HEURISTIC
Vol. 23, No. 2, October 2026
ISSN(p) 1693-8232
ISSN(e) 2723-1585

Energy-Efficient Multi-Objective Permutation Flow Shop Scheduling Using Bird of Prey-Based Optimization

Dana Marsetiya Utama[1], Tegar Pratama Enisuyitno[2]*
[1,2] Department of Industrial Engineering, Universitas Muhammadiyah Malang.

***Corresponding Author**: tegarp259gmailcom@webmail.umm.ac.id

Article history:
Received: 28 September 2026
Accepted: 28 October 2026
Published: 30 October 2026

Keywords:
Birds of Prey-Based Optimization;
Energy Consumption;
Epsilon Constraint;
Makespan;
Permutation Flow Shop Scheduling

ABSTRACT

Permutation flow shop scheduling is a critical issue in sustainable manufacturing because it affects makespan and energy consumption. Pareto-based approaches may generate multiple trade-off solutions requiring additional selection. This study develops a multi-objective optimization framework integrating the epsilon-constraint method and Bird of Prey-Based Optimization (BPBO) to minimize energy consumption while controlling makespan. Three scenarios were evaluated: 10 jobs and 6 machines, 50 jobs and 6 machines, and 100 jobs and 10 machines. Makespan was treated as a constraint, while energy consumption was the primary objective. BPBO was compared with Beluga Whale Optimization and Goat Optimization Algorithm based on makespan, energy consumption, and computation time. BPBO achieved the best performance. In the small scenario, all algorithms obtained a makespan of 144 and energy consumption of 4047. In the medium scenario, BPBO achieved a makespan of 547 and energy consumption of 72245. In the large scenario, BPBO obtained a makespan of 1233 and energy consumption of 533761 with efficient computation time. The methodological contribution lies in integrating the epsilon-constraint method with BPBO to prioritize energy minimization under a makespan constraint, extending constrained optimization to energy-efficient permutation flow shop scheduling. This framework provides a structured approach to evaluating scheduling performance when energy consumption and production completion time must be considered simultaneously. From a practical perspective, the framework can assist production planners in evaluating energy-efficient schedules that satisfy completion-time requirements. These findings indicate that the framework can support energy-aware scheduling and contribute to sustainable manufacturing by incorporating energy considerations into scheduling decisions.

INTRODUCTION

Production scheduling plays a strategic role in manufacturing systems because it determines the level of efficiency in resource utilization, including machinery, labor, materials, production time, and energy. In the context of manufacturing operations, scheduling decisions not only relate to the sequence of tasks but also determine a company's ability to meet production targets on time and efficiently. Various common performance criteria are used in scheduling problems, such as makespan, total flow time, tardiness, machine utilization, energy consumption, and environmental emissions [1].

Among these criteria, makespan is the primary indicator because it represents the total time required to complete all jobs in the production system. Minimizing makespan directly contributes to reducing wait times, increasing throughput, optimizing machine capacity, and meeting customer delivery deadlines [2] [3]. However, the demands of modern manufacturing are no longer solely focused on production speed. Rising energy costs and pressure to reduce environmental impact have made energy consumption a key objective in production scheduling. Inefficient production schedules can increase machine idle time, lead to greater unproductive energy use, and raise operating costs [4]. High energy consumption also contributes to increased carbon emissions and environmental degradation, making energy efficiency a key component of sustainable manufacturing systems [5]. Therefore, scheduling optimization that simultaneously considers makespan and total energy consumption is essential to achieving a balance between economic efficiency and environmental sustainability.

Various multi-objective optimization approaches have been developed to solve production scheduling problems that involve more than one objective and often have conflicting characteristics. Widely used approaches include the weighted sum method, goal programming, Pareto-based evolutionary algorithms such as the Non-dominated Sorting Genetic Algorithm II and the Strength Pareto Evolutionary Algorithm 2, as well as constraint-based approaches such as the epsilon-constraint method [6] [7] [8]. In recent years, these methods have been increasingly applied to scheduling problems that integrate makespan minimization and total energy consumption, as both of these indicators represent both production efficiency and environmental sustainability [5] [9]. The weighted sum method and goal programming work by combining multiple objectives into a single aggregate function, but these approaches have limitations in accurately representing trade-offs because they are highly dependent on weight values and risk introducing bias toward one of the objectives [10]. Meanwhile, Pareto-based approaches can generate a set of non-dominated solutions that illustrate various trade-off alternatives. However, they present new challenges in the process of selecting the final solution and in computational efficiency, particularly for large-scale scheduling problems [11]. Unlike these approaches, the epsilon constraint method offers a more structured optimization framework because one objective is optimized directly, while the other is treated as a constraint [7]. This characteristic is relevant for energy-efficient scheduling because the makespan can be explicitly controlled as an operational constraint. At the same time, total energy consumption is minimized as the primary objective function. This approach aligns with practical industry needs that require meeting production targets while reducing energy consumption.

As the complexity of scheduling problems increases, nature-inspired metaheuristic algorithms are being used more and more frequently due to their high capability in solving complex and NP-hard optimization problems. Metaheuristic algorithms offer flexibility, adaptability, and effectiveness in exploring large and complex search spaces [11]. In the context of permutation flow shop scheduling, the complexity increases because each job must be processed on all machines in the same sequence, resulting in an extremely large number of schedule combinations as the number of jobs and machines grows. Several recent algorithms have demonstrated great potential in solving large-scale optimization problems. The Beluga Whale Optimization Algorithm has garnered attention for its strong balance of exploration and exploitation, enabling it to perform global search while effectively refining solutions [12]. Additionally, modern metaheuristics exhibit fast convergence rates and resistance to local optima traps, which are critical characteristics for solving complex optimization problems [13]. In this study,

the Goat Optimization Algorithm and the Birds of Prey-Based Optimization were also considered because both have shown promising performance in recent optimization studies [13] [14].The characteristics of these algorithms make them relevant for solving energy-aware flow shop scheduling problems, particularly since this problem involves nonlinear relationships, a large solution space, and production constraints that must be satisfied simultaneously.

Previous research indicates that the permutation flow shop scheduling problem has been extensively studied using various objective functions and optimization algorithms. Most studies prioritize makespan as the primary criterion, as this metric directly reflects the efficiency of completing all tasks within the production system. Furthermore, there is a growing focus on energy consumption, as seen in the research by Tasgetiren, et al. [15], Gu, et al. [16], Zeng, et al. [17], Utama and Febrita [18], Huang, et al. [19], Liang, et al. [20], Pesaru, et al. [21], Utama and Al Imron [22], Lu, et al. [23], Risma, et al. [24], Ghasemi, et al. [13], Lu, et al. [25], Ma and Zhao [26], Du and Zhou [27], and Yin, et al. [28]. These studies demonstrate that the integration of time efficiency and energy efficiency has become a key direction in sustainable manufacturing scheduling. Several studies extend the optimization criteria by incorporating tardiness, such as Yüksel, et al. [29], Dong, et al. [30], Ying, et al. [31], Cai and Chen [32], Mimouni, et al. [33], Sun, et al. [34], and Tsiftsoglou, et al. [35], whilst the aspect of carbon emissions remains relatively limited and only appears in the study Gu, et al. [16] and Utama and Febrita [18]. In terms of methodology, previous research has employed a variety of approaches, ranging from the artificial bee colony [15], mixed-integer linear programming and metaheuristics [16], the hybrid cuckoo search algorithm [17], deep reinforcement learning [26] [29], improved NSGA-II [19], the giant trevally optimiser and African vultures optimization [20], the bi-role co-evolutionary algorithm [21], grey wolf optimisation and variable neighbourhood search [23], improved salp swarm [28], enhanced particle swarm optimization [30], to the hybrid tyrannosaurus optimization algorithm [35]. Although these developments represent significant progress, the literature remains dominated by metaheuristic and Pareto-based optimization approaches, while the integration of the epsilon-constraint method with newer metaheuristic algorithms, particularly Birds of Prey-Based Optimization (BPBO), remains insufficiently explored. Furthermore, there has been no comparative study that systematically evaluates BPBO, Beluga Whale Optimization, and the Goat Optimization Algorithm within a single energy-efficient flow-shop scheduling framework. This situation highlights a research gap for developing more structured, controlled, and relevant optimization approaches for sustainable production decision-making.

Although research on the permutation flow shop scheduling problem has advanced significantly, determining a production schedule that minimizes total energy consumption while satisfying a specified makespan constraint remains a key challenge in energy-efficient manufacturing. First, most previous research has focused on Pareto-based optimization approaches to address conflicting objectives, particularly makespan and energy consumption. While this approach can generate a set of non-dominated solutions, selecting a single operational schedule from the resulting set may require additional decision-making before implementation. Second, the integration of the epsilon-constraint method with nature-inspired metaheuristic algorithms remains limited, particularly in approaches that directly minimize energy consumption while treating makespan as an explicit constraint. Third, comparative analyses that evaluate the Beluga Whale Optimization, Goat Optimization Algorithm, and Birds of Prey-Based Optimization within a single integrated optimization framework have not been

extensively developed, particularly in the context of energy-efficient flow-shop scheduling. Fourth, research on the sensitivity of the epsilon value and its impact on solution stability, algorithm robustness, and the trade-off between makespan and energy consumption remains insufficient. Previous studies have also confirmed that structured integration between optimization frameworks and metaheuristic algorithms in the context of energy-efficient scheduling remains limited [36] [37]. Accordingly, the specific problem addressed in this study is the need for an optimization framework that minimizes total energy consumption under a makespan constraint while enabling a consistent evaluation of alternative metaheuristic algorithms. To address this problem, this study integrates the epsilon-constraint method with BPBO and compares its performance with BWO and GOA across three PFSP problem sizes. The algorithms are evaluated in terms of makespan, total energy consumption, and computational time to determine whether the proposed framework provides competitive energy-efficient scheduling solutions while satisfying the specified makespan constraint.

Based on this gap, this study proposes an energy-aware flow-shop scheduling approach by integrating the epsilon-constraint method and a nature-inspired metaheuristic algorithm. The primary objective of this study is to minimize total energy consumption while still satisfying the makespan constraint, so that the production system can achieve energy efficiency without compromising job completion targets. Birds of Prey-Based Optimization is used as the main algorithm because it possesses strong exploration and exploitation capabilities, good convergence speed, and robustness in avoiding local optima. To evaluate the effectiveness of the proposed approach, the performance of Birds of Prey-Based Optimization is systematically compared with the Goat Optimization Algorithm and the Beluga Whale Optimization Algorithm under identical experimental conditions. The novelty of this study lies in the integration of the epsilon-constraint method and BPBO within an energy-aware permutation flow shop scheduling framework, together with a systematic comparison of BPBO, GOA, and BWO across three problem sizes. In contrast to approaches that use Pareto-based optimization and require a subsequent selection among alternative solutions, the proposed formulation explicitly treats makespan as a constraint and total energy consumption as the optimization objective. This enables the scheduling problem to be approached from a decision-oriented perspective in which energy efficiency is pursued within a specified production completion limit. Thus, the originality of this study lies in the integration and empirical evaluation of existing optimization techniques rather than in the development of a new metaheuristic algorithm.

From a methodological and theoretical perspective, this study extends the empirical understanding of energy-aware permutation flow shop scheduling by examining how an epsilon-constrained formulation can be combined with different nature-inspired metaheuristics. By prioritizing energy minimization subject to a production completion limit, the framework offers an alternative to formulations that treat makespan and energy consumption as competing objectives requiring a final Pareto-solution selection. The findings provide a basis for further investigation of constraint-handling strategies, algorithm scalability, and the relationship between energy consumption and scheduling performance. The contribution should therefore be interpreted as a methodological and empirical extension of existing scheduling research rather than the introduction of an entirely new optimization theory.

Accordingly, the main contributions of this study are threefold: (1) the formulation of an energy-aware permutation flow shop scheduling framework that minimizes total energy consumption subject to a makespan constraint; (2) a comparative evaluation of

BPBO, GOA, and BWO across three problem sizes, providing empirical insights into their scheduling performance under consistent experimental conditions; and (3) the identification of the proposed framework's potential to support energy-conscious production scheduling. Together, these contributions clarify the study's methodological originality, scientific relevance, and practical implications while identifying opportunities for future research involving statistical validation, broader constraint-sensitivity analysis, and industrial implementation.

MATERIALS AND METHODS

Mathematical Model

This study formulates the permutation flow shop scheduling problem as an energy-minimization model subject to a makespan constraint implemented using the epsilon-constraint method. The notation and mathematical model for this problem are outlined as follows:

Table 1. Notations and Definitions Used in the Mathematical Model

Notations	Definitions
$N = (1,2,...,n)$	The set of jobs to be scheduled.
$M = (1, 2, ..., m)$	The set of machines in the flowshop system.
p_{ij}	Processing time of job *i* on machine *j*.
e_j^p	Energy consumption of machine *j* during production per unit of time.
e_j^i	Machine *j's* energy consumption whilst idle per unit of time.
D	Big-M.
ε	Maximum permitted makespan.
x_{ik}	Equals 1 if job *i* is processed before job *k*; 0 otherwise, for $k > i$.
C_{ij}	Completion time of job *i* on machine *j*.
C_{max}	The makespan or maximum completion time for all jobs.
θ_j	Total idle time of machine *j*.
TEC	Total energy consumption.

Objective function

$$\min Z = TEC \tag{1}$$

Subject to

$$C_{i1} \geq p_{i1} \qquad \forall i \in N \tag{2}$$

$$C_{ij} \geq C_{i,j-1} + p_{ij} \qquad \forall i \in N,\ \forall j \in M,\ j \geq 2 \tag{3}$$

$$C_{ij} \geq C_{kj} + p_{ij} - Dx_{ik} \qquad \forall i,k \in N,\ k > i,\ \forall j \in M \tag{4}$$

$$C_{kj} \geq C_{ij} + p_{kj} - D(1 - x_{ik}) \qquad \forall i,k \in N,\ k > i,\ \forall j \in M \tag{5}$$

$$C_{\max} \geq C_{im} \qquad \forall i \in N \tag{6}$$

$$C_{\max} \leq \varepsilon \tag{7}$$

$$\theta_j = C_{\max} - \textstyle\sum_{i \in N} p_{ij} \qquad \forall j \in M \tag{8}$$

$$TEC = \sum_{i \in N} \sum_{j \in M} p_{ij}\, e_j^p + \sum_{j \in M} \theta_j\, e_j^i \tag{9}$$

$$x_{ik} \in \{0,1\} \qquad \forall i,k \in N,\ k > i \tag{10}$$

$$C_{ij},\ C_{\max},\ \theta_j,\ TEC \geq 0 \qquad \forall i \in N,\ \forall j \in M \tag{11}$$

Equation (1) defines the primary objective function, which is to minimize total energy consumption. Energy consumption is calculated from two main components: the energy used during the production process and the energy consumed when the machine is idle. Equation (2) ensures that the completion time of each job on the first machine is not less than its processing time. Equation (3) ensures the processing precedence constraints across consecutive machines, requiring each job to complete processing on machine j-1 before processing begins on machine j. Equations (4) and (5) are big-M-based disjunctive constraints that ensure two jobs are not processed simultaneously on the same machine and maintain a consistent job sequence across all machines. Equation (6) defines the makespan as the maximum completion time of all jobs on the last machine. Equation (7) is the main component of the epsilon constraint, which limits the makespan value so that it does not exceed the specified ε value. With this constraint, the optimization process is directed toward finding a schedule with minimum energy consumption without compromising the production completion limit. Equation (8) calculates the total idle time on each machine based on the difference between the makespan and the total processing time of all jobs on that machine. Equation (9) calculates the total energy consumption as the sum of processing energy and idle energy. Equation (10) defines the job-sequence variable as a binary variable. At the same time, Equation (11) specifies that all time and energy variables must be non-negative. This model provides a structured mathematical framework for managing the trade-off between production time efficiency and energy-efficiency in permutation flow shop scheduling.

Proposed Algorithm

The proposed algorithm adopts Birds of Prey-Based Optimization to solve the energy-efficient permutation flow shop scheduling problem. Each bird of prey represents a candidate solution, while the prey represents the current best solution (X_{prey}). An initial population of$N$ individuals is randomly generated in the search space using Equation (12).

$$X_i = X_{\min} + rand(1, D) \times (X_{\max} - X_{\min}) \quad (12)$$

In Equation (12),X_i represents the position of the i-th individual, $X_{\min}$ and $X_{\max}$ are the lower and upper bounds of the decision variables, $rand(1, D)$ is a random vector of dimension D, and D is the number of jobs. Once the population is formed, each solution is evaluated to determine the best solution as X_{prey} and the worst solution as X_{worst} .

The search process is carried out through four main mechanisms. The first mechanism is individual search, which involves each predatory bird moving toward the current best solution. This mechanism reinforces local exploration and is formulated in Equation (13). ` $K_{i1}$ ` is a random number between 1 and 2 that determines whether the hunter attacks the target from the front or the back. If its value is 2, the hunter will attack the prey from the front; if its value is 1, the hunter will attack the prey from behind the i-th hunter.

$$X_i^{\text{new}} = X_i + \text{rand}(1, D) \times \left(X_{\text{prey}} - K_{i1} \times X_i\right) \quad (13)$$

The second mechanism is group hunting. At this stage, individual positions are updated by taking into account the population average X_{mean} . This mechanism enhances global exploration and accelerates convergence, as shown in Equation (14).

$$X_i^{\text{new}} = X_{\text{mean}} + \text{rand}(1, D) \times \left(X_{\text{prey}} - K_{i2} \times X_{\text{mean}}\right) \quad (14)$$

The third mechanism is population competition. Individuals move away from the worst solution (X_{worst}) so that the quality of the population improves at each iteration. This process is expressed in Equation (15).

$$X_i^{new} = X_i + \text{rand}(1, D) \times (X_i - K_{i3} \times X_{worst}) \quad (15)$$

The fourth mechanism is random relocation. This step moves individuals to new areas within the search space to maintain solution diversity. This relocation is expressed in Equation (16).

$$X_i^{new} = X_i + \text{rand}_1 \times \left(X_{\min} + \text{rand} \times (X_{\max} - X_{\min})\right) \quad (16)$$

In Equations (13) - (16),K_{i1} ,K_{i2} , and K_{i3} take random values of 1 or 2. These values determine whether the attack comes from the front or the back. The parametersrand_1 and$rand$ are random numbers between 0 and 1. All position-updating mechanisms are selected randomly at each iteration to maintain a balance between exploration and exploitation.

In summary, the algorithm begins with population initialization, objective function evaluation, and determination of X_{prey} , individual position updates, solution boundary checks, and updates to the best solution. This process repeats until the maximum number of iterations is reached. The pseudocode for the optimization procedure is presented in Algorithm 1.

Algorithm 1. Pseudocode of Birds of Prey-Based Optimization

1: Input: $\text{X}_{\min}, \text{X}_{\max}, \text{D}, \text{Iter}_{\max}$, and N
2: Output: the best solution (X_{prey})
3. Generate the initial birds of prey using $\text{X}_\text{i} = \text{X}_{\min} + \text{rand}(1, \text{D}) \times (\text{X}_{\max} - \text{X}_{\min})$ (12)
namely X
4. Evaluate the fitness of each bird of prey, f (X)
5. Iter ← 0
6. while Iter ≤ $\text{Iter}_{\max}$ do
7. Iter ← Iter + 1
8. for i = 1 to N do
9. Select X_{prey} and X_{worst}, and identify the factor P_i for i=1,2,….,N
10. if rand < P_i then
11. if rand < rand then
12. $\text{X}_\text{i}^{\text{new}} = \text{X}_\text{i} + \text{rand}(1, \text{D}) \times (\text{X}_{\text{prey}} - \text{K}_{\text{i1}} \times \text{X}_\text{i})$ (13)
13. else if rand < rand then
14. $\text{X}_\text{i}^{\text{new}} = \text{X}_{\text{mean}} + \text{rand}(1, \text{D}) \times \left(\text{X}_{\text{prey}} - \text{K}_{\text{i2}} \times \text{X}_{\text{mean}}\right)$ (14)
15. else
16. $\text{X}_\text{i}^{\text{new}} = \text{X}_\text{i} + \text{rand}(1, \text{D}) \times \text{X}_\text{i} - \text{K}_{\text{i3}} \times \text{X}_{\text{worst}})$ (15)
17. end
end
18. else
19. $\text{X}_\text{i}^{\text{new}} = \text{X}_\text{i} + \text{rand} \times (\text{X}_{\min} + \text{rand}(1, \text{D}) \times (\text{X}_{\max} - \text{X}_{\min}))$ (16)
20. end
21. if $\text{f}(\text{X}_\text{i}^{\text{new}}) < \text{f}(\text{X}_\text{i})$then
22. $\text{X}_\text{i} = \text{X}_\text{i}^{\text{new}}$ and $\text{f}(\text{X}_\text{i}) = \text{f}(\text{X}_\text{i}^{\text{new}})$
23. if $\text{f}(\text{X}_\text{i}) < \text{f}\left(\text{X}_{\text{prey}}\right)$ then
24. $\text{X}_{\text{prey}} = \text{X}_\text{i}$ and $\text{f}\left(\text{X}_{\text{prey}}\right) = \text{f}(\text{X}_\text{i})$
25. end
26. end
27. end
28. end

Data and experimental results

This study uses a numerical simulation approach to model a permutation flow shop scheduling system. The primary data used consists of the processing times for each job on each machine, organized in matrix form. This processing time matrix serves as the main input for the optimization process because it determines the processing order, the completion time of each job, the makespan, machine idle time, and total energy consumption. To comprehensively evaluate the algorithm's performance, the experiments were divided into three scenarios based on the problem's complexity level. The first scenario represents a small scale with 10 jobs and 6 machines. The second scenario represents a medium scale with 50 jobs and 6 machines. The third scenario represents a large scale with 100 jobs and 10 machines. This division of scenarios is designed to test the algorithm's ability to handle an increasing number of jobs and machines, so that stability, computational efficiency, and solution quality can be analyzed step by step. Simulation data is generated in a controlled manner to accurately depict the characteristics of a flow-shop production system while enabling a comparison of algorithm performance under consistent experimental conditions.

The experiments were conducted using MATLAB software as the primary computational platform. Each scenario was tested with algorithm parameters set based on the problem's complexity level. In the small-scale scenario, the maximum number of iterations was set to 500 with a population size of 200. In the medium-scale scenario, the maximum number of iterations was increased to 1000 with a population size of 700. In large-scale scenarios, the maximum number of iterations was set to 1200 with a population size of 800. Adjustments to the number of iterations and population size were made to provide a broader search capacity for problems with more complex solution spaces. In this experiment, the Birds of Prey-Based Optimization algorithm was used as the main algorithm.

In contrast, the Beluga Whale Optimization and Goat Optimization Algorithms were used as comparison algorithms. All three algorithms were run under the same scenario with identical input structures, constraints, and evaluation metrics. This setup ensures that the comparison of algorithm performance is conducted objectively and consistently. The analysis procedure in this study employs a multi-objective optimization approach based on the epsilon constraint method. Within this framework, total energy consumption is defined as the primary objective function. At the same time, makespan is treated as an explicit constraint. The computational process is carried out in two stages. In the first stage, the algorithm is run to minimize makespan, thereby obtaining the optimal makespan value as a reference for production limits. The results of the first stage yielded a makespan value of 144 for the 10-job and 6-machine scenario, 564 for the 50-job and 6-machine scenario, and 1246 for the 100-job and 10-machine scenario. These values were then used as the epsilon bounds in the second stage. In the second stage, the optimization process focuses on minimizing total energy consumption while ensuring that the makespan does not exceed the epsilon limit obtained in the first stage. This two-stage approach allows for explicit control of the trade-off between production time efficiency and energy efficiency, ensuring that the resulting solution meets the production completion target while minimizing energy consumption.

The algorithms were implemented by running programs in MATLAB. The computational process began with the initialization of process time data, energy parameters, population size, maximum number of iterations, and the epsilon threshold. Once initialization was complete, each algorithm generated an initial population of solutions in the form of a sequence of jobs. These solutions were then evaluated based on

makespan and total energy consumption according to the formulated mathematical model. During the iterative process, the algorithms update the solutions through their respective search mechanisms until they reach the termination condition determined by the maximum number of iterations. Birds of Prey-Based Optimization represents the hunting strategy of birds of prey through global and local search mechanisms. Global search is used to expand the exploration of the solution space, while local search is used to refine the best candidate solutions through more targeted movements. These characteristics allow the algorithm to maintain a balance between exploration and exploitation, reduce the risk of getting stuck at a local optimum, and improve the quality of solutions for complex scheduling problems.

The algorithms' performance was evaluated using three key indicators: makespan, total energy consumption, and computational time. Makespan is used to assess the algorithms' ability to generate schedules with efficient completion times. Total energy consumption is used to measure the algorithms' effectiveness in reducing process energy consumption and idle energy. Computational time is used to evaluate each algorithm's computational efficiency in solving the test scenarios. All experimental outputs were recorded and presented in tabular form to allow for a systematic analysis of the comparisons between algorithms. Through this experimental design, the performance of the Birds of Prey-Based Optimization, Beluga Whale Optimization, and Goat Optimization Algorithms can be directly compared within the framework of energy-based flow-shop scheduling, while also demonstrating the effectiveness of the epsilon constraint approach in generating efficient and controlled production schedules. Performance of the proposed algorithm and other algorithms is also measured using Efficiency Index Percentage (17) and Relative Eror Percentage (18).

$$EIP = \frac{\bar{T}_{proposed\ algorithm}}{\bar{T}_{another\ algorithm}} \times 100\% \quad (17)$$

$$REP = \frac{\bar{T}_{another\ algorithm} - \bar{T}_{proposed\ algorithm}}{\bar{T}_{proposed\ algorithm}} \times 100\% \quad (18)$$

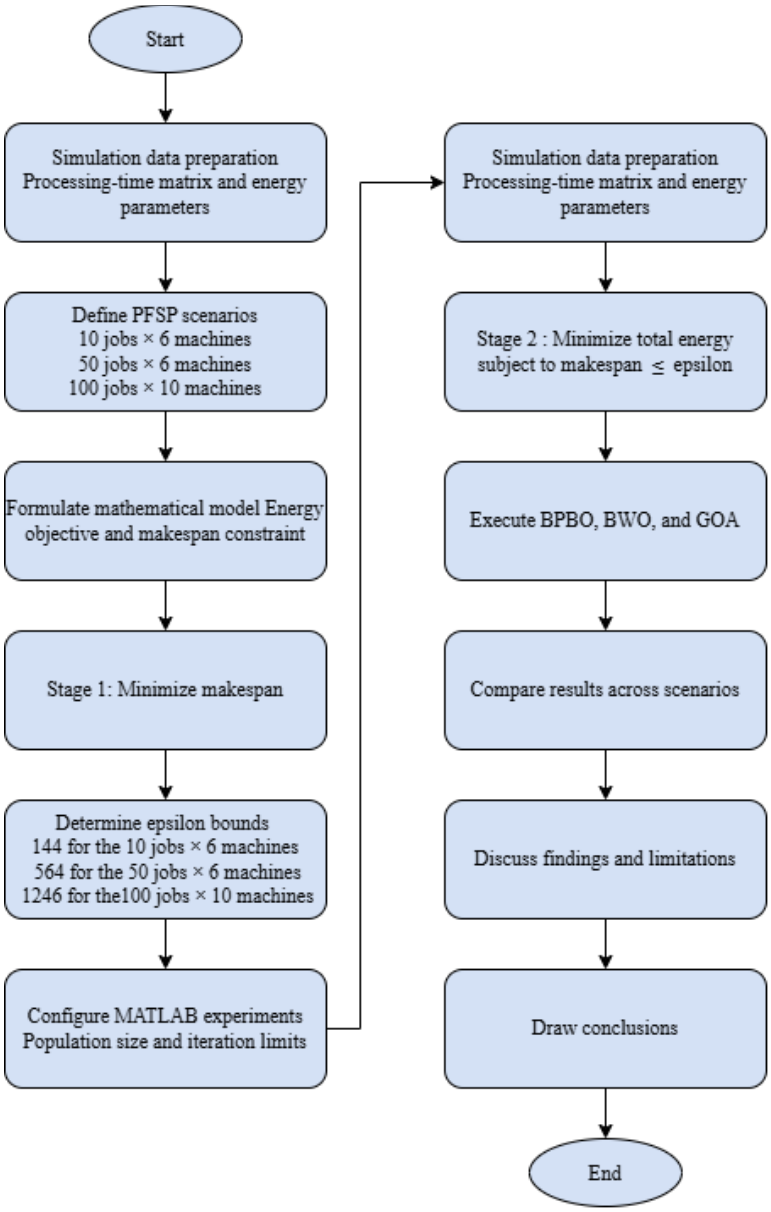

Figure 1. Research Methodology and Workflow for Energy-Efficient Permutation Flow Shop Scheduling

To provide a systematic overview of the research procedure, a research methodology flowchart is presented in Figure 1. The workflow begins with simulation data preparation, including the processing-time matrix and energy parameters, followed by the definition of three permutation flow shop scheduling scenarios. The mathematical model is then formulated to minimize total energy consumption subject to a makespan constraint. The optimization procedure consists of two stages: makespan minimization to determine the epsilon bounds, followed by energy minimization under the corresponding makespan limits. The BPBO, BWO, and GOA algorithms are subsequently implemented in MATLAB and evaluated across the three problem scenarios. Their performance is assessed using makespan, total energy consumption, computational time, Efficiency Index Percentage (EIP), and Relative Error Percentage (REP). The experimental results are then compared and discussed to determine the relative performance of the algorithms and to formulate conclusions regarding energy-efficient permutation flow shop scheduling. Figure 1 summarizes the sequence of these activities and clarifies the

relationship between data preparation, optimization, performance evaluation, and conclusion drawing.

RESULTS AND DISCUSSIONS

Results

The experimental results show that the Birds of Prey-Based Optimization algorithm delivers the best performance in solving the energy-based permutation flow shop scheduling problem. The main finding of this study is that the integration of the epsilon-constraint method and the Birds of Prey-Based Optimization algorithm is capable of generating production schedules with lower energy consumption, more competitive makespans, and more efficient computation times compared to the Beluga Whale Optimization and Goat Optimization Algorithms. New insights gained from these results indicate that the adaptive exploration and exploitation mechanisms in Birds of Prey-Based Optimization are effective for managing the trade-off between production time efficiency and energy efficiency, particularly as problem complexity increases. A summary of the data processing results is presented in Table 2 and Table 3, which includes a comparison of algorithm performance based on the number of jobs and machines, the number of iterations, population size, computational time, makespan, and total energy consumtion.

Table 2. Comparative Results of Makespan and Total Energy Consumption for BPBO, BWO, and GOA Across Three Scheduling Scenarios

Research	Number of machines and jobs	iteration	population	method	Makespan	Best energy
CASE 1	10 jobs and 6 machines	500	200	BWO	144	4047
				GOA	144	4047
				BPBO	144	4047
CASE 2	50 jobs and 6 machines	1000	700	BWO	550	72487
				GOA	564	72656
				BPBO	547	72245
CASE 3	100 jobs and 10 machines	1200	800	BWO	1234	540839
				GOA	1246	545923
				BPBO	1233	533761

Table 3. Comparative Computational Time and Scheduling Performance of BPBO, BWO, and GOA Across Three Scheduling Scenarios

Research	Number of machines and jobs	iteration	population	method	Makespan	Computation time
CASE 1	10 jobs and 6 machines	500	200	BWO	144	43207
				GOA	144	51827
				BPBO	144	36258
CASE 2	50 jobs and 6 machines	1000	700	BWO	550	252353
				GOA	564	274788
				BPBO	547	248794
CASE 3	100 jobs and 10 machines	1200	800	BWO	1234	431543
				GOA	1246	447274
				BPBO	1233	418154

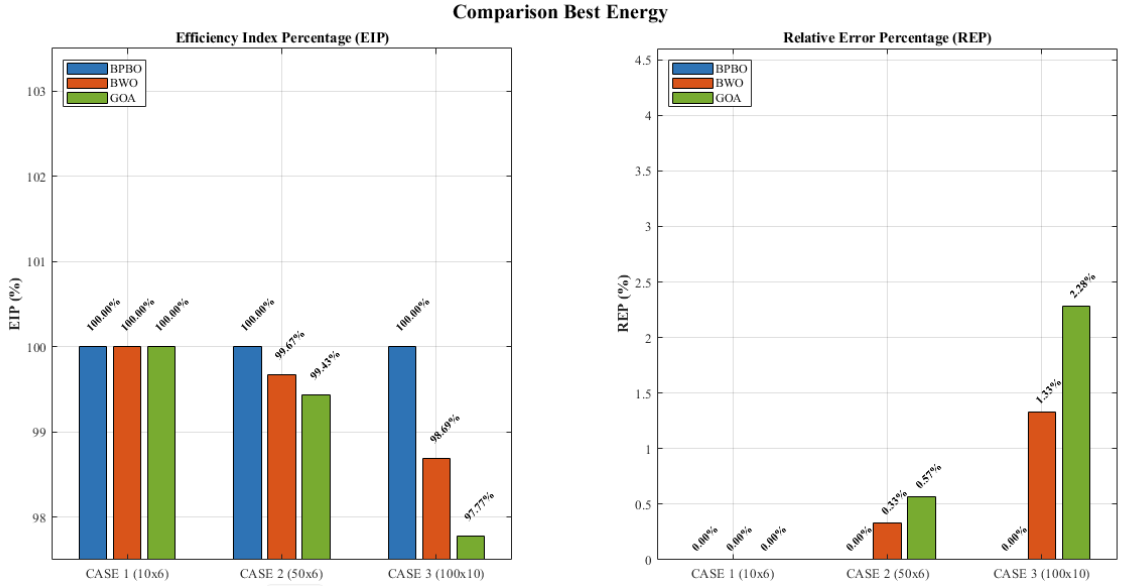

Figure 2. Comparison of Total Energy Consumption Among BPBO, BWO, and GOA

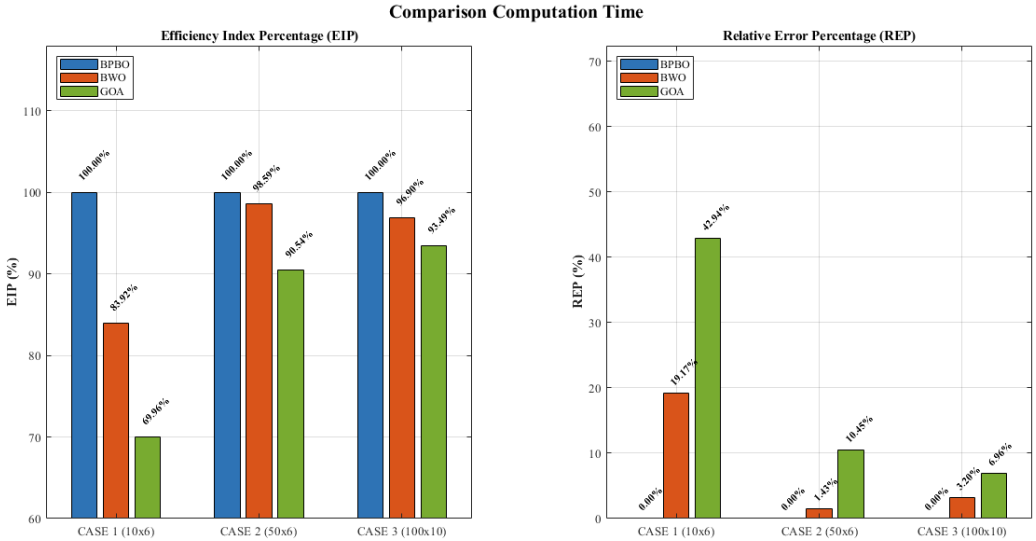

Figure 3. Comparison of Computational Time Among BPBO, BWO, and GOA

Figure 2 and Figure 3 present a comparative evaluation of BWO, GOA, and BPBO in terms of energy consumption and computational time across the three problem scenarios. The Efficiency Index Percentage (EIP) and Relative Error Percentage (REP) are included to complement the absolute performance values reported in Table 2 and Table 3 by providing a relative comparison among the algorithms. These indicators facilitate the identification of performance differences that may not be immediately apparent from the absolute values alone.

Discussions

Based on Table 2 and Table 3, the algorithm's performance is directly influenced by the problem's complexity. An increase in the number of jobs and machines expands the combinatorial solution space, making effective exploration and exploitation strategies increasingly important for identifying high-quality schedules. Under these conditions, the optimization algorithm must possess stable and efficient search capabilities to identify

promising solutions within a reasonable computational time. The experimental results show that Birds of Prey-Based Optimization achieved the lowest reported makespan and total energy consumption in the medium- and large-scale scenarios, as well as the lowest reported computation time across all three scenarios. However, these findings demonstrate BPBO's performance under the tested conditions rather than conclusively proving that it avoids local optima or is universally superior to other optimization algorithms.

In terms of makespan, the results show that in a small-scale scenario with 10 jobs and 6 machines, all algorithms produced the same value 144. This result indicates that the three algorithms obtained identical scheduling outcomes for this particular instance; however, it does not necessarily establish that the problem is simple or that all algorithms reached a proven global optimum. Performance differences began to emerge in the medium-scale scenario with 50 jobs and 6 machines. In this scenario, the Birds of Prey-Based Optimization algorithm produced a makespan of 547, outperforming the Beluga Whale Optimization algorithm 550 and the Goat Optimization Algorithm 564. In the large-scale scenario with 100 jobs and 10 machines, the Birds of Prey-Based Optimization once again produced the best makespan of 1233, followed by the Beluga Whale Optimization at 1234 and the Goat Optimization Algorithm at 1246. Relative to BWO, BPBO reduced makespan by approximately 0.55% in the medium-scale scenario and 0.08% in the large-scale scenario. Relative to GOA, the reductions were approximately 3.01% and 1.04%, respectively. Although the difference between BPBO and BWO in the large-scale scenario is relatively small, BPBO consistently achieved the lowest reported makespan in the medium- and large-scale scenarios. These results indicate that BPBO maintained favorable production completion times while minimizing energy consumption. Furthermore, all reported makespan values satisfied the corresponding epsilon limits of 144, 564, and 1246 for the small-, medium-, and large-scale scenarios, respectively. This confirms compliance with the specified makespan constraints in the reported experiments, although it does not establish that the same performance will be maintained under all possible operating conditions.

In terms of computation time, BPBO demonstrated the lowest reported computational time across all scenarios. In the small-scale scenario, this algorithm recorded a value of 36258, compared with 43207 for BWO and 51827 for GOA. In the medium-scale scenario, BPBO recorded 248794, compared with 252353 for BWO and 274788 for GOA. In the large-scale scenario, BPBO recorded 418154, compared with 431543 for BWO and 447274 for GOA. Based on the reported values, BPBO reduced computational time relative to BWO by approximately 16.08%, 1.41%, and 3.10% in the small-, medium-, and large-scale scenarios, respectively. Compared with GOA, the reductions were approximately 30.04%, 9.46%, and 6.51%, respectively. These results indicate that BPBO was computationally more efficient than the two comparison algorithms under the experimental settings used in this study. In real-world manufacturing, lower computational time may be beneficial because production systems often require scheduling decisions to be updated in response to changes in demand, machine availability, or job priorities. Nevertheless, the computation time unit and measurement procedure must be verified against the original MATLAB outputs. A fair runtime comparison also requires consistent hardware, software environments, stopping criteria, and implementation procedures. Therefore, practical real-time capability should not be inferred from the reported values alone.

In terms of energy consumption, the primary objective of this study, Birds of Prey-Based Optimization, demonstrated the most dominant performance. In the small scenario,

all algorithms produced the same energy value, namely 4047. This indicates that the algorithms obtained identical energy outcomes for the tested small-scale instance, but does not establish that the entire solution space was easily explored. In the medium scenario, Birds of Prey-Based Optimization produced an energy consumption of 72245, lower than Beluga Whale Optimization at 72487 and the Goat Optimization Algorithm at 72656. In the large-scale scenario, the superiority of Birds of Prey-Based Optimization becomes even clearer, with an energy consumption of 533761, lower than that of Beluga Whale Optimization at 540839 and the Goat Optimization Algorithm at 545923. These results correspond to energy consumption reductions of approximately 0.33% and 0.57% relative to BWO and GOA, respectively, in the medium-scale scenario. In the large-scale scenario, the reductions were approximately 1.31% relative to BWO and 2.23% relative to GOA. This reduction in energy consumption indicates that the Birds of Prey-Based Optimization is capable of scheduling jobs more effectively, thereby reducing machine idle time and improving process synchronization between machines. In a flow shop system, energy consumption is determined not only by active processing time but also by the duration during which machines remain idle. Therefore, the algorithm's ability to minimize idle time is a key factor in generating an energy-efficient production schedule.

The potential advantages of BPBO can be discussed in relation to its adaptive search mechanism, as described in the methodological literature [13]. An adaptive metaheuristic may balance exploration of different regions of the solution space with exploitation of promising candidate solutions. This balance is relevant to scheduling problems because the number of possible job sequences increases as the problem size grows. Nevertheless, the present experiments do not isolate the contribution of individual search mechanisms or directly measure population diversity. Therefore, the results support BPBO's comparative performance against BWO and GOA under the tested conditions, rather than proving that a particular search mechanism caused the observed improvements. In energy-efficient flow-shop scheduling, an effective search strategy is important because the job sequence affects makespan and, depending on the energy model, total energy consumption. Thus, the results of this study demonstrate that Birds of Prey-Based Optimization not only excels at finding solutions with minimum energy consumption but is also capable of simultaneously maintaining high makespan quality and computational efficiency.

These findings can be contextualized through comparison with previous studies on energy-efficient flow-shop scheduling. Tasgetiren, et al. [15] investigated an energy-efficient no-wait flow-shop scheduling problem by considering total flow time and total energy consumption. Their study shares the energy-efficiency focus of the present research, although its scheduling formulation and optimization approach differ. Gu, et al. [16] investigated a multi-objective permutation flow-shop scheduling problem using an improved hybrid cuckoo search algorithm, focusing on makespan and total carbon emissions. Although their work also addresses production scheduling and environmental performance, the environmental objective differs from the total energy consumption objective used in this study. Zeng, et al. [17] examined distributed no-wait flow-shop scheduling with sequence-dependent setup times and considered makespan and total energy consumption using an improved NSGA-II algorithm. This study is particularly relevant because it considers both production completion time and energy consumption; however, its distributed scheduling environment and Pareto-based approach differ from the epsilon-constraint formulation adopted here. Yüksel, et al. [29] investigated a bi-objective no-wait permutation flow-shop scheduling problem involving total tardiness and total energy consumption. Furthermore, these results expand the literature on

permutation flow shop scheduling by demonstrating that the integration of the epsilon-constraint method and adaptive metaheuristics can serve as a more structured alternative compared to Pareto-based approaches, which generate numerous non-dominated solutions and require an additional selection process.

Taken together, these studies demonstrate that energy-related objectives can be incorporated into flow-shop scheduling through different optimization formulations. The present study differs primarily in its use of an epsilon-constraint approach and its comparative evaluation of BPBO, BWO, and GOA across three problem sizes within a common experimental framework. The reported results show that BPBO achieved the lowest energy consumption in the medium- and large-scale scenarios and the lowest reported computation time across all three scenarios. However, numerical results from the previous studies should not be compared directly with the present results because the benchmark instances, problem assumptions, objective functions, and computational settings differ. The comparison therefore clarifies the methodological similarities and differences rather than establishing a numerical ranking among algorithms evaluated under different conditions. To make these relationships explicit, Table 4 summarizes the problem formulations, optimization methods, objectives, and contributions of selected previous studies in relation to the present research.

Table 4. Comparison of Previous Studies and the Present Study

Study	Scheduling problem	Optimization method	Objective or research focus	Relationship to the present study
Ghasemi, et al. [13]	General optimization problems	Birds of Prey-Based Optimization (BPBO)	Metaheuristic optimization performance	Provides the methodological basis for BPBO; the present study evaluates its application to energy-efficient PFSP scheduling.
Tasgetiren, et al. [15]	Energy-efficient no-wait flow-shop scheduling	Discrete artificial bee colony and other multi-objective approaches	Total flow time and total energy consumption	Shares the energy-efficiency focus but differs in scheduling formulation and optimization framework.
Gu, et al. [16]	Multi-objective PFSP-related scheduling	Improved hybrid cuckoo search	Makespan and total carbon emissions	Shares the production-scheduling context but focuses on carbon emissions rather than total energy consumption.
Zeng, et al. [17]	Distributed no-wait flow-shop	Improved NSGA-II	Makespan and total energy consumption	Shares the energy and makespan focus but differs in

	scheduling with sequence-dependent setup times			scheduling assumptions, algorithm, and optimization formulation.
Yüksel, et al. [29]	Bi-objective no-wait permutation flow-shop scheduling	Mathematical programming and metaheuristic approaches	Total tardiness and total energy consumption	Shares the energy-minimization focus but considers total tardiness rather than a makespan constraint.
Present study	Energy-efficient permutation flow-shop scheduling	BPBO, compared with BWO and GOA	Minimize total energy consumption subject to an epsilon-based makespan constraint	Provides a comparative evaluation across three problem sizes within a common constrained optimization framework.

The comparison in Table 4 highlights the main contribution of the present study: the application and comparative evaluation of BPBO within an energy-efficient PFSP framework that minimizes total energy consumption subject to a predefined makespan limit. The study also compares BPBO with BWO and GOA across small-, medium-, and large-scale scenarios. The findings provide empirical evidence of BPBO's performance under the tested conditions, but they do not establish universal superiority over the algorithms proposed in previous studies because those algorithms were not all evaluated using the same benchmark instances and computational conditions

Theoretically, the results of this study are relevant to the principle of balancing exploration and exploitation in metaheuristic optimization. Deb, et al. [6] and Zitzler, et al. [8] provide a theoretical context for considering solution quality and diversity in multi-objective optimization. However, because the present study does not directly measure population diversity or compare Pareto-front quality, these concepts should be treated as theoretical context rather than as mechanisms empirically verified by the experiments. Furthermore, the use of the epsilon-constraint method provides an explicit structure for controlling the relationship between energy consumption and production completion time. Energy consumption is minimized as the primary objective while makespan is maintained as a constraint. This formulation is potentially useful when production planners must satisfy a specified completion-time requirement while seeking to reduce energy use. The contribution is therefore the application and evaluation of this constrained framework, rather than proof that it is universally superior to Pareto-based approaches.

In practical terms, these findings indicate that BPBO has potential for application in manufacturing systems seeking to improve energy efficiency while satisfying production completion requirements. Production planners could specify an acceptable makespan limit and use the optimization process to identify a schedule with lower

modeled energy consumption. The results for the medium- and large-scale scenarios suggest that BPBO is a promising candidate under the tested conditions. The lower reported computational-time values may also be beneficial when schedules need to be recalculated in response to changing production conditions, provided that the runtime measurements are verified and meet the response-time requirements of the intended application. However, practical deployment requires validation using industrial production data, realistic machine-energy profiles, and relevant operational constraints. Further analysis of machine idle time would be needed before attributing the observed energy reductions specifically to reduced idle periods or improved machine synchronization.

Several limitations should also be acknowledged. The experiments cover three problem scenarios, which may not represent the full range of PFSP scheduling conditions. In addition, inferential statistical tests and detailed sensitivity analyses of algorithm parameters and epsilon values are not reported. Consequently, the statistical significance, stability, and generalizability of the observed performance differences remain to be established. Future research should evaluate more benchmark instances, conduct repeated independent runs, report descriptive statistics such as the mean and standard deviation, and compare the algorithms using consistent computational settings. Sensitivity analyses of epsilon values and algorithm parameters would also help establish how changes in production completion requirements affect energy consumption and solution quality. Overall, BPBO achieved the lowest reported energy consumption in the medium- and large-scale scenarios and the lowest reported computation time across all three scenarios. Comparison with previous studies clarifies that the present research contributes an epsilon-constraint formulation and a comparative evaluation of BPBO, BWO, and GOA within a common PFSP framework. These findings support the potential of the proposed approach for energy-efficient scheduling under a makespan limit. Nevertheless, broader benchmark testing, statistical evaluation, and sensitivity analysis are required before stronger conclusions about robustness and generalizability can be drawn.

CONCLUSION

This study evaluates an energy-efficient permutation flow shop scheduling framework that combines the epsilon-constraint method with Birds of Prey-Based Optimization (BPBO) to minimize total energy consumption while satisfying specified makespan limits. BPBO was compared with Beluga Whale Optimization (BWO) and the Goat Optimization Algorithm (GOA) across three problem scenarios: 10 jobs and 6 machines, 50 jobs and 6 machines, and 100 jobs and 10 machines. The results indicate that all three algorithms achieved the same makespan of 144 and total energy consumption of 4047 in the small-scale scenario. In the medium-scale scenario, BPBO achieved a makespan of 547 and total energy consumption of 72245, compared with 550 and 72487 for BWO and 564 and 72656 for GOA. In the large-scale scenario, BPBO obtained a makespan of 1233 and total energy consumption of 533761, whereas BWO achieved 1234 and 540839, and GOA achieved 1246 and 545923, respectively. Thus, among the three algorithms evaluated, BPBO produced the lowest makespan and total energy consumption in the medium- and large-scale scenarios. The reported computational-time values were also lower for BPBO in all three scenarios, although this comparison should be interpreted in accordance with the verified time units and experimental conditions. These findings are limited to the tested instances and do not establish that BPBO is universally superior to other optimization algorithms. The epsilon-constraint method enables makespan to be imposed as a production-related constraint while total energy consumption is minimized.

All reported solutions met the specified makespan limits for their respective scenarios. The main contribution of this study is the empirical evaluation of BPBO against BWO and GOA within a common energy-efficient permutation flow shop scheduling framework. From a practical perspective, the findings suggest that BPBO is a promising alternative for the tested scheduling problems when both energy consumption and production completion limits are considered. Further validation is required before applying the framework to actual industrial operations. This study has several limitations, including the use of deterministic simulation data, the absence of inferential statistical testing, and the lack of an in-depth sensitivity analysis of the epsilon value and algorithm parameters. Consequently, the statistical significance and generalizability of the observed performance differences have not yet been established. Future research should evaluate the framework using real production data, conduct repeated independent runs and appropriate statistical tests, and examine the sensitivity of the results to epsilon values and algorithm parameters. Further investigations could also explore hybrid metaheuristics and machine-learning-based methods to evaluate their potential to improve solution quality and computational efficiency across a wider range of scheduling problems.

ACKNOWLEDGMENT

The author would like to express his gratitude to the academic supervisors at the Department of Industrial Engineering, University of Muhammadiyah Malang, for their invaluable dedication and guidance during this research.

REFERENCES

[1] N. Absi and W. van den Heuvel, "Worst case analysis of relax and fix heuristics for lot-sizing problems," *European Journal of Operational Research,* vol. 279, no. 2, pp. 449-458, 2019. doi: https://doi.org/10.1016/j.ejor.2019.06.010

[2] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Mathematics of operations research,* vol. 1, no. 2, pp. 117-129, 1976. doi: https://doi.org/10.1287/moor.1.2.117

[3] S. M. Johnson, "Optimal two-and three-stage production schedules with setup times included," *Naval research logistics quarterly,* vol. 1, no. 1, pp. 61-68, 1954. doi: https://doi.org/10.1002/nav.3800010110

[4] A. Z. Dragicevic, "Concentric framework for sustainability assessment," *Journal of Cleaner Production,* vol. 248, p. 119268, 2020. doi: https://doi.org/10.1016/j.jclepro.2019.119268

[5] C.-H. V. Chen and Y.-C. Chen, "Assessment of enhancing employee engagement in energy-saving behavior at workplace: an empirical study," *Sustainability,* vol. 13, no. 5, p. 2457, 2021. doi: https://doi.org/10.3390/su13052457

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation,* vol. 6, no. 2, pp. 182-197, 2002. doi: https://doi.org/10.1109/4235.996017

[7] G. Mavrotas, "Effective implementation of the ε-constraint method in multi-objective mathematical programming problems," *Applied mathematics computation,* vol. 213, no. 2, pp. 455-465, 2009. doi: https://doi.org/10.1016/j.amc.2009.03.037

[8] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *TIK report,* vol. 103, 2001. doi: https://doi.org/10.3929/ethz-a-004284029

[9] A. Mokhtari-Moghadam, T. T. Nguyen, and M. Mohsendokht, "Energy-aware flexible flow-shop scheduling for sustainable manufacturing: A multi-objective approach," *Process Integration Optimization for Sustainability,* pp. 1-16, 2026. doi: https://doi.org/10.1007/s41660-026-00705-0

[10] G. Mavrotas and K. Florios, "An improved version of the augmented ε-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems," *Applied mathematics computation,* vol. 219, no. 18, pp. 9652-9669, 2013. doi: https://doi.org/10.1016/j.amc.2013.03.002

[11] X.-S. Yang, "Nature-inspired optimization algorithms: Challenges and open problems," *Journal of computational science,* vol. 46, p. 101104, 2020. doi: https://doi.org/10.1016/j.jocs.2020.101104

[12] C. Zhong, G. Li, and Z. Meng, "Beluga whale optimization: A novel nature-inspired metaheuristic algorithm," *Knowledge-based systems,* vol. 251, p. 109215, 2022. doi: https://doi.org/10.1016/j.knosys.2022.109215

[13] M. Ghasemi *et al.*, "Birds of prey-based optimization (BPBO): A metaheuristic algorithm for optimization," *Evolutionary Intelligence,* vol. 18, no. 4, p. 88, 2025. doi: https://doi.org/10.1007/s12065-025-01052-8

[14] H. Nozari, H. Abdi, and A. Szmelter-Jarosz, "Goat optimization algorithm: a novel bio-inspired metaheuristic for global optimization," *arXiv preprint arXiv:.02331,* 2025. doi: https://doi.org/10.48550/arXiv.2503.02331

[15] M. F. Tasgetiren, D. Yüksel, L. Gao, Q.-K. Pan, and P. Li, "A discrete artificial bee colony algorithm for the energy-efficient no-wait flowshop scheduling problem," *Procedia Manufacturing,* vol. 39, pp. 1223-1231, 2019. doi: https://doi.org/10.1016/j.promfg.2020.01.347

[16] W. Gu, Z. Li, M. Dai, and M. Yuan, "An energy-efficient multi-objective permutation flow shop scheduling problem using an improved hybrid cuckoo search algorithm," *Advances in Mechanical Engineering,* vol. 13, no. 6, p. 16878140211023603, 2021. doi: https://doi.org/10.1177/16878140211023603

[17] Q.-q. Zeng, J.-q. Li, R.-h. Li, T.-h. Huang, Y.-y. Han, and H.-y. Sang, "Improved NSGA-II for energy-efficient distributed no-wait flow-shop with sequence-dependent setup time," *Complex Intelligent Systems,* vol. 9, no. 1, pp. 825-849, 2023. doi: https://doi.org/10.1007/s40747-022-00830-6

[18] D. M. Utama and C. Febrita, "Low-carbon no-idle permutation flow shop schedulling problem: giant trevally optimizer vs African vultures optimization algorithm," *Int J Adv Appl Sci,* vol. 12, no. 3, pp. 195-204, 2023. doi: https://doi.org/10.11591/ijaas.v12.i3.pp195-204

[19] K. Huang, R. Li, W. Gong, R. Wang, and H. Wei, "BRCE: bi-roles co-evolution for energy-efficient distributed heterogeneous permutation flow shop scheduling with flexible machine speed," *Complex Intelligent Systems,* vol. 9, no. 5, pp. 4805-4816, 2023. doi: https://doi.org/10.1007/s40747-023-00984-x

[20] P. Liang, P. Xiao, Z. Li, M. Luo, and C. Zhang, "A novel deep reinforcement learning-based algorithm for multi-objective energy-efficient flow-shop scheduling," *IET Collaborative Intelligent Manufacturing,* vol. 6, no. 4, p. e12121, 2024. doi: https://doi.org/10.1049/cim2.12121

[21] V. Pesaru, V. Saddikuti, and M. Janardhanan, "Hybrid Grey-Wolf Algorithm approach for Solving energy efficient permutation flowshop scheduling problem with Setup Times," *Procedia Computer Science,* vol. 241, pp. 352-359, 2024. doi: https://doi.org/10.1016/j.procs.2024.08.047

[22] D. M. Utama and C. N. Al Imron, "A systematic literature review on no-idle flow shop scheduling problem," in *Operations Research Forum*, 2024, vol. 5, no. 1, p. 18: Springer.

[23] Y. Lu, Y. Yuan, A. Sitahong, Y. Chao, and Y. Wang, "An Optimization Method for Green Permutation Flow Shop Scheduling Based on Deep Reinforcement Learning and MOEA/D," *Machines,* vol. 12, no. 10, p. 721, 2024. doi: https://doi.org/10.3390/machines12100721

[24] Y. M. Risma, D. M. U. Utama, and I. Amallynda, "Energy-efficient No-idle Flowshop Scheduling Optimization Using African Vultures Algorithm," *International Journal of Industrial Engineering Engineering Management,* vol. 6, no. 1, pp. 27-33, 2024. doi: https://doi.org/10.24002/ijieem.v6i1.8335

[25] Y. Lu, Y. Yuan, J. Yasenjiang, A. Sitahong, Y. Chao, and Y. Wang, "An optimized method for solving the Green Permutation Flow Shop Scheduling Problem using a combination of deep reinforcement learning and improved genetic algorithm," *Mathematics,* vol. 13, no. 4, p. 545, 2025. doi: https://doi.org/10.3390/math13040545

[26] T. Ma and C. Zhao, "An Enhanced Particle Swarm Optimization Algorithm for the Permutation Flow Shop Scheduling Problem," *Symmetry,* vol. 17, no. 10, p. 1697, 2025. doi: https://doi.org/10.3390/sym17101697

[27] X. Du and Y. Zhou, "A novel hybrid differential evolutionary algorithm for solving multi-objective distributed permutation flow-shop scheduling problem," *International Journal of Computational Intelligence Systems,* vol. 18, no. 1, p. 67, 2025. doi: https://doi.org/10.1007/s44196-025-00793-9

[28] K. Yin, Z. Li, M. Li, Y. Xue, and Y. Chen, "A Novel Imperialist Competitive Algorithm for Energy-Efficient Permutation Flow Shop Scheduling Problem Considering the Deterioration Effect of Machines," *Mathematics,* vol. 13, no. 24, p. 3973, 2025. doi: https://doi.org/10.3390/math13243973

[29] D. Yüksel, M. F. Taşgetiren, L. Kandiller, and L. Gao, "An energy-efficient bi-objective no-wait permutation flowshop scheduling problem to minimize total tardiness and total energy consumption," *Computers Industrial Engineering,* vol. 145, p. 106431, 2020. doi: https://doi.org/10.1016/j.cie.2020.106431

[30] Z. Dong, T. Ren, J. Weng, F. Qi, and X. Wang, "Minimizing the late work of the flow shop scheduling problem with a deep reinforcement learning based approach," *Applied Sciences,* vol. 12, no. 5, p. 2366, 2022. doi: https://doi.org/10.3390/app12052366

[31] K.-C. Ying, P. Pourhejazy, and C.-E. Sung, "Two-agent proportionate flowshop scheduling with deadlines: polynomial-time optimization algorithms," *Annals of Operations Research,* vol. 343, no. 1, pp. 543-558, 2024. doi: https://doi.org/10.1007/s10479-024-06275-z

[32] Y. Cai and H. Chen, "An improved salp swarm algorithm for permutation flow shop vehicle routing problem," *Scientific Reports,* vol. 15, no. 1, p. 6704, 2025. doi: https://doi.org/10.1038/s41598-025-86054-3

[33] H. Mimouni, A. Jalid, and S. Aqil, "Enhanced Optimization Strategies for No-Wait Flow Shop Scheduling with Sequence-Dependent Setup Times: A Hybrid NEH-GRASP Approach for Minimizing the Total Weighted Flow Time and Energy Cost," *Sustainability,* vol. 17, no. 17, p. 7599, 2025. doi: https://doi.org/10.3390/su17177599

[34] R. Sun *et al.*, "A Multiple-Objective Memetic Algorithm for the Energy-Efficient Scheduling of Distributed Assembly Flow Shops," *Symmetry,* vol. 18, no. 2, p. 315, 2026. doi: https://doi.org/10.3390/sym18020315

[35] M. Tsiftsoglou, Y. Marinakis, and M. Marinaki, "Optimizing the Classic and the Energy-Efficient Permutation Flowshop Scheduling Problem with a Hybrid Tyrannosaurus Rex Optimization Algorithm," *Biomimetics,* vol. 11, no. 4, p. 262, 2026. doi: https://doi.org/10.3390/biomimetics11040262

[36] M. Dai, D. Tang, A. Giret, M. A. Salido, and W. D. Li, "Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm," *Robotics computer-integrated manufacturing,* vol. 29, no. 5, pp. 418-429, 2013. doi: https://doi.org/10.1016/j.rcim.2013.04.001

[37] D. M. Utama, D. S. Widodo, and Informatics, "An energy-efficient flow shop scheduling using hybrid Harris hawks optimization," *Bulletin of Electrical Engineering,* vol. 10, no. 3, pp. 1154-1163, 2021. doi: https://doi.org/10.11591/eei.v10i3.2958

(This page is intentionally left blank)