EGRET SWARM OPTIMIZATION ALGORITHM FOR PRODUCTION OPTIMIZATION: A PERFORMANCE EVALUATION STUDY

ĐÁNH GIÁ HIỆU NĂNG CỦA THUẬT TOÁN EGRET SWARM OPTIMIZATION TRONG TỐI ƯU HÓA SẢN XUẤT

Pham Khac Son, Nguyen Thanh Long, Nguyen Cong Hanh*

The University of Danang - University of Science and Technology, Vietnam

*Corresponding author: nchanh@dut.udn.vn

(Received: May 22, 2026; Revised: July 11, 2026; Accepted: July 15, 2026)

DOI: 10.31130/ud-jst.2026.24(7A).361E

Abstract - This study evaluates the effectiveness of the Egret Swarm Optimization Algorithm (ESOA) by comparing it with Particle Swarm Optimization (PSO), Differential Evolution (DE), and Grey Wolf Optimizer (GWO) on the CEC-2017 benchmark suite. The results demonstrate that ESOA achieves high solution quality, strong stability, and reliable convergence. In addition, the algorithm was applied to a real-world resource allocation problem, using data collected from the Manufacturing Execution System (MES) of a manufacturing enterprise over a three-month period, incorporating nonlinear cost fluctuations and operational constraints. Experimental results show that ESOA maintains stable performance across multiple runs and exhibits efficient search capability in large-scale solution spaces. However, the study also reveals certain limitations related to the characteristics of the dataset and the number of iterations required to achieve convergence.

l**Key words -** Optimization; Egret Swarm Optimization Algorithm; Benchmark; Scheduling; Assignment Problem

Tóm tắt – Nghiên cứu này đánh giá hiệu quả của thuật toán Egret Swarm Optimization Algorithm (ESOA) thông qua so sánh với các thuật toán Particle Swarm Optimization (PSO), Differential Evolution (DE) và Grey Wolf Optimizer (GWO) trên bộ dữ liệu chuẩn CEC-2017. Kết quả cho thấy ESOA đạt chất lượng nghiệm tốt, độ ổn định cao và khả năng hội tụ đáng tin cậy. Bên cạnh đó, thuật toán được áp dụng vào bài toán thực tế về phân bổ nguồn lực sản xuất, sử dụng dữ liệu thu thập từ hệ thống Manufacturing Execution Systems (MES) của một doanh nghiệp sản xuất trong ba tháng, có tích hợp yếu tố biến động chi phí phi tuyến tính và các ràng buộc vận hành. Kết quả thực nghiệm cho thấy ESOA duy trì hiệu năng ổn định qua nhiều lần chạy và thể hiện khả năng tìm kiếm hiệu quả trong không gian nghiệm quy mô lớn. Tuy nhiên, nghiên cứu cũng chỉ ra một số hạn chế liên quan đến đặc tính của bộ dữ liệu và số lượng vòng lặp cần thiết để đạt trạng thái hội tụ.

Từ khóa – Tối ưu hóa; Thuật toán tối ưu hóa Egret; Benchmark; Lập lịch; Bài toán phân công.

1. Introduction

In modern manufacturing environments, operational plans are typically established in advance based on demand forecasts, equipment capacity, and delivery schedules. According to McKinsey & Company [1], optimizing production schedules can improve productivity by 10-30% while reducing production cycle time by 20-50%. However, such static plans can rapidly become infeasible when disruptions occur, including unexpected machine breakdowns, order changes, or deviations in process progress across production stages. Under these conditions, enterprises are required to promptly reallocate resources in order to maintain output levels, minimize delivery delays, and control operating costs. Accordingly, production scheduling and resource allocation are widely recognized as critical decision-making problems that strongly influence the overall efficiency of manufacturing systems [2-3].

Alongside this trend, the emergence of smart manufacturing has significantly transformed factory management practices. Many enterprises are progressively deploying Industrial Internet of Things (IIoT) platforms and Manufacturing Execution Systems (MES) to collect real-time data from machines, customer orders, and production processes [4-5]. IDC reported that the global volume of industrial data is growing at an annual rate exceeding 23%, while the majority of large-scale manufacturers have adopted digital solutions to varying extents [6]. Nevertheless, converting operational data into optimal decisions remains a substantial challenge. In practice, many production plans are still developed based on managerial experience or simplified heuristic rules, resulting in suboptimal performance and limited responsiveness to changing system conditions [7-9].

From a methodological perspective, conventional optimization techniques such as linear programming and dynamic programming provide strong theoretical foundations and can generate high-quality solutions in many deterministic settings [10-11]. However, when problem size increases or nonlinear and discrete constraints are introduced, the computational burden of these approaches may rise considerably [12]. As a result, metaheuristic algorithms have attracted growing attention due to their ability to identify near-optimal solutions within acceptable computational time. Popular approaches such as: Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Differential Evolution (DE), Grey Wolf Optimizer (GWO) have been widely applied to industrial optimization problems [13-17].

Recent studies have shown that metaheuristic methods can significantly enhance solution quality in production scheduling problems, particularly in job shop and flow shop environments [18-24]. However, these methods still suffer from several common limitations, including premature convergence, entrapment in local optima, and an insufficient balance between exploration and exploitation capabilities [25-28]. Such weaknesses become more pronounced in real manufacturing environments, where data are often dynamic and multiple constraints must be satisfied simultaneously.

Within the growing family of nature-inspired optimization techniques, Chen et al. proposed the Egret Swarm Optimization Algorithm (ESOA), which is inspired by the foraging behavior of egrets [29]. This algorithm integrates both local and global search mechanisms to improve convergence speed while reducing the risk of stagnation in local optima. Initial studies have demonstrated the potential of ESOA on benchmark functions and several engineering optimization tasks. However, existing research has primarily focused on algorithmic validation or parameter tuning, whereas applications of ESOA to production scheduling and resource allocation problems using real industrial data remain limited.

Motivated by this research gap, the present study investigates the practical applicability of ESOA to real-world production optimization problems. Specifically, two categories of data are considered: (i) the CEC-2017 benchmark suite, used to evaluate the search capability, robustness, and convergence behavior of ESOA relative to widely used metaheuristic algorithms; and (ii) real operational data extracted from the MES of a manufacturing enterprise over a three-month period, used to formulate a production resource allocation and operating cost minimization problem.

The primary objective of this study is to develop a data-driven decision support tool capable of rapidly identifying efficient production plans under practical operational constraints. The main contributions of this paper are as follows:

(1) To evaluate the performance of ESOA on the CEC-2017 benchmark suite using metrics related to solution quality, statistical robustness, and convergence behavior;

(2) To construct a production resource allocation optimization model based on real MES data, incorporating constraints related to machine capacity, working shifts, and operating costs;

(3) To validate the applicability of ESOA in reducing total production cost and supporting rapid rescheduling under changing operational conditions.

2. Overview of the ESOA

In this study, ESOA is employed as the optimization engine for the proposed resource allocation and production rescheduling problem. Since this work adopts the original algorithm without modification, only a brief overview of its main principles is presented, while the detailed mathematical formulations and implementation can be found in Chen et al. [29]. Furthermore, ESOA is compared with several representative metaheuristic algorithms using standard benchmark functions to validate its search performance before its application to the real-world production optimization problem.

ESOA is a population-based metaheuristic inspired by the hunting behaviors of the Great Egret and Snowy Egret. It combines the Sit-and-Wait strategy for local exploitation and the Aggressive strategy for global exploration, enabling an effective balance between solution refinement and search diversity. Candidate solutions are evaluated and selected after each iteration to preserve high-quality solutions while maintaining population diversity. Owing to these characteristics, ESOA is well suited for constrained nonlinear production optimization and is therefore adopted as the optimization engine in the proposed production scheduling framework.

3. Performance Validation on the CEC-2017

Table 1. *Summary of 29 functions of the CEC-2017 test suite*

Func Type	Number of Func	Func IDs	Fmin Range
Unimodal	3	F1 – F3	100 – 300
Multimodal	7	F4 – F10	400–1000
Hybrid	10	F11 – F20	1100–2000
Composition	9	F21 – F29	2100–2900

In this section, the ESOA is evaluated and compared with the PSO, DE, and GWO algorithms through the results of the CEC-2017 benchmark test. This test includes 29 optimization problems with the content summarized in Table 1.

Table 2. *Parameter settings*

Algorithm	Parameters
Common	D = 50, N = 50, T_max = 1000, Runs = 30, Range = [-100,100]
ESOA	β =1.5, w: 0.5-0 (linear), α: 2-0 (exp)
PSO	$C_1 = C_2 = 1.5$, w: 0.9-0.4 (linear)
DE	F = 0.8, CR = 0.9, DE/rand/1/bin
GWO	a: 2-0 (linear)

All algorithms were evaluated under identical conditions on the CEC-2017 benchmark suite (range [-100,100]): dimension D = 50, population size N = 50, T_max = 1,000 iterations (Max_FES = N × (T_max + 1) = 50,050) and 30 independent runs/function, with termination based solely on the fixed evaluation budget (no convergence-tolerance criterion). Each run used a distinct, reproducible seed (seed = run × 137 + function_id). Algorithm-specific parameters are given in Table 2.

Experiments were implemented in Python 3 on Google Colaboratory (Intel Xeon CPU, 12 GB RAM), using NumPy for numerical computation, SciPy for statistical analysis, pandas for data processing, Matplotlib for visualization and opfunu for the CEC-2017 benchmark functions.

3.1.1. Friedman Test Result

To reliably assess performance differences between ESOA and three reference algorithms (PSO, DE, GWO), this study employs the Friedman tests. This approach suits the multimodal optimization context, where result distributions often violate normality assumptions, making parametric statistical methods difficult to justify.

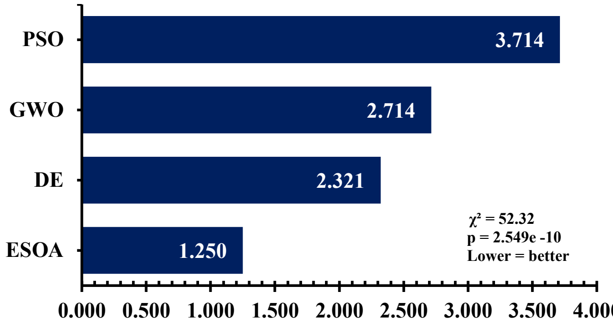

Figure 1. *Friedman Test Result (CEC-2017, D=50, 29 Func)*

The Friedman test results in Figure 1 yield $\chi^2 = 52.32$ with a p-value of approximately 2.549×10^{-10}. At the significance level of 0.05, the null hypothesis that all algorithms perform equivalently is rejected, indicating that the observed performance differences are statistically significant. In terms of mean ranks, ESOA achieves the best performance with the lowest mean rank of 1.250, followed by DE (2.321), GWO (2.714) and PSO (3.714). Compared with ESOA, the mean rank of DE is higher by 1.071, while those of GWO and PSO are higher by 1.464 and 2.464, respectively. These consistent rank differences indicate that ESOA generally outperforms the competing algorithms across the CEC-2017 benchmark suite rather than achieving superior performance only on a small subset of benchmark functions.

3.1.2. Nemenyi post-hoc test

Table 3. *Nemenyi post-hoc test results*

	ESOA	PSO	DE	GWO
ESOA	1.0000	5.51×10^{-12}	1.02×10^{-2}	1.29×10^{-4}
PSO	5.51×10^{-12}	1.0000	3.16×10^{-4}	1.96×10^{-2}
DE	1.02×10^{-2}	3.16×10^{-4}	1.0000	6.65×10^{-1}
GWO	1.29×10^{-4}	1.96×10^{-2}	6.65×10^{-1}	1.0000

The Nemenyi post-hoc test results in Table 3 provide a more granular view of the pairwise relationships among the algorithms, complementing the Wilcoxon test results presented above. Specifically, ESOA differs with very high statistical significance from all three remaining algorithms ($p = 5.51\times10^{-12}$ versus PSO; $p \approx 1.02 \times 10^{-2}$ versus DE; $p = 1.29\times10^{-4}$ versus GWO, all below the significance threshold $\alpha = 0.05$), thereby firmly confirming that the performance superiority of ESOA observed in both the convergence curves and the Wilcoxon test is not attributable to chance but reflects a robust statistical effect across the entire benchmark function set. Similarly, PSO differs with clear statistical significance from DE ($p = 3.16\times10^{-4}$) and GWO ($p \approx 1.96 \times 10^{-2}$), reinforcing the conclusion that PSO is the weakest-performing algorithm among the four compared. Most notably, the DE-GWO pair yields a p-value of 6.65×10^{-1}, far exceeding the 0.05 threshold, indicating no statistically significant difference between these two algorithms - a finding entirely consistent with the relatively balanced win/tie/loss ratios observed in the Wilcoxon test (40/36/8 and 34/40/10, respectively).

This confirms that, although DE accumulates a slightly higher number of wins than GWO in pairwise function-level comparisons, the overall gap between the two algorithms is not sufficiently large to conclude that one is statistically superior to the other. Taken together, the Nemenyi results confirm that ESOA constitutes a distinct, statistically separable performance group relative to the remaining three algorithms (ESOA > PSO, DE, GWO), whereas DE and GWO can be regarded as belonging to a statistically equivalent performance group, with PSO ranking lowest and differing significantly from both DE and GWO.

3.1.3. Wilcoxon Test Result

Table 4. *Wilcoxon Signed-Rank Result*

Algorithm	win	tie	loss
ESOA	76	2	6
PSO	6	4	74
DE	40	8	36
GWO	34	10	40

The Wilcoxon signed-rank test results ($\alpha = 0.05$, 84 pairwise comparisons) in Table 4 strongly corroborate the observations drawn from the convergence curves: ESOA achieves a dominant win rate of 76/84 (~90.5%), confirming its statistically significant superiority over all three competing algorithms across the majority of benchmark functions. Conversely, PSO exhibits the weakest performance, losing in 74/84 comparisons, consistent with the slow convergence behavior previously observed. DE and GWO show relatively balanced outcomes (40 wins/36 losses and 34 wins/40 losses, respectively, with comparatively high tie counts of 8 and 10), reflecting GWO's tendency toward premature convergence despite its fast initial descent enabling it to outperform DE on certain simpler functions while losing on more complex ones. Overall, the statistical tests confirm the general performance ranking ESOA > DE $\gtrsim$ GWO > PSO, with ESOA's advantage over the remaining three algorithms being clear and statistically significant, whereas the DE-GWO gap remains comparatively marginal and warrants more cautious interpretation in the Discussion section.

3.1.4. Convergence Curve Analysis

Figure 2 presents the convergence curves for a representative subset of the CEC-2017 benchmark functions rather than the complete set of 29 test functions. These functions were selected to illustrate the convergence characteristics of the compared algorithms across optimization problems of varying difficulty and landscape complexity. Presenting convergence curves for every benchmark function would substantially increase the length of the manuscript while providing largely redundant information, as many functions exhibit similar convergence behavior. Although only a subset of functions is presented for visual analysis, the performance of all algorithms was comprehensively evaluated on the complete set of 29 CEC-2017 benchmark functions using quantitative performance metrics and non-parametric statistical analyses (the Wilcoxon signed-rank test and the Friedman test). The conclusions drawn in this study are therefore based on the overall benchmark evaluation rather than solely on the selected convergence plots.

The analysis of the representative convergence curves (F1, F7, F12, F13, F18, F21) reveals that ESOA exhibits a clear and consistent advantage over the other three algorithms, not only in terms of convergence speed during the early stage (iterations 0-200) but, more importantly, in its continued capacity to improve the solution during the later stage, as evidenced by the slight downward slope still observable in the zoomed inset (iterations 950-1000), in contrast to the nearly flat trajectories exhibited by GWO and DE-thereby confirming ESOA's superior local exploitation capability even after the other algorithms have effectively stagnated. The most noteworthy finding is the premature convergence exhibited by GWO, clearly visible on F1, F12, F13, and F18, where the algorithm decreases rapidly within the first 50-100 iterations but subsequently plateaus almost entirely, in some cases (F18, F21) stabilizing at a fitness level even higher than that of PSO and DE a phenomenon that can be attributed to the position-update mechanism based on the alpha/beta/delta leader hierarchy, which causes the population to lose diversity and cluster too rapidly around the current best solution, resulting in entrapment in local optima, an effect that is further exacerbated in the D = 50 search space due to the curse of dimensionality. When examined in relation to the characteristics of each objective function, F1, being unimodal, allows the performance gap among algorithms to purely reflect exploitation capability, whereas F13 and F18, as hybrid/composition functions containing Katsuura-based oscillatory components, cause all algorithms to drop almost vertically within the first few dozen iterations before becoming trapped within a shared basin of attraction, such that the true order-of-magnitude differences among algorithms can only be discerned through the log-scale zoomed insets rather than through visual inspection on a linear scale. F21, in particular, emerges as the most discriminative function, with ESOA reaching deeply negative fitness values (approximately -5000 to -6000) while PSO oscillates around zero and both DE and GWO remain trapped in positive territory (~2000-3000), a gap that does not narrow even after the full 1000 iterations, indicating a fundamental disparity in final solution quality rather than merely a difference in convergence rate. Regarding PSO and DE, both algorithms display slower yet smoother convergence trajectories with fewer abrupt inflections, reflecting a comparatively conservative exploration-exploitation strategy; although both are generally outperformed by ESOA across most functions, PSO demonstrates notable competitiveness on F7 during the mid-stage of the search (iterations 200-600), suggesting that it may be better suited to landscapes with shifted/rotated multimodal structures. Collectively, these results demonstrate the consistency of ESOA's performance across diverse function categories, underscoring its robustness; the premature convergence observed in GWO is consistent with limitations previously reported in the original GWO literature. In particular, the results on F21 illustrate the most pronounced solution-quality gap among the four algorithms, a finding further substantiated by the Wilcoxon signed-rank test between ESOA and each competing algorithm, confirming that the observed differences are statistically significant ($p < 0.05$) rather than merely apparent from visual inspection of the convergence curves.

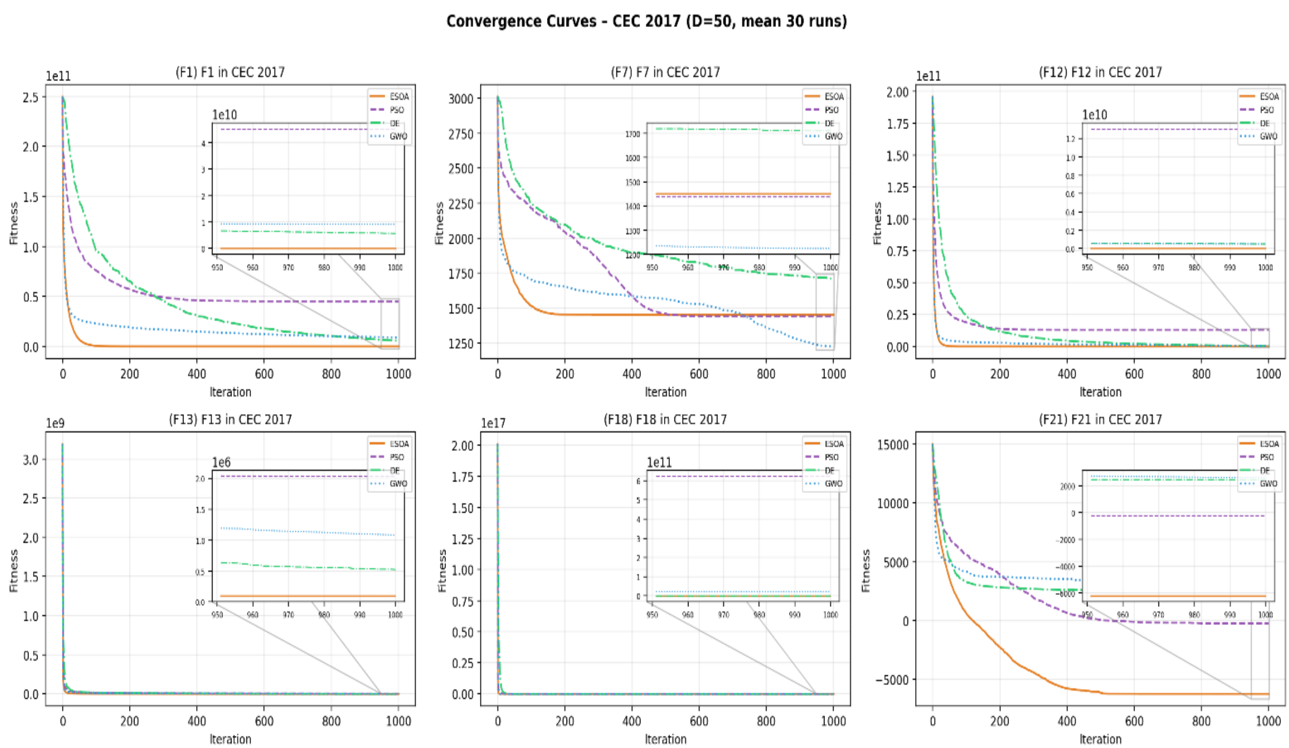

Figure 2. Convergence Curves of ESOA on CEC-2017

3.1.5. Overall Summary

Based on the above analyses, ESOA demonstrates competitive overall performance compared with PSO, DE, and GWO on the CEC-2017 benchmark. Its strengths are reflected not only in the final solution quality but also in its convergence behavior, showing rapid improvement during the early search stage while still maintaining progress in later iterations. In addition, ESOA performs favorably across different categories of benchmark functions rather than relying on a specific problem type, indicating a relatively strong adaptability to diverse optimization landscapes. From a practical perspective, these results suggest that ESOA can utilize computational resources efficiently by achieving high-quality solutions within a reasonable number of iterations.

In general, the benchmark results are consistent with the original design concept of ESOA, which aims to maintain a balance between exploration and exploitation. Through this mechanism, the algorithm is able to reduce several common weaknesses observed in other metaheuristics, such as premature convergence or excessive sensitivity to randomness in certain scenarios, particularly when handling higher-dimensional optimization problems.

4. Case Study

4.1. Data and Problem Context

The dataset used for the applied research is directly extracted from the MES system of a manufacturing enterprise over a period of three months. The resource allocation and cost optimization problem is modeled at a scale of 31 machines and 15 product types. The input parameters include:

- Production demand (D_p): Required quantities for 15 product types; shortages incur a high penalty cost (ω).

- Machine compatibility matrix ($A_{m,p}$): Binary parameter indicating whether machine m can process product p $\left(A_{m,p} \in \{0,1\}\right)$.

Time parameters: Cycle time (CT_m) and setup time (ST_m) for machine m.

- Cost structure and shift system: The base operating cost of each machine (C_m) is determined by its characteristics. However, the system operates under a three-tier shift structure with increasing cost coefficients: regular shift (max 80h, K1 = 1.0), night shift (max 40h, K2 = 1.2) and weekend shift (max 24h, K3 = 1.35).

Overall, this dataset belongs to a class of complex problems characterized by a large search space, multiple stringent constraints, and the nonlinear nature of the cost function when overtime shifts are activated. Solving the full combinatorial space of 31×15 to obtain a static optimal schedule using the MILP approach reveals limitations in flexibility and prolonged computation time, especially under significant fluctuations in input data (e.g., unexpected machine failures).

The inherent complexity and noise of this real-world dataset create an ideal testing environment to evaluate the balance between exploration and exploitation capabilities of the ESOA. Expected to provide responsive scheduling solutions within a short time while minimizing operational costs, thereby effectively overcoming the limitations of traditional linear programming approaches.

4.2. Baseline Schedule

To accurately evaluate the effectiveness of the ESOA in a dynamic production environment, the study first establishes a static Baseline schedule. This schedule is designed to identify a production allocation plan that fully satisfies 100% of the demand (Dp).

The MILP model is defined with a set of core decision variables, including:

- m∈{1,2,…,31} denotes the machine index.

- p∈{1,2,…,15} denotes the product type index.

- t∈{1,2,3} denotes the shift index corresponding to regular, night, and weekend shifts.

- $x_{m,p} \geq 0$: Integer variable representing the quantity of product p processed on machine m.

- $y_{m,p} \in \{0,1\}$: Binary variable indicating the setup state of machine m for product p.

- $h_{m,t} \geq 0$: Continuous variable representing the operating hours of machine m in shift t (with $t \in \{1,2,3\}$, corresponding to regular, night, and weekend shifts).

- $z_{m,t} \in \{0,1\}$: Binary variable equal to 1 if machine m is activated in shift t.

- $U_p \geq 0$: Integer variable representing the shortage quantity of product p.

- $l_{max}, l_{min} \geq 0$: Continuous variables representing the maximum and minimum workload levels in the workshop.

The objective function of the system is formulated to minimize the total operational cost (TC), including shortage penalty costs, lot-splitting penalties, and workload imbalance penalties.

$$Min\, TC = \sum_{m\in M} \sum_{t\epsilon T} \left(C_m K_t h_{m,t}\right) + \sum_{p\, \epsilon P} (\omega U_p) + \sum_{(m,p)\in V} \left(\alpha y_{m,p}\right) + \gamma\left(l_{max} - l_{min}\right) \quad (1)$$

In which ω, α, and γ are penalty coefficients to ensure that the system consistently prioritizes demand fulfillment. This process is governed by a set of strict technical constraints:

Demand satisfaction: The total processed quantity of each product across all machines must exactly match the required quantity specified by the MES system.

$$\sum_{m\, \epsilon M\, |(m,p)\in V} x_{m,p} + U_p = D_p, \forall p \in P \quad (2)$$

Time balancing: The total production time (including ST_m and CT_m) must not exceed the total allocated operating hours across all shifts, while also complying with the maximum time limit of each shift (L_t).

$$\frac{\sum_p (x_{m,p} CT_m + y_{m,p} ST_m)}{60} = \sum_t h_{m,t}, \forall m \in M \quad (3)$$

Shift activation and minimum runtime (R_{min}): Ensures sequential shift activation and requires that a machine must operate for at least 4 hours once an overtime shift is activated.

$$z_{m,2} \leq z_{m,1} \text{ and } z_{m,3} \leq z_{m,2} \quad (4)$$

$$R_{min}\, z_{m,t} \leq h_{m,t} \leq L_t\, z_{m,t} \quad (5)$$

Setup logical constraint: A machine can process product p ($x_{m,p} > 0$) only if it has been successfully set up ($y_{m,p} = 1$).

$$x_{m,p} \leq \min(D_p, \frac{\Sigma_{t\epsilon T} L_t 60}{CT_m})\, y_{m,p} \quad (6)$$

$$y_{m,p} \leq \sum_{t\,\epsilon\,T} Z_{m,t} \quad (7)$$

Constraint on product variety per machine: Each machine is allowed to process at most five different product types, in order to limit operational complexity and reduce setup and switching costs associated with handling multiple product types.

$$\sum_{p\epsilon\, P|(m,p)\in V} y_{m,p} \leq 5 \text{ , } \forall m \in M \quad (8)$$

Load balancing: The actual load of each machine m must lie within the interval $[l_{min}, l_{max}]$.

Minimum batch constraint B_{min}: To avoid the case where a machine is set up only to process an excessively small quantity of products.

$$x_{m,p} \geq B_{min}\, y_{m,p}, \forall (m,p) \in V \quad (9)$$

The mathematical solver will explore the solution space to return a locally optimal static production allocation matrix (31×15). This output not only serves as a benchmark in terms of ideal cost but also acts as a high-quality "seed individual". This individual will be directly injected into the initial population of the ESOA swarm-based algorithm in the subsequent stage. This enables ESOA to bypass the initial blind search phase, converge more rapidly, and be ready to handle the dynamic rescheduling problem when disruptions occur that break the static structure of the Baseline.

4.3. Dynamic Rescheduling Optimization Using ESOA

In practice, unexpected disruptions may completely invalidate the feasibility of a static schedule. The problem is to reallocate the unfinished workload to other machines in order to optimize the increased cost and satisfy demand.

To address this dynamic situation, ESOA is applied using a hybrid approach. The optimal allocation matrix from the Baseline is reused as a seed individual after shifting the workload of the failed machine and randomly compensating it across compatible machines. This approach enables the ESOA to initialize directly within a high-quality solution region instead of restarting the search from scratch. The objective function of ESOA evaluates the quality of each adjustment plan based on the total cost, combined with a penalty function method to handle inequality constraints.

$$F(x) = \sum_{m=1}^{M} Cost(h_m) + \omega'(V_{demand} + V_{compat} + V_{broken}) \quad (10)$$

In this formulation, $\mathrm{Cost}(h_m)$ denotes a stepwise cost function associated with the working shifts of each machine. A penalty function with weight ω is imposed when any of the following constraints is violated: production shortage (V_{demand}), assignment of incompatible product types (V_{compat}), or allocation of tasks to a failed machine (V_{broken}).

The search process for a responsive rescheduling solution leverages the core mechanisms of ESOA illustration in Figure 3:

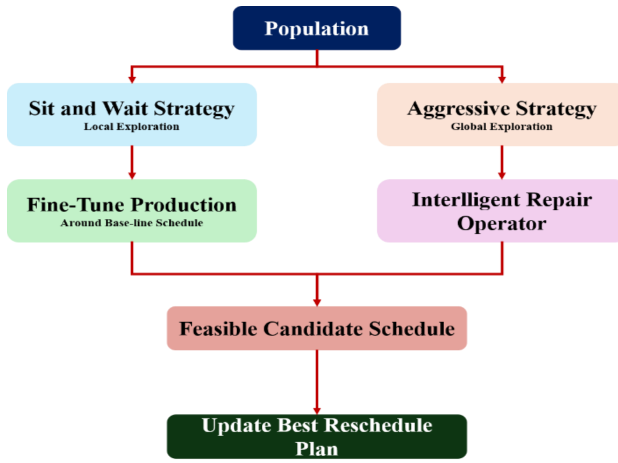

Figure 3. *Data Flow*

- Sit-and-wait strategy: Guided by local search around elite individuals, candidate solutions perform exploitation in the neighborhood of the Baseline schedule to identify minimal and cost-efficient production adjustments.

- Aggressive Strategy: A global search mechanism combined with an intelligent repair operator. When infeasibilities such as shortages or surpluses arise due to mutation, this operator reallocates production from the most loaded machine to the least loaded one, facilitating feasibility restoration and enabling escape from local optima.

This mechanism allows ESOA to rapidly generate a feasible reallocation of workload, maintaining system operation with minimal disruption cost.

4.4. Result Comparison

The performance of the algorithms is comprehensively evaluated based on four main criteria: (i) solution quality (total operating cost), (ii) stability (standard deviation of the solutions), (iii) convergence speed (average number of iterations to reach the convergence threshold) and (iv) computational cost (average execution time per run). Detailed statistical metrics for the solution quality criterion, including Best, Worst, Mean and Standard Deviation (Std) are summarized in Table 5.

Table 5. *Performance comparison of optimization algorithms*

Algorithm	Best	Worst	Mean	Std
ESOA	715.1	807.7	764.2	17.7
PSO	705.2	810.9	765.7	25.4
DE	742.9	799.9	766.7	12.6
GWO	789.6	790.1	789.8	10.1

Best-case Performance: The PSO algorithm achieved the lowest best objective value (705.2), indicating strong exploration capability and the ability to identify highly competitive solutions. This value is 1.399% lower than ESOA and 5.336% lower than DE.

Average Performance: ESOA obtained the best mean result (764.2), followed closely by PSO (765.7) and DE (766.7). The differences among these three algorithms are relatively small, suggesting comparable overall optimization capability. In contrast, GWO showed clearly weaker performance, with a substantially higher mean value (789.8).

Stability: DE demonstrated the highest robustness among the leading algorithms, achieving the lowest standard deviation (12.6), followed by ESOA (17.7) and PSO (25.4). This indicates that DE produced more consistent results across repeated runs, whereas PSO exhibited greater variability despite obtaining the best single solution. GWO also showed a small standard deviation (10.1), but this was associated with the worst mean performance, suggesting premature convergence to a low-quality local optimum with limited ability to escape during later search stages.

Table 6. *Average runtime and iterations to reach the threshold*

Algorithm	Time/run (s)	Average Iterations to Reach Convergence Threshold
ESOA	9.1	113.2
PSO	7.1	40.5
DE	10.2	120.9
GWO	11.4	20.0

All experiments were conducted under identical hardware and software conditions, as a result, in Table 6; the iteration count therefore serves as an objective metric for assessing convergence speed. GWO converges fastest at 20.0 iterations, followed by PSO (40.5), ESOA (113.2), and DE (120.9). However, GWO's rapid convergence reflects premature entrapment in a low-quality local optimum rather than genuine search efficiency, as evidenced by its highest mean objective value (789.8) in Table 5. PSO, despite its relatively fast convergence, exhibits the highest standard deviation (25.4), indicating inconsistent solution quality across runs. In contrast, ESOA requires more iterations but achieves the best mean performance (764.2) with moderate stability, reflecting a more effective balance between exploration and exploitation. When solution quality and consistency are prioritized over convergence speed, ESOA showed promising reliability under the tested conditions.

5. Conclusion

This study presents a comprehensive evaluation of the ESOA through comparisons with representative metaheuristic methods, including PSO, DE and GWO, on the CEC-2017 benchmark suite. The experimental results demonstrate that ESOA achieves competitive overall performance, characterized by high-quality solutions, low result variability, and reliable convergence behavior.

Furthermore, the study extends the application of ESOA to a real-world problem involving production resource allocation and operating cost optimization, using data collected from a manufacturing enterprise's MES system over a three-month period. The formulated problem reflects complex characteristics, including nonlinear cost fluctuations and practical operational constraints. The results indicate that ESOA not only delivers competitive solution quality compared to well-established algorithms but also maintains stable performance across multiple runs. In addition, the experimental results suggest that ESOA has the potential to provide effective exploration and exploitation capabilities compared with the conventional optimization methods considered in this study.

However, certain limitations were identified during the course of the study, primarily related to the characteristics of the dataset and the number of iterations required to achieve convergence. Future research will focus on developing enhanced variants of ESOA to address these limitations, thereby improving the algorithm's robustness, computational efficiency and broader applicability.

REFERENCES

[1] K. de Backer, M. Mancini, and A. Sharma, "Optimizing back-end semiconductor manufacturing through Industry 4.0," *McKinsey & Company,* Feb. 2017. [Online]. Available: https://www.mckinsey.com/industries/semiconductors/our-insights/optimizing-back-end-semiconductor-manufacturing-through-industry-40. [Accessed: Jan. 12, 2026].

[2] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 5th ed. Cham, Switzerland: Springer, 2016. doi: 10.1007/978-3-319-26580-3.

[3] S. Nahmias and T. Olsen, *Production and Operations Analysis*, 7th ed. Long Grove, IL, USA: Waveland Press, 2015. ISBN: 1478628243.

[4] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng*., vol. 6, no. 4, pp. 239–242, Jun. 2014, doi: 10.1007/s12599-014-0334-4.

[5] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.,* vol. 10, no. 4, pp. 2233–2243, Nov. 2014, doi: 10.1109/TII.2014.2300753.

[6] IDC, "Worldwide IDC Global DataSphere forecast, 2022–2026: Enterprise organizations driving most of the data growth," *https://www.marketresearch.com,* May 2022. [Online]. Available: https://www.marketresearch.com/IDC-v2477/Worldwide-IDC-Global-DataSphere-Forecast-31469143/. [Accessed: Jan. 1, 2026].

[7] C. Li, Y. Chen, and Y. Shang, "A review of industrial big data for decision making in intelligent manufacturing," *Eng. Sci. Technol., Int. J.,* vol. 29, p. 101021, 2021, doi: 10.1016/j.jestch.2021.06.001.

[8] A. El Azzaoui, M. M. Salim, and J. H. Park, "Secure and reliable big-data-based decision making using quantum approach in IIoT systems," *Sensors,* vol. 23, no. 10, p. 4852, 2023, doi: 10.3390/s23104852.

[9] A. Banerjee, S. Roy, and D. Mukherjee, "An IIoT machine model for achieving consistency in product quality in manufacturing plants," *arXiv*:2109.12964 [cs.LG], Sep. 2021.

[10] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, 10th ed. New York, NY, USA: McGraw-Hill, 2015.

[11] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1957.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

[13] J. H. Holland, "Genetic algorithms," *Sci. Amer*., vol. 267, no. 1, pp. 66–73, Jul. 1992.

[14] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *in Proc. IEEE Int. Conf. Neural Netw. (ICNN’95), Perth, Australia,* Nov.–Dec. 1995, pp. 1942–1948.

[15] N. C. Hanh, N. C. Quy, T. V. H. Thanh, D. N. C. Cong, L. V. Trung, and V. T. Hanh, "Lithium-Ion Battery Cell Production Process Simulation Using Tecnomatix Plant Simulation Software,"*The University of Danang - Journal of Science and Technology*, vol. 22, no. 1, pp. 36–41, Jan. 2024.

[16] T. P. Q. Nguyen and T. H. T. Le, "An innovative genetic algorithm-based master schedule to optimize job shop scheduling problem," *The University of Danang - Journal of Science and Technology*, vol. 22, no. 12, pp. 7–12, 2024, doi: 10.31130/ud-jst.2024.422E.

[17] R. Storn and K. V. Price, “Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces,” *J. Global Optim*., vol. 11, no. 4, pp. 341–359, 1997.

[18] W. Zhang, X. Bao, X. Hao, and M. Gen, “Metaheuristics for multi-objective scheduling problems in industry 4.0 and 5.0: A state-of-the-art survey,” *Front. Ind. Eng*., vol. 3, p. 1540022, 2025, doi: 10.3389/fieng.2025.1540022.

[19] B. Benaissa, M. Kobayashi, M. Al Ali, T. Khatir, and M. E. A. E. Elmeliani, “Metaheuristic optimization algorithms: An overview,” *HCMCOU J. Sci. Adv. Comput. Struct.,* vol. 14, no. 1, pp. 33–61, 2024, doi: 10.46223/HCMCOUJS.acs.en.14.1.47.2024.

[20] A. G. Gad, “Perspective chapter: Metaheuristic optimization in Industry 4.0 – foundations, applications, and future horizons,” *in Industry 4.0 - Transforming the Future Beyond Manufacturing - Volume 1: Digital Technologies and Smart Industrial Systems. London, U.K.: IntechOpen,* 2026, doi: 10.5772/intechopen.1014314.

[21] M. Maciąg, M. Szeląg, and P. Sowa, “Optimization of electric drive system using metaheuristic algorithms in virtual commissioning environment,” *Energies,* vol. 19, no. 4, p. 1057, 2026, doi: 10.3390/en19041057.

[22] B. Jeong, J.-H. Han, and J.-Y. Lee, “Metaheuristics for a flow shop scheduling problem with urgent jobs and limited waiting times,” *Algorithms*, vol. 14, no. 11, p. 323, 2021, doi: 10.3390/a14110323.

[23] S. Dauzère-Pérès, J. Ding, L. Shen, and K. Tamssaouet, “The flexible job shop scheduling problem: A review,” *Eur. J. Oper. Res.,* vol. 314, no. 2, pp. 409–432, 2024, doi: 10.1016/j.ejor.2023.05.017.

[24] H. N. Trieu, H. P. Nguyen, and T. M. H. Phan, "Two-stage dynamic scheduling model for a flexible flow shop using a genetic algorithm: A case study in a truck body manufacturing company," *The University of Danang - Journal of Science and Technology*, vol. 24, no. 3, pp. 101–107, 2026, doi: 10.31130/ud-jst.2026.24(3).721E.

[25] Y. Wang, Z. Jia, and X. Zhang, “A hybrid meta-heuristic for the flexible flow shop scheduling with blocking,” *Swarm Evol. Comput.,* vol. 75, p. 101195, 2022, doi: 10.1016/j.swevo.2022.101195.

[26] A. Seyyedabbasi, “A reinforcement learning-based metaheuristic algorithm for solving global optimization problems,” *Adv. Eng. Softw.,* vol. 178, p. 103411, 2023, doi: 10.1016/j.advengsoft.2023.103411.

[27] E. V. Altay, O. Altay, and Y. Özçevik, “A comparative study of metaheuristic optimization algorithms for solving real-world engineering design problems*,” Comput. Model. Eng. Sci.,* vol. 139, no. 1, pp. 1039–1094, 2024, doi: 10.32604/cmes.2023.029404.

[28] T. Ivanovski, M. Brkić Bakarić, and M. Matetić, “Recent advances in metaheuristic algorithms,” *Algorithms,* vol. 19, no. 1, p. 19, 2026, doi: 10.3390/a19010019.

[29] Z. Chen *et al.*, “Egret swarm optimization algorithm: An evolutionary computation approach for model free optimization,” *Biomimetics,* vol. 7, no. 4, p. 144, 2022, doi: 10.3390/biomimetics7040144.