JOURNAL OF MECHANICAL ENGINEERING AND SCIENCES
ISSN: 2289-4659 e-ISSN: 2231-8380
VOLUME 20, ISSUE 2, 2026, 11164 - 11177
DOI: https://doi.org/10.15282/jmes.20.2.2026.3.0871

RESEARCH ARTICLE

Optimisation of energy and machine balance in the hybrid flowshop scheduling problem using differential evolution

W. C. Yang, M. F. F. A. Rashid, H. A. Salaam, M. A. N. Mu'tasim*

Faculty of Mechanical and Automotive Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Pahang, Malaysia

Abstract - One of the most essential processes in manufacturing is continuous operations without any downtime. The Energy Balanced Hybrid Flow Shop Scheduling Problem (EMBHFSP) is an interesting study as it provides huge impact on machine effectiveness while balancing between minimizing completion time and energy consumption. Limited study has focused on hybrid flow shops (HFS) with a concentration on energy-machine balanced production. This paper aims to develop a computational model and evaluate the exploration effectiveness of Differential Evolution in optimizing the EMBHFSP. The most popular as well as latest optimization algorithms including Simulated Annealing, Grey Wolf Optimization, Henry Gas Solubility Optimization, Harmony Search, Imperialist Competitive Algorithm, Multi-verse Optimizer, and Thermal Exchange Optimization were evaluated against Differential Evolution utilizing the EMBHFSP model across 20 Optimization repetitions. The experimental results indicate that the Differential evolution algorithm surpassed others by 45% in mean fitness value and exhibited a 67.5% enhancement in standard deviation across all benchmark problems. In addition, a case study was performed in a manufacturing facility to validate the applicability of the model. Three different scheduling solutions, optimizing makespan, energy balance and machine utilization balance are generated using differential evolution. The results show the ability of the model to solve trade-offs between efficiency of production and sustainability in real industrial environments.

Article History
Received : 30 July 2025
Revised : 26 February 2026
Accepted : 9 June 2026
Published : 30 June 2026

Keywords
Hybrid flowshop
Energy balanced
Differential evolution (DE)
Scheduling

1. Introduction

The hybrid flow shop scheduling problem (HFSP) is a complex NP-hard problem that exists in advanced manufacturing and production systems. Conventional models try to minimise the makespan, the total time to finish all jobs, since it directly impacts throughput and delivery efficiency [1]. Other objectives, such as energy consumption, have recently been considered in the scheduling framework due to growing interest in sustainable and cost-effective operations [2]. In HFSP, several studies have explored the trade-off between makespan and energy. Zhang et al. [3] proposed a multi-objective HFSP model to minimise makespan and energy consumption, using a hybrid multi-objective evolutionary algorithm to obtain a good schedule in green manufacturing. Wang et al. [4] designed an energy-efficient HFSP model with dynamic machine-speed regulation and idle-power modelling to minimise total energy consumption while maintaining production efficiency. Li et al. [5] proposed a multi-objective HFSP model for machine load balancing, including a load-balancing metric to ensure a fair division of work among parallel machines. They found that if the workload isn't properly balanced, the machines are overused, require more maintenance, and wear unevenly. However, there are few studies on the energy usage balance, which is important to peak load management and energy cost regulation, although research in this area is growing. Lu et al. [6] studied energy balancing for unrelated parallel machine scheduling by a bi-objective genetic algorithm, but did not consider the constraints of HFSP. Shao et al. [7] further extended this work by considering energy variance as a criterion in flexible manufacturing systems, but an overall framework for hybrid flow shops remains missing.

Metaheuristic algorithms play a significant role in solving complex variants of the HFSP, particularly when multiple conflicting objectives, such as makespan and energy consumption, are present. The earliest and most popular methods are differential evolution (DE)[8], particle swarm optimisation (PSO)[9] and the genetic algorithm (GA)[10]. These classical algorithms have proven to be robust and versatile for large-scale optimisation problems. These algorithms are still used in HFSP applications and have been demonstrated to be stable in performance, especially for dual-objective scheduling problems. Recent studies have presented innovative and adaptive metaheuristics, including the Grey Wolf Optimiser (GWO)[11], Henry gas solubility optimisation (HGSO) [12], Harmony Search (HS)[13], Imperialist Competitive Algorithm (ICA)[14], Multi-Verse Optimiser (MVO)[15], and Thermal Exchange Optimisation (TEO)[16]. These algorithms seek to improve the exploration-exploitation balance. Despite their potential, their efficacy in multi-criteria decision-making remains predominantly unsubstantiated. Recent research has provided innovative and adaptive metaheuristics such as Grey Wolf Optimiser [11], Henry Gas Solubility Optimisation [12], Harmony Search (HS) [13], Imperialist Competitive Algorithm (ICA) [14], Multi-Verse Optimiser [15], and Thermal Exchange Optimisation [16]. These algorithms aim to optimise the exploration-exploitation tradeoff. Despite their potential, their effectiveness in multi-criteria decision-making is largely unverified.

Ab Rashid and Mu'tasim [17] applied the Tiki-Taka Algorithm (TTA) to the HFSP bi-objective problem, a recent innovation and a contextually related one. TTA has been competitive in bi-objective HFSP models, maximising both makespan and energy consumption. The extension in cost-based HFSP models (Ab Rashid & Mu'tasim [18] also holds promise for solving many-objective problems. At the same time, improved versions of conventional algorithms have also

***CORRESPONDING AUTHOR | M. A. N. Mu'tasim** | ✉ ammar@umpsa.edu.my

been developed. In many-objective problems, such as energy-efficient HFSP models, the Non-Dominated Sorting Genetic Algorithm III (NSGA-III) has attracted attention for its ability to maintain a well-distributed Pareto front. Mu'tasim and Ab Rashid [19] have shown that it is effective at optimising energy consumption while maintaining solution diversity under different constraints. Simultaneously, revised iterations of traditional algorithms have also been developed. The Non-Dominated Sorting Genetic Algorithm III (NSGA-III) has garnered interest for its capacity to sustain a well-distributed Pareto front in many-objective problems, including energy-efficient HFSP models. Mu'tasim and Ab Rashid [19] illustrated its effectiveness in optimising energy consumption while maintaining solution diversity amidst various constraints. However, comparative studies between classical algorithms (DE, PSO, GA) and modern metaheuristics (TTA, GWO, HSGO, TEO) for many-objective HFSP formulations, such as EMBHFSP, remain limited. This limitation is important because recent manufacturing studies have emphasised the need for responsive, efficient, and cost-effective production systems, while also highlighting the lack of mathematical models, simulations, and optimisation techniques to support decision-making in complex manufacturing environments [20-21]. However, there is a lack of a comprehensive benchmarking framework for evaluating these algorithms using standardised problem settings and consistent metrics (e.g., makespan, total energy consumption, machine workload equilibrium, and energy variance). This highlights an important research gap that this study seeks to address by evaluating the performance of both traditional and novel metaheuristics in the context of a four-objective EMBHFSP.

The objectives of this study are to develop a mathematical model to minimise makespan, energy consumption, balance machine load and machine energy to extend machine life, to verify the model with metaheuristics algorithms from the most cited algorithms to relatively new algorithms using benchmark problems and to validate the model with real-world problems using a case study from a local manufacturing shop in Johor Bahru, Malaysia. Multi-objective scheduling in manufacturing has received increasing attention, but the literature remains seriously deficient. Most recent research focuses on makespan and energy consumption in parallel machine systems or flexible job shops, but largely ignores factors such as machine load balance and the long-term durability of equipment [22]. The trade-off between makespan and energy consumption is also discussed in the review paper on energy-aware metaheuristics [23]. But it does not deal with matters of load distribution and wear and tear of machinery." There are several studies [24]-[26] on machine breakdowns, but a balanced load distribution among machines has not yet been achieved. In other domains, such as the information technology sector, research has explored lifespan extension alongside energy efficiency and makespan optimisation in cloud systems [27]. However, the use of such integrated approaches is still relatively unexplored in the manufacturing domain. Therefore, there is a major gap in the literature, particularly the absence of a unified framework that simultaneously accounts for machine load balancing, makespan, and energy consumption. We propose a new framework for the Energy-aware machine balance hybrid flowshop scheduling problem to minimise makespan and energy consumption, thereby enhancing equipment lifespan by balancing machines. In addition, the mathematical model will be evaluated using the benchmark problem by comparing different types of mechanisms in meta-heuristic algorithms under standard parameters. Finally, a real-world case study will be conducted at a manufacturing factory in Johor to validate the mathematical model and bridge the gap between theory and practice.

2. Materials and Methods

2.1 Computational Modelling

A methodical, multi-stage modelling approach is used to develop a computational framework for solving the Energy-aware machine-balance hybrid flowshop scheduling problem. This framework is proposed to optimise job scheduling in a hybrid flowshop manufacturing system. Meanwhile, makespan minimisation, energy efficiency, machine balance and energy balance in HFSP are considered. Figure 1 shows a typical hybrid flowshop configuration, with jobs flowing through several stages, each with several machines in sequence. For this case study, the scheduling process begins with the initialisation of a predefined job sequence using a predefined dispatching rule approach, the first-come, first-served approach, {J1, J2, J3, J4, J5}. This first step determines the sequence in which jobs are to be processed on the production line.

The second step is to allocate specific machines to each job at each production stage after the initialisation phase. As shown in Table 1, it is necessary to ensure that each job is assigned to an available machine using a deterministic assignment rule, with the machine allocations and the corresponding processing times for each job at each stage listed. In this scheduling model, two fundamental constraints must be satisfied for the schedule to be feasible and valid. First, the machine availability constraint ensures that each job can be processed only on machines available at the given stage, thereby avoiding overlapping resource usage. Second, the precedence constraint requires that each job complete its processing in a specific step before proceeding to the next, thereby ensuring a sequential and uninterrupted workflow. These constraints are needed to maintain the logical sequence of operations and to represent the practical constraints of real-world hybrid flowshop systems. The third stage is the scheduling computation in which the model computes the makespan, that is, the total time to complete all jobs, with strict satisfaction of the machine availability and job precedence constraints. This ensures that no two jobs are processed simultaneously on the same machine and that each job passes through the stages in the correct sequence. The calculated start time (ST), processing time (PT) and finish time (FT) for each job at each stage are tabulated in Table 2. The resulting makespan is 154 minutes. Once the schedule is fixed, the energy consumption for each operation is calculated as the product of the processing time and the corresponding machine's power rating. Such a quantification provides information about the energy requirements of the entire production schedule. The detailed energy consumption per job and stages, as well as the machine utilised and the energy consumption in watt-minutes (W·min), are shown in Table 3. This results in a total energy consumption of 4175 W·min.

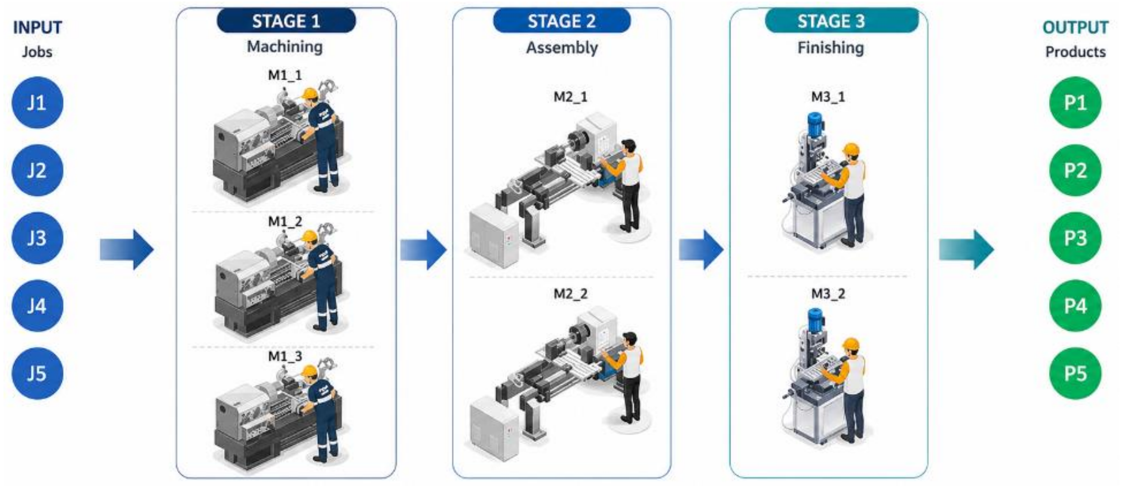


Figure 1. Hybrid flowshop scheduling setup

Table 1. Assign jobs to the machine in the job sequence

Jobs	Machine used on each stage			Jobs	Processing time (Minutes)		
	Stage 1	Stage 2	Stage 3		Stage 1	Stage 2	Stage 3
1	2	1	2	1	5	30	17
2	3	2	2	2	18	41	6
3	2	1	1	3	10	38	18
4	1	2	2	4	3	35	19
5	3	2	2	5	16	45	15

Table 2. Makespan calculations

Jobs	Stage 1			Stage 2			Stage 3		
	ST	PT	FT	ST	PT	FT	ST	PT	FT
1	0	5	5	5	30	32	3	17	52
2	0	18	18	15	41	59	59	16	75
3	5	10	15	35	38	73	73	18	91
4	0	3	3	59	35	94	94	19	113
5	18	16	34	94	45	139	139	15	154

Table 3. Calculation of energy utilised

Jobs	Stage	Machine	PT	Power rate	EE (W.min)
1	1	2	5	7	35
1	2	1	30	12	360
1	3	2	17	15	255
2	1	3	18	6	108
2	2	2	41	14	574
2	3	2	16	15	240
3	1	2	10	7	70
3	2	1	38	12	456
3	3	1	18	19	342
4	1	1	3	3	9
4	2	2	35	14	490
4	3	2	19	15	285
5	1	3	16	6	96
5	2	2	45	14	630
5	3	2	15	15	225
Total energy consumption					4175

To evaluate the effectiveness and overall quality of the computed schedule, two additional performance metrics are used: the Normalised Load Imbalance Metric (V_{load}) and the Energy Variance (CV_{energy}). The V_{load} metric measures how evenly job assignments are distributed across all machines; a lower value indicates more uniform utilisation. In this case, the calculated Vload is 0.3568, indicating a moderate level of load balancing across the machines. On the other hand, the CV_{energy} metric captures the variability in energy consumption across different machines. High CV_{energy} indicates high inconsistency of energy use and, therefore, inefficiency. The observed CV_{energy} value of 94.53% indicates a large variation in the energy distribution, indicating that energy demand in the system is uneven. The performance indicators are shown in Figure 2, which gives a complete view of operational efficiency and energy-related performance. It shows the job schedule timeline and the calculated values of V_{load} and CV_{energy}.

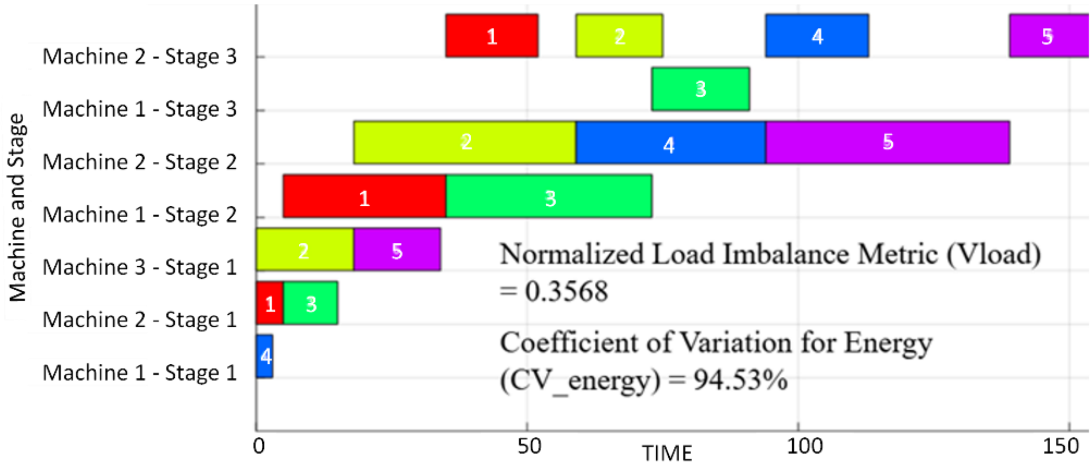


Figure 2. Production schedule with V_{load} and CV $_{energy}$ output

The EMBHFSP mathematical model takes into account several performance objectives for optimising the production efficiency and sustainability. The summary list of notations is presented as in Table 4 to explain the equation. The main objectives of the proposed approach are to minimise the makespan, reduce total energy consumption, and achieve a balanced utilisation of machines in terms of workload and energy consumption.

Table 4. The notation list

Abbreviation	Descriptions
J	Set of jobs to be processed, where $j \in \{1, 2, \dots, n\}$
S	Set of manufacturing stages, where $s \in \{1, 2, \dots, k\}$
M_s	Set of parallel machines available at stage s, where $m \in M_s$
Pj,s	Power consumption rate when processing job j at stage s
$T_{j,s}$	Processing time required for job j at stage s
$L_{s,m}$	Total workload (processing time) assigned to machine m at stage s
$E_{s,m}$	Total energy consumed by machine m at stage s
$C_{j,s}$	Completion time of job j at stage s
C_{max}	Maximum completion time across all jobs and stages (Makespan)
EE	Total cumulative energy consumption
V_{load}	Normalized load imbalance metric (Workload Variance)
CV_{energy}	Coefficient of Variation for energy usage

These goals are reflected in four key objective components as follows: The main goal in flowshop scheduling is to minimise the makespan (Cmax), which is the total time to finish all jobs through all stages, as given in Eq. (1). The makespan is 154 minutes, as in the example above.

$$c_{max} = \max_{j \in J, s \in S} C_{j,s} \tag{1}$$

Eq. (2), which accounts for the cumulative energy consumption across all manufacturing stages and tasks, is used to determine the overall energy consumption (EE) in the hybrid flowshop scheduling problem. This formula calculates EE for each job j at each stage s of the production sequence by multiplying the power consumption (Pj,s) by the processing time (Tj,s). The EE in the example above is calculated as 4175 W.min.

$$EE = \sum_{j \in J} \sum_{s \in S} P_{j,s} \, . T_{j,s} \tag{2}$$

To evaluate how evenly the workload is distributed among the machines, the normalized load imbalance metric (V_{load}) is introduced. The square root of the variation in machine loads, divided by the mean of machine loads, is the load variance ratio (V_{load}), as indicated in Eq. (3). This indicator helps assess how evenly the workload is distributed among the various HFSP machines. In the example above, V_{load} was computed as 0.3568, indicating moderate load imbalance.

$$V_{load} = \frac{\sqrt{Var(I_{s,m})}}{Mean(I_{s,m})} \tag{3}$$

The coefficient of variation for energy (CV_{energy}), which is the square root of the variance of machine energy consumption divided by the mean energy consumption multiplied by 100, can be calculated as shown in Equation (4). This metric measures the relative differences in energy usage among the HFSP machines. The CV_{energy} is the difference of energy consumption of each machine in the hybrid flowshop system. A high CV_{energy} (eg, 94.53% in the example) indicates a large disparity and possible inefficiencies.

$$CV_{energy} = \frac{\sqrt{Var(E_{s,m})}}{Mean(E_{s,m})} \times 100 \tag{4}$$

Since this problem is multi-objective in nature, the a priori approach (weighted sum method) is adopted to scalarize the four objectives to a single composite function. This opens up the possibility of solving the model as a single-objective optimisation problem. In the context of machine utilisation, energy consumption is prioritised, with an emphasis on the interdependencies among makespan, machine load, and machine variance. The objectives Cmax, V_{load}, and CV_{energy} are on different scales; thus, they are normalised to a common scale [0, 1] based on their corresponding global peaks. Each objective was evaluated separately by tuning the weight coefficients w_1, w_2, w_3 and w_4 (representing C_{max}, EE, V_{load} and CV_{energy}, respectively), as shown in Eq. (5). More specifically, we tested each objective in isolation by assigning the corresponding weight to 1, setting all other coefficients to 0, and repeating this iteratively for all 4 functions.

$$\min f = w_1 C_{max} + w_2 EE + w_3 V_{load} + w_3 CV_{energy} \tag{5}$$

2.2 Differential Evolution

Differential Evolution is a population-based metaheuristic optimisation algorithm for continuous search spaces, first introduced by Storn and Price [28]. The algorithm employs a mutation-crossover-selection scheme in which new candidate solutions are generated by perturbing existing vectors via scaled differences of randomly selected population members (mutation), merging parameters (crossover), and applying competitive selection. DE has only three control parameters: population size (NP), mutation scale factor (F), and crossover rate (CR), and thus is especially useful for complex, nonlinear optimisation problems. Its differential mutation operator is unique, improving exploration while preserving convergence and outperforming many evolutionary algorithms for multimodal function optimisation. For each target vector x_i in the current population, a mutant vector v_i is generated using the differential strategy in Eq. (6), where x_{r1}, x_{r2}, x_{r3} are randomly selected distinct vectors from the population. F is the mutation scale factor and is usually between [0 and 2]. This factor controls the amplification of Differential variation and is suitable for an exploration strategy. This operation introduces diversity and guides the search towards promising areas in the solution space. The mutation technique is similar to GA and is particularly effective at finding a global optimum and avoiding a local optimum.

$$v_i = x_{r1} + \mathrm{F}.(x_{r2} - x_{r3}) \tag{6}$$

The crossover parameter is used to increase variability and exploitation strategy. Crossover DE exploits the current knowledge stored in the current iteration and applies a binomial crossover to mix the mutated vector v_i with the original target vector x_i. As a result, it becomes a trial vector, u_i as presented in Eq. (7).

$$u_{i,j} = \begin{cases} v_{i,j} \; if \; rand_j(0,1) \leq CR \; or \; j = j_{rand} \\ \quad x_{i,j} \; otherwise \end{cases} \tag{7}$$

The CR is the crossover probability with bounds of [0, 1] and $rand_j(0,1)$ is the uniformly distributed random number. j_{rand} is the parameter that ensures at least one parameter is taken from the mutant vector. This crossover operation balances exploration and exploitation by combining known-good traits from both the parent and mutant vectors. In the final parameter of DE, the selection parameter uses greedy selection. This technique ensures that only the better solution between the trial vector $u_{i,j}$ and the original vector $x_{i,j}$ survives into the next iterations as presented in Eq. (8). In this research, the parameter set for the DE algorithms (which are NP) is as follows: 50, a mutation scale factor lower bound of 0.2, an upper bound of 0.8, and CR of 0.2.

$$x_i^{(t+1)} = \begin{cases} u_i \; if \; f(u_i) \leq f(x_i) \\ \quad x_i \; otherwise \end{cases} \tag{8}$$

3. Results and Discussion

3.1 Benchmark Problem Results

In this section, we present a computational framework for evaluating the energy-balanced hybrid flowshop scheduling problem using different metaheuristic algorithms. The main algorithm of the study, Differential Evolution, is compared with several state-of-the-art optimisation algorithms, including Simulated Annealing, Grey Wolf Optimiser, Hybrid Sine-Gordon Optimisation, Harmony Search, Imperialist Competitive Algorithm, Multi-Verse Optimiser, and Thermal Exchange Optimisation. The aim of this comparative experiment is to evaluate the performance of each algorithm in solving the multi-objective optimisation problem of minimising makespan, total energy consumption, and energy and load imbalance. For the computational experiment, we use a benchmark set of 12 problems from Carlier and Néron [29] as shown in Table 5. The benchmark is well known for hybrid flowshop scheduling problems, ensuring a robust and representative evaluation. These benchmarks consist of test cases across different jobs, stages, and machine configurations, each with varying complexity. Also, the number of jobs varies from 10 to 15, the number of stages varies from 5 to 10, and the number of parallel machines allocated to each stage depends on the machine configuration. This diversity is important for evaluating the algorithm's performance and adaptability across different real-world manufacturing scenarios. Standard benchmark problems provide a baseline for the algorithm's performance and enable meaningful comparison with the existing body of literature.

Selecting a normal distribution to generate processing times in the range [3], [28] reflects a real-world manufacturing environment where task durations vary within a reasonable range. This approach ensures that the algorithms do not compensate for outlier processing times; instead, they focus on solving the core scheduling and energy optimisation challenges of the problem by excluding extreme values that may skew the optimisation process. Moreover, the limited distribution keeps the computational complexity under control. The range of job sizes allows the algorithms to show their adaptability to different production constraints. All algorithms were executed with the same experimental parameters to allow comparability. The population size for all algorithms was set to 50, a balanced value that provides sufficient diversity while avoiding high computational cost. In general, larger populations increase the diversity of solutions and thus the quality of the search, but also increase computation time and the number of evaluations. After testing and fine-tuning, NP = 50 was found to provide a good trade-off between computational efficiency and performance. 1000 iterations were sufficient for each algorithm to converge and achieve efficiency. The number of iterations was determined by the problem's characteristics and the typical convergence properties of the tested metaheuristic algorithms. Given the complexity of the EMHFSP and the variety of benchmarks, 1000 iterations ensured that all algorithms had sufficient time to converge to good results. To account for the stochastic nature of the algorithms, each optimisation was run 20 times for each problem instance. This repetition is important because metaheuristics often yield different results across runs, and a single run may produce unrepresentative or suboptimal results. The stability and robustness of each method can be evaluated through multiple algorithm runs and statistical analysis. The performance of the algorithms can be estimated more accurately.

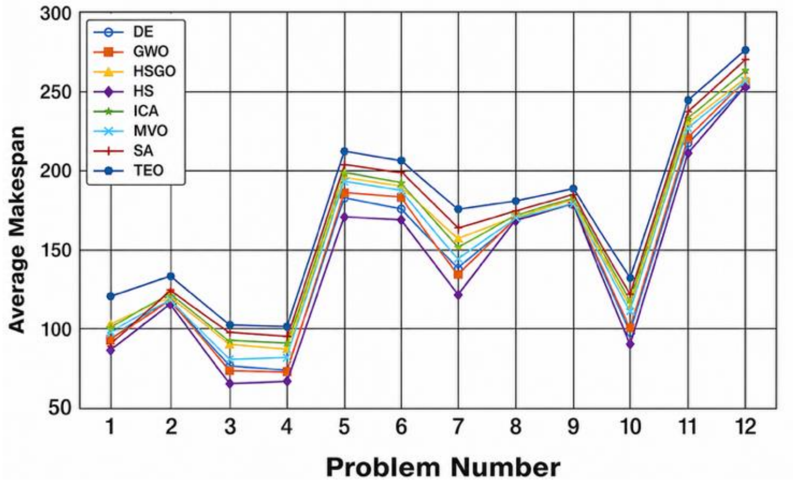
Table 5. Benchmark problems

No.	Problem	Jobs	Stages	Machine configuration	No.	Problem	Jobs	Stages	Machine configuration
1	J10c5a2	10	5	2 2 1 2 2	7	J10c10c1	10	10	3 3 3 3 2 3 3 3 3 3
2	J10c5b1	10	5	1 2 2 2 2	8	J15c5a1	15	5	3 3 1 3 3
3	J10c5c1	10	5	3 3 2 3 3	9	J15c5b1	15	5	1 3 3 3 3
4	J10c5c1	10	5	3 3 3 3 3	10	J15c5c1	15	5	3 3 2 3 3
5	J10c10a2	10	10	2 2 2 2 1 2 2 2 2 2	11	J15c10a2	15	10	3 3 3 3 1 3 3 3 3 3
6	J15c10b1	15	10	1 3 3 3 3 3 3 3 3 3	12	J15c10b1	15	10	1 3 3 3 3 3 3 3 3 3

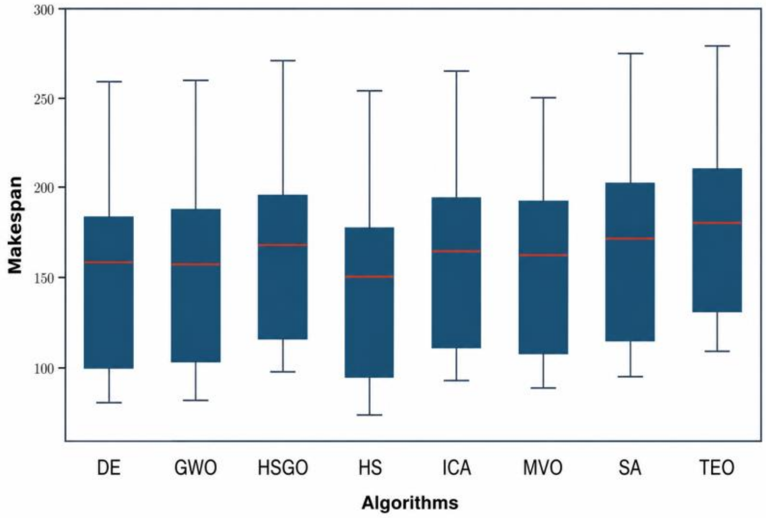
The overall optimisation performance of each algorithm was evaluated by averaging the minimum values of these objectives across 20 runs for each benchmark problem. The evaluation should be based on the algorithm's ability to consistently find high-quality solutions. Thus, the mean of the minimum value is a good metric as it captures the best solution obtained over multiple runs. In addition to the minimum average values, the standard deviation of the objective values was calculated. The standard deviation is a measure of the stability (or variability) of the algorithm's performance. A higher standard deviation indicates that the algorithm's performance is more unpredictable and potentially unreliable, whereas a lower standard deviation indicates that the algorithm reliably produces solutions close to the most well-known values. For industrial applications, it is very important to have consistent results. The low standard deviation values for Makespan and $CV_{\text{energy indicate}}$ that the algorithm is stable and produces high-quality solutions.

3.1.1 Algorithm comparisons

Across all 12 problems, the DE algorithm provides the best average Makespan, $V_{\text{load,}}$ and CV_{energy} fitness values. The DE algorithms had the lowest standard deviation fitness values for Makespan and CV_{energy} across all 12 challenges. In addition, among the twelve benchmark problems, the TEO had the largest mean and standard deviation. fitness solution. Across all algorithms, TEO had the fastest runtime for running 12 benchmark problems 20 times. Figure 3 presents the mean. In terms of makespan, HS consistently outperforms the other algorithms, achieving the best average performance in 10 out of 12 benchmark instances. This is followed by DE and GWO, with DE securing two first-place rankings. In contrast, TEO demonstrates the weakest performance overall, ranking lowest across most cases. Similarly, SA and HSGO also show consistently poor performance throughout all benchmark problems. For the SD of makespan, DE demonstrates the most consistent performance, ranking first in 7 out of 12 benchmark problems, followed by HS with 5 first-place rankings, and SA showing relatively stable results. In contrast, TEO ranks the lowest, indicating the highest variability in makespan across runs. Similarly, ICA and MVO exhibit poor consistency, with higher standard deviations across all 12 benchmark cases.



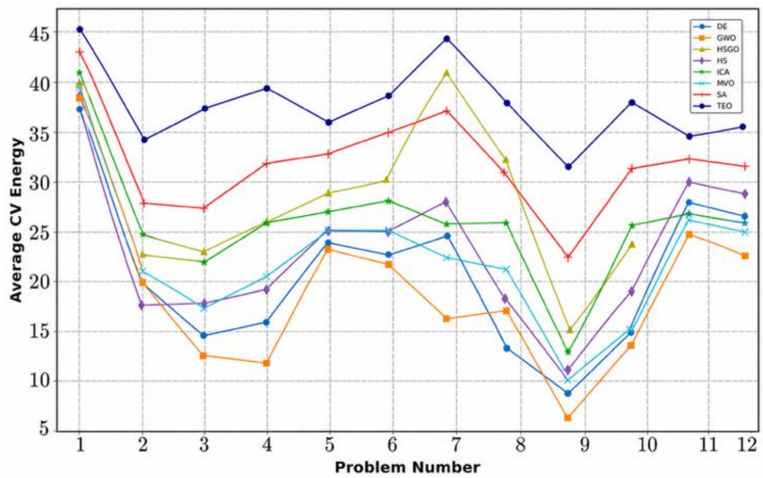
(a) Average Makespan against 12 benchmark problems



(b) Performance comparison (Makespan) among the algorithms

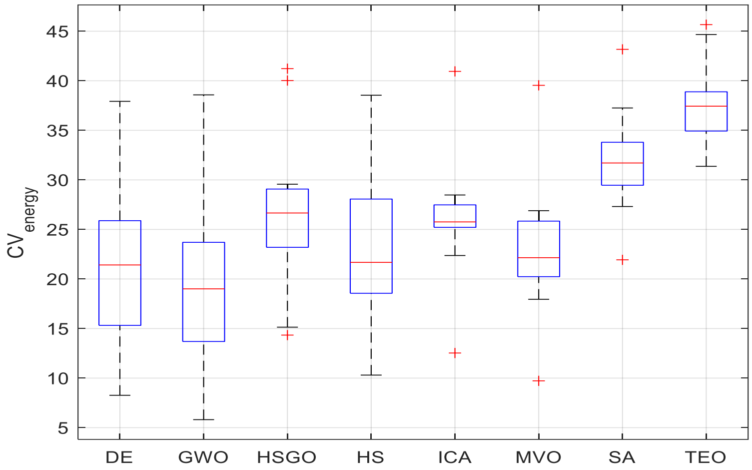
Figure 3. The best mean makespan achieved by each algorithm on benchmark problems

Figure 4 shows the mean of CV_{energy}. GWO shows the best performance in terms of average CV_{energy}, followed by DE and MVO. GWO ranks first in 6 of 12 benchmark problems, while DE ranks first in 4. TEO, on the other hand, is always the worst-performing model, ranking last in all tests. Moreover, SA and HSGO perform poorly in terms of average CV_{energy} across all 12 benchmarks. In terms of the SD of CV_{energy}, DE performs best, followed by HS and GWO. DE wins 4 out of 12 benchmark problems, while HS wins 4 as well. The worst-performing algorithm is TEO, which is always the worst. For the standard deviation of CV_{energy}, SA and ICA perform poorly across all 12 benchmarks.



(a) Average CV_{energy} against 12 benchmark problems

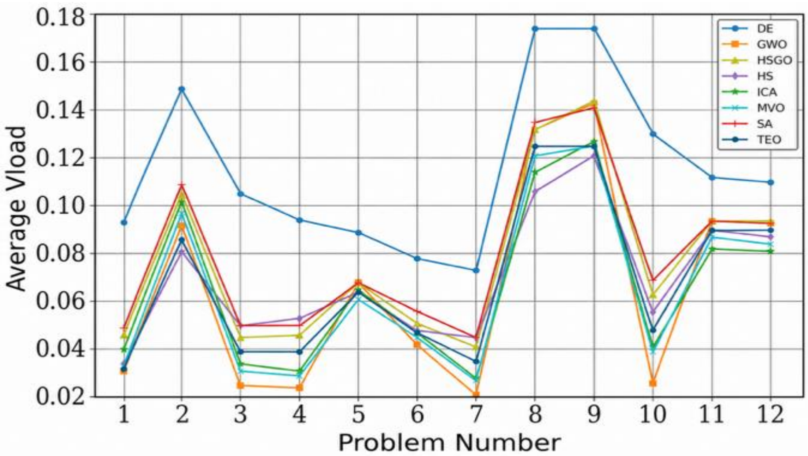
Figure 4. Mean for CV_{energy} achieved by each algorithm on benchmark problems



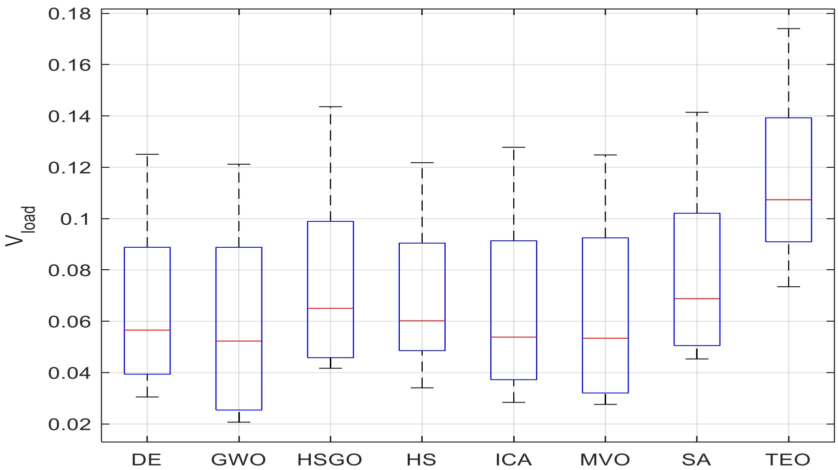
(b) Mean value for CV_{energy} achieved among the algorithms

Figure 4. (cont.)

Based on Figure 5 below, average of V_{load}, GWO achieves the best average V_{load}, followed by MVO and ICA. GWO ranks first in 7 out of 12 benchmark problems, while ICA secures first place in 2 of the 12 benchmarks. The worst-performing algorithm is TEO, which consistently ranks last. Additionally, SA and HSGO show poor performance in terms of average V_{load} across all 12 benchmarks. For the SD of $V_{load,}$ DE achieves the best performance in terms of the standard deviation of V_{load}, followed by HS and GWO. DE ranks first in 10 out of 12 benchmark problems, while HS and GWO each secure first place in 1 benchmark problem. The worst-performing algorithm is TEO, consistently placing last. SA and ICA also demonstrate poor performance in terms of the standard deviation of V_{load} across all 12 benchmarks.



(a) Average V_{load} against 12 benchmark problems

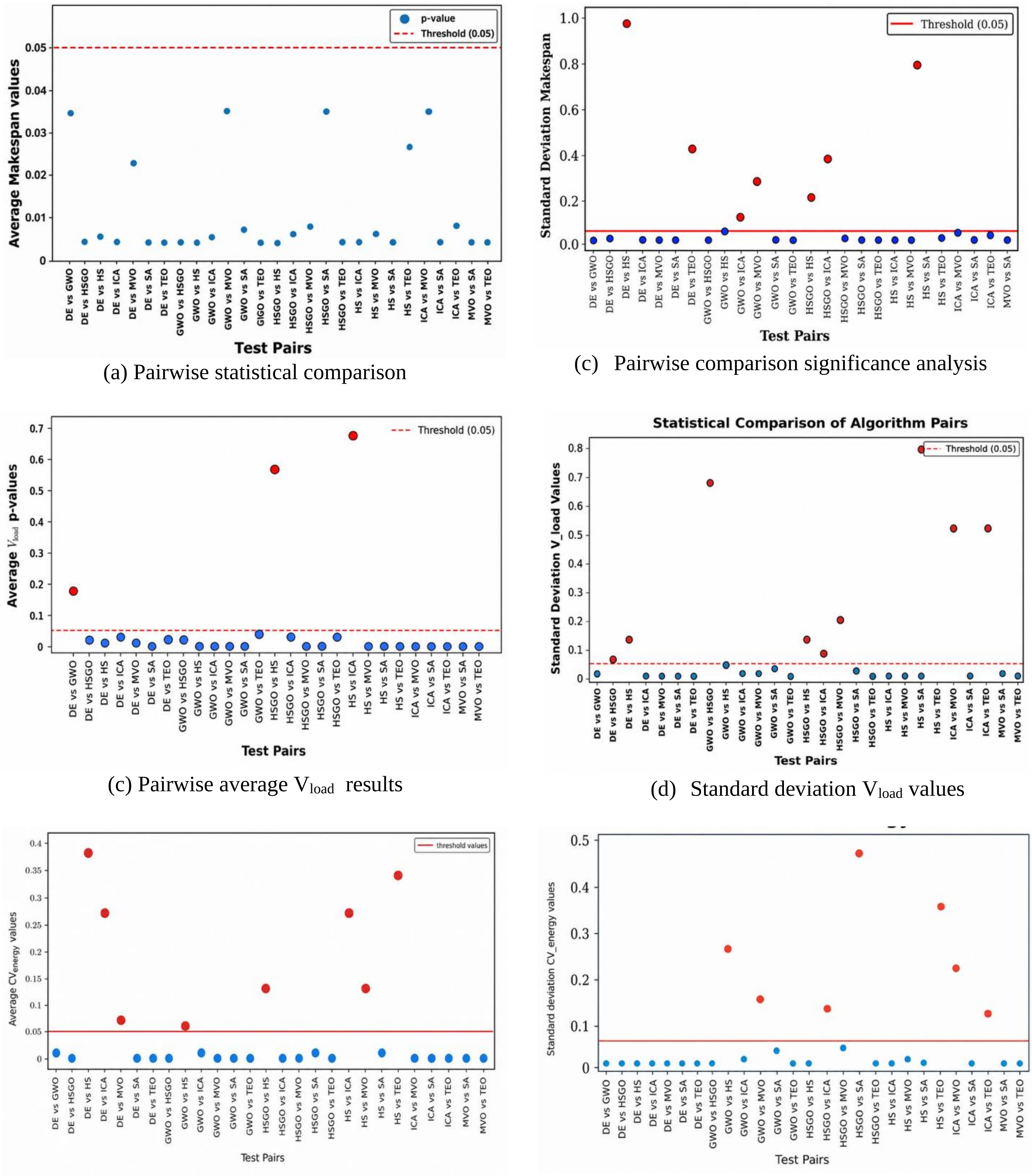


(b) Mean value for V_{load} achieved among the algorithms

Figure 5. Mean for V_{load} achieved by each algorithm on benchmark problems

3.1.2 Overall Comparisons

A non-parametric pairwise comparison was performed for all the optimisation objectives to rigorously validate the computational efficacy of the tested metaheuristics. Figure 6 presents the test pairs and tests the relative performance profiles for the mean and standard deviation values of the objective functions. The critical significance value of α = 0.05 is the horizontal reference baseline. Blue markers indicate that p-values are strictly below this threshold, meaning that one algorithm is statistically significantly better than its competitor and effectively rejects the null hypothesis. The coordinates plotted above the threshold (marked in red) are, on the other hand, pairs where the performance differences are statistically insignificant ($p > 0.05$), i.e. the algorithms reached a state of numerical convergence parity.



(a) Pairwise statistical comparison

(c) Pairwise comparison significance analysis

(c) Pairwise average V_{load} results

(d) Standard deviation V_{load} values

(e) Average CV_{energy} values

(e) Average standard deviation

Figure 6. Statistical analysis

In terms of statistical output, differential evolution exhibits a markedly different performance profile for makespan optimisation. In the average. In makespan analysis, all DE test pairs are strictly below the critical threshold, implying statistically significant differences in temporal efficiency relative to all competing metaheuristics. In terms of V_{load}, DE is competitive in terms of mean optimisation and has a unique profile of consistency. The pairwise test of average workload distribution between DE and GWO yields a p-value above the significance threshold, indicating that the workload distribution of DE is not significantly different from that of GWO. On the other hand, DE remains statistically different from the other metaheuristics in this average dimension. In terms of operational consistency, the standard deviation of Vload indicates that DE achieves statistical parity with ICA, meaning both algorithms are similarly robust when stabilising machine workloads. DE exhibits a prominent convergence plateau for the Energy Consumption (CV_{energy}) objective. The average CVenergy subplot shows that DE achieves statistical parity with most benchmark algorithms, with no significant differences from HSGO, ICA, and MVO. This high density of elevated markers demonstrates that DE's mathematical quality in minimising mean energy overhead is matched by these competing metaheuristics.

3.1.3 Case Study

This case study focuses on the production of cable hangers at a manufacturing facility in Johor Bahru. The designed component shown in Figure 7 has several support arms with adjustable openings (11× 35 mm slots) for versatile cable management and wall mounting. The hangers are manufactured from 50 mm flat bars; the overall size is 470 × 200 mm, and they are produced in batches of 10,000 (300 per pallet). There are two design variants: the main variant with several bent arms and the secondary variant with three arms, one of which is unbent. The product's structural design provides reliable cable support and enhances installation efficiency through its versatile mounting system.

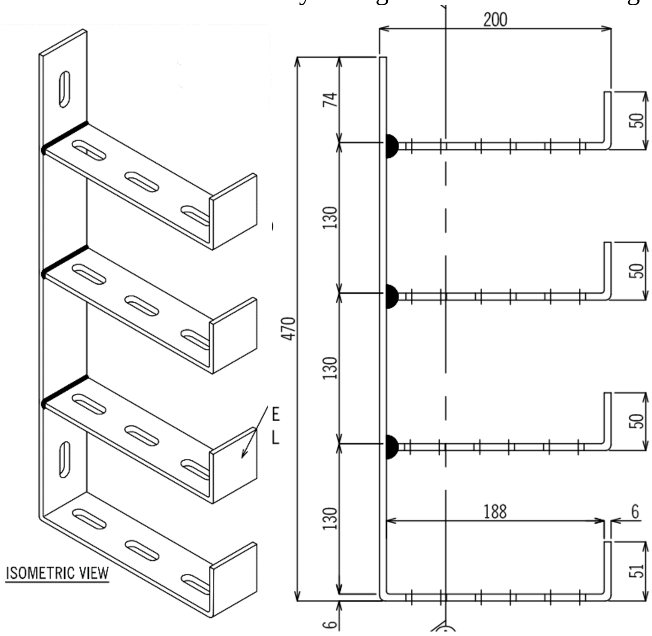


Figure 7. Cable hanger

In stage 1, CNC cutting machines cut the steel flat bars to the necessary dimensions for the cable hanger production (470 × 200 mm). Step 2: Punching the cut blanks to make 11 × 35 mm slotted apertures and mounting holes using hydraulic presses. Stage 3 refines the surface through precision grinding to remove burrs and sharp edges. Stage 4: Blanking (Support arms and hooks are bent to exact specifications by special press brakes). In stage 5, jig fixtures are used to fit the components, and the structural join is made by welding (Stage 6). Powder coating for corrosion protection and aesthetics. Surface preparation (degreasing and shot blasting). This research considers eight different jobs differentiated by three parameters: design (Design 1: normal configuration; Design 2: simplified three-arm variant), material (Material 1: mild steel; Material 2: stainless steel 304), and thickness (Thickness 1: 4.5 mm; Thickness 2: 6.0 mm). The case study results using all algorithms are depicted in Table 6. The scheduling chart for makespan (HS solution), CV_{energy} (DE solution), and V_{load} (DE solution) is presented in Figure 8.

Table 6. Ranking for overall

		DE	GWO	HGSO	HS	ICA	MVO	SA	TEO
Makespan	Mean	2.1667	3.2500	5.6667	1.1667	4.9167	4.0833	6.5833	8.000
	SD	1.5000	4.9167	4.9167	1.5833	6.0833	5.6667	4.0000	7.250
CV_{energy}	Mean	2.1667	1.7500	5.2500	3.5000	5.0000	3.4167	6.9167	8.000
	SD	1.9167	3.5000	3.9167	2.3333	5.5833	5.25	5.7500	7.750
V_{load}	Mean	3.7500	1.5833	5.9167	4.3333	3.0833	2.6667	6.6667	8.000
	SD	1.1667	3.5833	6.0000	2.3333	5.1667	4.0833	5.6667	8.000
Overall	Mean	2.1111	3.0972	5.2778	2.5417	4.9722	4.1944	5.9306	7.8333
	Rank	1	3	6	2	5	4	7	8

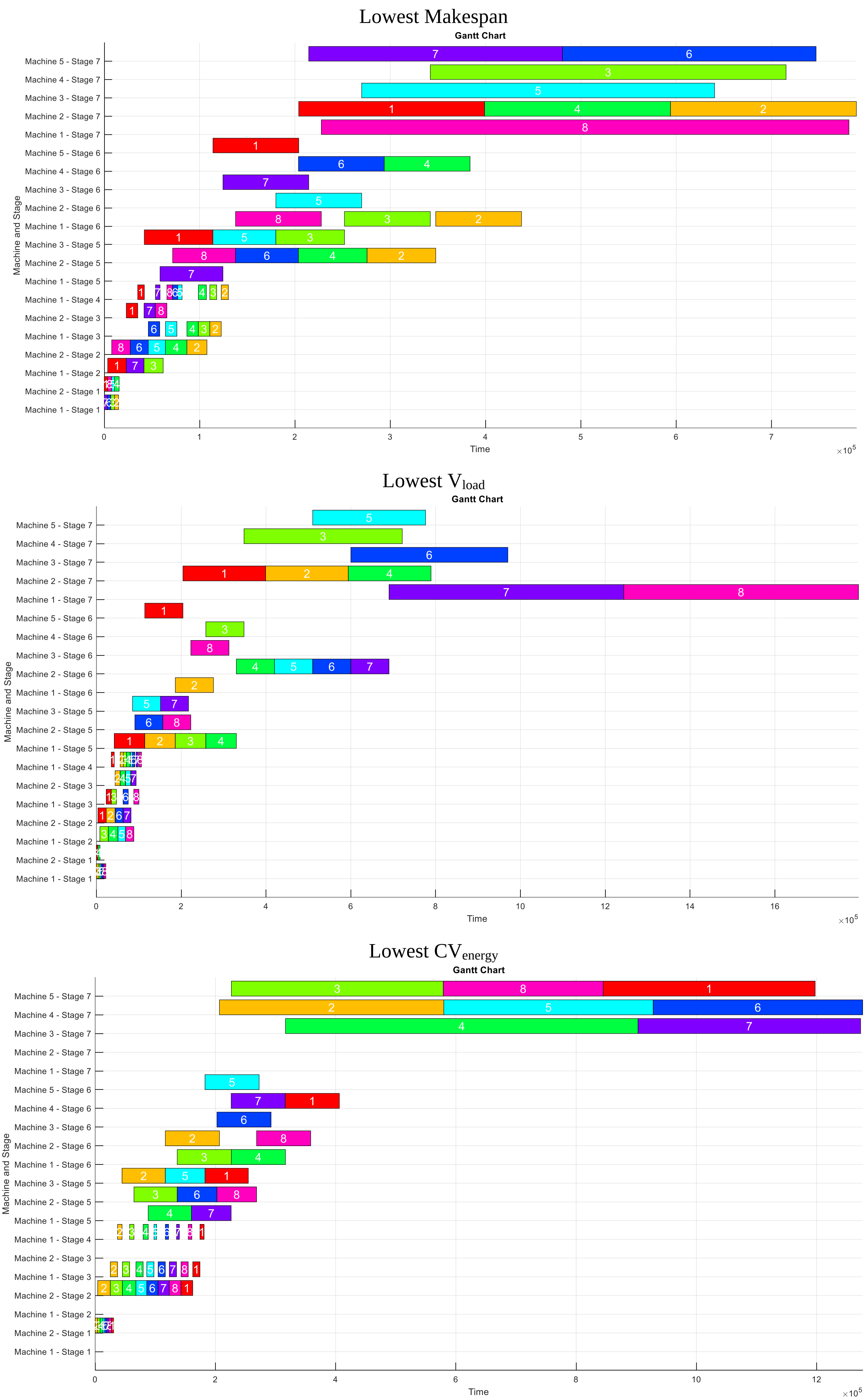


Figure 8. Scheduling chart for all 3 objectives

As shown in Table 7, the DE achieves the best results for CV_{energy} and V_{load} in both the mean and standard deviation, while the HS yields the lowest makespan in both cases. The maximum value for three objectives for the two statistical measures, which are made by the TEO algorithms. Figure 8 shows the scheduling charts for each objective, giving key insights into the optimisation strategies used. The Makespan chart clearly indicates that the main objective is to reduce the completion time. In this case, the algorithm assigns resources (i.e., machines) irrespective of power rate, preferring to use all available machines. The V_{load} scheduling chart shows a different priority. The objective here is to maintain a balanced workload distribution across all machines, without considering energy consumption or total completion time. The chart shows the trade-off in the algorithm's power usage optimisation for time balancing across machines. The CV_{energy} chart takes a more nuanced approach, looking at both machine utilisation and power rate. Interestingly, three machines are used for CV_{energy} in Stage 7, which may seem counterintuitive at first. However, this strategy shows that the algorithm seeks to optimise the energy distribution in the system. This limits the operation to three machines, thereby reducing oscillations in energy consumption and providing a more stable, efficient energy profile. This method also means that the algorithm can avoid potential clashes or downtime that may occur in a non-optimised schedule by focusing on a shorter list of tasks. What the algorithm is essentially doing is trying to ensure that no single machine experiences excessive energy demand. The result is a more level and sustainable energy usage profile with fewer spikes in consumption.

Table 7. Case study data

Algorithms	Makespan (min)		CV_{energy} (min)		V_{load} (min)	
	mean	SD	mean	SD	mean	SD
DE	793830	6871.3820	50.8616	1.7011	0.1622	0.0011
GWO	812820	18459.2467	53.9946	4.6929	0.1647	0.0029
HSGO	889560	35050.9914	65.2791	2.3152	0.1894	0.0065
HS	789855	2468.6402	53.0373	2.9665	0.1650	0.0030
ICA	824175	19925.8857	61.8084	3.0170	0.1715	0.0049
MVO	817425	23484.8411	57.8956	3.7664	0.1690	0.0025
SA	833250	31766.3924	62.9151	3.2243	0.1834	0.0064
TEO	868155	48913.6347	66.2323	2.3504	0.1970	0.0073

5. Conclusions

In this paper, the energy-balance hybrid flow shop scheduling problem was considered, and several metaheuristic algorithms, including Simulated Annealing, Differential Evolution, Grey Wolf Optimiser, and Harmony Search, were examined. The results for all 12 benchmark problems indicated that DE was the most efficient algorithm, which outperformed all other metaheuristics in terms of makespan and CV_{energy}. Moreover, DE showed the lowest standard deviation for both makespan and CV_{energy}, indicating its ability to provide consistent results. However, GWO was best for average V_{load}, while DE still had a better standard deviation for V_{load}. Consequently, DE was the best algorithm overall, with HS in second and GWO in third. But GWO outperformed DE on average value, indicating DE was worse for V_{load}. However, DE is a strong scheduling method for hybrid flow shops, with high consistency across benchmark instances and a balanced multi-objective performance. From a practical perspective, the results highlight the importance of applying energy-aware scheduling approaches in modern manufacturing to reduce operational expenses and environmental impacts. The proposed DE-based approach minimises energy consumption and the distribution of machine workload, which is beneficial for sustainable production planning. For large-scale industrial applications, future research should consider incorporating real-world constraints such as machine breakdowns, setup times, and equipment availability to improve the model's scalability. Moreover, hybrid strategies that combine DE with other metaheuristics can improve solution robustness and quality.

The selection of parameters for DE is very important for its optimisation performance. These parameter settings were optimised for solving EMBHFSP and were found to be important for ensuring both DE convergence and solution quality. The parameters influenced the algorithm's efficiency in exploring the solution space, its ability to converge to high-quality solutions, and the stability and consistency of results across the benchmark instances. In the case study, DE yielded the best results for CV_{energy} and V_{load}, while HS achieved the best average makespan after 20 repetitions. The model successfully demonstrated the trade-offs among makespan and CV_{energy}. For low makespan, DE used all machines, resulting in higher idle time at the end of the scheduling process, but maximising throughput. For V_{load}, the load was balanced by using multiple machines, resulting in longer scheduling time but a more balanced distribution of load across machines. Finally, CV_{energy} experienced a more energy-efficient process with fewer machines, resulting in less idle time and more balanced energy consumption across the machines. In summary, the study demonstrated that DE is an efficient algorithm that outperforms others by 45% in mean fitness value and achieves a 67.5% improvement in standard deviation across all benchmark problems. It efficiently handled the trade-offs among low makespan, low $V_{load,}$ and low $CV_{energy,}$ yielding a reliable scheduling solution for hybrid flow-shop operations.

Acknowledgements
We would like to thank and acknowledge Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA) for funding this project.

Funding
The authors would like to thank Universiti Malaysia Pahang Al-Sultan Abdullah for funding this project using RDU240315.

Declaration of Competing Interest
The author declares no conflicts of interest.

Credit Authorship Contribution Statement
W.C. Yang: Methodology; Analysis; Writing – original draft
M. F. F. Rashid: Conceptualisation: Analysis; Validation; Supervision
H. A. Salaam: Analysis, Writing -revise & review
M. A . N. Mu'tasim: Conceptualisation; Supervision; Writing-revise & review

Availability of Data and Materials
The data supporting this study's findings are available on request from the corresponding author.

Ethics Declarations
This study did not involve human participants or animals. Ethical approval was therefore not required.

Generative Artificial Intelligence Declarations
The authors claim that artificially intelligent-assisted technologies, such as generative AI, were not used to generate content, ideas, or theories. We have just utilised AI to enhance readability and refine the language. This was used with extreme human control and oversight. The authors take full responsibility for reviewing and approving the content.

References
[1] J. S. Neufeld, S. Schulz, and U. Buscher, "A systematic review of multi-objective hybrid flow shop scheduling," *European Journal of Operational Research*, vol. 309, no. 1, pp. 1-23, 2023.
[2] A. Missaoui, C. Ozturk, B. O. Sullivan, and M. Garraffa, "Energy efficient manufacturing scheduling: A systematic literature review," *Handbook of Formal Optimization*, p. 1-26, 2023.
[3] C. Li, B. Zhang, Y. Y. Han, Y. Wang, J. Q. Li, and K. Gao, "Energy-efficient hybrid flowshop scheduling with consistent sublots using an improved cooperative coevolutionary algorithm," *Mathematics*, vol. 11, no. 1, p. 77, 2022.
[4] Z. Wang, L. Shen, X. Li, and L. Gao, "An improved multi-objective firefly algorithm for energy-efficient hybrid flowshop rescheduling problem," *Journal of Cleaner Production*, vol. 385, p. 135738, 2023.
[5] H. Li, Q. Yan, and H. Wang, "Self-Learning artificial bee colony algorithm for distributed hybrid flow shop scheduling with preventive maintenance," in *36th Chinese Control and Decision Conference*, IEEE, pp. 179-183, 2024.
[6] C. Lu, J. Zhou, L. Gao, X. Li, and J. Wang, "Modeling and multi-objective optimization for energy-aware scheduling of distributed hybrid flow-shop," *Applied Soft Computing*, vol. 156, p. 111508, 2024.
[7] W. Shao, Z. Shao, D. Pi, and J. Gao, "Energy-efficiency oriented distributed heterogeneous hybrid flow shop scheduling with multilevelled mixed-model assembly," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2025.
[8] W. Zhang, Y. Wang, Y. Yang, and M. Gen, "Hybrid multiobjective evolutionary algorithm based on differential evolution for flow shop scheduling problems," *Computers & Industrial Engineering*, vol. 130, pp. 661-670, 2019.
[9] M. Marichelvam, M. Geetha, and Ö. Tosun, "An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors–A case study," *Computers & Operations Research*, vol. 114, p. 104812, 2020.
[10] O. Engin, G. Ceran, and M. K. Yilmaz, "An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems," *Applied Soft Computing*, vol. 11, no. 3, pp. 3056-3065, 2011.
[11] Y. Yuan-yuan, Y. Chun-ming, and Y. Feng, "Solving bi-objective reentrant hybrid flow shop scheduling problemsby a hybrid discrete grey wolf optimizer," *Operations Research and Management Science*, vol. 28, no. 8, p. 190, 2019.
[12] D. M. Utama, B. N. I. Farida, U. Fitriani, M. F. Ibrahim, and D. S. Widodo, "Hybrid Henry gas solubility optimization: An effective algorithm for fuel consumption vehicle routing problem," *Jurnal Ilmiah Teknik Industri*, vol. 20, no. 2, pp. 141-152, 2021.
[13] M. Marichelvam and M. Geetha, "Application of novel harmony search algorithm for solving hybrid flow shop scheduling problems to minimise makespan," *International Journal of Industrial and Systems Engineering*, vol. 23, no. 4, pp. 467-481, 2016.

[14] R. Zhou, D. Lei, and X. Zhou, “Multi-objective energy-efficient interval scheduling in hybrid flow shop using imperialist competitive algorithm,” *IEEE Access*, vol. 7, pp. 85029-85041, 2019.
[15] D. M. Utama, M. F. Ibrahim, D. S. Wijaya, D. S. Widodo, and M. D. Primayesti, “A novel hybrid multi-verse optimizer algorithm for energy-efficient permutation flow shop scheduling problem,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 2394, no. 1, p. 012006, 2022.
[16] X. Cai, X. Li, N. Razmjooy, and N. Ghadimi, “Breast cancer diagnosis by convolutional neural network and advanced thermal exchange optimization algorithm,” *Computational and Mathematical Methods in Medicine*, vol. 2021, no. 1, p. 5595180, 2021.
[17] M. F. F. Ab Rashid, M. A. N. Mu'tasim, , “A novel Tiki-Taka algorithm to optimize hybrid flow shop scheduling with energy consumption,” *Engineering and Applied Science Research*, vol. 49, no. 2, pp. 189-200, 2022.
[18] M. F. F. and M. A. Nik Mu’tasim, “Cost-based hybrid flow shop scheduling with uniform machine optimization using an improved tiki-taka algorithm,” *Journal of Industrial and Production Engineering*, vol. 41, no. 2, pp. 182-99, 2023.
[19] M. A. N. Mu'Tasim and M. F. F. A. Rashid, “Hybrid flow shop scheduling problem with energy utilization using non-dominated sorting genetic algorithm-III (NSGA-III) optimization,” *International Journal of Automotive and Mechanical Engineering*, vol. 20, no. 4, pp. 10862–10877, 2023.
[20] N. Nelfiyanti, A. S. H. Md Yasir, N. Mohamed, and M. F. A. Ahmad. “Adoption and impact of Quick Response Manufacturing across industry sector,” *Operations Research Forum*, vol. 6, no. 2, p. 64, 2025.
[21] N. H. Zakaria, N. M. Z. Nik Mohamed, M. F. F. Ab Rahid, and A. N. Mohd Rose. “Lean manufacturing implementation in reducing waste for electronic assembly line,” In *MATEC Web of Conferences*, vol. 90, p. 01048, 2017.
[22] Zuo, Yang, Fuqing Zhao, and Yang Yu. “Two-stage learning scatter search algorithm for the distributed hybrid flow shop scheduling problem with machine breakdown,” *Expert Systems with Applications*, vol. 259, p.125344, 2025.
[23] R. Stewart, A. Raith, and O. Sinnen, “*Optimising makespan and energy consumption in task scheduling for parallel systems,” Computers & Operations Research*, vol. 154, p. 106212, 2023.
[24] M. Parichehreh, H. Gholizadeh, A. M. Fathollahi-Fard, K. Y. Wong, “An energy-efficient unrelated parallel machine scheduling problem with learning effect of operators and deterioration of jobs,” *International Journal of Environmental Science and Technology*, vol. 21, no. 15, p. 9651-9676, 2024.
[25] H. Zhang, C. Qin, G. Xu, Y. Chen, and Z. Gao, “An energy-saving distributed flexible job shop scheduling with machine breakdowns,” *Applied Soft Computing*, vol. 167, p. 112276, 2024.
[26] I. Rastgar, J. Rezaeian, I. Mahdavi, and P. Fattahi, “Modeling and solving the multi-objective energy-efficient production planning and scheduling with imperfect maintenance activities,” *Journal of Quality in Maintenance Engineering*, vol. 30, no. 1, p. 26-50, 2024.
[27] Y. J. Xiong, S. Y. Cheng, and B. Chen, “Mitigating lifetime-energy-makespan issues in reliability-aware workflow scheduling for big data,” *Journal of Circuits, Systems and Computers*, vol. 31, no. 1, p. 2250012, 2022.
[28] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” Journal of global optimization, vol. 11, pp. 341-359, 1997.
[29] J. Carlier and E. Néron, “An exact method for solving the multi-processor flow-shop,” RAIRO. Operations Research, vol. 34, no. 1, pp. 1–25, 2000.