INTERNATIONAL JOURNAL OF AUTOMOTIVE AND MECHANICAL ENGINEERING
ISSN: 2229-8649 e-ISSN: 2180-1606
VOL. 20, ISSUE 4, 2023, 10862 – 10877
DOI: https://doi.org/10.15282/ijame.20.4.2023.05.0840

RESEARCH ARTICLE

Hybrid Flow Shop Scheduling Problem with Energy Utilization using Non-Dominated Sorting Genetic Algorithm-III (NSGA-III) Optimization

M. A. N. Mutasim* and M. F. F. A. Rashid

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

ABSTRACT - Hybrid flow shop scheduling (HFS) is an on sought problem modelling for production manufacturing. Due to its impact on productivity, researchers from different backgrounds have been attracted to solve its optimum solution. The HFS is a complex dilemma and provides ample solutions, thus inviting researchers to propose niche optimization methods for the problem. Recently, researchers have moved on to multi-objective solutions. In real-world situations, HFS is known for multi-objective problems, and consequently, the need for optimum solutions in multi-objective HFS is a necessity. Regarding sustainability topic, energy utilization is mainly considered as one of the objectives, including the common makespan criteria. This paper presents the existing multi-objective approach for solving energy utilization and makespan problems in HFS scheduling using Non-Dominated Sorting Genetic Algorithm-III (NSGA-III), and a comparison to other optimization models was subjected for analysis. The model was compared with the most sought algorithm and latest multi-objective algorithms, Strength Pareto Evolutionary Algorithm 2 (SPEA - II), Multi-Objective Algorithm Particle Swarm Optimization (MOPSO), Pareto Envelope-based Selection Algorithm II (PESA-II) and Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D). The research interest starts with problem modelling, followed by a computational experiment with an existing multi-objective approach conducted using twelve HFS benchmark problems. Then, a case study problem is presented to assess all models. The numerical results showed that the NSGA-III obtained 50% best overall for distribution performance metrics and 42% best in convergence performance metrics for HFS benchmark problems. In addition, the case study results show that NSGA-III obtained the best overall convergence and distribution performance metrics. The results show that NSGA-III can search for the best fitness solution without compromising makespan and total energy utilization. In the future, these multi-objective algorithms' potential can be further investigated for hybrid flow shop scheduling problems.

ARTICLE HISTORY

Received	:	13th June 2023
Revised	:	19th Sept 2023
Accepted	:	15th Oct 2023
Published	:	26th Dec 2023

KEYWORDS
Hybrid flow shop
NSGA-III
Scheduling
Optimization

1.0 INTRODUCTION

The essential aspect of the manufacturing area is the production schedule. This activity involves many stakeholders and resources, such as raw materials, machines, spare parts, and human resources. In many manufacturing industries, this scheduling often occurs through job and machine assignments. One primary production environment that includes job and machine scheduling is the hybrid flow shop (HFS). HFS is a combination of parallel machines and flow shops. The flows are being processed in stages. A parallel machine is present for each stage to reduce manufacturing bottlenecks. Since flow shops have a strict flow model that allows only one job at a single stage, parallel machines with the same capability allow multiple movements of jobs per stage. Hence, modellers must understand the constraints of each model. First, being a predetermined sequence; second, the precedence must finish first prior to the following job process; and third, the machines must be available for jobs to process [1]. Many have studied multi-objective HFS scheduling problems before [2]. The most basic version is an identical parallel machine, where all machines in every stage are identical. The second version is a uniform parallel machine in which the processing time depends on the machine's speed. The last version is the unrelated parallel machine, which does not depend on another machine. The processing for the same process might be different because of model variation[3].

In this paper, a study on unrelated parallel machines in the flow shop environment and the study of energy utilization is presented. The other objective concerns minimizing the total completion time or makespan. Previous studies had other prominent objectives, such as maximum lateness, machine utilization, penalty, and tardiness [4]. A new goal concerning environmental factors in manufacturing has since had much of the focus in the world as the energy crisis emerges. For example, the goal is to minimize energy costs, total carbon emissions, and noise pollution in HFS scheduling. Thus, this study focuses on investigating energy utilization in HFS machines. A single-objective analysis is abundant; however, multi-objective case studies are more reasonable and require more endeavor in real-world case studies. In recent years, multi-objective problems have been the leading research topic. Recently, several articles have focused on multi-objective HFS with energy efficiency (EE-HFS). A study by Li et al. [4] gave a model of EE-HFS with machines in both the standby and processing states. This model considered the setup time (gaps) between each job. On the other hand, a different study

***CORRESPONDING AUTHOR | M. A. N. Mutasim** | ✉ ammar@umpsa.edu.my

reveals that EE's challenges are finding solutions to energy prices while the power rate is in effect [2]. In another work, research on reducing EE-HFS exposure during off-peak processing hours was also in attention [5].

The weighted sum and Pareto-based methods are the main approaches to managing a multi-objective optimization issue. A significant number of the algorithms in EE-HFS make use of the problem. In the past five years, an algorithm based on genetic algorithms (GA) has been used to solve the EE-HFS problem [6] and then improvised with a Pareto-based method using a similar algorithm [7]. Other evolutionary algorithms (EA) were applied in similar years by Jiang and Zhang and Gong et al. to tackle the emotional EE-HFS problem [8, 9]. In recent years, various alternative algorithms, such as the moth flame algorithm [10], the tiki-taka algorithm [11], and the firefly algorithm [12], have been utilized in order to solve EE-HFS. The performance of these brand-new algorithms was compared to that of well-known algorithms like the GA, EA, ant colony optimization method, particle swarm optimization algorithm, and artificial bee colony algorithm. All the comparisons demonstrated the extraordinary capabilities of each algorithm based on several different criteria, such as the best fitness, the amount of processing time, and the objective values. In the scenario of the Pareto-based algorithm, it is common knowledge that there needs to be more work done to date on the Pareto-based approach to resolving EE-HFS problems.

Based on earlier investigations, numerous optimization techniques have been developed to maximize HFS energy usage. One of the reasons for algorithm variety is the fact that no one algorithm can consistently outperform another. In other words, the efficiency of an optimization technique depends on a number of factors, such as the program's coding, the computational representation of a problem, and the test, problem, or case study used during the experiment. Researchers often tweak well-known EA and GA algorithms like SPEA-II, PESA-II, and NSGA-II in addition to developing new algorithms to meet current challenges and issues. The study selects the most sought-after and well-liked algorithms to resolve HFS energy challenges for the current research subject. The model was validated using the most widely recognized benchmarks in the HFS field. The analysis also used actual industrial case studies. By carefully examining each algorithm's capabilities and comparing them using performance metrics tailored for multi-objective situations, this work aims to fill the gap in the algorithms (most frequently cited algorithms).

As a result, the results from the chosen algorithms can be used to develop a better method to address HFS with energy concerns. Additionally, this study uses the Non-Dominated Sorting Genetic Algorithm III (NSGA-III), an improved version of NSGA-II with a superior selection mechanism to maximize crowding distance to achieve population variety. This paper discusses the results of computer experiments and the case study. The remainder of this paper is organized as follows: Section 2 briefly presents an overview of the optimization model. Section 3 explains the NSGA-III algorithm, and Section 4 provides the experimental settings and discusses the experimental results and case study, while Section 5 concludes the paper.

2.0 MODELLING OF HYBRID FLOW SHOP SCHEDULING

2.1 Hybrid Flow Shop Scheduling

A hybrid flow shop (HFS) scheduling problem combines a flow shop scheduling problem with a scheduling issue for parallel machines. For HFS, *n* jobs must be processed in *k* steps. If there are no other jobs on that machine, the job can be processed at any stage 1 machine before moving on to any stage 2 machine. Jobs must come before one another and go through the procedures at each stage, as shown in Figure 1.

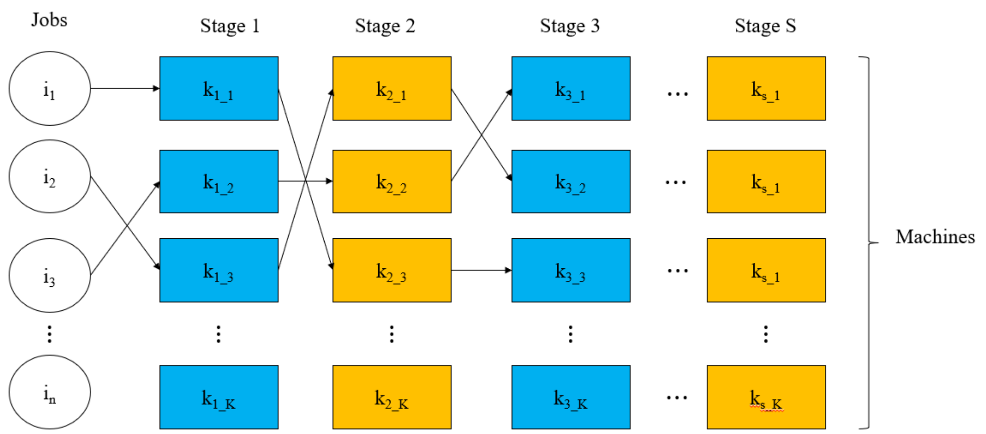

Figure 1. Hybrid flow shop environment

There could be different machines at each level. There must be at least one machine at each stage. Additionally, there are three habitats for the machines, including uniform, identical, and unrelated machines. In the stages of this case study, there are nonidentical machines, which suggests that their processing speeds and power ratings are different. The reduction of completion time (makespan) and energy consumption are the objectives of EE-HFS in this work. The total energy varies depending on the job assignment for scheduling due to the varying power ratings for the equipment. For idle machines, no power was calculated. Due to varying models, power consumption, and speed variations, the energy and time numbers for each machine in a stage in the cited problem are varied. Since the hybrid flow shop has multiple stages, its flow is unidirectional. HFS requires it to adhere to two procedures. The first is the order in which jobs are processed, and the second is the availability of machines, where jobs and machines are assigned before processing. These notations display the algorithm's steps.

Equation (1) represents the process's utmost completion time and minimizes the makespan from optimization for objective one. The second objective is to reduce energy consumption; Eq. (2) pertains to the total energy calculated by multiplying the job operation time by the machine's power consumption for the entire processing time. However, when machines are idle, they are deemed to be turned off. Applying the mathematical model from Rashid and Mutasim [11], Eq. (1) and (2), there are two primary objectives to consider: minimizing the completion time C_{max} and the total usage of machine energy (EE). The constraints (3), task, and I can only be processed at a single machine per stage. Equation (4) is the continuous matrix representation for the HFS problem. The minimum value of each row is represented by Equation (5), and Equation (6) determines the ascending sorting of the task sequence based on Equation (7). In the following problem statement, an illustration of the representation of equations is shown below.

$$C_{\max} = \max\left\{C_i\right\}_{i=1,2,3,\ldots,n} \tag{1}$$

where C_{max} is the makespan time in minutes, C_i is the completion time of job i and n is the number of jobs.

$$EE = \sum_{i=1}^{n}\sum_{s=1}^{S}\sum_{k_s=1}^{K_s} t_{ik_s}.E_{ik_s}.y_{ik_s} \tag{2}$$

where EE is the total energy utilized by multiplying the processing time on machine k at stage s with the rate of power on machine k at stage s in Watt and then multiplied by the availability of the machine either 0 or 1 for off or on respectively.

Subjected to:

$$y_{ik_s}\begin{cases}1\\0\end{cases}$$

1 is the if job, i is processed at machine k_s and 0 is else

$$\begin{aligned}&\sum_{k_s=1}^{K_s} y_{ik_s} = 1,\\&\forall i = 1,2,\ldots,n;\\&s = 1,2,\ldots,S\end{aligned} \tag{3}$$

where S is the number of stages

$$X = \begin{pmatrix} x_{1,1,1} & \cdots & x_{1,S,K_s} \\ \vdots & \ddots & \vdots \\ x_{n,1,1} & \cdots & x_{n,S,K_s} \end{pmatrix} \tag{4}$$

$$X_{i_{\min}} = \begin{bmatrix} x_{1_{\min}} \\ x_{2_{\min}} \\ \cdots \\ x_{n_{\min}} \end{bmatrix} \tag{5}$$

$$X'_j = ascending\left\{X_{i_{\min}}\right\} \tag{6}$$

$$J = \left\{j\right\} \; \forall i = 1,2,\ldots,n; \tag{7}$$

where J is the number of jobs.

3.0 NON-DOMINATED SORTING GENETIC ALGORITHM III (NSGA-III)

The previous NSGA II proposed by Deb et al,[13] calculates the crowding distance of everyone by measuring their diversity and thus preserve a set of diverse solutions. The offspring is created, by selecting the best N member from the parent then combined into a 2N population of parents and offspring R_t , $P_t \cup Q_t$. This keeps the elite members of the parents as well as diverse the offspring. The offspring R_t is sorted by choosing a non-dominated offspring in a rank of the objective functions. The non-dominated is selected one at a time to construct a new population S_t. The selection starts from F_t until S_t until it reaches the size of N population. The solution that maximizes the diversity of the Pareto front will be chosen. The preservation of the population is done through a niche operator that calculates the crowding distance of each member. The final output of the NSGA II will present a set of non-dominated solutions that approximate Pareto front with diverse solutions.

The basic process of NSGA-III presented by Deb and Jain [14] remains like the NSGA-II. The difference between the NSGA-II and NSGA-III is the substantial change in the selection mechanism. Unlike NSGA-II, the NSGA-III maintains its diversity by updating a well spread reference point. To ensure the identification of the non-dominated front, reference points using normalized hyperplane a (M-1)-dimensional unit simplex are chosen during the optimization process. The members of the population are chosen based on the points of reference shown in equation 8 below.

$$H = \binom{M + p - 1}{p} \tag{8}$$

H refers to the number of reference points and M refers to the number of objectives. Say for example a three-objective problem (M = 3), assuming there are four divisions (p = 4) are chosen for each objective axis. $H = \binom{6}{4}$ or 15 reference points shown in Figure 2 below. The generation process of NSGA-III can be referred to in Figure 3.

At first, the ideal point of population S_t is identified with the minimum value of each objective function, then it is translated by subtracting the objective function value with the ideal point. The process is done until it finds the extreme farthest point solution to an objective function through weight vector identification. Once the other furthest objective function is found, it is then intercepted together in a single axis of a hyperplane using a normalized equation 9 below.

$$f_i^n(x) = \frac{f_i^{'}(x)}{a_i - z_i^{\min}} = \frac{f_i^{'}(x) - z_i^{\min}}{a_i - z_i^{\min}}, \quad \text{for I} = 1,2\ldots,M \tag{9}$$

where the $f_i^n(x)$ is the objective function and $f_i^{'}(x)$ is the objective direction at i^{th} objective axis and $z_i^{\min}$ is the ideal point.

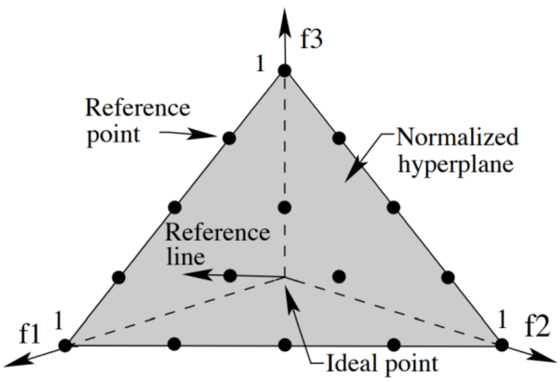

Figure 2. 15 reference points for 3 objectives and p =4

Once normalizing is done, next, a reference line is defined on the hyperplane by joining the reference point with the origin. Then, each population member is then associated with the reference point calculated using a perpendicular distance to the reference line as shown in Figure 3.

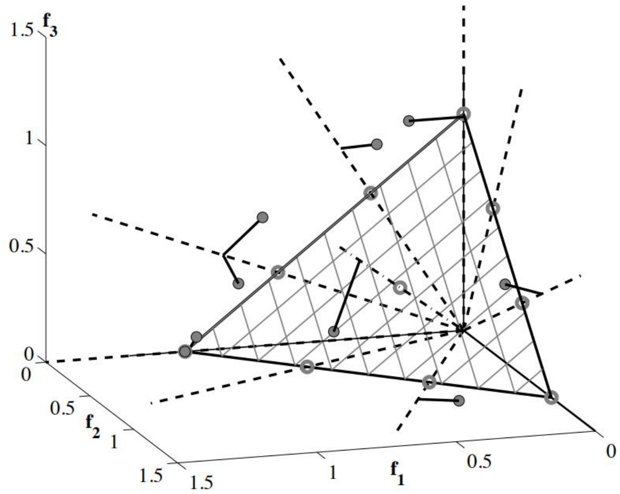

Figure 3. Population member is associated with the reference point using a reference line

The niche preservation of the algorithm is measured by the association of the population members to a reference point. Sometimes, there can be many members to a point and sometimes it can be none. In the case of the former, the reference line is increased by P_{t+1}. Whereas for the latter, the reference points are excluded from further consideration for the current generation. Once the P_{t+1} is created, a new offspring population Q_{t+1} is created using the same operator. This allows the algorithm to maintain its diversity. The full flowchart of NSGA-III is shown in Figure 4 below.

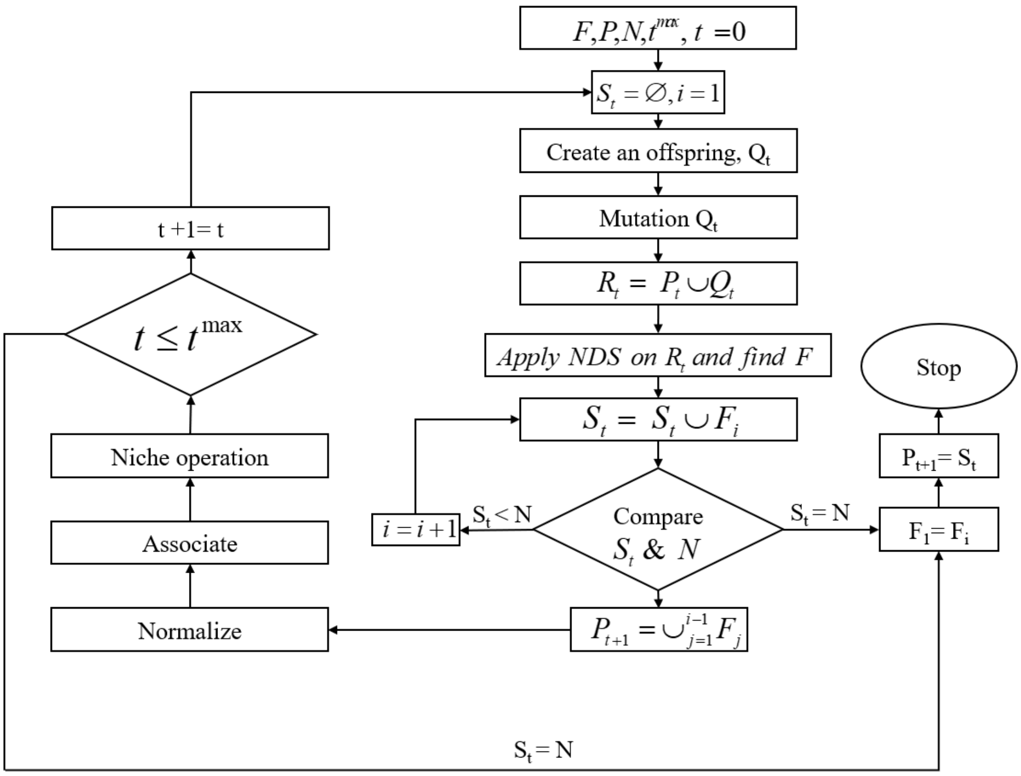

Figure 4. NSGA-III generation process

4.0 COMPUTATIONAL EXPERIMENT

A computational experiment with Carlier and Neron benchmark problem has been conducted [15]. The problem consisted of 10 to 15 jobs and 5 to 10 stages with different machine configurations on each stage. The benchmark problem is made for comparison purposes for each algorithm. Meanwhile, the case study problem is meant to verify the algorithms for real-life problems. The optimization is repeated 20 times for computational experiments, with a population number of 100 and maximum iterations of 300 for each algorithm. Table 1 shows the benchmark test problem configurations by Carlier and Neron.

The benchmark problem configured by Carlier and Neron was utilized as a hypothetical test problem for many flow shop scheduling problems. The detail of the problem consists of processing times that are randomly generated using normal distribution between [3, 20]. For this case, objective 2, the energy utilization was included and randomly generated as well by varying the values of [1.0 1.5, 2.0, 2.5, 3 3.5, 4.0] kW for each machine.

Table 1. Benchmark problem configurations by Carlier and Neron

No.	Benchmark Problem	No. of Jobs	No. of stage	Machine configuration for each stage	Normal Distribution of Cmax (min) / EE (kW)
1.	j10c5a2	10	5	2 2 1 2 2	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
2.	j10c5b1	10	5	1 2 2 2 2	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
3.	j10c5c1	10	5	3 3 2 3 3	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
4.	j10c5d1	10	5	3 3 3 3 3	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
5.	j10c10a2	10	10	2 2 2 2 1 2 2 2 2 2	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
6.	j10c10b1	10	10	1 2 2 2 2 2 2 2 2 2	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
7.	j10c10c1	10	10	3 3 3 3 2 3 3 3 3 3	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
8.	j15c5a1	15	5	3 3 1 3 3	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
9.	j15c5b1	15	5	1 3 3 3 3	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
10.	j15c5c1	15	5	3 3 2 3 3	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
11.	j15c10a2	15	10	3 3 3 3 1 3 3 3 3 3	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]
12.	j15c10b1	15	10	1 3 3 3 3 3 3 3 3 3	[3, 20] / [1 2.5, 3.0, 3.5, 4 2.5, 3.0, 3.5, 4]

Five Pareto-based multi-objective algorithms were chosen and compared in the optimization process. Seven performance metrics were used adopted from deb [16] Bezerra et al.[17] and Yoosefelahi et al. [18] for a comparative study between the five different algorithms. The performance is based on solutions as close to the Pareto front solutions. Moreover, second, to find diverse solutions in the non-dominated solutions. The extent performance measures the minimum optimum value in the Pareto optimal solutions. For this case, the minimum value of makespan and energy utilization from Pareto optimal solutions taken from the 20 times run will be measured along with the value of the other objective. The second performance is the range which is the relative distance of each Pareto optimal solution. Seven performance metrics were used to quantify the results from 5 non-dominated solutions. These are (i) Pareto percentage (%), (ii) Error ratio (ER); (iii) Generational Distance (GD); (iv) Inverted Generational Distance (IGD); (v) Spacing (Sp), (vi) Maximum Spread (MS) and Normalized Hypervolume (NHV). All metrics are obtained for measuring the solutions' cardinality, convergence, and distribution.

Cardinality is a measurement to quantify non-dominated points generated by an algorithm. The metrics used for this indicator are Pareto percentage (%) and Error Ratio (ER). The Pareto percentage shown in equation (10) below is the percentage of Pareto solution from the algorithm over the total number of Pareto fronts. The Error Ratio represents the non-dominated solution from the algorithm that is not on the Pareto front. The equation for Error ratio is shown in equation (11) below. The smaller the ER and Pareto % value, the better the performance.

$$Pareto\ Percentage\ =\ \left(Number\ of\ Non-Dominated\ Solutions\ /\ Total\ Number\ of\ Solutions\right)\ \times\ 100\% \tag{10}$$

$$ER = \frac{\sum_{i=1}^{|Q|} e_i}{|Q|} \tag{11}$$

where Q is the number of solutions of q algorithm that are not in the Pareto set,

The second indicator is the convergence of the solution. The metrics referring to this indicator are the Generational Distance (GD) also shown in equation (12) and the Inverted Generational Distance (IGD) as shown in equation (13) below. These measurements compute the proximity of the set of non-dominated solutions from the Pareto front. The better algorithm is the performance that obtained the smallest GD and IGD values.

$$GD_q = \frac{\sum_{i=1}^{S_q} d_i}{S_q} \tag{12}$$

where S_q is the q algorithm solutions and, $d_{i,}$ is the Euclidean distance in objective space between the solutions and the nearest member.

$d_i = \min_{k=1}^{P} \sqrt{\sum_{m=1}^{M} \left(f_m^{(i)} - f_m^{*(k)} \right)^2}$ with $f_m^{(i)}$ *m*th objective function value of solution i.

$$IGD_q(A,R) = GD_q(R,A) \tag{13}$$

where A is the approximation Pareto front and R is the reference Pareto front.

Finally, the last indicator is distribution and spread. These indicators refer to Spacing (Sp), the relative distance between the solution, as shown in equation (14) and Maximum Spread (MS), and the extent of distribution by two extreme points, as shown in equation (15). These indicators measure the distribution of non-dominated solutions and aspects related to the coverage of objective space and the spread of the points. The smaller the space between the non-dominated solutions and the larger the points spread, the better the results. Finally, normalized hypervolume (NHV) shown in equation (16), determines the diversity of the solution. Each objective function's maximum solution (worst solution) is considered similar for each algorithm. For HV, the most significant area covered by the algorithm has the best performance of the solution.

$$Sp = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} \left(d_i - \overline{d} \right)^2} \tag{14}$$

where $\overline{d}$ is the mean value of the above distance d_i, $\overline{d} = \sum_{i=1}^{|Q|} d_i / |Q|$

$$MS = \sqrt{\sum_{i=1}^{M|} \left(\min f_i - \max f_i \right)^2} \tag{15}$$

$$HV = volume(\cup_{i=1}^{|Q|} v_i) \tag{16}$$

where v_i is a hypercube constructed by a reference point and solutions i's as the corners of the hypercubes.

4.1 Algorithm Comparison with Benchmarks

Consequently, computational experiments were conducted to compare NSGA III performance against other multi-objective Pareto-based algorithms. In this study, five, including NSGA-III, were chosen to optimize the benchmark and the case study problem. The algorithms were the fitness assignment dominance rank elitist, Non-Dominated Sorting Genetic Algorithm (NSGA-III), Strength Pareto Evolutionary Algorithm (SPEA-II) [19], Multi-objective Particle Swarm Optimization (MOPSO) [20], Pareto Envelope-based Selection Algorithm II (PESA-II) [21] and Multi-objective Evolutionary Algorithm by Decomposition (MOEA/D) [22]. These algorithms differ based on their fitness assignment rank, performance, and reputation for solving multi-objective problems. The SPEA-II and PESA-II are referred for comparison because they are well-established algorithms and have reasonable cardinality solutions. The MOEA/D is one of the most well-liked algorithms for comparison. MOEA/D algorithm can give good convergence results. The MOPSO is a popular algorithm introduced by Coello and Lechuga. It is a dominant algorithm and used throughout the HFS scheduling problem. Next, after the benchmark, a case study in a machine shop is evaluated using these algorithms. This work is vital to explain the capability of the algorithms as well as the capacity to solve efficiently in real-life uses.

Tables 2-8 below present the optimization result based on the seven metrics explained above. The values in bold and asterisks represent the best results for each metric and the benchmark problem. It is important to note that no algorithms stand out with better results than others for every indicator. All algorithms have their capabilities according to the indicators that were provided.

For the ranking system, the hierarchy is based on standard competition systems. For bolded results, it indicates that the rank is number one. The worst result is assigned by number 5. If a similar number appears, that means the ranks are tied. The following rank will be visually left empty. The Table indicates the mean ranking by the algorithms for all metrics. Table 9 shows the performance of algorithms based on three leading indicators: mainly the cardinality solution, convergence, and distribution solution.

4.1.1 Cardinality Performance Metric

Based on the benchmark results, SPEA-II was recognized as the best cardinality by claiming both the highest Pareto % and lowest Error Ratio metric. According to the benchmark problems, about 42% or 5 out of 12 benchmarks of the cases have shown that SPEA-II ranked first, followed closely by NSGA-III, with 33% of the benchmarks obtained as first place. The other algorithms share the other 15 % of first place. SPEA-II and NSGA-III have also achieved first and second place on over 50% of benchmark problems. These algorithms can provide more Pareto-front solutions compared to other algorithms. Hence, it shows that SPEA-II and NSGA-III can obtain good cardinality in the objective space.

Table 2. Metric solution using multi-objective optimization algorithms for Error Ratio (*ER*)

Problems	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
j10c5a2	0.5560*	0.9000	0.8182	0.7500	0.7778
j10c5b1	0.9375	0.8000	0.2000*	1.0000	0.7143
j10c5c1	0.9000	0.6000*	0.7777	1.0000	0.8750
j10c5d1	0.8751	0.8750	0.0000*	1.0000	1.0000
j10c10a2	0.6667	0.6000*	0.6400	1.0000	0.6000*
j10c10b1	1.0000	0.5000*	0.7232	0.7500	0.8333
j10c10c1	0.5714	0.8000	0.7500	1.0000	0.3750*
j15c5a1	0.3333*	0.6667	0.7500	1.0000	1.0000
j15c5b1	1.0000	0.5714	0.2000*	1.0000	0.8000
j15c5c1	0.8333	0.5000*	0.8000	0.5000*	0.5000*
j15c10a2	0.7857	0.6000	0.5000*	1.0000	1.0000
j15c10b1	0.8889	0.5714*	0.6667	1.0000	0.6000

Table 3. Metric solution using multi-objective optimization algorithms for Pareto Percentage (*%*)

Problems	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
j10c5a2	44.44*	10.00	18.18	25.00	22.22
j10c5b1	6.25	20.00	80.00*	0.00	28.57
j10c5c1	10.00	40.00*	22.22	0.00	12.50
j10c5d1	14.29	12.50	100.00*	0.00	0.00
j10c10a2	33.33	40.00*	36.36	0.00	40.00*
j10c10b1	0.00	50.00*	27.27	25.00	16.67
j10c10c1	42.86	20.00	25.00	0.00	62.50*
j15c5a1	66.67*	33.33	25.00	0.00	0.00
j15c5b1	0.00	42.86	80.00*	0.00	20.00
j15c5c1	16.67	50.00*	20.00	50.00*	50.00*
j15c10a2	21.43	40.00	50.00*	0.00	0.00
j15c10b1	11.11	42.86*	33.33	0.00	40.00

4.1.2 Convergence Performance Metric

Meanwhile, two metrics were analyzed for the average convergence indicator: the generational distance (GD) and the inverted generational distance (IGD). The GD, founded by Van Veldhuizen and Lamont [23] is a metric to calculate the distance between the non-dominated solutions with the Pareto optimal front. The algorithm that has the lowest GD has the best convergence with the optimal front. GD metric, the NSGA-III, PESA II and SPEA-II all achieved similar first-place results of 25% each. When combined with first and second place, MOEA/D obtained the best rank of more than 29% of all 12 benchmarks. The result also shows that when combined first place and second place positions, NSGA-III received 25% and PESA-II 21%. These indications show that MOEA/D non-dominated solutions approach near the Pareto front with the smallest GD values. However, for IGD, PESA-II and SPEA-II were shown to have better 1st-place IGD values for 12 benchmarks, with both obtaining a similar 33% each. PESA-II also achieved 33% of first and second rank in JGD, followed by NSGA-III with 29%. When combined with GD and IGD ranks, PESA-II obtained an average of 2.375 in the overall average rank position, followed by NSGA-III with 2.45833 in the average rank position. PESA-II and NSGA-III also obtained 27% each in the first and second rank for smaller convergence values for benchmark problems.

Table 4. Metric solution using multi-objective optimization algorithms for Generational Distance (*GD*)

Problems	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
j10c5a2	1.9222*	4.8201	10.9182	21.5250	5.9111
j10c5b1	6.9877	4.2011	0.2200*	8.9715	2.6000
j10c5c1	26.9858	4.4707*	22.7924	29.0654	19.9895
j10c5d1	8.9572	19.3313	0.1200*	11.0320	8.7900
j10c10a2	5.8750*	22.4951	13.5182	15.3417	8.7700

Table 4. (Cont.)

Problems	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
j10c10b1	12.6500	17.5000	5.1729	37.1125	4.0417*
j10c10c1	3.2501	20.2750	60.5460	85.9140	1.2500*
j15c5a1	0.7167*	25.7167	13.8250	16.8800	16.9750
j15c5b1	11.6300	8.7000	4.4400*	23.3200	7.0246
j15c5c1	7.8167	25.6333	50.5800	2.0000*	11.5500
j15c10a2	34.0143	10.6350*	38.4188	17.2143	19.5143
j15c10b1	18.2389	4.1500*	13.2500	19.0364	16.1000

Table 5. Metric solution using multi-objective optimization algorithms for Inverted Generational Distance (*IGD*)

Problems	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
j10c5a2	1.1200*	6.4901	4.1900	15.5500	3.3400
j10c5b1	3.8902	1.7516*	4.1800	10.2500	7.9200
j10c5c1	17.4576*	27.8310954	22.8536	37.4587	21.3080
j10c5d1	7.9286	17.1357	3.6429*	15.7360	11.3357
j10c10a2	13.3792*	23.9251	15.4500	29.4125	68.5708
j10c10b1	10.4950	7.9900	6.5000*	46.0300	19.1300
j10c10c1	13.1808	18.7269	10.4577	26.2962	6.1192*
j15c5a1	11.8625*	23.0500	12.6250	27.5500	50.4125
j15c5b1	17.7500	8.9250*	15.0875	37.6500	46.5779
j15c5c1	16.5857	10.6857*	19.1286	32.1143	30.5000
j15c10a2	9.6228	8.5545*	15.5818	21.5227	37.7909
j15c10b1	37.9333	45.5722	12.3000*	37.6667	94.0000

4.1.3 Distribution Performance Metric

Finally, three metrics were measured for all five algorithms using the benchmark problems for distribution. The Spacing (SP) [24] refers to the non-dominated solution distributions on the objective space. The lowest value of the SP leads to the best uniform distribution in the Pareto front. For the spacing metric, NSGA-III and MOEA/D both obtain 33% first rank. However, when combined with first and second-rank positions, the MOEA/D obtained an overall 33%, followed by NSGA-III with 20.8%. For the maximum spread (MS) metric, SPEA-II achieved the best position from 5 of the 12 benchmark problems. In the second position, NSGA-III attained 4 of the 12 benchmark problems in the first position. Lastly, for hypervolume metric (HV), NSGA-III attained first rank from 5 from 12 benchmark problems, followed by three other algorithms, PESA-II, MOEA/D and MOPSO, with 2 out of 12 problems in the first rank. As represented by Sp, MS and Normalized HV metrics, both first ranked for Sp and NHV metrics were obtained by NSGA-III, while SPEA-II obtained ranked first for MS.

Table 6. Metric solution using multi-objective optimization algorithms for Spacing (Sp)

Problems	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
j10c5a2	3.5136*	6.0233	15.1541	39.7108	10.7836
j10c5b1	17.5040	5.3453	10.2270	1.8731*	4.1014
j10c5c1	19.2848	11.6283	14.3395	5.6279*	30.2827
j10c5d1	6.6373	24.2190	13.4759	18.8560	4.6799*
j10c10a2	24.9777	25.5270	5.9803*	47.8562	10.3715
j10c10b1	11.5359	12.5241	9.3419*	99.7870	17.1887
j10c10c1	8.9497	4.4977*	10.5119	8.6575	27.3765
j15c5a1	41.6750	37.5368	13.2247	32.0145	9.5525*
j15c5b1	18.8065	26.6898	3.9909*	21.9958	8.8746
j15c5c1	3.8223	18.9741	29.0145	2.0450*	2.0450*
j15c10a2	26.0276	15.5046	59.9746	14.9563	7.3748*
j15c10b1	88.5219	48.0206	14.4596*	66.3033	28.1469

Table 7. Metric solution using multi-objective optimization algorithms for Maximum Spread (MS)

Problems	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
j10c5a2	70.52	73.27	134.76*	115.00	101.10
j10c5b1	141.43*	116.85	75.27	83.79	52.90
j10c5c1	226.77*	148.29	147.31	132.75	186.87
j10c5d1	120.94	178.65*	123.23	141.05	104.05
j10c10a2	248.35	408.25*	268.65	273.05	133.30
j10c10b1	236.75	317.65*	200.90	295.70	156.00
j10c10c1	105.35	184.91	339.85*	327.05	201.95
j15c5a1	237.75	443.82*	305.20	198.80	132.90
j15c5b1	202.36	272.72*	217.20	244.30	89.60
j15c5c1	7.82	25.63	50.58*	2.00	11.55
j15c10a2	34.01	10.64	38.49*	17.21	19.51
j15c10b1	18.24	4.15	13.25	19.04*	16.10

Table 8. Metric solution using multi-objective optimization algorithms for Normalized Hyper Volume (NVH)

Problems	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
j10c5a2	74340.70	80760.30	85554.80*	34053.00	75685.40
j10c5b1	141400.01*	116800.01	75200.00	83700.01	52900.01
j10c5c1	134993.70	85493.70	146973.60*	92327.80	127176.10
j10c5d1	152505.75	170722.05	125564.40	182476.05*	106287.65
j10c10a2	181494.90	273180.45	292086.85*	131704.60	135724.50
j10c10b1	307691.00	295316.90	312921.25*	116989.70	160782.80
j10c10c1	308873.75	391134.15	439618.40	575054.05*	343429.65
j15c5a1	80447.70	79780.00	122576.60*	53387.90	56124.80
j15c5b1	159097.20*	118111.20	89283.40	65171.70	76750.20
j15c5c1	105355.40	166330.50*	116161.70	52731.60	51712.70
j15c10a2	513159.05	395179.65	303680.20	226484.40	5103511.00*
j15c10b1	327844.75	285462.65	318823.15	397100.15	6800165.00*

4.1.4 Discussion

Based on the visual explanation and the computational experiment results, the NSGA-III shows second best in all five metrics: Pareto %, ER, GD, IGD and SP. While for MS and HV, NSGA-III outperformed other algorithms with an overall first rank in both metrics. As for averaging all metrics, the NSGA-III obtained the best average rank of 2.48, followed by SPEA-II with 2.80 and closely followed by PESA-II with 2.90 in the third rank. The overall average shows that the performance of NSGA-III for all metrics ranked three and below for all benchmark problems.

Table 9. Average ranking obtained by multi-objective algorithms

Metrics	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
Pareto %	3.25	2.17*	2.33	3.92	2.67
ER	3.25	2.17*	2.33	3.92	2.67
GD	2.75	3.08	2.75	4.08	2.33*
IGD	2.00*	2.75	2.17	4.33	3.75
SP	3.33	3.25	2.83	3.08	2.42*
MS	3.17	3.42	2.58*	2.83	3.92
HV	2.58	2.75	2.33*	3.75	3.58

It is also shown in Table 10 below that when divided into cardinality indicator, convergence indicator and distribution indicator; the NSGA-III managed to attain at least rank two and below compared to the other algorithms. Based on the benchmark problems, it is shown that by averaging all three indicators, the NSGA-III slightly outperformed all other algorithms, maintained its rank overall, and indicates that the algorithm is stable in exploring and exploiting optimum solutions in the hybrid flow shop scheduling problem.

Table 10. Average ranking obtained by multi-objective algorithms

	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
Cardinality	4	1	2	5	3
Convergence	1	3	2	5	4
Distribution	3	2	1	4	5
Overall Rank	3	2	1	4	5

As shown in Figure 5 below, all algorithms have the ability to obtain near-optimal solutions. All five algorithms can obtain non-dominated solutions and form the Pareto front. The solution from all algorithms can be improved by increasing the maximum iteration. Since the maximum iteration was set to 300, the solution from all algorithms converged to some specific objective value. If given more iterations to run on, better and more significant achievement can be seen differently, especially from MOPSO, since it has achieved the fastest computational time for these benchmark problems. It can also be seen that all algorithms do not have outstanding performance. From a critical point of view, SPEA-II performs better in more extensive jobs and more prominent stages than other algorithms. Whereas for more minor case problems, the performance of these five algorithms has similar outcomes. Based on the evaluation study, after filtering the points and removing the same points, the raw value of the objective function shows that NSGA-III keeps the highest raw data, followed by MOPSO, PESA-II and SPEA-II. These can be improved by increasing the number of archived sizes of the multi-objective algorithms.

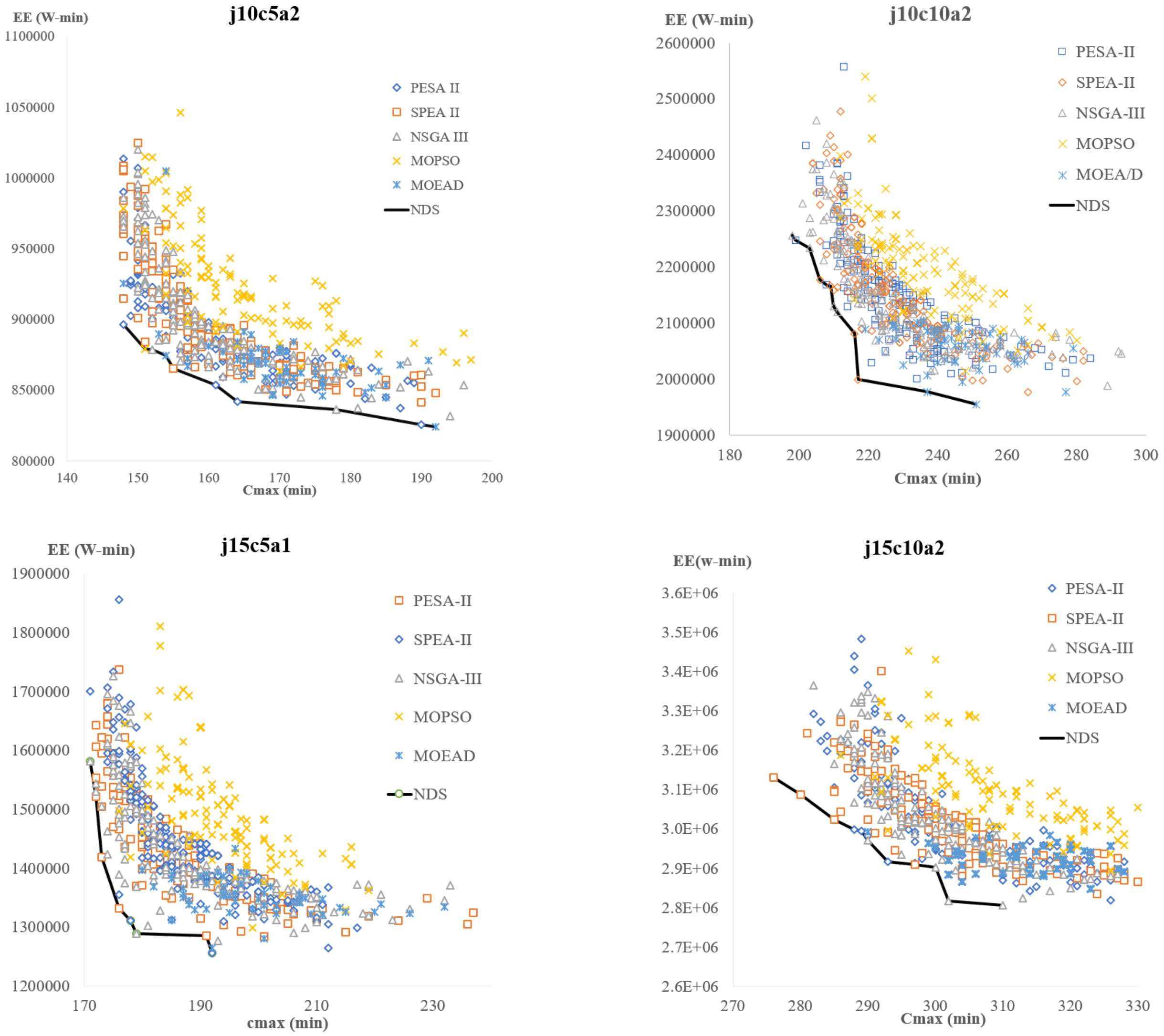

Figure 5. Non-Dominated Solutions and dominated solutions from selected benchmarks

5.0 INDUSTRIAL CASE STUDY

The purpose of conducting the case study was to validate the applicability of HFS model and NSGA-III algorithm in real life problem. The industrial case study was conducted at a company located in Kuantan, Malaysia. The organization manufactures a stainless-steel block. The store processes customized materials for use in various products. Each block is attired in accordance with its dimensions, shape, and profundity. Multiple machines and machine varieties are used to manufacture the product. Four machines were used to manufacture the stainless-steel base block: a bending machine, a lathe machine, a machining machine, and a wire-cutting machine. According to the number of available machines at the facility, there is one wire-cutting machine, two lathe machines, two milling machines, and three bending machines available in the facility. Each machine within a given category is unique, has a different setup and runs in its own processing time.

Additionally, the energy consumption of each machine varies too. Regarding the case study, the requirement from the client was to manufacture 27 stainless steel blocks. According to the design given, the process begins with bending, lathe, milling, and wire-cutting machines; these four machines must execute the jobs in a unidirectional fashion.

In the first stage, three bending devices are used to cut the material to the required dimensions. In the second and third stages, there are two lathe machines and two milling machines for eradicating surface roughness and face milling, respectively. In the final stage, one wire-cutting equipment is used for hole design. The scheduling procedure will only take these assets into account. The tasks to be processed contain holes of various shapes and sizes. Therefore, each task has a unique processing time. The dimensions of the block also differ from one to another. When contemplating running a similar process on a different machine type, capacity and power rate vary.

The staging process and the machine scheduling for manufacturing 27 block jobs are shown in Figure 6 below. The case study is focused on scheduling the jobs to the available machines. The processing time and energy usage vary according to the machine setup. The bending machine processing time to manufacture one job can vary between 11 to 22 minutes, with power rates ranging between 1000 to 1600 watts. The lathe machines in the workshop can process the block within 20 to 50 minutes and consume power from 1000 to 1500 watts. The milling process time is also like the lathe processing time, between 18 to 42 minutes for different jobs. The milling machine's power rate is between 1500 to 1800 watts. Finally, the wire cut machine to produce the holes in the block processing time ranges from 50 to 80 minutes depending on the hole size, multiple holes, and depth. The power rate of the machine is approximately 700 watts. The calculated time and power ratings obtained from the case study were tabulated. For this study, the proposed work is determining the best scheduling management for the following cycle arrangement. In addition, the scheduling can minimize energy consumption by utilizing the schedule of the machines.

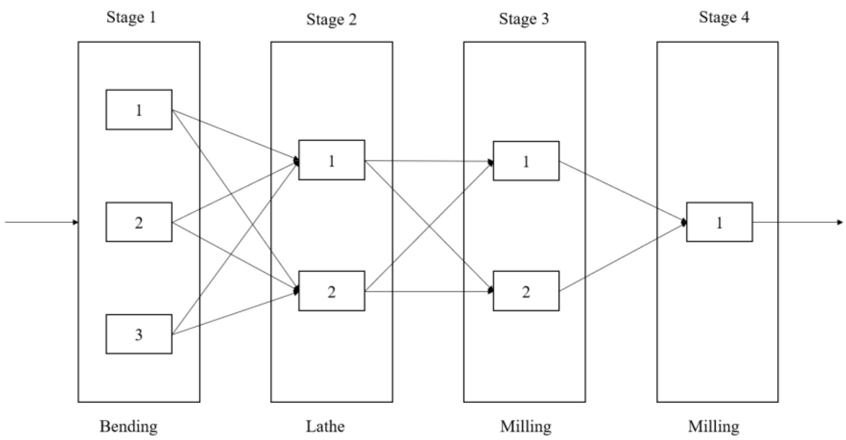

Figure 6. Machine shop case study layout

For the case study problem, five similar algorithms were used to form the optimization results. The analysis was repeated with 30 repetitions to ensure the best results were achieved. Based on the computational experiment, all 30 repetitions of the algorithm solution were combined, and all duplicate solutions were removed. The final filtered solutions were computationally experimented with again to obtain the non-dominated solutions (NDS). Each algorithm's five non-dominated solutions are combined and explored to achieve the final Pareto Front (PF). These algorithms' performance was then compared together using the same metrics.

As shown previously in the benchmark problems, the overall performance of all algorithms was judged based on their non-dominated (NDS) results that are close to the Pareto front, also known as the cardinality factor, the convergence factor and the diversity factor of their results in the objectives space as well as the computational time to solve the problems. All comparison algorithms have similar settings; the population size is 300, the maximum iterations are 300, and the maximum archive size is 100 for each algorithm. Finally, all the NDS solutions from all algorithms are combined again to obtain the actual Pareto front (PF).

Table 11 presents the optimization result for the case study problem. Based on Table 11, the NSGA-III performance shows the best overall result, followed by the MOEA/D, SPEA-II, PESA-II, and MOPSO. The metrics used for comparison are Pareto Percentage (%P), Error ratio (ER), Generational Distance (GD), Inverted General Distance (IGD), Spacing (Sp), Maximum Spread (MS), Hyper Volume (HV) and Computational time from 30 repetitions (s). It shows that SPEA-II achieves the best in the Pareto % and ER. For GD and IGD, MOEA/D achieved rank 1 in both metrics. For spacing metrics also, MOEA/D achieved the best rank position, while for MS and HV, the best rank was obtained by MOPSO and NSGA-III. MOPSO also obtained the first rank in computational time.

Table 11. Algorithms comparison

Metrics	Algorithms				
	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
Pareto %	0.125	0.400*	0.125	0.000	0.333
Error Ratio (ER)	0.875	0.600*	0.875	1.000	0.667
Generational Distance (GD)	8.6003	12.5601	9.8689	36.3091	2.5267*
Inverted Generational Distance (IGD)	15.7003	5.4801	8.3703	26.4533	3.2512*
Maximum Spread (MS)	258.5002	227.2001	267.4001	345.0006*	125.0011
Spacing (Sp)	10.2515	11.9293	14.5935	33.3849	7.7997*
Hyper Volume (HV)	1715051.4	2235772.8	3415475*	1954066	2154022
Computational time (s)	1030.66	1249.43	1430.24	875.67*	1760.39

The metrics indicator for all five algorithms was averaged according to the rank system, with rank one being the best and rank five being the worst. The rank will be shared if a similar value is obtained from the algorithms. Based on Table 12 below, the average rank for all five algorithms is presented. SPEA-II obtained the best average of 2.38, while NSGA-III obtained a rank of 2.

Table 12. Metric ranks

Algorithms	%PF	ER	GD	IGD	SP	MS	HV	AVERAGE
PESA-II	4	3	2	3	2	3	5	3.14
SPEA-II	1	1	4	1	3	4	2	2.38
NSGA-III	3	3	3	2	4	2	1	2.57
MOPSO	5	5	5	4	5	1	4	4.14
MOEA/D	2	2	1	5	1	5	3	2.71

For overall comparison, the best way to interpret the quality of the solution is through ranking. According to Table 12 below, the best algorithm based on the rankings is tied between NSGA-III and SPEA-II. Both algorithms were equally ranked using cardinality, convergence, and distribution factors. SPEA-II algorithm achieved the best rank in the cardinality component, while in convergence and distribution, the factor was topped by NSGA-III rank. Despite the average ranking result shown in Tables 11 and 12, no specific algorithm outperformed one or the other on these seven metrics and the three-element solutions. Each optimal algorithm can find a near-optimal solution and can produce the optimum results.

Table 13. Overall ranks

	PESA-II	SPEA-II	NSGA-III	MOPSO	MOEA/D
Cardinality	4	1	3	5	2
Convergence	2	2	1	5	4
Distribution	4	2	1	4	2
Overall rank	4	1	1	5	3

5.1 Discussion of the case study

Through visualization in Figure 7, it has revealed that the obtained Pareto front (dotted line) for all five algorithms presented almost similar distribution along the optimal Pareto front. The converged solution for each algorithm also presents an almost similar convergence point arrangement from the starting point (minimum point of Cmax) until the final point (maximum point of Cmax) against the optimal Pareto front. However, it can be visualized that no multi-objective algorithm outperformed the other algorithms.

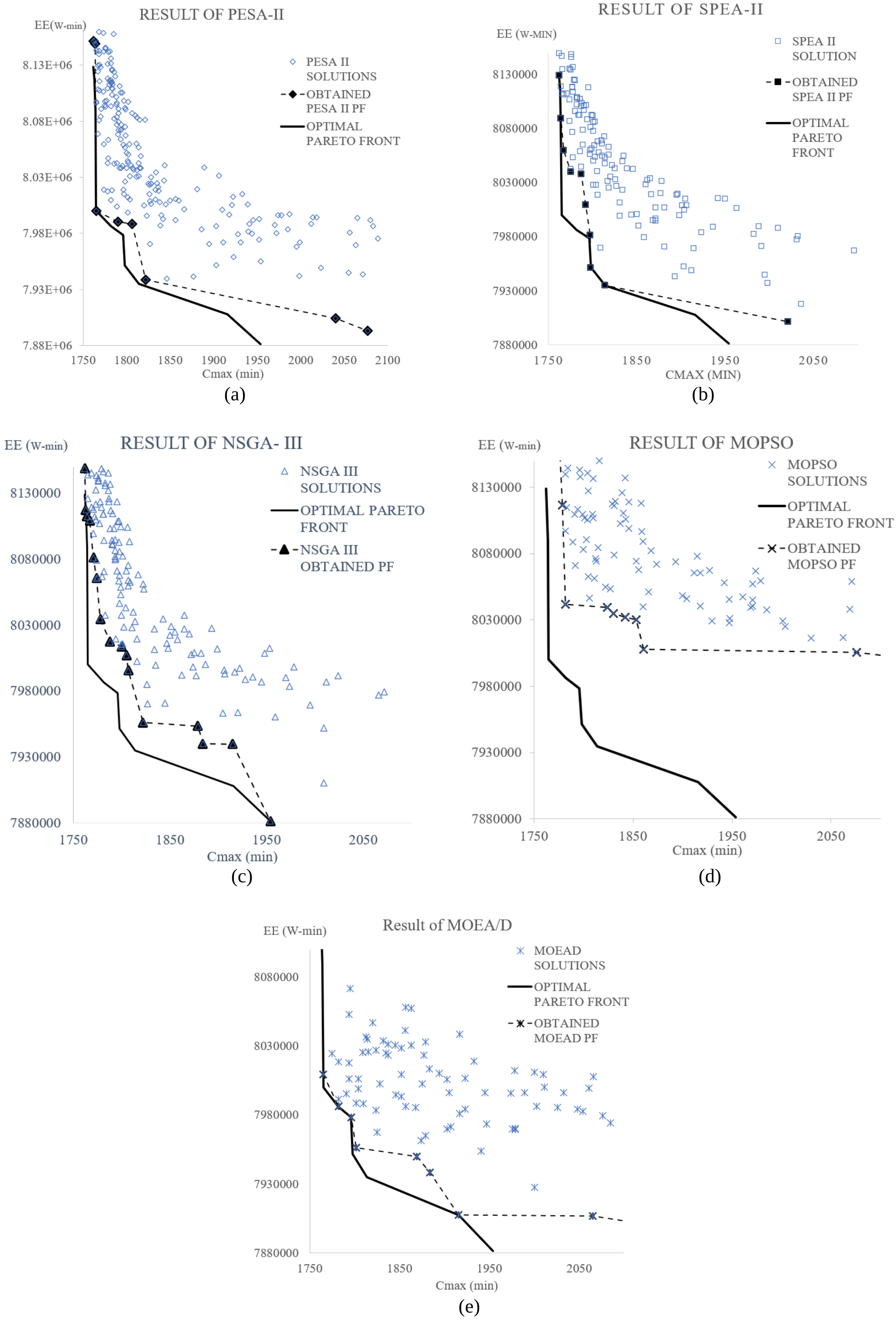

Figure 7. Convergence and distribution points of the obtained Pareto front with the optimal Pareto front

Finally, to summarize the case study, a box plot presenting the minimum value, maximum value range and mean for the makespan and energy solution are shown in Figure 8 below. The minimum value of Cmax obtained was 1762 minutes for all three algorithms (PESA-II, SPEA-II and NSGA-III). The minimum value of EE was 79,545.6 kW-min obtained by MOEA/D. The result can vary, and it depends on the interest of the manager's choice. That is if the requirements of the solution are interested in minimizing the makespan first and then the EE efficient value after. The other solutions from these algorithms on the Pareto optimal solution can also be chosen based on the manager's objective interest. The results may compromise the Cmax and EE energy values if one considers them over the other. Solutions can also significantly improve if all algorithms increase the archive storing of the Pareto value and maximize the iterations and the population solution.

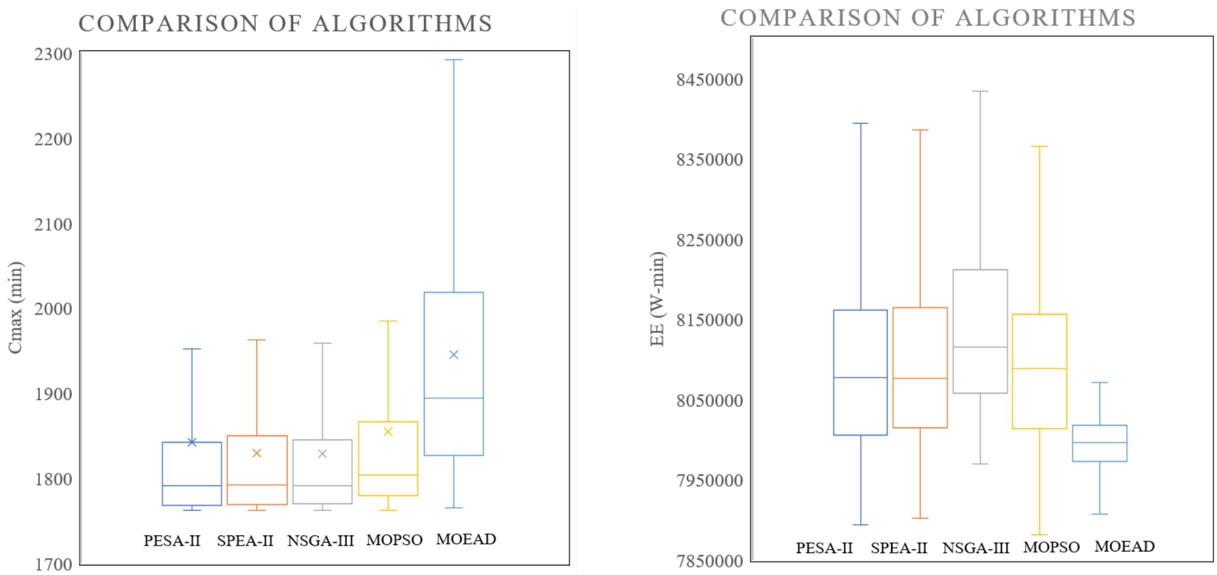

Figure 8. Values of Cmax and EE for all 5 algorithms

6.0 CONCLUSION

This research concludes by discussing the scheduling of hybrid flow shops while taking energy and makespan into account. This study's main contribution is based on NSGA-III's aptitude for handling hybrid flow shops with multi-objective issues. The hybrid flow shop is an assembly line primarily found in a variety of regional industries, including textiles, electronics, machining, and the production of auto parts. Twelve hybrid flow shop model problems were used as benchmarks for this works on a computer experiment using a four-stage hybrid flow shop case study model. The fundamental goal of computational algorithm experimentation is to compare the performance of a given model's algorithm to that of other well-known models. According to the findings, the NSGA-III outperformed other algorithms in terms of both the convergence indicator and the distribution indicator. Other algorithms, in particular SPEA-II, have a higher cardinality status than NSGA-III. Because NSGA-III uses an elitist approach, it can show a variety of solutions and converge on a single one. On cardinality, the NSGA-III can be modified. By reducing the inaccuracy on the Pareto solutions, it may be made better. Large numbers of Pareto Front values can be obtained with NSGA-III. However, the chosen solutions have the potential to raise the error ratio and lower Pareto percentage values.

The case study results for this work also showed how well the algorithm handled real-world issues and suggested ideas that could help the company enhance the production line. The NSGA-III has demonstrated its ability to provide solutions with greater convergence and distribution. Future studies should use the NSGA-III with SPEA-II cardinality performance for the HFS topic. Researchers are also urged to think about developing a superior multi-objective algorithm that can outperform the solution standard's cardinality, convergence, and distribution. Additionally, maximizing other objective functions, such as lateness, machine breakdown, idle machines, and makespan and energy difficulties in the sector, can be incorporated as part of the primary work.

7.0 ACKNOWLEDGEMENT

The authors would like to be obliged to Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA) for providing laboratory facilities and financial assistance under grant no. RDU223017.

8.0 REFERENCES

[1] V. Fernandez-Viagas, P. Perez-Gonzalez, and J. M. Framinan, "Efficiency of the solution representations for the hybrid flow shop scheduling problem with makespan objective," *Computers Operations Research*, vol. 109, pp. 77-88, 2019.

[2] S. Schulz, J. S. Neufeld, and U. Buscher, "A multi-objective iterated local search algorithm for comprehensive energy-aware hybrid flow shop scheduling," *Journal of Cleaner Production*, vol. 224, pp. 421-434, 2019.

[3] Ö. Tosun, M. K. Marichelvam, and N. Tosun, "A literature review on hybrid flow shop scheduling," *International Journal of Advanced Operations Management,* vol. 12, no. 2, pp. 156-194, 2020.

[4] J.-q. Li, H.-y. Sang, Y.-y. Han, C.-g. Wang, and K.-z. Gao, "Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions," *Journal of Cleaner Production,* vol. 181, pp. 584-598, 2018.

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

[6] S. Schulz, "A genetic algorithm to solve the hybrid flow shop scheduling problem with subcontracting options and energy cost consideration," in *Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology–ISAT 2018: Part III,* 2019: Springer, pp. 263-273.

[7] S. Schulz, M. Schönheit, and J. S. Neufeld, "Multi-objective carbon-efficient scheduling in distributed permutation flow shops under consideration of transportation efforts," *Journal of Cleaner Production,* vol. 365, p. 132551, 2022.

[8] Z. Liu, J. Yan, Q. Cheng, C. Yang, S. Sun, and D. Xue, "The mixed production mode considering continuous and intermittent processing for an energy-efficient hybrid flow shop scheduling," *Journal of Cleaner Production,* vol. 246, p. 119071, 2020.

[9] G. Gong *et al.*, "Energy-efficient flexible flow shop scheduling with worker flexibility," *Expert Systems with Applications,* vol. 141, p. 112902, 2020.

[10] A. N. Mohd Rose and 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*: Springer, 2022, pp. 77-86.

[11] M. F. F. Ab Rashid and M. A. N. Mu'tasim, "A novel Tiki-Taka algorithm to optimize hybrid flow shop scheduling with energy consumption," *Engineering Applied Science Research,* vol. 49, no. 2, pp. 189-200, 2022.

[12] M. F. F. A. Rashid and M. A. H. Osman, "Optimisation of energy efficient hybrid flowshop scheduling problem using firefly algorithm," in *2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2020: IEEE, pp. 36-41.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, pp. 182-197, 2002.

[14] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation,* vol. 18, no. 4, pp. 577-601, 2013.

[15] J. Carlier and E. Neron, "An exact method for solving the multi-processor flow-shop," *J RAIRO-Operations Research-Recherche Opérationnelle*, vol. 34, no. 1, pp. 1-25, 2000.

[16] K. Deb, "Multi-objective optimisation using evolutionary algorithms: an introduction," in *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*: Springer, 2011, pp. 3-34.

[17] L. C. Bezerra, M. López-Ibánez, and T. Stützle, "An empirical assessment of the properties of inverted generational distance on multi-and many-objective optimization," in *International Conference on Evolutionary Multi-criterion Optimization*, 2017: Springer, pp. 31-45.

[18] A. Yoosefelahi, M. Aminnayeri, H. Mosadegh, and H. D. Ardakani, "Type II robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model," *Journal of Manufacturing Systems,* vol. 31, no. 2, pp. 139-151, 2012.

[19] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *J TIK-Report, ETH Zurich,* vol. 103, 2001.

[20] C. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, 2002, vol. 2: IEEE, pp. 1051-1056.

[21] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 2001, pp. 283-290.

[22] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation,* vol. 11, no. 6, pp. 712-731, 2007.

[23] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a Pareto front," in *Late Breaking Papers at the Genetic Programming 1998 Conference*, 1998: Citeseer, pp. 221-228.

[24] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *International Conference on Evolutionary Multi-criterion Optimization*, 2003: Springer, pp. 376-390.