JOURNAL OF MECHANICAL ENGINEERING AND SCIENCES
ISSN: 2289-4659 e-ISSN: 2231-8380
VOLUME 18, ISSUE 3, 2024, 10171 - 10180
DOI: https://doi.org/10.15282/jmes.18.3.2024.6.0803

RESEARCH ARTICLE

Energy-aware scheduling optimization in hybrid flow shops using artificial bee colony algorithm

M. A. H Osman[1], M. F. F. Ab Rashid[2]*, N. M. Z. Nik Mohamed[1], M. A. N. Mu'tasim[2]

[1] Faculty of Manufacturing and Mechatronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Malaysia
[2] Faculty of Mechanical and Automotive Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Malaysia
Phone: +6094316257; Fax.: +6094315017

ABSTRACT - Hybrid flow shop scheduling (HFS) involves optimizing production processes, where different manufacturing stages have varying capacities, combining parallel machine and flow shop scheduling to improve efficiency and reduce production time. Incorporating energy considerations into HFS problems has emerged as a critical area of research, driven by the growing emphasis on environmental sustainability and cost-effectiveness in manufacturing operations. This study addresses the hybrid flow shop scheduling with energy consideration (HFSE) problem, aiming to simultaneously optimize makespan and total energy consumption, two conflicting objectives. An Artificial Bee Colony (ABC) algorithm is proposed as an effective solution methodology for tackling the HFSE problem. Through an extensive computational experiment involving a well-known benchmark suite, the ABC algorithm demonstrated remarkable performance, consistently outperforming several popular metaheuristic algorithms, including Genetic Algorithms, Particle Swarm Optimization, Memetic Algorithms, and Whale Optimization Algorithm in 75% of the problems. The proposed approach's ability to efficiently explore the search space and balance the trade-offs between makespan minimization and energy consumption reduction contributed to its superior results. The ABC algorithm reduces makespan and energy consumption by 2.95% and 3.43%, respectively. This finding suggests potential benefits for manufacturing operations, including decreased production time and lower operational costs.

ARTICLE HISTORY

Received	:	03rd June 2024
Revised	:	05th Sept. 2024
Accepted	:	14th Sept. 2024
Published	:	30th Sept. 2024

KEYWORDS
Production scheduling
Hybrid flow shop
Artificial bee colony
Energy optimization

1. INTRODUCTION

Hybrid flow shop scheduling (HFS) is a production scheduling problem that combines elements of flow shop and parallel machine scheduling. In a flow shop, jobs must pass through multiple production stages in a specific order, ensuring a streamlined and sequential process [1]. On the other hand, parallel machine scheduling involves assigning jobs to multiple identical machines operating at the same stage, allowing for the simultaneous processing of different tasks [2]. This hybrid approach integrates the sequential nature of flow shops with the flexibility of parallel machine scheduling, making it suitable for various manufacturing environments [3]. Historically, HFS has been a well-researched area, with numerous studies addressing different aspects of the problem [4]. These studies have explored various strategies and algorithms to optimize scheduling to improve efficiency and reduce production times in industries such as electronics, automotive parts, plastics, and garments [5-7]. The primary focus of these traditional approaches has been minimizing makespan, reducing job tardiness, and maximizing machine utilization [8]. Recently, researchers have begun integrating energy considerations into HFS, driven by the growing importance of environmental sustainability, cost savings, and regulatory compliance [9]. This new dimension, known as Hybrid Flow Shop Scheduling with Energy Consideration (HFSE), aims to optimize the production schedule and the energy consumption associated with manufacturing processes. Including energy factors reflects a broader awareness of industrial operations' environmental impact and economic implications. However, integrating energy considerations into the HFS problem significantly increases complexity [10]. Unlike traditional flow shop scheduling, HFSE necessitates a meticulous investigation to identify suitable and efficient algorithms for optimization. The dual objectives of minimizing makespan (total completion time) and optimizing energy consumption stand a remarkable challenge [11]. Notably, as the problem size grows, finding optimal solutions becomes exponentially time-consuming due to the vast search space. Researchers continue exploring innovative approaches to balance production efficiency and energy conservation within the HFSE framework.

The literature explores using Genetic Algorithms (GA) to optimize different objectives in the hybrid flow shop scheduling problem while accounting for energy consumption. Meng et al. [9] focused on minimizing total energy usage by developing an Improved Genetic Algorithm (IGA) that incorporated a new energy-conscious decoding method [9]. This approach aimed to enhance the energy efficiency of the scheduling solution. Schulz tackled the problem from a multiobjective perspective, simultaneously optimizing production, energy, and transportation costs [12]. Their GA implementation utilized a detailed matrix encoding procedure to address the non-deterministic polynomial-time hard (NP- hard) complexity, intelligent swap, and right-shifting procedures to enhance the algorithm's performance. More recently, Lu et al. [13] proposed an Efficient Adaptive Genetic Algorithm (EAGA) that employed layered strategies and an enhanced adaptive adjustment method. This approach aimed to balance exploration and exploitation, ultimately leading to better convergence and higher-quality scheduling solutions while considering energy consumption. Jiang et al. [14] proposed an Energy-Oriented Multiobjective Optimization (EOMO) algorithm based on the Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) framework to minimize tardiness and energy usage simultaneously. Their approach incorporated an external archive population (EAP), local search, and a discrete-event system (DES) simulation procedure to enhance algorithmic efficiency. More recently, researchers tackled the problem of optimizing production efficiency and energy consumption using an improved MOEA/D algorithm (IMOEA/D) [15]. Their contributions included a discrete integer encoding scheme, a heuristic decoding approach based on processing paths, and the introduction of a variable neighborhood search mechanism to enhance the local search capability, ultimately leading to better convergence and solution quality. Zhang et al. [16] tackled the multiobjective optimization of makespan and energy consumption using a novel particle swarm optimization (PSO) algorithm designed for efficient global search. Their approach employed a multi-group strategy to enhance the exploration capability. Additionally, two local search strategies were incorporated to exploit problem-specific knowledge. Notably, a Q-learning mechanism guided a variable neighborhood search, balancing exploration and exploitation during the optimization process. Luo [17] proposed a new multiobjective ant colony optimization (MOACO) algorithm to simultaneously optimize production efficiency and electric power costs. Their approach introduced a right-shift procedure to adjust the start times of operations, aiming to improve the quality of solutions. This work demonstrated the applicability of swarm intelligence techniques to the energy-aware hybrid flow shop scheduling problem.

Geng et al. [18] focused on minimizing makespan and energy consumption through an improved memetic algorithm. Their contributions included three heuristic rules for population initialization and various neighborhood search methods. These enhancements aimed to balance diversification and intensification, leading to better convergence and solution quality. Cai et al. [19] explored a shuffled frog-leaping algorithm (SFLA) for the multiobjective optimization of makespan and energy. Their cooperated SFLA approach featured an effective cooperation process between the best and worst memeplexes based on evaluation results. Additionally, an adaptive population shuffling mechanism was adopted to enhance search efficiency, potentially leading to improved scheduling solutions. In 2023, Wang et al. [20] proposed an improved multiobjective firefly algorithm (MO-FA) to simultaneously optimize makespan, energy consumption, and production stability in the hybrid flow shop scheduling environment. Their contributions included the implementation of a population updating rule inspired by the location updating law of fireflies, coupled with a variable neighborhood search mechanism to avoid premature convergence to local optima. They also introduced fast, non-dominated sorting and elite individual reserving strategies to effectively guide the population evolution process. By incorporating these enhancements, the improved MO-FA demonstrated superior performance in navigating the complex trade-offs between the three conflicting objectives, ultimately providing high-quality scheduling solutions that balanced makespan minimization, energy efficiency, and production stability. Despite a growing number of research addressing energy utilization in HFS, the proportion of studies focusing on energy considerations remains relatively small compared to the extensive work on traditional HFS problems. Additionally, the inherent complexity of the HFSE problem, characterized by its combinatorial nature, discrete decision variables, and conflicting objectives, explains the need for more efficient and robust optimization approaches. Existing methods often fall short of effectively balancing these complexities, highlighting the necessity for further exploration of advanced algorithms to achieve better performance in optimizing makespan and energy consumption. This paper investigates the performance of the Artificial Bee Colony (ABC) algorithm in optimizing HFSE. Inspired by bees' foraging behavior, ABC has been successfully applied to various problems [21]. However, its application to combinatorial problems remains limited. The main contribution of this work is developing an efficient computational model and algorithm capable of simultaneously optimizing both makespan and energy utilization.

2. MATERIALS AND METHODS

2.1 Hybrid Flow Shop Scheduling with Energy Consumption

Hybrid flow shop scheduling involves processing *n* jobs through, *S* stages, whereby each stage may consist of multiple machines capable of performing similar operations but with different capacities. All the jobs must strictly follow the same processing flow. When the energy factor is considered an objective function or a constraint, the problem is known as HFSE. In hybrid flow shop scheduling, *n* jobs are processed through *S* stages. Figure 1 illustrates an example of the HFSE problem, where jobs j_1 through j_n are processed through three stages: turning (Stage 1) with two machines, milling (Stage 2) with one machine, and cutting (Stage 3) with three machines. The term M_{sk} used in Figure 1 refers to the k^{th} machine on stage, *s*. Each machine at a stage has different capabilities, which affect the processing times. Each job must be processed on only one machine at each stage. Consequently, the machine assignment decision directly impacts the processing time and the makespan. Additionally, since machines may have different power rates, the machine assignment also affects the total energy consumption in the process.

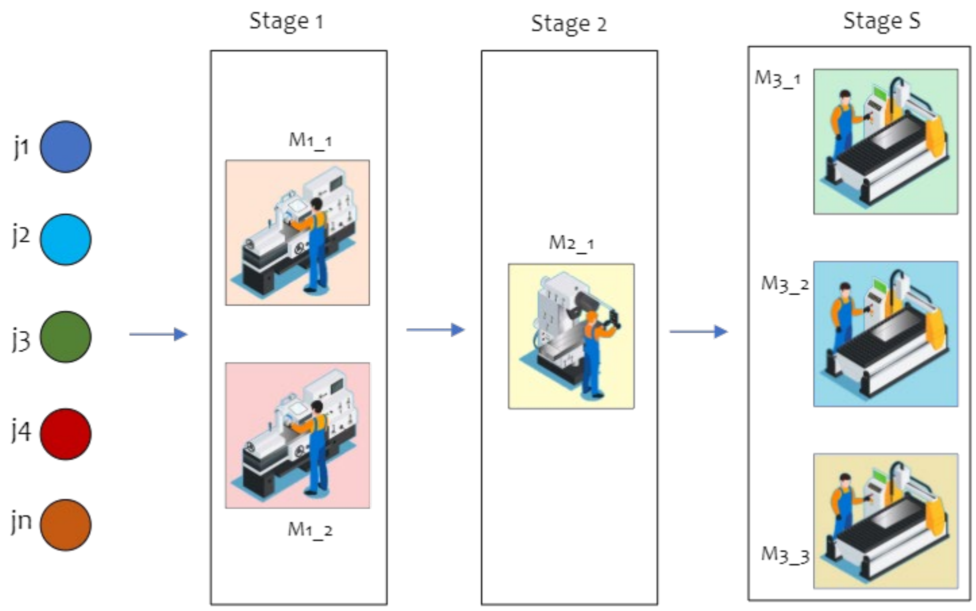

Figure 1. Production layout of hybrid flow shop scheduling

In HFSE, processing times for each task on each machine are known and fixed. During task execution, no interruptions or pre-emptive actions are allowed, ensuring that each job is completed without breaks. Each machine can handle only one job at a time, maintaining exclusive machine-job assignments. At each process stage, only one machine is permitted to handle each job, adhering to a strict one-to-one machine-job correspondence throughout the scheduling stages. The HFSE problem involves sequencing a set of jobs through a series of processing stages, where each job must strictly follow the predetermined sequence of processes. In this problem, process bypass is permissible, allowing jobs to skip unnecessary processes, which can lead to more efficient scheduling. Additionally, the setup times between different jobs on the same machine are considered negligible, simplifying the scheduling process. All jobs are available for scheduling from the start, as there are no constraints on job release times. Furthermore, all machines are accessible and ready for operation at the beginning of the scheduling period. Notably, during idle periods, the machines are assumed to be turned off, eliminating the need to consider idle energy consumption. These characteristics of the HFS problem encompass the key assumptions and constraints that must be considered when developing scheduling algorithms and optimizing the system's overall performance. Several additional characteristics should be considered in the HFSE problem. Firstly, there is an unlimited buffer between consecutive processes, allowing for the temporary storage of jobs as they move from one process to the next. This flexibility can facilitate more efficient scheduling and prevent bottlenecks caused by limited buffer capacities. Furthermore, there are no constraints on job due dates, meaning that jobs can be completed within any reasonable timeframe without prioritizing certain jobs over others based on strict deadlines. The problem also imposes no limitations on the number of jobs, machines, or stages in the production process, allowing for considering diverse manufacturing scenarios with varying complexities. Finally, transportation times between machines or stages are considered negligible, simplifying the scheduling calculations and enabling a focus on the processing times and resource allocation aspects of the problem.

The following HFSE mathematical model is applied to represent and optimize the problem. The optimization objectives are to minimize:

$$f_1 = \min C_{max} \tag{1}$$

$$f_2 = \min TEC \tag{2}$$

$$C_{max} = \max \{C_j\}, \qquad j = 1,2,\dots,n \tag{3}$$

$$TEC = \sum_{j=1}^{J}\sum_{s=1}^{S}\sum_{m=1}^{M} t_{j,s,m}p_{s,m}y_{j,s,m} \qquad y_{j,s,m} = \begin{cases} 1, & \text{If job } j \text{ is processed at machine } m \text{ on stage } s \\ 0, & \text{Otherwise} \end{cases} \tag{4}$$

Based on Eqs. (1) – (4), J denotes the total number of jobs that need to be scheduled, while S represents the total number of stages in the production process. M_s indicates the number of machines available at a specific stage, s. The processing time of job j on machine m at stage, s is represented by $t_{j,s,m}$, measured in minutes. C_j represents the completion time of job j in minutes, which is a crucial metric for evaluating the scheduling performance. C_{max}, also known as the makespan, is the maximum completion time among all jobs and is typically the primary objective to be minimized in HFS problems. The power rate for machine m at stage s is given by p_{sm} and is measured in Watts. This parameter is relevant when considering energy consumption or environmental impact as part of the scheduling objectives or constraints. The model presented in Eqs. (1) through (4) forms the basis of the computational model used in this work. It will determine the accuracy of the results obtained in the computational experiments section.

2.2 Artificial Bee Colony Algorithm

The Artificial Bee Colony algorithm is a population-based optimization technique inspired by the foraging behavior of honeybees to solve optimization problems [22]. Each phase plays a vital role: initialization sets the stage, evaluation assesses solution quality, employed bees refine solutions, onlooker bees enhance exploitation, and scout bees ensure continuous exploration [23]. Together, these phases contribute to the algorithm's robustness and effectiveness in finding optimal solutions. The flowchart of the ABC algorithm is presented in Figure 2.

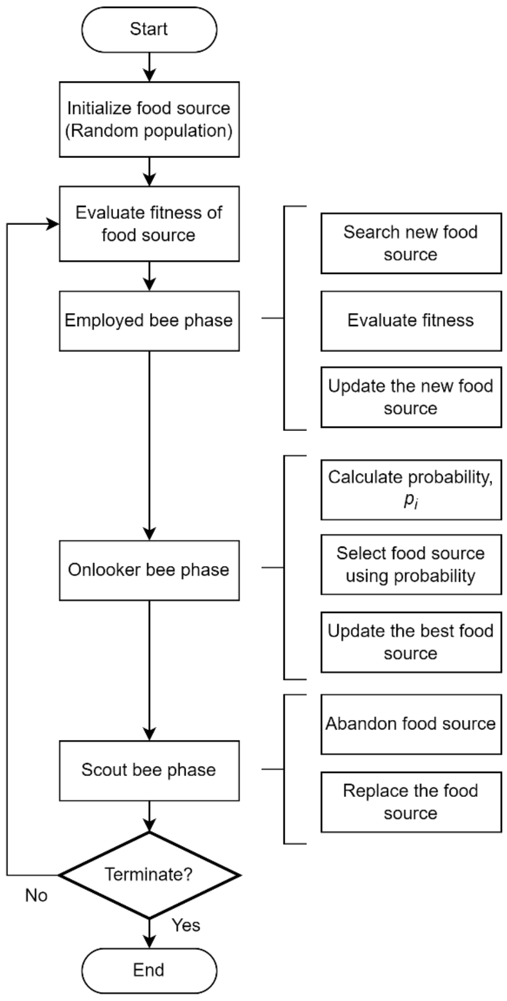
Start
Initialize food source (Random population)
Evaluate fitness of food source
Employed bee phase
Search new food source
Evaluate fitness
Update the new food source
Onlooker bee phase
Calculate probability, p_i
Select food source using probability
Update the best food source
Scout bee phase
Abandon food source
Replace the food source
Terminate?
No
Yes
End

Figure 2. Flow chart of ABC algorithm

2.2.1 Initialization phase

The algorithm's parameters, such as population size and the number of iterations, are defined in the initialization phase. Typically, the population size ranges from 20 to 50. Random solutions, referred to as food sources, are generated within the lower and upper bounds of the problem's search space. These initial solutions are crucial as they set the starting point for the algorithm's exploration and exploitation of the search space.

2.2.2 Evaluation phase

Once the initial solutions are generated, they undergo evaluation. Each solution is assessed using a fitness function, as Section 2 of the relevant study details. This function helps determine the quality of each solution. Solutions that meet or exceed certain criteria are considered good food sources. This filtering process ensures that only promising solutions are carried forward for further improvement. The fitness function for solution x_i is denoted as fit_i and is computed in the following manner:

$$fit_i = \begin{cases} \dfrac{1}{1+f_i} & \text{if } f_i \geq 0 \\ 1 + abs(f_i) & \text{if } f_i < 0 \end{cases} \tag{5}$$

2.2.3 Employed phase

In the employed bee phase, each employed bee searches for new food sources based on information gathered during the evaluation phase. In the optimization context, this phase involves generating new candidate solutions by modifying existing ones. The fitness of these new solutions is then calculated, and a greedy selection process is applied. This means that if the new solution has better fitness than the previous one, it replaces the old solution. This phase is critical for refining solutions and improving their quality iteratively.

$$v_{i,j} = x_{i,j} + \Phi_{ij}(x_{i,j} - x_{k,j}) \tag{6}$$

The notation $x_{i,j}$ denotes the ith solution for the jth dimension. The variable Φ represents a random number within the range of [-1,1]. Additionally, x_k denotes a solution selected at random from the current population.

2.2.4 Onlooker bees phase

Unemployed bees are divided into two groups: onlooker bees and scout bees. Employed bees share information about their food sources with onlooker bees. The onlooker bees then choose a food source based on a probability related to the fitness of the solutions. Techniques such as the roulette wheel selection method are often used. Greedy selection is again applied, with more onlooker bees recruited to richer food sources. This phase enhances the algorithm's ability to exploit good solutions by focusing more resources on them. Next, the probability value, p_i for solution x_i, is calculated using the equation below. According to this equation, a higher fitness of the solution will result in a greater probability value.

$$P_i = \frac{fit_i}{\sum_{i=1}^{N_{swarm}/2} fit_i} \tag{7}$$

2.2.5 Scout bees phase

Scout bees, a subset of the unemployed bees, choose their food sources randomly. This phase is activated when certain solutions do not improve over a specified number of iterations, known as Lmax. These stagnant solutions are replaced by new, randomly generated solutions within the problem's bounds. This ensures that the search space is continuously explored, preventing the algorithm from becoming stuck in local optima and promoting the discovery of potentially better solutions.

3. RESULTS AND DISCUSSION

An extensive computational experiment has been conducted to evaluate the performance of the proposed ABC algorithm for solving the HFSE problem. The well-known benchmark test suite originally introduced by Neron and Carlier for the classical HFS problem has been adopted [24]. This set comprises 12 distinct problem instances, encompassing scenarios with 10 jobs and 5 stages, 10 jobs and 10 stages, 15 jobs and 5 stages, and 15 jobs and 10 stages. This benchmark problem is widely recognized and has been extensively used to assess the performance of algorithms on HFS problems, thereby facilitating a standardized comparison with existing approaches. Four other popular metaheuristic algorithms have been selected to benchmark the proposed ABC algorithm: the GA, PSO, MA, and WOA. These algorithms are well-established nature-inspired optimization techniques that have demonstrated impressive performance across a wide range of combinatorial optimization problems [2-4]. On the other hand, WOA is a relatively newer bio-inspired algorithm proposed by Mirjalili and Lewis [5], which has gained significant attention due to its promising results on various complex optimization tasks. In order to ensure an unbiased comparison, all five algorithms were configured with identical parameter settings. Specifically, a population size of 30 individuals (candidate solutions) and a maximum iteration limit of 500 were used. To mitigate the potential effects of pseudo-randomness inherent in the stochastic nature of these algorithms, each problem instance was independently solved 20 times with different random seeds. The average fitness value across these 20 runs was recorded as the final performance metric for that particular instance. This approach of multiple independent runs helps to capture the algorithms' behavior more reliably and reduces the influence of random variations. Table 1 presents the average fitness results obtained from the computational experiment across different problem instances. On the other hand, Table 2 indicates the rank of the algorithms for each of the problem instances. The ranking is based on the average fitness values, where the algorithm with the best (minimum) average fitness is assigned rank 1, while the algorithm with the worst (maximum) average fitness is given rank 5.

The ABC algorithm exhibits outstanding overall performance, consistently securing the top rank in 9 out of the 12, or 75% of the problems. This remarkable consistency demonstrates the algorithm's efficiency and suitability for tackling most of the problems considered in this study. The MA also emerges as a strong contender, achieving ranks 2 or 3 across all problem instances, reflecting its reliable and consistent performance. In contrast, the GA and PSO algorithms demonstrate moderate performance. While GA secures the top rank in a few instances, its rankings exhibit variability across problems, indicating a degree of inconsistency. Similarly, PSO generally occupies the middle ranks, suggesting a lack of consistency compared to the top-performing algorithms. Notably, the WOA consistently ranks the lowest, indicating its relative inefficiency for the problems considered in this study. A detailed analysis of individual problems reveals in-depth observations of each algorithm's performance profile. The ABC algorithm demonstrates exceptional efficacy in Problem 1, claiming the top rank, followed closely by the MA and WOA. Meanwhile, the GA and PSO underperform compared to ABC. The ABC maintains its superiority in Problem 2, with MA as a close contender. GA and PSO exchange positions, highlighting their sensitivity to specific problem characteristics. Problem 3 sees an unexpected shift, with GA surpassing all others, while ABC and WOA deliver moderate performances, and PSO and MA are unexpectedly left behind. ABC consistently dominates for the remaining Problems 4 through 12, closely trailed by MA. GA and PSO display inconsistent rankings, indicating variable efficiency across different scenarios. WOA, however, consistently occupies the lowest rank throughout these problem instances.

Table 1. Average fitness from optimization

Problem No.	GA	PSO	MA	ABC	WOA
1	0.2683	0.2637	0.2407	0.2389	0.2549
2	0.2367	0.2470	0.2292	0.2240	0.2467
3	0.2312	0.2520	0.2388	0.2320	0.2543
4	0.2233	0.2396	0.2265	0.2199	0.2456
5	0.2712	0.2766	0.2701	0.2617	0.2894
6	0.2783	0.2688	0.2738	0.2638	0.2956
7	0.2503	0.2639	0.2746	0.2568	0.2936
8	0.2996	0.3157	0.3030	0.2984	0.3223
9	0.3546	0.3692	0.3604	0.3544	0.3746
10	0.2814	0.2900	0.3241	0.3195	0.3407
11	0.2953	0.2900	0.2853	0.2817	0.3098
12	0.2752	0.2586	0.2578	0.2425	0.2829

Table 2. Optimization algorithm ranks according to Table 1

Problem No.	GA	PSO	MA	ABC	WOA
1	5	4	2	1	3
2	3	5	2	1	4
3	1	4	3	2	5
4	2	4	3	1	5
5	3	4	2	1	5
6	4	2	3	1	5
7	1	3	4	2	5
8	2	4	3	1	5
9	2	4	3	1	5
10	1	2	4	3	5
11	4	3	2	1	5
12	4	3	2	1	5
Average Rank	2.67	3.50	2.75	1.33	4.75

Several notable trends and observations can be drawn from the performance data. The ABC algorithm demonstrates remarkable consistency and superiority across most problems, making it a strong candidate for optimization tasks in similar contexts. Its ability to efficiently explore the search space and effectively balance multiple objectives contributes to its outstanding performance. The MA algorithm shows reliable performance, often ranking just behind ABC, suggesting that it is also a good choice when considering consistency and robustness across diverse problem instances. The GA and PSO algorithms exhibit varied performance, with GA sometimes achieving the best rank but generally fluctuating in its rankings across problems. PSO tends to occupy the middle ranks, indicating a lack of consistent performance compared to the top-performing algorithms. The WOA algorithm consistently ranks lowest across all problem instances, indicating that it may not be well-suited for the types of problems considered in this study or may require further tuning and enhancements to improve its performance in this specific context.

Overall, the computational results highlight the proposed ABC algorithm's superiority and the MA algorithm's reliable performance for the problems considered in this study. While GA and PSO demonstrate moderate performance, their inconsistencies across different problem instances suggest room for improvement. The WOA algorithm, being a relatively newer approach, exhibits limited effectiveness for the specific optimization tasks explored here, necessitating further investigation and potential enhancements.

Table 3. Optimization results of C_{max} and TEC

Problem No.	Indicator	GA	PSO	MA	WOA	ABC
1	Average C_{max}	150.3	150.4	149.7	147.4	146
	Min C_{max}	146	146	146	146	146
	Average TEC	9.65	9.28	8.80	9.37	8.50
	Min TEC	8.44	8.80	8.46	8.76	8.40
2	Average C_{max}	104.9	108.2	103.2	107	103
	Min C_{max}	103	103	103	103	103
	Average TEC	9.56	9.71	9.31	9.92	8.95
	Min TEC	8.81	9.05	9.00	9.40	8.69
3	Average C_{max}	84.3	89.4	86.4	91.2	83.8
	Min C_{max}	69	79	82	80	81
	Average TEC	8.49	9.48	8.80	9.34	8.67
	Min TEC	7.35	8.64	8.12	7.71	7.93
4	Average C_{max}	80.3	86.6	81.5	88.2	79
	Min C_{max}	72	73	75	84	76
	Average TEC	5.02	5.31	5.08	5.53	4.95
	Min TEC	4.02	4.27	4.62	5.07	4.20
5	Average C_{max}	192.5	193.9	191.6	197.2	188.7
	Min C_{max}	177	183	188	191	186
	Average TEC	18.26	18.55	18.52	19.24	18.47
	Min TEC	16.44	17.11	17.57	17.95	17.35
6	Average C_{max}	197.4	193.1	194.7	201.4	190.3
	Min C_{max}	181	179	190	191	186
	Average TEC	17.03	17.49	17.61	18.20	18.02
	Min TEC	15.21	16.20	16.73	17.11	17.38
7	Average C_{max}	150.7	152.6	168.1	160	151.1
	Min C_{max}	136	141	152	152	145
	Average TEC	17.20	18.28	18.09	19.04	17.98
	Min TEC	15.66	16.57	17.32	17.74	16.65
8	Average C_{max}	192.6	194.2	193.5	194.9	191.2
	Min C_{max}	191	191	191	191	191
	Average TEC	11.13	12.21	11.87	12.65	11.33
	Min TEC	10.58	11.50	11.25	11.47	11.03
9	Average C_{max}	221.4	222.1	223.2	223	220
	Min C_{max}	220	220	220	220	220
	Average TEC	9.60	10.75	10.48	11.04	9.89
	Min TEC	9.18	10.27	9.73	9.99	9.51
10	Average C_{max}	114.4	115.9	129.6	128.5	124.1
	Min C_{max}	101	110	120	124	121
	Average TEC	11.00	11.33	11.77	12.11	11.52
	Min TEC	9.72	10.58	10.70	10.94	10.88
11	Average C_{max}	261.1	257.9	277.3	261.7	253.4
	Min C_{max}	242	250	251	253	249
	Average TEC	29.96	30.33	31.66	31.78	30.75
	Min TEC	28.68	29.05	29.88	30.01	29.73
12	Average C_{max}	255.3	248.7	246.9	255.1	243.4
	Min C_{max}	247	235	241	237	241
	Average TEC	29.17	29.16	29.68	30.32	28.76
	Min TEC	26.36	27.43	28.68	28.16	27.42

Table 3 presents the results of the computational experiment in terms of the individual objective function values: the minimized makespan, C_{max}, and the minimized total energy consumption (TEC). Noted that the C_{max} is in minutes, while TEC is in the Watt unit. The computational results highlight the remarkable performance of the proposed ABC algorithm in optimizing both makespan, C_{max}, and TEC objectives. Across most problem instances, ABC achieves the lowest average Cmax values, signifying its effectiveness in minimizing the overall completion time of jobs. Across all problem instances, the ABC achieved an average Cmax reduction of 2.95% compared to all algorithms. The highest reduction was 4.7% compared to the WOA, while the lowest reduction was 1.17% compared to GA. This impressive makespan efficiency demonstrates ABC's ability to generate schedules that ensure timely task completion, a critical factor in many real-world manufacturing and production scenarios. Additionally, the GA and MA also exhibit strong performance in terms of makespan optimization. Their average Cmax values often come close to those achieved by ABC, suggesting that these algorithms are capable of achieving low completion times under various problem conditions. This consistent performance across different instances underscores the robustness and reliability of GA and MA in minimizing makespan. In contrast, the PSO and WOA generally show higher average Cmax values compared to the top-performing algorithms. This indicates that PSO and WOA are relatively less effective in minimizing completion times, which may limit their applicability in time-critical scheduling scenarios.

Turning to the TEC objective, the ABC algorithm once again excels by frequently obtaining the lowest average and minimum TEC values across multiple problem instances. This dual efficiency in time and energy optimization highlights ABC's remarkable capability to generate schedules that balance productivity and energy efficiency. For TEC, the ABC algorithm achieved an average reduction of 3.43%. Despite its strong performance, the energy consumption achieved by the GA was slightly lower than ABC, with a difference of only 0.14%. By minimizing energy consumption while maintaining low completion times, ABC emerges as a highly desirable solution for energy-aware scheduling problems. The GA also demonstrates competitive performance in reducing TEC, often achieving the lowest or near-lowest TEC values in several problem instances. This strong energy efficiency and respectable makespan performance make GA a formidable contender for generating energy-efficient schedules without compromising productivity. The MA provides a balanced performance, with respectable TEC values that, while not as low as those of ABC and GA, still contribute to energy savings. This characteristic of MA may be advantageous in scenarios where a trade-off between energy efficiency and other objectives is required. In contrast to the top-performing algorithms, PSO and WOA generally exhibit higher TEC values, indicating their relatively lower efficiency in minimizing energy consumption. This limitation may restrict their applicability in energy-conscious scheduling contexts, where minimizing completion time and energy usage is critical.

Overall, the computational results highlight the superiority of the ABC algorithm in optimizing both makespan and energy consumption objectives simultaneously. The consistent performance of GA and MA in either makespan or energy optimization, or both, also establishes them as viable alternatives depending on the specific requirements and priorities of the scheduling problem. There are a few factors that contributed to the superior ABC performance. One of the factors is its simplicity and ease of implementation. The straightforward structure reduces the complexity of coding and implementation, making the algorithm accessible even to those with limited experience in optimization algorithms. The reduced number of parameters, such as the number of employed and onlooker bees and the scout bee limit, simplifies the setup and minimizes the need for extensive parameter tuning. Besides that, the ABC also effectively balances exploration and exploitation through its distinct bee roles [25]. Employed bees focus on refining existing solutions, while onlooker bees use information from employed bees to further enhance these solutions. Scout bees contribute to exploration by introducing new random solutions when current solutions become stagnant. This balance helps maintain an effective search process, ensuring that the algorithm explores new regions while also refining known good solutions. This algorithm is also known for its efficient global search capability, facilitated by scout bees that explore new regions when existing solutions fail to improve [26]. This mechanism helps prevent the algorithm from getting stuck in local optima by diversifying the search and enhancing its ability to find global optima. ABC's global search efficiency is crucial for handling complex optimization problems with multiple local optima. Finally, the dynamic population management in the ABC algorithm is a significant feature that helps maintain diversity and avoid premature convergence [27]. When a solution becomes stagnant, scout bees introduce new random solutions, which prevents the population from becoming homogeneous and ensures continued exploration of the search space. This dynamic approach enhances the algorithm's ability to find optimal solutions by avoiding local optima and maintaining diverse solutions.

4. CONCLUSIONS

Integrating energy considerations into the hybrid flow shop scheduling problem, termed HFSE, has emerged as a critical area of research driven by environmental sustainability, cost savings, and regulatory compliance goals. This study explored the application of the ABC algorithm for addressing the multiobjective HFSE problem, aiming to minimize makespan and total energy consumption simultaneously. Through an extensive computational experiment involving a well-established benchmark suite, the proposed ABC algorithm demonstrated remarkable performance, consistently securing top ranks across the majority of problem instances. The algorithm's ability to efficiently explore the search space and effectively balance the conflicting objectives of makespan minimization and energy consumption reduction contributed to its outstanding results. Comparative analyses with other popular metaheuristic algorithms, including GA, PSO, MA, and the WOA, further highlighted the superiority of the ABC approach. While GA and MA exhibited reliable performance in either makespan or energy optimization, PSO and WOA were relatively less effective, indicating potential limitations in their applicability to HFSE problems. The computational results underscore the significance of the ABC algorithm as a powerful optimization technique for energy-aware scheduling in hybrid flow shop environments. According to the results of the computational experiments, the ABC algorithm successfully reduced the total makespan and energy utilization simultaneously by 2.95% and 3.43%, respectively. By generating schedules that strike an optimal balance between productivity and energy efficiency, the proposed approach addresses the growing demand for sustainable and cost-effective manufacturing practices.

Future research could explore the integration of additional real-world constraints and objectives, such as machine availability, setup times, and production costs, into the HFSE framework. Furthermore, developing parallel or distributed implementations of the ABC algorithm could enhance its computational efficiency, enabling its application to larger-scale and more complex scheduling problems encountered in modern manufacturing settings. Overall, this study contributes to advancing energy-conscious scheduling methodologies and paves the way for adopting efficient and sustainable practices in hybrid flow shop manufacturing environments.

ACKNOWLEDGEMENTS

The Universiti Malaysia Pahang Al-Sultan Abdullah supported this work under the internal research grant RDU223017.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

AUTHORS CONTRIBUTION

M. A H. Osman (Computational experiment; Result analysis; Draft writing)

M. F. F. Ab. Rashid (Original coding; Writing improvement; supervision)

N. M. Z. Nik Mohamed (Supervision)

M. A. N. Mutasim (Code troubleshooting; writing improvement)

AVAILABILITY OF DATA AND MATERIALS

The data supporting this study's findings are available on request from the corresponding author.

ETHICS STATEMENT

Not applicable

REFERENCES

[1] M. Khatami, A. Salehipour, T. C. E. Cheng, "Flow-shop scheduling with exact delays to minimize makespan," *Computers & Industrial Engineering*, vol. 183, p. 109456, 2023.

[2] X. An, G. Si, T. Xia, D. Wang, E. Pan, L. Xi, "An energy-efficient collaborative strategy of maintenance planning and production scheduling for serial-parallel systems under time-of-use tariffs," *Applied Energy*, vol. 336, p. 120794, 2023.

[3] Y. J. Wang, G. G. Wang, F. M. Tian, D. W. Gong, W. Pedrycz, "Solving energy-efficient fuzzy hybrid flow-shop scheduling problem at a variable machine speed using an extended NSGA-II," *Engineering Applications of Artificial Intelligence*, vol. 121, p. 105977, 2023.

[4] I. Ribas, R. Leisten, J. M. Framiñan, "Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective," *Computers & Operations Research*, vol. 37, no. 8, pp. 1439–1454, 2010.

[5] J. Trojanowska, J. Machado, M. L. R. Varela, S. Carmo-Silva, N. M. L. Costa, "Comparative simulation study of production scheduling in the hybrid and the parallel flow," *Management and Production Engineering Review*, vol. 1, no. 2, pp. 10 – 20, 2017.

[6] Y. Kuang, X. Wu, Z. Chen, W. Li, "A two-stage cross-neighborhood search algorithm bridging different solution representation spaces for solving the hybrid flow shop scheduling problem," *Swarm and Evolutionary Computation*, vol. 84, p. 101455, 2024.

[7] S. Wang, H. Zhang, "A metaheuristic for flowshop scheduling with batch processing machines in textile manufacturing," *Applied Soft Computing*, vol. 145, p. 110594, 2023.

[8] J. S. Neufeld, S. Schulz, U. Buscher, "A systematic review of multiobjective hybrid flow shop scheduling," *European Journal of Operational Research*, vol. 309, no. 1, pp. 1–23, 2023.

[9] L. Meng, C. Zhang, X. Shao, Y. Ren, C. Ren, "Mathematical modelling and optimisation of energy-conscious hybrid flow shop scheduling problem with unrelated parallel machines," *International Journal of Production Research*, vol. 57, no. 4, pp. 1119–1145, 2019.

[10] J. Behnamian, S. M. T. Fatemi Ghomi, "Hybrid flowshop scheduling with machine and resource-dependent processing times," *Applied Mathematical Modelling*, vol. 35, no. 3, pp. 1107–1123, 2011.

[11] M. F. F. Ab. Rashid, A. N. Mohd Rose, N. M. Z. Nik Mohamed, "Hybrid flow shop scheduling with energy consumption in machine shop using moth flame optimization," in *Recent Trends in Mechatronics Towards Industry 4.0*, pp. 77–86, 2020.

[12] S. Schulz, "A genetic algorithm to solve the hybrid flow shop scheduling problem with subcontracting options and energy cost consideration," *Advances in Intelligent Systems and Computing*, vol. 854, pp. 263–273, 2019.

[13] H. Lu, F. Qiao, "An efficient adaptive genetic algorithm for energy saving in the hybrid flow shop scheduling with batch production at last stage," *Expert Systems*, vol. 39, no. 2, p. e12678, 2022.

[14] S. L. Jiang, L. Zhang, "Energy-oriented scheduling for hybrid flow shop with limited buffers through efficient multiobjective optimization," *IEEE Access*, vol. 7, pp. 34477–34487, 2019.

[15] X. Lian, Z. Zheng, C. Wang, X. Gao, "An energy-efficient hybrid flow shop scheduling problem in steelmaking plants," *Computers & Industrial Engineering*, vol. 162, p. 107683, 2021.

[16] W. Zhang, C. Li, M. Gen, W. Yang, G. Zhang, "A multiobjective memetic algorithm with particle swarm optimization and Q-learning-based local search for energy-efficient distributed heterogeneous hybrid flow-shop scheduling problem," *Expert Systems with Applications*, vol. 237, p. 121570, 2024.

[17] H. Luo, B. Du, G. Q. Huang, H. Chen, X. Li, "Hybrid flow shop scheduling considering machine electricity consumption cost," *International Journal of Production Economics*, vol. 146, no. 2, pp. 423–439, 2013.

[18] K. Geng, C. Ye, "A memetic algorithm for energy-efficient distributed re-entrant hybrid flow shop scheduling problem," *Journal of Intelligent & Fuzzy Systems*, vol. 41, pp. 3951–3971, 2021.

[19] J. Cai, D. Lei, "A cooperated shuffled frog-leaping algorithm for distributed energy-efficient hybrid flow shop scheduling with fuzzy processing time," *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2235–2253, 2021.

[20] Z. Wang, L. Shen, X. Li, L. Gao, "An improved multiobjective firefly algorithm for energy-efficient hybrid flowshop rescheduling problem," *Journal of Cleaner Production*, vol. 385, p. 135738, 2023.

[21] D. Karaboga, B. Gorkemli, "A combinatorial Artificial Bee Colony algorithm for traveling salesman problem," in *International Symposium on Innovations in Intelligent Systems and Applications*, pp. 50–53, 2011.

[22] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical Report-TR06*, Erciyes University, Kayseri, Turkey, 2005.

[23] B. Akay, D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1001–1014, 2012.

[24] J. Carlier, E. Neron, "An exact method for solving the multi-processor flow-shop," *RAIRO-Operation Research*, vol. 34, no. 1, pp. 1–25, 2000.

[25] J. Wang, Y. Liu, S. Rao, X. Zhou, J. Hu, "A novel self-adaptive multi-strategy artificial bee colony algorithm for coverage optimization in wireless sensor networks," *Ad Hoc Networks*, vol. 150, p. 103284, 2023.

[26] M. Li, C. T. Chang, Z. Liu, "A discrete artificial bee colony algorithm and its application in flexible flow shop scheduling with assembly and machine deterioration effect," *Applied Soft Computing*, vol. 159, p. 111593, 2024.

[27] X. Zhou, X. Zhang, W. Gao, H. Wang, Y. Ma, "Adaptive multi-population artificial bee colony algorithm based on fitness landscape analysis," *Applied Soft Computing*, vol. 164, p. 111952, 2024.