavior. In addition, the applicability of EDBO was demonstrated through a real-world case study on operating cost optimization in a solar-connected residential power system, where the proposed method obtained favorable solutions compared with the competing algorithms. Overall, the results indicate that the proposed EDBO provides a well-balanced exploration–exploitation mechanism and improved convergence

reliability, making it a competitive alternative for solving both benchmark and real-world optimization problems.

ACKNOWLEDGMENTS

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

REFERENCES

- [1] E. H. Houssein, M. K. Saeed, G. Hu, and M. M. Al-Sayed, "Metaheuristics for Solving Global and Engineering Optimization Problems: Review, Applications, Open Issues and Challenges," *Archives of Computational Methods in Engineering*, vol. 31, no. 8, pp. 4485–4519, 2024, doi: 10.1007/s11831-024-10168-6.
- [2] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019, doi: <https://doi.org/10.1016/j.future.2019.02.028>.
- [3] J.-K. Xue and B. Shen, "Dung beetle optimizer: a new meta-heuristic algorithm for global optimization," *J Supercomput*, vol. 79, Nov. 2022, doi: 10.1007/s11227-022-04959-6.
- [4] A. Das, R. Rai, B. Sasmal, K. G. Dhal, R. A. Khurma, and R. Saha, "Metaheuristic Algorithms Since 2020: Development, Taxonomy, Analysis, and Applications," *Archives of Computational Methods in Engineering*, 2025, doi: 10.1007/s11831-025-10408-3.
- [5] H. T. Vy and H. P. H. Anh, "Application of Dung Beetle Optimizer Algorithm for Solving Optimization Problem in Smart Residential Power Grid," in *Computational Intelligence Methods for Green Technology and Sustainable Development*, Y.-P. Huang, W.-J. Wang, H.-G. Le, and A.-Q. Hoang, Eds., Cham: Springer Nature Switzerland, 2024, pp. 148–159.
- [6] X. Liu, J.-S. Wang, S.-B. Zhang, X.-Y. Guan, and Y.-Z. Gao, "Optimization scheduling of off-grid hybrid renewable energy systems based on dung beetle optimizer with convergence factor and mathematical spiral," *Renew Energy*, vol. 237, p. 121874, 2024, doi: <https://doi.org/10.1016/j.renene.2024.121874>.
- [7] R. Quan, Z. Qiu, H. Wan, Z. Yang, and X. Li, "Dung beetle optimization algorithm-based hybrid deep learning model for ultra-short-term PV power prediction," *iScience*, vol. 27, no. 11, p. 111126, 2024, doi: <https://doi.org/10.1016/j.isci.2024.111126>.
- [8] Yue, Y., Ren, H., Liu, D., & Zhang, L. "Optimal Scheduling of Microgrids Based on an Improved Dung Beetle Optimization Algorithm," *Applied Sciences*, vol. 15, no. 2, p. 975, 2025, doi: <https://doi.org/10.3390/app15020975>.
- [9] Y. Chen, W. Ning, and W. Du, "A study of scheduling strategies for microgrids based on the non-dominated sorting dung beetle optimization algorithm," *Sci Rep*, vol. 15, Dec. 2025, doi: 10.1038/s41598-025-02446-5.
- [10] W. Gu and F. Wang, "A multi-strategy improved dung beetle optimisation algorithm and its application," *Cluster Comput*, vol. 28, no. 1, p. 49, 2024, doi: 10.1007/s10586-024-04704-z.
- [11] W. Tang, H. Qin, and S. Kang, "Dung beetle optimizer based on mean fitness distance balance and multi-strategy fusion for solving practical engineering problems," *Sci Rep*, vol. 15, Dec. 2025, doi: 10.1038/s41598-025-02937-5.
- [12] R. Fang, T. Zhou, B. Yu, Z. Li, L. Ma, and Y. Zhang, "Dung Beetle Optimization Algorithm Based on Improved Multi-Strategy Fusion," *Electronics (Basel)*, vol. 14, p. 197, Dec. 2025, doi: 10.3390/electronics14010197.
- [13] Y. Zhang, T. Li, T. Ma, D. Yang, and X. Sun, "Short-Term Photovoltaic Power Prediction Based on Extreme Learning Machine with Improved Dung Beetle Optimization Algorithm," *Energies (Basel)*, vol. 17, no. 4, p. 960, 2024, doi: <https://doi.org/10.3390/en17040960>.
- [14] F. Zhu, G. Li, H. Tang, Y. Li, X. Lv, and X. Wang, "Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems," *Expert Syst Appl*, vol. 236, p. 121219, 2024, doi: <https://doi.org/10.1016/j.eswa.2023.121219>.
- [15] J. E. Dennis and W. J. E., "Optimization on Microcomputers: The Nelder-Mead Simplex Algorithm," 1985. [Online]. Available: <https://api.semanticscholar.org/CorpusID:61018648>
- [16] F. Gao and L. Han, "Implementing the Nelder-Mead simplex algorithm with adaptive parameters," *Comput Optim Appl*, vol. 51, no. 1, pp. 259–277, 2012, doi: 10.1007/s10589-010-9329-3.
- [17] S. Takenaga, S. Watanabe, M. Nomura, Y. Ozaki, M. Onishi, and H. Habe, "Evaluating Initialization of Nelder-Mead Method for Hyperparameter Optimization in Deep Learning," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 3372–3379. doi: 10.1109/ICPR48806.2021.9412240.
- [18] S. Wessing, "Proper initialization is crucial for the Nelder–Mead simplex search," *Optim Lett*, vol. 13, no. 4, pp. 847–856, 2019, doi: 10.1007/s11590-018-1284-4.
- [19] S. Takenaga, Y. Ozaki, and M. Onishi, "Practical initialization of the Nelder–Mead method for computationally expensive optimization problems," *Optim Lett*, vol. 17, no. 2, pp. 283–297, 2023, doi: 10.1007/s11590-022-01953-y.
- [20] G. Wu, R. Mallipeddi, and P. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization," Dec. 2016.
- [21] T. Eftimov, P. Korošec, and B. Koroušić Seljak, "A Novel Approach to statistical comparison of meta-heuristic stochastic optimization algorithms using deep statistics," *Inf Sci (N Y)*, vol. 417, pp. 186–215, 2017, doi: <https://doi.org/10.1016/j.ins.2017.07.015>.

- [22] J. Liu and Y. Xu, "T-Friedman Test: A New Statistical Test for Multiple Comparison with an Adjustable Conservativeness Measure," *International Journal of Computational Intelligence Systems*, vol. 15, no. 1, p. 29, 2022, doi: 10.1007/s44196-022-00083-8.
- [23] J. Demsar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, Dec. 2006.
- [24] H. T. Vy and H. P. H. Anh, "Optimal Renewable Energy Management System Using Hybrid Evolutionary Neural Q-Learning Technique," *Journal of Control, Automation and Electrical Systems*, 2025, doi: 10.1007/s40313-025-01192-3.
- [25] X. Zhang, J. Jiang, H. Zheng, and J. Zhang, "Optimization Scheme of Integrated Community Energy Utilization System Based on Improved Sine-Cosine Algorithm," *Energy Engineering*, vol. 119, pp. 1117–1140, Dec. 2022, doi: 10.32604/ee.2022.017288.
- [26] PowerTechSystems, "Data for depreciation cost," [Online] <https://www.powertechsystems.eu/home/tech-corner/lithium-ion-vs-lead-acid-cost-analysis/>.
- [27] NREL, "Data of load demand from National Renewable Energy Laboratory (NREL), USA," [Online] Available at: <https://data.openei.org/submissions/153> [Accessed 2023].
- [28] NREL, "Data of solar energy from National Renewable Energy Laboratory (NREL), USA," [Online] Available at: <https://www.nrel.gov/grid/solar-resource/renewable-resource-data.html>.
- [29] U.S.Bureau of Labor Statistics, "Data for buying electricity price," [Online] https://www.bls.gov/regions/midwest/data/averageenergyprices_selectedareas_table.htm.

BIOGRAPHY OF AUTHORS



HUYNH TUYET VY completed her BS and MSc degrees in electronics and telecommunication engineering from Ho Chi Minh City University of Technology (HCMUT) in 2003 and 2007, respectively. She is currently a PhD student at the Faculty of Electronics Technology of the Industrial University of Ho Chi Minh City (IUH). She has previous job experience as a project manager. Renewable energy, artificial intelligence, evolutionary and meta-heuristic optimization techniques, and their applications in the planning, operation, and control of smart grids are among her research interests.



Ho Pham Huy Anh received the B.S. and the M.Sc. degrees in the Department of Electrical and Electronics Engineering from HCM City University of Technology in 1987 and in 1993, respectively. He received the Ph.D. degree from University of Ulsan, Korea in 2008. He is currently a Senior Lecturer in the Faculty of Electrical and Electronics (FEEE), HCM City University of Technology, VNU-HCM, HCM City, Viet Nam. Up to now, he co-authored 5 books and published over 160 papers on national and international journals and conference proceedings. His current research interests include intelligent control, robotics, renewable energy applications, modeling and identification of nonlinear dynamic system, deep machine-learning and soft-computing techniques.

APPENDIX

Average ranking of algorithms with dimensions=30, and agents=30.												
Func.	HHO		DBO		RUN		SMA		AO		EDBO	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
C1	1.002E+02	3	8.522E+02	6	1.097E+02	4	6.432E+02	5	1.000E+02	2	1.000E+02	1
C2	2.019E+02	3	3.801E+03	6	2.084E+02	4	1.067E+03	5	2.008E+02	2	2.000E+02	1
C3	3.009E+02	4	3.232E+02	6	3.001E+02	3	3.016E+02	5	3.000E+02	2	3.000E+02	1
C4	4.000E+02	1	4.028E+02	6	4.000E+02	4	4.001E+02	5	4.000E+02	2	4.000E+02	3

C5	5.000E+02	2	5.036E+02	6	5.000E+02	1	5.030E+02	5	5.000E+02	3	5.000E+02	4
C6	6.000E+02	4	6.000E+02	3	6.000E+02	2	6.000E+02	6	6.000E+02	5	6.000E+02	1
C7	7.000E+02	4	7.000E+02	5	7.000E+02	2	7.010E+02	6	7.000E+02	3	7.000E+02	1
C8	8.000E+02	3	8.131E+02	6	8.000E+02	2	8.000E+02	5	8.000E+02	4	8.000E+02	1
C9	9.000E+02	3	9.070E+02	6	9.000E+02	2	9.000E+02	5	9.000E+02	4	9.000E+02	1
C10	1.000E+03	3	1.057E+03	6	1.000E+03	2	1.000E+03	5	1.000E+03	4	1.000E+03	1
C11	1.100E+03	3	1.143E+03	6	1.100E+03	1	1.100E+03	5	1.100E+03	2	1.100E+03	4
C12	1.200E+03	3	1.201E+03	5	1.200E+03	2	1.201E+03	4	1.202E+03	6	1.200E+03	1
C13	1.300E+03	1	1.301E+03	5	1.300E+03	2	1.301E+03	6	1.300E+03	3	1.300E+03	4
C14	1.400E+03	3	1.400E+03	1	1.400E+03	2	1.400E+03	4	1.401E+03	6	1.400E+03	5
C15	1.500E+03	4	1.501E+03	5	1.500E+03	2	1.502E+03	6	1.500E+03	1	1.500E+03	3
C16	1.600E+03	3	1.607E+03	5	1.602E+03	2	1.603E+03	3	1.610E+03	6	1.603E+03	4
C17	1.711E+03	3	2.741E+03	6	1.723E+03	5	1.719E+03	4	1.700E+03	2	1.700E+03	1
C18	2.432E+03	4	4.980E+04	6	1.813E+03	3	2.762E+03	5	1.801E+03	2	1.801E+03	1
C19	1.908E+03	3	1.916E+03	6	1.901E+03	1	1.908E+03	5	1.908E+03	4	1.905E+03	2
C20	2.055E+03	5	2.062E+03	6	2.016E+03	4	2.010E+03	3	2.004E+03	2	2.003E+03	1
C21	2.106E+03	3	2.827E+03	6	2.108E+03	4	2.115E+03	5	2.106E+03	2	2.103E+03	1
C22	2.207E+03	5	2.221E+03	6	2.203E+03	2	2.204E+03	4	2.204E+03	3	2.202E+03	1
C23	2.302E+03	4	2.302E+03	3	2.302E+03	2	2.306E+03	6	2.302E+03	5	2.302E+03	1
C24	2.400E+03	3	2.508E+03	6	2.400E+03	2	2.402E+03	5	2.400E+03	4	2.400E+03	1
C25	2.500E+03	3	2.664E+03	6	2.500E+03	2	2.514E+03	5	2.500E+03	4	2.500E+03	1
C26	2.600E+03	4	2.600E+03	3	2.600E+03	2	2.605E+03	6	2.600E+03	5	2.600E+03	1
C27	2.705E+03	4	3.168E+03	6	2.705E+03	2	2.726E+03	5	2.705E+03	3	2.705E+03	1
C28	2.802E+03	3	4.140E+03	6	2.802E+03	1	2.859E+03	5	2.802E+03	2	2.802E+03	4
C29	2.987E+03	3	1.701E+04	6	3.141E+03	4	3.213E+03	5	2.987E+03	2	2.987E+03	1
C30	3.012E+03	3	8.645E+04	6	3.058E+03	4	3.102E+03	5	3.004E+03	2	3.004E+03	1
Average		3.17	5.37		2.50		4.93		3.23		1.80	

Average ranking of algorithms with dimensions=30, and agents=50.

Func.	HHO		DBO		RUN		SMA		AO		EDBO	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
C1	1.001E+02	3	6.822E+02	6	1.117E+02	4	2.004E+02	5	1.000E+02	2	1.000E+02	1
C2	2.007E+02	4	4.200E+02	6	2.007E+02	3	6.456E+02	5	2.000E+02	2	2.000E+02	1
C3	3.002E+02	4	3.018E+02	6	3.000E+02	2	3.003E+02	5	3.000E+02	3	3.000E+02	1
C4	4.000E+02	1	4.014E+02	6	4.000E+02	3	4.001E+02	5	4.000E+02	2	4.000E+02	4
C5	5.000E+02	1	5.047E+02	6	5.013E+02	4	5.031E+02	5	5.000E+02	2	5.000E+02	3
C6	6.000E+02	5	6.000E+02	3	6.000E+02	1	6.000E+02	6	6.000E+02	4	6.000E+02	2

C7	7.000E+0 2	4	7.000E+0 2	5	7.000E+0 2	2	7.010E+02	6	7.000E+0 2	3	7.000E+0 2	1
C8	8.000E+0 2	3	8.027E+0 2	6	8.000E+0 2	2	8.000E+02	5	8.000E+0 2	4	8.000E+0 2	1
C9	9.000E+0 2	3	9.129E+0 2	6	9.000E+0 2	2	9.000E+02	5	9.000E+0 2	4	9.000E+0 2	1
C10	1.000E+0 3	4	1.254E+0 3	6	1.000E+0 3	2	1.000E+03	5	1.000E+0 3	3	1.000E+0 3	1
C11	1.100E+0 3	3	1.100E+0 3	6	1.100E+0 3	2	1.100E+03	5	1.100E+0 3	4	1.100E+0 3	1
C12	1.200E+0 3	3	1.201E+0 3	6	1.200E+0 3	4	1.200E+03	5	1.200E+0 3	2	1.200E+0 3	1
C13	1.300E+0 3	1	1.301E+0 3	6	1.300E+0 3	3	1.301E+03	4	1.301E+0 3	5	1.300E+0 3	2
C14	1.400E+0 3	3	1.400E+0 3	1	1.400E+0 3	4	1.400E+03	2	1.401E+0 3	6	1.401E+0 3	5
C15	1.500E+0 3	1	1.504E+0 3	6	1.500E+0 3	4	1.501E+03	5	1.500E+0 3	2	1.500E+0 3	3
C16	1.600E+0 3	2	1.606E+0 3	5	1.602E+0 3	3	1.603E+03	4	1.610E+0 3	6	1.600E+0 3	1
C17	1.747E+0 3	5	2.755E+0 3	6	1.714E+0 3	3	1.714E+03	4	1.700E+0 3	2	1.700E+0 3	1
C18	1.897E+0 3	4	4.514E+0 4	6	1.813E+0 3	3	2.058E+03	5	1.802E+0 3	2	1.800E+0 3	1
C19	1.908E+0 3	3	1.910E+0 3	5	1.901E+0 3	1	1.914E+03	6	1.908E+0 3	4	1.901E+0 3	2
C20	2.042E+0 3	5	2.047E+0 3	6	2.029E+0 3	4	2.008E+03	3	2.004E+0 3	2	2.002E+0 3	1
C21	2.106E+0 3	4	2.680E+0 3	6	2.108E+0 3	5	2.106E+03	3	2.106E+0 3	2	2.102E+0 3	1
C22	2.207E+0 3	5	2.222E+0 3	6	2.204E+0 3	2	2.204E+03	4	2.204E+0 3	3	2.203E+0 3	1
C23	2.302E+0 3	5	2.302E+0 3	2	2.302E+0 3	3	2.306E+03	6	2.302E+0 3	4	2.302E+0 3	1
C24	2.400E+0 3	3	2.434E+0 3	6	2.400E+0 3	2	2.426E+03	5	2.400E+0 3	4	2.400E+0 3	1
C25	2.500E+0 3	3	2.591E+0 3	6	2.500E+0 3	2	2.509E+03	5	2.500E+0 3	4	2.500E+0 3	1
C26	2.600E+0 3	4	2.600E+0 3	2	2.600E+0 3	3	2.604E+03	6	2.600E+0 3	5	2.600E+0 3	1
C27	2.705E+0 3	4	3.065E+0 3	6	2.705E+0 3	2	2.708E+03	5	2.705E+0 3	3	2.705E+0 3	1
C28	2.802E+0 3	3	3.999E+0 3	6	2.802E+0 3	1	2.837E+03	5	2.802E+0 3	4	2.802E+0 3	2
C29	4.751E+0 3	5	1.519E+0 4	6	3.059E+0 3	3	4.386E+03	4	2.987E+0 3	2	2.987E+0 3	1
C30	3.006E+0 3	3	9.814E+0 4	6	3.091E+0 3	4	7.491E+04	5	3.004E+0 3	2	3.004E+0 3	1
Average		3.37		5.37		2.77		4.77		3.23		1.50

Average ranking of algorithms with dimensions=30, and agents=100.

Func.	HHO		DBO		RUN		SMA		AO		EDBO	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
C1	1.000E+0 2	2	1.403E+0 2	6	1.068E+0 2	4	1.084E+02	5	1.000E+0 2	3	1.000E+0 2	1
C2	2.001E+0 2	2	4.880E+0 3	6	2.001E+0 2	3	1.025E+03	5	2.002E+0 2	4	2.000E+0 2	1
C3	3.000E+0 2	5	3.003E+0 2	6	3.000E+0 2	2	3.000E+02	4	3.000E+0 2	3	3.000E+0 2	1
C4	4.000E+0 2	1	4.018E+0 2	6	4.000E+0 2	4	4.000E+02	5	4.000E+0 2	2	4.000E+0 2	3
C5	5.000E+0 2	2	5.048E+0 2	6	5.000E+0 2	1	5.027E+02	5	5.000E+0 2	3	5.000E+0 2	4
C6	6.000E+0 2	4	6.000E+0 2	1	6.000E+0 2	3	6.000E+02	6	6.000E+0 2	5	6.000E+0 2	2
C7	7.000E+0 2	4	7.000E+0 2	5	7.000E+0 2	2	7.008E+02	6	7.000E+0 2	3	7.000E+0 2	1
C8	8.000E+0 2	3	8.001E+0 2	6	8.000E+0 2	2	8.000E+02	5	8.000E+0 2	4	8.000E+0 2	1

C9	9.000E+0 2	3	9.010E+0 2	6	9.000E+0 2	2	9.000E+02	5	9.000E+0 2	4	9.000E+0 2	1
C10	1.000E+0 3	3	1.017E+0 3	6	1.000E+0 3	2	1.000E+03	5	1.000E+0 3	4	1.000E+0 3	1
C11	1.100E+0 3	3	1.100E+0 3	5	1.100E+0 3	2	1.100E+03	6	1.100E+0 3	4	1.100E+0 3	1
C12	1.200E+0 3	3	1.201E+0 3	6	1.200E+0 3	4	1.200E+03	5	1.200E+0 3	2	1.200E+0 3	1
C13	1.300E+0 3	3	1.301E+0 3	5	1.300E+0 3	2	1.301E+03	4	1.301E+0 3	6	1.300E+0 3	1
C14	1.400E+0 3	1	1.400E+0 3	3	1.400E+0 3	5	1.400E+03	2	1.401E+0 3	6	1.400E+0 3	4
C15	1.500E+0 3	1	1.502E+0 3	6	1.500E+0 3	4	1.501E+03	5	1.500E+0 3	2	1.500E+0 3	3
C16	1.600E+0 3	2	1.606E+0 3	6	1.602E+0 3	5	1.602E+03	4	1.600E+0 3	3	1.600E+0 3	1
C17	1.702E+0 3	3	3.175E+0 3	6	1.708E+0 3	5	1.705E+03	4	1.700E+0 3	2	1.700E+0 3	1
C18	1.834E+0 3	4	2.747E+0 4	6	1.811E+0 3	3	1.844E+03	5	1.801E+0 3	2	1.800E+0 3	1
C19	1.907E+0 3	3	1.911E+0 3	6	1.903E+0 3	2	1.911E+03	5	1.908E+0 3	4	1.903E+0 3	1
C20	2.029E+0 3	6	2.021E+0 3	4	2.022E+0 3	5	2.007E+03	3	2.004E+0 3	2	2.003E+0 3	1
C21	2.105E+0 3	3	2.759E+0 3	6	2.108E+0 3	5	2.102E+03	2	2.106E+0 3	4	2.102E+0 3	1
C22	2.206E+0 3	5	2.220E+0 3	6	2.202E+0 3	2	2.202E+03	1	2.204E+0 3	4	2.204E+0 3	3
C23	2.302E+0 3	4	2.302E+0 3	2	2.302E+0 3	3	2.304E+03	6	2.302E+0 3	5	2.302E+0 3	1
C24	2.400E+0 3	3	2.400E+0 3	5	2.400E+0 3	2	2.401E+03	6	2.400E+0 3	4	2.400E+0 3	1
C25	2.500E+0 3	3	2.535E+0 3	6	2.500E+0 3	2	2.501E+03	5	2.500E+0 3	4	2.500E+0 3	1
C26	2.600E+0 3	5	2.600E+0 3	2	2.600E+0 3	3	2.602E+03	6	2.600E+0 3	4	2.600E+0 3	1
C27	2.705E+0 3	3	3.102E+0 3	6	2.705E+0 3	2	2.707E+03	5	2.705E+0 3	4	2.705E+0 3	1
C28	2.802E+0 3	3	3.946E+0 3	6	2.802E+0 3	2	2.831E+03	5	2.802E+0 3	4	2.802E+0 3	1
C29	2.987E+0 3	2	2.019E+0 4	6	3.024E+0 3	4	3.384E+03	5	2.987E+0 3	3	2.987E+0 3	1
C30	3.006E+0 3	3	5.249E+0 4	6	3.066E+0 3	4	3.117E+03	5	3.004E+0 3	2	3.004E+0 3	1
Average		3.07		5.27		3.03		4.67		3.53		1.43

Average ranking of algorithms with dimensions=50, and agents=30.

Func.	HHO		DBO		RUN		SMA		AO		EDBO	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
C1	1.003E+0 2	2	8.806E+0 3	5	1.494E+0 3	4	1.325E+04	6	1.005E+0 2	3	1.000E+0 2	1
C2	2.037E+0 2	3	6.756E+0 3	5	2.000E+0 2	1	6.707E+04	6	2.005E+0 2	2	2.046E+0 2	4
C3	3.017E+0 2	4	4.207E+0 2	6	3.002E+0 2	3	3.128E+02	5	3.002E+0 2	2	3.000E+0 2	1
C4	4.000E+0 2	1	4.032E+0 2	5	4.000E+0 2	4	4.032E+02	6	4.000E+0 2	2	4.000E+0 2	3
C5	5.000E+0 2	2	5.036E+0 2	6	5.036E+0 2	5	5.035E+02	4	5.000E+0 2	1	5.000E+0 2	3
C6	6.000E+0 2	4	6.000E+0 2	5	6.000E+0 2	1	6.001E+02	6	6.000E+0 2	3	6.000E+0 2	2
C7	7.000E+0 2	2	7.000E+0 2	5	7.000E+0 2	4	7.011E+02	6	7.000E+0 2	3	7.000E+0 2	1
C8	8.000E+0 2	4	8.184E+0 2	6	8.000E+0 2	2	8.002E+02	5	8.000E+0 2	3	8.000E+0 2	1
C9	9.000E+0 2	4	9.094E+0 2	6	9.000E+0 2	1	9.001E+02	5	9.000E+0 2	3	9.000E+0 2	2
C10	1.000E+0 3	4	1.140E+0 3	6	1.000E+0 3	1	1.001E+03	5	1.000E+0 3	3	1.000E+0 3	2

C11	1.100E+0 3	3	1.184E+0 3	6	1.100E+0 3	1	1.103E+03	5	1.100E+0 3	4	1.100E+0 3	2
C12	1.200E+0 3	3	1.201E+0 3	5	1.200E+0 3	4	1.203E+03	6	1.200E+0 3	1	1.200E+0 3	2
C13	1.300E+0 3	1	1.301E+0 3	5	1.300E+0 3	3	1.301E+03	4	1.301E+0 3	6	1.300E+0 3	2
C14	1.400E+0 3	2	1.400E+0 3	1	1.400E+0 3	4	1.400E+03	3	1.401E+0 3	6	1.401E+0 3	5
C15	1.500E+0 3	2	1.506E+0 3	6	1.500E+0 3	4	1.503E+03	5	1.500E+0 3	1	1.500E+0 3	3
C16	1.600E+0 3	1	1.612E+0 3	6	1.604E+0 3	4	1.604E+03	5	1.600E+0 3	2	1.600E+0 3	3
C17	1.701E+0 3	3	6.284E+0 3	6	1.735E+0 3	4	2.501E+03	5	1.701E+0 3	2	1.700E+0 3	1
C18	1.972E+0 3	4	1.101E+0 5	6	1.822E+0 3	3	1.096E+04	5	1.801E+0 3	2	1.801E+0 3	1
C19	1.914E+0 3	4	1.925E+0 3	6	1.912E+0 3	1	1.921E+03	5	1.914E+0 3	3	1.912E+0 3	2
C20	2.105E+0 3	5	2.206E+0 3	6	2.050E+0 3	4	2.035E+03	3	2.007E+0 3	2	2.005E+0 3	1
C21	2.110E+0 3	3	3.131E+0 3	6	2.212E+0 3	4	2.313E+03	5	2.110E+0 3	2	2.107E+0 3	1
C22	2.211E+0 3	5	2.225E+0 3	6	2.205E+0 3	1	2.209E+03	4	2.206E+0 3	2	2.206E+0 3	3
C23	2.303E+0 3	4	2.354E+0 3	6	2.303E+0 3	1	2.328E+03	5	2.303E+0 3	3	2.303E+0 3	2
C24	2.400E+0 3	4	2.592E+0 3	6	2.400E+0 3	1	2.411E+03	5	2.400E+0 3	3	2.400E+0 3	2
C25	2.500E+0 3	4	2.709E+0 3	6	2.500E+0 3	2	2.558E+03	5	2.500E+0 3	3	2.500E+0 3	1
C26	2.600E+0 3	4	2.604E+0 3	5	2.600E+0 3	1	2.621E+03	6	2.600E+0 3	3	2.600E+0 3	2
C27	2.705E+0 3	3	3.337E+0 3	6	2.705E+0 3	1	2.824E+03	5	2.705E+0 3	2	2.706E+0 3	4
C28	2.803E+0 3	2	5.703E+0 3	6	2.803E+0 3	1	2.906E+03	5	2.803E+0 3	3	2.804E+0 3	4
C29	2.989E+0 3	3	2.486E+0 4	6	3.505E+0 3	4	8.121E+03	5	2.989E+0 3	2	2.988E+0 3	1
C30	3.007E+0 3	3	1.526E+0 5	6	3.177E+0 3	4	3.241E+03	5	3.007E+0 3	2	3.006E+0 3	1
Average		3.10		5.57		2.60		5.00		2.63		2.10

Average ranking of algorithms with dimensions=50, and agents=50.

Func.	HHO		DBO		RUN		SMA		AO		EDBO	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
C1	1.002E+0 2	2	2.125E+0 3	6	1.767E+0 2	4	1.643E+03	5	1.006E+0 2	3	1.000E+0 2	1
C2	2.009E+0 2	3	8.292E+0 3	5	2.000E+0 2	2	1.935E+05	6	2.111E+0 2	4	2.000E+0 2	1
C3	3.002E+0 2	4	3.988E+0 2	6	3.000E+0 2	3	3.038E+02	5	3.000E+0 2	2	3.000E+0 2	1
C4	4.000E+0 2	1	4.024E+0 2	6	4.000E+0 2	3	4.019E+02	5	4.000E+0 2	2	4.000E+0 2	4
C5	5.000E+0 2	1	5.053E+0 2	6	5.035E+0 2	5	5.035E+02	4	5.000E+0 2	2	5.000E+0 2	3
C6	6.000E+0 2	3	6.000E+0 2	5	6.000E+0 2	2	6.000E+02	6	6.000E+0 2	4	6.000E+0 2	1
C7	7.000E+0 2	3	7.000E+0 2	5	7.000E+0 2	2	7.011E+02	6	7.000E+0 2	4	7.000E+0 2	1
C8	8.000E+0 2	4	8.026E+0 2	6	8.000E+0 2	2	8.001E+02	5	8.000E+0 2	3	8.000E+0 2	1
C9	9.000E+0 2	3	9.102E+0 2	6	9.000E+0 2	2	9.002E+02	5	9.000E+0 2	4	9.000E+0 2	1
C10	1.000E+0 3	3	1.040E+0 3	6	1.000E+0 3	2	1.002E+03	5	1.000E+0 3	4	1.000E+0 3	1
C11	1.100E+0 3	3	1.856E+0 3	6	1.100E+0 3	2	1.122E+03	5	1.100E+0 3	4	1.100E+0 3	1
C12	1.200E+0 3	3	1.201E+0 3	5	1.201E+0 3	4	1.201E+03	6	1.200E+0 3	1	1.200E+0 3	2

C13	1.300E+0 3	1	1.301E+0 3	4	1.300E+0 3	3	1.301E+03	6	1.301E+0 3	5	1.300E+0 3	2
C14	1.400E+0 3	3	1.400E+0 3	1	1.400E+0 3	5	1.400E+03	2	1.401E+0 3	6	1.400E+0 3	4
C15	1.500E+0 3	1	1.506E+0 3	6	1.500E+0 3	4	1.502E+03	5	1.500E+0 3	2	1.500E+0 3	3
C16	1.600E+0 3	2	1.611E+0 3	6	1.604E+0 3	4	1.605E+03	5	1.600E+0 3	3	1.600E+0 3	1
C17	1.701E+0 3	3	4.815E+0 3	6	1.718E+0 3	4	1.909E+03	5	1.700E+0 3	2	1.700E+0 3	1
C18	1.816E+0 3	3	9.553E+0 4	6	1.818E+0 3	4	1.153E+04	5	1.802E+0 3	2	1.801E+0 3	1
C19	1.914E+0 3	3	1.924E+0 3	5	1.912E+0 3	2	1.987E+03	6	1.914E+0 3	4	1.911E+0 3	1
C20	2.056E+0 3	5	2.117E+0 3	6	2.055E+0 3	4	2.039E+03	3	2.006E+0 3	2	2.005E+0 3	1
C21	2.110E+0 3	3	3.429E+0 3	6	2.160E+0 3	4	2.257E+03	5	2.110E+0 3	2	2.106E+0 3	1
C22	2.212E+0 3	5	2.224E+0 3	6	2.205E+0 3	2	2.211E+03	4	2.206E+0 3	3	2.204E+0 3	1
C23	2.303E+0 3	3	2.354E+0 3	6	2.303E+0 3	2	2.321E+03	5	2.303E+0 3	4	2.303E+0 3	1
C24	2.400E+0 3	4	2.490E+0 3	6	2.400E+0 3	2	2.422E+03	5	2.400E+0 3	3	2.400E+0 3	1
C25	2.500E+0 3	3	2.686E+0 3	6	2.500E+0 3	2	2.538E+03	5	2.500E+0 3	4	2.500E+0 3	1
C26	2.600E+0 3	4	2.603E+0 3	5	2.600E+0 3	2	2.623E+03	6	2.600E+0 3	3	2.600E+0 3	1
C27	2.705E+0 3	3	3.345E+0 3	6	2.705E+0 3	1	2.782E+03	5	2.705E+0 3	2	2.706E+0 3	4
C28	2.803E+0 3	2	5.786E+0 3	6	2.803E+0 3	1	2.874E+03	5	2.803E+0 3	3	2.804E+0 3	4
C29	2.988E+0 3	2	2.312E+0 4	6	3.281E+0 3	4	5.066E+03	5	2.989E+0 3	3	2.988E+0 3	1
C30	3.011E+0 3	3	6.876E+0 4	6	3.147E+0 3	4	3.181E+03	5	3.007E+0 3	2	3.006E+0 3	1
Average		2.87		5.57		2.90		5.00		3.07		1.60

Average ranking of algorithms with dimensions=50, and agents=100.

Func.	HHO		DBO		RUN		SMA		AO		EDBO	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
C1	1.000E+0 2	3	1.442E+0 3	5	1.085E+0 2	4	4.695E+03	6	1.000E+0 2	2	1.000E+0 2	1
C2	2.002E+0 2	2	4.317E+0 3	5	2.014E+0 2	4	5.162E+04	6	2.014E+0 2	3	2.000E+0 2	1
C3	3.000E+0 2	3	3.020E+0 2	6	3.000E+0 2	2	3.011E+02	5	3.000E+0 2	4	3.000E+0 2	1
C4	4.000E+0 2	2	4.044E+0 2	6	4.000E+0 2	4	4.008E+02	5	4.000E+0 2	1	4.000E+0 2	3
C5	5.000E+0 2	1	5.050E+0 2	6	5.010E+0 2	4	5.034E+02	5	5.000E+0 2	2	5.000E+0 2	3
C6	6.000E+0 2	4	6.000E+0 2	5	6.000E+0 2	2	6.000E+02	6	6.000E+0 2	3	6.000E+0 2	1
C7	7.000E+0 2	4	7.000E+0 2	5	7.000E+0 2	2	7.011E+02	6	7.000E+0 2	3	7.000E+0 2	1
C8	8.000E+0 2	3	8.003E+0 2	6	8.000E+0 2	2	8.001E+02	5	8.000E+0 2	4	8.000E+0 2	1
C9	9.001E+0 2	4	9.027E+0 2	6	9.000E+0 2	2	9.001E+02	5	9.000E+0 2	3	9.000E+0 2	1
C10	1.000E+0 3	3	1.010E+0 3	6	1.000E+0 3	2	1.002E+03	5	1.000E+0 3	4	1.000E+0 3	1
C11	1.100E+0 3	3	1.100E+0 3	5	1.100E+0 3	2	1.102E+03	6	1.100E+0 3	4	1.100E+0 3	1
C12	1.200E+0 3	3	1.201E+0 3	5	1.200E+0 3	4	1.201E+03	6	1.200E+0 3	2	1.200E+0 3	1
C13	1.300E+0 3	1	1.301E+0 3	4	1.300E+0 3	3	1.301E+03	5	1.301E+0 3	6	1.300E+0 3	2
C14	1.400E+0 3	3	1.400E+0 3	4	1.400E+0 3	5	1.400E+03	1	1.401E+0 3	6	1.400E+0 3	2

C15	1.500E+0 3	1	1.508E+0 3	6	1.500E+0 3	4	1.501E+03	5	1.500E+0 3	2	1.500E+0 3	3
C16	1.600E+0 3	2	1.612E+0 3	6	1.604E+0 3	4	1.604E+03	5	1.600E+0 3	3	1.600E+0 3	1
C17	1.703E+0 3	3	5.062E+0 3	6	1.717E+0 3	4	1.952E+03	5	1.700E+0 3	2	1.700E+0 3	1
C18	1.804E+0 3	3	7.947E+0 4	6	1.817E+0 3	4	1.742E+04	5	1.801E+0 3	1	1.801E+0 3	2
C19	1.913E+0 3	3	1.922E+0 3	6	1.902E+0 3	1	1.921E+03	5	1.914E+0 3	4	1.912E+0 3	2
C20	2.056E+0 3	5	2.072E+0 3	6	2.047E+0 3	4	2.019E+03	3	2.006E+0 3	2	2.003E+0 3	1
C21	2.110E+0 3	3	3.247E+0 3	6	2.153E+0 3	5	2.126E+03	4	2.110E+0 3	2	2.106E+0 3	1
C22	2.208E+0 3	5	2.223E+0 3	6	2.205E+0 3	2	2.207E+03	4	2.206E+0 3	3	2.205E+0 3	1
C23	2.303E+0 3	4	2.358E+0 3	6	2.303E+0 3	2	2.320E+03	5	2.303E+0 3	3	2.303E+0 3	1
C24	2.400E+0 3	3	2.494E+0 3	6	2.400E+0 3	2	2.411E+03	5	2.400E+0 3	4	2.400E+0 3	1
C25	2.500E+0 3	3	2.587E+0 3	6	2.500E+0 3	2	2.511E+03	5	2.500E+0 3	4	2.500E+0 3	1
C26	2.600E+0 3	3	2.600E+0 3	4	2.600E+0 3	2	2.618E+03	6	2.600E+0 3	5	2.600E+0 3	1
C27	2.705E+0 3	4	3.337E+0 3	6	2.705E+0 3	1	2.749E+03	5	2.705E+0 3	2	2.705E+0 3	3
C28	2.803E+0 3	2	5.287E+0 3	6	2.803E+0 3	1	2.840E+03	5	2.803E+0 3	3	2.804E+0 3	4
C29	2.988E+0 3	2	2.958E+0 4	6	3.058E+0 3	4	4.395E+03	5	2.988E+0 3	3	2.988E+0 3	1
C30	3.007E+0 3	3	2.378E+0 5	6	3.142E+0 3	5	3.101E+03	4	3.007E+0 3	2	3.006E+0 3	1
Average		2.93		5.60		2.97		4.93		3.07		1.50

Average ranking of algorithms with dimensions=100, and agents=30.

Func.	HHO		DBO		RUN		SMA		AO		EDBO	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
C1	1.009E+0 2	3	6.331E+0 4	6	1.863E+0 3	4	5.876E+04	5	1.003E+0 2	2	1.001E+0 2	1
C2	2.122E+0 2	2	5.213E+0 6	5	2.500E+0 2	4	8.525E+06	6	2.020E+0 2	1	2.318E+0 2	3
C3	3.015E+0 2	4	1.609E+0 3	6	3.008E+0 2	3	3.437E+02	5	3.000E+0 2	2	3.000E+0 2	1
C4	4.000E+0 2	2	4.173E+0 2	6	4.000E+0 2	3	4.114E+02	5	4.000E+0 2	1	4.001E+0 2	4
C5	5.000E+0 2	1	5.051E+0 2	6	5.036E+0 2	5	5.036E+02	4	5.000E+0 2	2	5.000E+0 2	3
C6	6.000E+0 2	2	6.000E+0 2	5	6.000E+0 2	1	6.002E+02	6	6.000E+0 2	4	6.000E+0 2	3
C7	7.000E+0 2	4	7.010E+0 2	5	7.000E+0 2	2	7.014E+02	6	7.000E+0 2	3	7.000E+0 2	1
C8	8.000E+0 2	4	8.201E+0 2	6	8.000E+0 2	1	8.047E+02	5	8.000E+0 2	2	8.000E+0 2	3
C9	9.000E+0 2	3	9.077E+0 2	6	9.000E+0 2	1	9.040E+02	5	9.000E+0 2	2	9.000E+0 2	4
C10	1.000E+0 3	3	1.390E+0 3	6	1.000E+0 3	1	1.066E+03	5	1.000E+0 3	4	1.000E+0 3	2
C11	1.100E+0 3	2	1.295E+0 3	6	1.100E+0 3	1	1.173E+03	5	1.100E+0 3	4	1.100E+0 3	3
C12	1.200E+0 3	3	1.203E+0 3	5	1.203E+0 3	4	1.206E+03	6	1.200E+0 3	1	1.200E+0 3	2
C13	1.300E+0 3	2	1.301E+0 3	4	1.301E+0 3	3	1.301E+03	5	1.301E+0 3	6	1.300E+0 3	1
C14	1.400E+0 3	2	1.400E+0 3	3	1.400E+0 3	4	1.400E+03	1	1.401E+0 3	6	1.400E+0 3	5
C15	1.500E+0 3	2	1.525E+0 3	6	1.500E+0 3	3	1.520E+03	5	1.500E+0 3	1	1.500E+0 3	4
C16	1.600E+0 3	1	1.627E+0 3	6	1.608E+0 3	4	1.617E+03	5	1.600E+0 3	2	1.601E+0 3	3

C17	1.701E+0 3	3	4.339E+0 4	6	1.852E+0 3	4	7.351E+03	5	1.700E+0 3	1	1.700E+0 3	2
C18	1.803E+0 3	2	2.890E+0 5	6	2.140E+0 3	4	7.867E+03	5	1.808E+0 3	3	1.802E+0 3	1
C19	1.929E+0 3	3	1.984E+0 3	6	1.934E+0 3	4	1.955E+03	5	1.929E+0 3	2	1.927E+0 3	1
C20	2.097E+0 3	3	2.694E+0 3	6	2.159E+0 3	4	2.380E+03	5	2.014E+0 3	2	2.012E+0 3	1
C21	2.120E+0 3	3	1.111E+0 4	6	2.217E+0 3	4	7.164E+03	5	2.120E+0 3	2	2.118E+0 3	1
C22	2.219E+0 3	4	2.253E+0 3	6	2.213E+0 3	3	2.219E+03	5	2.208E+0 3	1	2.210E+0 3	2
C23	2.306E+0 3	4	2.408E+0 3	6	2.306E+0 3	1	2.381E+03	5	2.306E+0 3	2	2.306E+0 3	3
C24	2.400E+0 3	4	2.854E+0 3	6	2.400E+0 3	1	2.522E+03	5	2.400E+0 3	2	2.400E+0 3	3
C25	2.500E+0 3	4	2.810E+0 3	6	2.500E+0 3	1	2.634E+03	5	2.500E+0 3	3	2.500E+0 3	2
C26	2.600E+0 3	3	2.694E+0 3	6	2.600E+0 3	1	2.683E+03	5	2.600E+0 3	4	2.600E+0 3	2
C27	2.705E+0 3	3	3.693E+0 3	6	2.705E+0 3	1	3.070E+03	5	2.705E+0 3	2	2.706E+0 3	4
C28	2.806E+0 3	2	9.631E+0 3	6	2.806E+0 3	1	3.089E+03	5	2.807E+0 3	3	2.807E+0 3	4
C29	2.993E+0 3	3	4.369E+0 4	5	5.171E+0 3	4	8.653E+04	6	2.992E+0 3	1	2.992E+0 3	2
C30	3.014E+0 3	3	7.722E+0 3	6	3.483E+0 3	4	3.879E+03	5	3.013E+0 3	1	3.013E+0 3	2
Average		2.80		5.67		2.70		5.00		2.40		2.43

Average ranking of algorithms with dimensions=100, and agents=50.

Func.	HHO		DBO		RUN		SMA		AO		EDBO	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
C1	1.003E+0 2	3	4.645E+0 4	5	4.995E+0 2	4	9.424E+04	6	1.000E+0 2	2	1.000E+0 2	1
C2	2.016E+0 2	2	1.239E+0 6	5	2.002E+0 2	1	5.972E+06	6	2.024E+0 2	3	2.052E+0 2	4
C3	3.002E+0 2	2	1.324E+0 3	6	3.003E+0 2	3	3.271E+02	5	3.004E+0 2	4	3.000E+0 2	1
C4	4.000E+0 2	2	4.160E+0 2	6	4.000E+0 2	3	4.129E+02	5	4.000E+0 2	1	4.000E+0 2	4
C5	5.000E+0 2	1	5.050E+0 2	6	5.017E+0 2	4	5.036E+02	5	5.000E+0 2	2	5.000E+0 2	3
C6	6.000E+0 2	3	6.000E+0 2	5	6.000E+0 2	1	6.002E+02	6	6.000E+0 2	2	6.000E+0 2	4
C7	7.000E+0 2	3	7.005E+0 2	5	7.000E+0 2	2	7.014E+02	6	7.000E+0 2	4	7.000E+0 2	1
C8	8.000E+0 2	2	8.069E+0 2	6	8.000E+0 2	1	8.061E+02	5	8.000E+0 2	3	8.000E+0 2	4
C9	9.000E+0 2	3	9.021E+0 2	5	9.000E+0 2	2	9.053E+02	6	9.000E+0 2	1	9.000E+0 2	4
C10	1.000E+0 3	2	1.816E+0 3	6	1.000E+0 3	1	1.084E+03	5	1.000E+0 3	4	1.000E+0 3	3
C11	1.100E+0 3	2	1.313E+0 3	6	1.100E+0 3	1	1.203E+03	5	1.100E+0 3	3	1.100E+0 3	4
C12	1.200E+0 3	3	1.202E+0 3	5	1.201E+0 3	4	1.204E+03	6	1.200E+0 3	2	1.200E+0 3	1
C13	1.300E+0 3	2	1.301E+0 3	4	1.300E+0 3	3	1.301E+03	5	1.301E+0 3	6	1.300E+0 3	1
C14	1.400E+0 3	1	1.400E+0 3	3	1.400E+0 3	4	1.400E+03	2	1.401E+0 3	6	1.400E+0 3	5
C15	1.500E+0 3	2	1.514E+0 3	5	1.500E+0 3	4	1.516E+03	6	1.500E+0 3	1	1.500E+0 3	3
C16	1.600E+0 3	1	1.628E+0 3	6	1.608E+0 3	4	1.611E+03	5	1.600E+0 3	2	1.600E+0 3	3
C17	1.700E+0 3	3	2.411E+0 4	6	1.802E+0 3	4	1.286E+04	5	1.700E+0 3	2	1.700E+0 3	1
C18	1.803E+0 3	3	2.661E+0 5	6	2.250E+0 3	4	1.839E+04	5	1.801E+0 3	1	1.802E+0 3	2

C19	1.929E+0 3	3	1.974E+0 3	6	1.936E+0 3	4	1.950E+03	5	1.929E+0 3	2	1.918E+0 3	1
C20	2.097E+0 3	3	2.612E+0 3	6	2.113E+0 3	4	2.347E+03	5	2.013E+0 3	2	2.012E+0 3	1
C21	2.120E+0 3	3	8.738E+0 3	6	2.141E+0 3	4	6.812E+03	5	2.120E+0 3	2	2.118E+0 3	1
C22	2.220E+0 3	4	2.238E+0 3	6	2.211E+0 3	3	2.228E+03	5	2.208E+0 3	1	2.210E+0 3	2
C23	2.306E+0 3	2	2.424E+0 3	6	2.306E+0 3	1	2.384E+03	5	2.306E+0 3	3	2.306E+0 3	4
C24	2.400E+0 3	3	2.615E+0 3	6	2.400E+0 3	1	2.523E+03	5	2.400E+0 3	2	2.400E+0 3	4
C25	2.500E+0 3	4	2.797E+0 3	6	2.500E+0 3	1	2.604E+03	5	2.500E+0 3	2	2.500E+0 3	3
C26	2.600E+0 3	3	2.668E+0 3	5	2.600E+0 3	1	2.675E+03	6	2.600E+0 3	2	2.600E+0 3	4
C27	2.705E+0 3	3	3.681E+0 3	6	2.705E+0 3	1	2.981E+03	5	2.705E+0 3	2	2.706E+0 3	4
C28	2.806E+0 3	3	9.150E+0 3	6	2.806E+0 3	1	3.021E+03	5	2.806E+0 3	2	2.807E+0 3	4
C29	2.992E+0 3	2	5.764E+0 4	5	3.961E+0 3	4	1.100E+05	6	2.993E+0 3	3	2.992E+0 3	1
C30	3.014E+0 3	3	5.382E+0 3	6	3.539E+0 3	4	4.328E+03	5	3.013E+0 3	1	3.013E+0 3	2
Average		2.53		5.53		2.63		5.20		2.43		2.67

Average ranking of algorithms with dimensions=100, and agents=100.

Func.	HHO		DBO		RUN		SMA		AO		EDBO	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
C1	1.001E+0 2	2	2.950E+0 4	5	3.082E+0 2	4	8.662E+04	6	1.003E+0 2	3	1.000E+0 2	1
C2	2.005E+0 2	2	9.168E+0 4	5	2.004E+0 2	1	4.516E+06	6	2.016E+0 2	3	2.019E+0 2	4
C3	3.001E+0 2	4	7.768E+0 2	6	3.001E+0 2	3	3.108E+02	5	3.000E+0 2	2	3.000E+0 2	1
C4	4.000E+0 2	1	4.147E+0 2	6	4.000E+0 2	3	4.123E+02	5	4.000E+0 2	2	4.000E+0 2	4
C5	5.000E+0 2	1	5.083E+0 2	6	5.035E+0 2	4	5.036E+02	5	5.000E+0 2	2	5.000E+0 2	3
C6	6.000E+0 2	2	6.000E+0 2	5	6.000E+0 2	1	6.002E+02	6	6.000E+0 2	4	6.000E+0 2	3
C7	7.000E+0 2	4	7.001E+0 2	5	7.000E+0 2	1	7.015E+02	6	7.000E+0 2	2	7.000E+0 2	3
C8	8.000E+0 2	3	8.063E+0 2	6	8.000E+0 2	1	8.047E+02	5	8.000E+0 2	2	8.000E+0 2	4
C9	9.000E+0 2	2	9.007E+0 2	5	9.000E+0 2	1	9.051E+02	6	9.000E+0 2	3	9.000E+0 2	4
C10	1.000E+0 3	3	1.183E+0 3	6	1.000E+0 3	1	1.072E+03	5	1.000E+0 3	2	1.000E+0 3	4
C11	1.100E+0 3	3	1.147E+0 3	5	1.100E+0 3	2	1.164E+03	6	1.100E+0 3	1	1.100E+0 3	4
C12	1.200E+0 3	3	1.201E+0 3	5	1.200E+0 3	4	1.203E+03	6	1.200E+0 3	1	1.200E+0 3	2
C13	1.300E+0 3	2	1.301E+0 3	5	1.300E+0 3	3	1.301E+03	4	1.301E+0 3	6	1.300E+0 3	1
C14	1.400E+0 3	4	1.400E+0 3	3	1.400E+0 3	5	1.400E+03	2	1.401E+0 3	6	1.400E+0 3	1
C15	1.500E+0 3	1	1.514E+0 3	6	1.500E+0 3	4	1.511E+03	5	1.500E+0 3	2	1.500E+0 3	3
C16	1.600E+0 3	1	1.625E+0 3	6	1.608E+0 3	4	1.612E+03	5	1.600E+0 3	2	1.600E+0 3	3
C17	1.701E+0 3	3	2.262E+0 4	6	1.770E+0 3	4	1.204E+04	5	1.700E+0 3	2	1.700E+0 3	1
C18	1.801E+0 3	3	2.934E+0 5	6	1.841E+0 3	4	2.351E+04	5	1.801E+0 3	1	1.801E+0 3	2
C19	1.929E+0 3	3	1.962E+0 3	6	1.925E+0 3	1	1.949E+03	5	1.929E+0 3	4	1.926E+0 3	2
C20	2.075E+0 3	4	2.370E+0 3	6	2.071E+0 3	3	2.125E+03	5	2.013E+0 3	2	2.012E+0 3	1

C21	2.120E+0 3	2	7.585E+0 3	5	2.189E+0 3	4	8.359E+03	6	2.120E+0 3	3	2.117E+0 3	1
C22	2.214E+0 3	4	2.232E+0 3	6	2.209E+0 3	2	2.225E+03	5	2.207E+0 3	1	2.209E+0 3	3
C23	2.306E+0 3	3	2.364E+0 3	5	2.306E+0 3	1	2.372E+03	6	2.306E+0 3	2	2.306E+0 3	4
C24	2.400E+0 3	3	2.511E+0 3	5	2.400E+0 3	1	2.548E+03	6	2.400E+0 3	2	2.400E+0 3	4
C25	2.500E+0 3	3	2.762E+0 3	6	2.500E+0 3	1	2.576E+03	5	2.500E+0 3	2	2.500E+0 3	4
C26	2.600E+0 3	2	2.646E+0 3	5	2.600E+0 3	1	2.675E+03	6	2.600E+0 3	4	2.600E+0 3	3
C27	2.705E+0 3	2	3.648E+0 3	6	2.705E+0 3	1	2.852E+03	5	2.705E+0 3	3	2.705E+0 3	4
C28	2.806E+0 3	3	7.083E+0 3	6	2.806E+0 3	2	2.950E+03	5	2.806E+0 3	1	2.807E+0 3	4
C29	2.992E+0 3	3	7.051E+0 4	5	3.616E+0 3	4	8.062E+04	6	2.992E+0 3	2	2.992E+0 3	1
C30	3.013E+0 3	2	6.284E+0 5	6	3.327E+0 3	4	5.065E+03	5	3.013E+0 3	3	3.013E+0 3	1
Average		2.60		5.47		2.50		5.27		2.50		2.67