JOURNAL OF MECHANICAL ENGINEERING AND SCIENCES
ISSN: 2289-4659 e-ISSN: 2231-8380
VOLUME 19, ISSUE 3, 2025, 10756 - 10769
DOI: https://doi.org/10.15282/jmes.19.3.2025.5.0843

RESEARCH ARTICLE

Optimization of hybrid flow shop scheduling in a machine shop: Achieving energy efficiency and minimizing machine idleness with multi-objective Tiki Taka optimization

S. N. H. H. M. Hata, M. A. N. Mu'tasim*, M. F. F. Rashid

Faculty of Mechanical and Automotive Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), 26600, Pekan, Pahang, Malaysia
Phone: +6094315406, Fax: +6094246222

ABSTRACT - Hybrid Flow Shop Scheduling (HFS) has garnered significant interest in terms of problem formulation and solution approaches. This work introduces an optimization approach for a case study on a hybrid flow shop scheduling problem. The objective is to minimize the makespan, energy consumption, and idle machines in the manufacturing shop. The HFS comprises multiple concurrent production lines, each composed of several machines that operate in one or more stages. A case study was conducted using fourteen jobs across three stages, which involved the use of lathes, milling machines, and deburring machines. The EE-HFS was optimized using Multi-Objective Tiki Taka Optimization (MOTTA). The study considered machine idle time as a key factor influencing energy efficiency, incorporating it into the scheduling evaluation. The optimization result was compared to established algorithms, such as the Non-dominated Sorting Genetic Algorithm-II, the Multi-Objective Evolutionary Algorithm Based on Decomposition, the Multi-Objective Particle Swarm Optimization, and the recent algorithm, the Multi-Objective Grey Wolf Optimizer. The metrics used for comparison include Error Ratio (ER), Pareto Percentage (%), Spacing, Maximum Spread, computational speed, Hyper Volume, Inverted Generational Distance (IGD), and Generational Distance (GD). The results indicate that MOTTA exhibits superior performance, with 78.42% as the best overall result, and 100% improvement in terms of convergence and domination compared to the case study solutions (ER, ND, GD, and IGD). Overall, the findings have important implications for Hybrid flow shop scheduling in terms of the energy utilization model, reducing idle machine time, and the promising potential of MOTTA for application in other combinatorial scheduling challenges. This case study provides substantial benefits to the organization by effectively reducing its daily energy consumption and equipment usage, while also enhancing resource management.

ARTICLE HISTORY
Received : 15th Feb. 2025
Revised : 24th July 2025
Accepted : 01st Aug. 2025
Published : 30th Sept. 2025

KEYWORDS
Hybrid flow shop
Multi-objective Tiki Taka
Optimization
Energy efficient
Idle machines

1. INTRODUCTION

Production scheduling is a critical decision-making process in both manufacturing and production systems. The term 'Hybrid Flowshop Scheduling (HFS)' refers to the problem of combining parallel machines with Flowshop scheduling. The research on scheduling challenges in a single factory has been considerable over the past few decades, employing precise methods, heuristics, and metaheuristic approaches. The fundamental focus of current scheduling research has been on multiple-machine scheduling. This paper extended the work by Ab Rashid et al. [1] on hybrid flow shop scheduling. The Hybrid Flowshop Scheduling Problem (HFSP) poses challenges for various production sectors, including electronics, textiles, and semiconductors [2]. In manufacturing systems, HFS differ from single-machine (1 stage, one machine) and Parallel Machines (1 stage, many machines), which all perform the same job. However, in the real world, industrial procedures often involve more than one type of operation, such as cutting, assembling, painting, and packaging, which cannot all be done simultaneously. Therefore, HFS requires processing problems with n tasks in *k* stages, where k is at least 2. Each stage contains at least one machine, but at least one stage contains a parallel machine. The machine processes only a single task at a time [3]. Initially, researchers only considered the HFS for machines on each level that were identical in design and function. This is the most fundamental version of the HFS problem that researchers have studied since the 1970s. As a result, researchers examine the HFS, considering machines that may not be identical at this stage. The model, capacity, pace, and power rating of the machine may vary.

A previous study aimed to determine the maximum lateness, machine usage, penalty, and tardiness. Recently, researchers introduced a new objective that pertains to environmental factors such as minimizing energy costs, total carbon emissions, and noise pollution [4, 5]. Moreover, the next step in energy consideration involves examining idle machines during the process. However, in real-world case studies, multi-objective cases are more practical and require more effort. In recent years, multi-objective problems have been the subject of the most extensive research. Recent

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

publications have focused on energy-efficient hybrid Flowshop scheduling (EE-HFS) with multiple objectives. Li et al. [6] developed a model for EE-HFS that considers machines in both standby and processing states, taking into account setup times (gaps) between tasks. However, a separate study identified EE issues associated with energy cost resolution during the power rate period [7]. Furthermore, studies have been conducted on minimizing EE-HFS during non-processing hours [8].

The weighted sum method and the Pareto-based method are two approaches that can be employed to manage a multi-objective optimization problem. Numerous algorithms in EE-HFS utilize the weighted sum method. The weighted sum approach applies an algorithm based on the MOGA to the EE-HFS problem [9]. Since then, new algorithms like the Multi-Objective Ant Lion Optimizer (MOALO) [10], Multi-Objective Keshtel Algorithm (MOKA), and Multi-Objective Keshtel and Social Engineering Optimizer (MOKSEA) have successfully solved numerous HFS with energy utilization, outperforming well-known algorithms like Non-dominated Sorting Genetic Algorithm II (NSGA II) [11] and Multi-Objective Particle Swarm Optimization (MOPSO) [12] in solving dynamic EE-HFS. Recent algorithms for solving EE-HFS include the Multi-Objective Artificial Bee Colony algorithm [13], the Multi-Objective Grey Wolf Optimization [14], and the combination of two algorithms to solve the problem [15], among others. These new algorithms were evaluated in comparison to well-established algorithms, such as the Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [16] and the Strength Pareto Evolutionary Algorithm II (SPEA II) [17]. All comparisons revealed the remarkable capabilities of each algorithm across various criteria, including optimal fitness, computational time, and objective values. In the case of the Pareto-based algorithm, it is believed that solving EE-HFS with the Pareto method has received increased attention over the past years. Recently, Mutasim and Rashid employed several comparisons, ranging from the most popular [18] to the latest algorithms, to compare Pareto-based algorithms using performance indicators [19]. Using a case study from a manufacturing facility, this paper introduces a new algorithm, Multi-Objective Tiki Taka Optimization (MOTTA) [20], to evaluate the C_{max} in the EE-HFS problem [1]. This work focuses on finding MOTTA's ability to solve problems using the a posteriori approach. At the same time, additional objectives include minimizing idle machine time. Most previous work used the weighted sum method, which has difficulties capturing diverse trade-offs in multi-objective issues, especially when the objectives clash or there are more than three goals. Additionally, many algorithms overlooked critical real-world factors, such as the amount of time machines spend idle, which resulted in less accurate estimates of energy efficiency. The MOTTA algorithm addresses these issues by employing a Pareto-based posteriori approach, which generates a broader range of balanced solutions and incorporates idle time as a goal, thereby better mimicking real production settings. This makes solutions in complex EE-HFS situations more effective and valuable. The need to manage multi-objective optimization problems with many objectives (typically more than three) prompted the development of new algorithms such as MOTTA, which will help enhance solution finding in the EE-HFS problem. It was designed to improve the efficacy and efficiency of MOTTA in multi-objective optimization scenarios.

This paper introduces a multi-objective Pareto selection, whereas the latter paper also introduces a weighted multi-objective approach. Additionally, this paper provides better choices for stakeholders compared to future research. One of the most challenging aspects of the HFS is allocating jobs to machines at different stages. Historically, we have assessed the effectiveness of a job assignment based on the duration required to complete it. Global warming and carbon emissions have increased the importance of sustainability worldwide. When strategizing manufacturing operations, it is crucial to consider sustainability. As for this paper, one of its primary aims is to reduce the overall completion time or duration of an unrelated HFS problem. Consequently, this study also investigates the utilization of energy in HFS machines.

2. MATERIAL AND METHODS

2.1 Case Study

A hybrid flow shop scheduling problem is a cross between a parallel machine scheduling problem and a flow shop scheduling problem. HFS requires processing a set of jobs in K phases. Any machine in stage one can initially process the task, and if there are no other jobs on that machine, any machine in stage two can also process it. As indicated, the job must be listed before the prior position and undergo the process at each location. The number of machines at each stage can vary. Additionally, the machines on stage are not identical, meaning their processing times and power ratings are not comparable. With this project, EE-HFS aims to reduce both the completion time (makespan) and energy consumption. Due to the machines' varying power ratings, the total energy consumption will differ depending on the work assignment and scheduling. A machine shop in Batu Pahat, Malaysia, conducted the case study [19]. The product undergoes manufacturing using a variety of machines. Three lathe machines were used for grooving and drilling, while three milling machines were used for fabricating the contour. Two deburring machines were used to remove rough edges and finish the product. Each machine within a given category is unique. Additionally, the energy consumption of each machine varies. Fourteen jobs were allocated. These machines, in three stages, must process the tasks in a unidirectional manner, beginning with the lathe, milling, and deburring machines.

The manufacturer must schedule 14 tasks on the available machines, as shown in Figure 1. The results of the case study's time and power calculations were tabulated. This study proposes to determine the optimal scheduling management for the following cycle arrangement. Additionally, scheduling can reduce energy consumption and minimize idle time on machines by utilizing their schedules effectively. Table 1 provides detailed information about the processing time of each job on each machine.

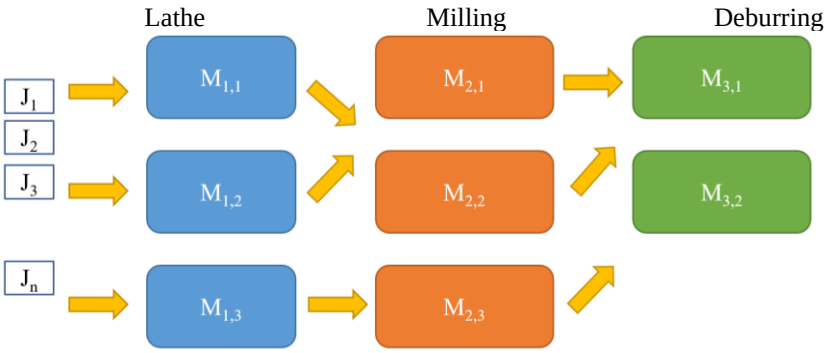

Figure 1. Layout of the case study company

2.2 Problem Formulation

This section describes how to represent the case study and calculate the EE-HFS objectives using a model from Rashid and Mutasim [1]. Additionally, the objective space includes minimizing idle time on machines. First, input the problem definition in Table 1, followed by 14 jobs with three stages and machine configurations [3 3 2]. To calculate the makespan, the task time in Table 1 is replaced with a random number, also known as a continuous value, as indicated in Eq. (1). The EE-HFS is expressed continuously. Due to the stochastic nature of the metaheuristics, the number is constant and chosen at random between the lower and upper bounds, which are between 0 and 1, as shown in Table 2.

Table 1. Job processing time on each machine (minutes)

	Stage 1 (Lathe)			Stage 2 (Milling)			Stage 3 (Deburring)	
Job	$M_{1,1}$	$M_{1,2}$	$M_{1,3}$	$M_{2,1}$	$M_{2,2}$	$M_{2,3}$	$M_{3,1}$	$M_{3,2}$
1	480	300	240	360	300	180	240	150
2	400	250	200	300	250	150	200	125
3	160	100	80	120	100	60	80	50
4	960	600	480	720	600	360	480	300
5	384	240	192	288	240	144	192	120
6	192	120	96	144	120	72	96	60
7	320	200	160	240	200	120	160	100
8	800	500	400	600	500	300	400	250
9	640	400	320	480	400	240	320	200
10	576	360	288	432	360	216	288	180
11	1280	800	640	960	800	480	640	400
12	768	480	384	576	480	288	384	240
13	960	600	480	720	600	360	480	300
14	400	250	200	300	250	150	200	125

Table 2. Continuous problem representation

	Stage 1 (Lathe)			Stage 2 (Milling)			Stage 3 (Deburring)	
Job	$M_{1,1}$	$M_{1,2}$	$M_{1,3}$	$M_{2,1}$	$M_{2,2}$	$M_{2,3}$	$M_{3,1}$	$M_{3,2}$
1	0.746	0.117*	0.509	0.169	0.831	0.928	0.169	0.884
2	0.388	0.383	0.271*	0.868	0.742	0.448	0.710	0.944
3	0.174	0.245	0.641	0.809	0.853	0.398	0.115	0.080*
4	0.360	0.829	0.215	0.791	0.655	0.026*	0.786	0.923
5	0.492	0.834	0.131*	0.760	0.926	0.833	0.259	0.213
6	0.522	0.397*	0.479	0.994	0.604	0.945	0.490	0.438
7	0.773	0.744	0.443	0.053	0.088	0.798	0.656	0.032*
8	0.557	0.720	0.110*	0.217	0.811	0.139	0.882	0.924
9	0.013*	0.377	0.168	0.540	0.102	0.039	0.933	0.972
10	0.361	0.644	0.068	0.208	0.040*	0.469	0.150	0.991
11	0.427	0.955	0.724	0.581	0.540	0.705	0.005*	0.783
12	0.927	0.008*	0.825	0.767	0.997	0.228	0.920	0.642
13	0.105	0.268	0.764	0.806	0.104*	0.470	0.219	0.923
14	0.320	0.858	0.260	0.878	0.188	0.759	0.032*	0.642

Since permutations cannot be used in metaheuristics, random numbers are selected and computed according to the magnitude of the job-machine assignment problem. These digits are decoded using the heuristic shown in Eqs. (2) and (3) using the heuristic rule (ascending order). Several dispatching rules are available, including FCFS, LPT, SPT, LILO, and descending order, among others. The job-to-machine assignment is selected based on the smallest value from each stage for each job. The smallest number in each row and stage is chosen, as it demonstrates which machine is best suited for that job. The system needs to examine other numbers to find better possibilities later, but only the lowest one indicates which machine to use. For example, for job 1, in stage 1 (lathe), the smallest value from these values [0.746, 0.117, 0.509] is 0.117. This means for job 1, $M_{1,2}$ is selected in stage 1. All bold values are chosen for job-to-machine assignment. Similarly, the job sequence is also represented using a heuristic rule in Eqs. (3) and (4). The smallest value in the jobs row is selected, as indicated by the symbol (*). In this example, the chosen job sequence is as follows: job 11, job 12, job 9, job 4, job 14, job 7, job 10, job 3, job 13, job 8, job 1, job 5, job 2, and job 6.

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

$$X \begin{bmatrix} x_{1_{min}} \\ x_{2_{min}} \\ \cdots \\ x_{n_{min}}[] \end{bmatrix}_{i_{min}} \tag{2}$$

$$X'_j = ascending\{X_{i_{min}}\{\}\} \tag{3}$$

$$J = \{j\} \forall i = 1,2,\ldots,n; \tag{4}$$

In Eqs. (5) to (8), there are two primary considerations: minimizing the completion time C_{max} and minimizing the total energy utilization EE_{total}. The EE calculates the total time of machine usage during the job process, and Tit calculates the time when the machine is idle. During idle machine, 10% power is consumed when the machine is idle. Due to the constraints in Eq. (8) and the nature of the task, only one machine can be used per stage.

$$C\,max\{C_i\}_{i=1,2,3,\ldots,n_{max}} \tag{5}$$

$$EE = \sum_{i=1}^{n} \sum_{s=1}^{S} \sum_{ks=1}^{Ks} (t_{iks} \,.\, E_{iks} . y_{iks}) \tag{6}$$

$$TIt = \sum_{s=1}^{S} \sum_{ks=1}^{Ks} \sum_{(J_i, J_j) \in J_{k,s}} max(0, ST_{jk} - FT_{ik}) \tag{7}$$

$$EE_{total} = \sum_{i=1}^{n} \sum_{s=1}^{S} \sum_{ks=1}^{Ks} (t_{iks} \,.\, E_{iks} . y_{iks}) + 0.1TIt \tag{8}$$

Subjected to:

y_{iks} {1 If job i, process at machine k_s
0 Else 0

$$\sum_{ks=1}^{Ks} y_{iks} = 1, \quad \forall i = 1,2,\ldots,n; \quad s = 1,2,\ldots,S \tag{9}$$

The calculation for the makespan for the selected job sequence and job-machine assignment is as shown in Table 3. The makespan value for this scheduling is 5328 minutes. The next step is to calculate the total amount of time the machine is idle. Using Eq. (3), the calculation of idle processing time involves monitoring the duration of idle time for each individual stage and machine. The first step is to determine the total number of machines being utilized in the current stage. Next, the jobs assigned to the current machine and stage can be identified by sorting them based on their start time. For example, initially, there are a total of three machines. The jobs processed in Machine 1 of Stage 1 are [11, 9, 3, 13]. Subsequently, the idle time is determined by subtracting the finish time of the previous job from the start time of the next job. In this scenario, the calculation of idle time results in zero. The technique is repeated with the subsequent machine at an analogous level. After the stage is over, the process is computed for the following step. The total idle time is the aggregate of all periods of inactivity during the scheduling process. In this situation, Table 4 excludes any idle time that

was handled in this sample. The ultimate total idle time is obtained by adding up all the idle times, resulting in a total of 4262 minutes. To calculate energy consumption, the operating duration of each machine must be determined. The duration of each machine's operation is then multiplied by its power rating and converted to kWh units, as shown in Table 5. According to the calculation, the EE consumption for Stage 1 is 61.05 kWh, for Stage 2 it is 85.88 kWh, and for Stage 3 it is 48.68 kWh. Finally, from Figure 2, the total calculation for EE_{total} is the combination of the energy consumed during processing time (195.61 kWh) and the total energy consumed during idle time, which is calculated using Table 4 and multiplied by the power rating given for each machine. The total idle energy consumed during idle time (0.1 of the total energy consumed) is 7.348 kWh. The EE_{total} is measured as 202.95kW-h.

Table 3. Calculation of makespan

	Stage 1			Stage 2			Stage 3		
Job	Machine	Start Time	Finish Time	Machine	Start Time	Finish Time	Machine	Start Time	Finish Time
11	$M_{1,1}$	0	1280	$M_{2,2}$	1280	2080	$M_{3,1}$	2080	2720
12	$M_{1,2}$	0	480	$M_{2,3}$	480	768	$M_{3,2}$	768	1008
9	$M_{1,1}$	1280	1920	$M_{2,3}$	1920	2160	$M_{3,1}$	2720	3040
4	$M_{1,3}$	0	480	$M_{2,3}$	2160	2520	$M_{3,1}$	3040	3520
14	$M_{1,3}$	480	680	$M2_{,2}$	2080	2330	$M_{3,1}$	3520	3720
7	$M_{1,3}$	680	840	$M_{2,1}$	840	1080	$M_{3,2}$	1080	1180
10	$M_{1,3}$	840	1128	$M_{2,2}$	2330	2690	$M_{3,1}$	3720	4008
3	$M_{1,1}$	1920	2080	$M_{2,3}$	2520	2580	$M_{3,2}$	2580	2630
13	$M_{1,1}$	2080	3040	$M_{2,2}$	3040	3640	$M_{3,1}$	4008	4488
8	$M_{1,3}$	1128	1528	$M_{2,3}$	2580	2880	$M_{3,1}$	4488	4888
1	$M_{1,2}$	480	780	$M_{2,1}$	1080	1440	$M_{3,1}$	4888	5128
5	$M_{1,3}$	1528	1720	$M_{2,1}$	1720	2008	$M_{3,2}$	2630	2750
2	$M_{1,3}$	1720	1920	$M_{2,3}$	2880	3030	$M_{3,1}$	5128	5328
6	$M_{1,2}$	780	900	$M_{2,2}$	3640	3760	$M_{3,2}$	3760	3820

Table 4. Total idle machine time

Stage	Machine	Idle End time	Idle Start time	Idle time
2	1	1440	1720	280
2	2	2690	3040	350
2	3	768	1920	1152
3	2	1008	1080	72
3	2	1180	2580	1400
3	2	2750	3760	1010

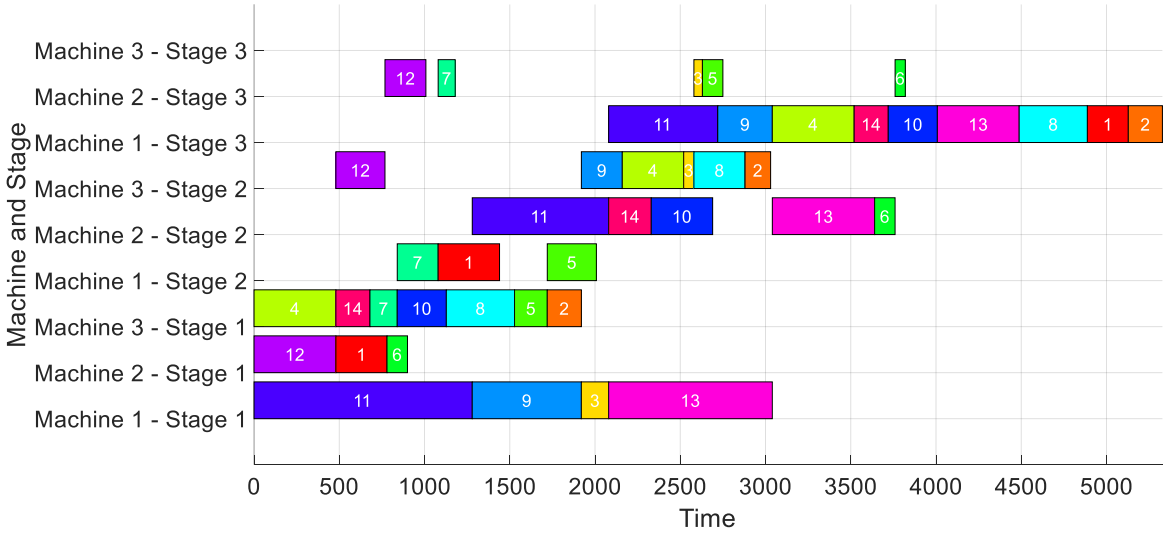

Figure 2. Schedule for EE-HFS example

Table 5. Total energy utilization for machines

Stage	Machine	Cumulative period (min)	Power rating (Watt)	kW-hr
Lathe	M1_1	3040	600	30.40
	M1_2	900	550	8.25
	M1_3	1920	700	22.40
Milling	M2_1	888	1200	17.76
	M2_2	2130	1000	35.50
	M2_3	1398	1400	32.62
Deburring	M3_1	3248	750	40.60
	M3_2	570	850	8.08

2.3 Multi-Objective Tiki Taka Algorithm

The Tiki-Taka algorithm [17] is introduced as a cost-centric strategy for hybrid flow shop scheduling, engineered to discover solutions that not only minimize makespan and enhance productivity but also minimize energy consumption. This positions it as a valuable instrument for advancing the sustainability and efficiency of manufacturing systems. The Tiki-Taka algorithm draws inspiration from a football playing style pioneered by Barcelona Football Club, emphasizing specific techniques such as short passing, strategic player positioning, movement, and possession control. In this tactical approach, a football team strategically forms triangles around opponents, engaging in short passing within the triangular formation to methodically build up their attack from a defensive position. The primary goal of the triangular formation is to attract multiple opponent players, prompting the creation of a new triangle elsewhere. Next, players try to find the key players who can read the game well and move quickly. Key players can capitalize on opportunities to score goals and position themselves strategically.

The Tiki-Taka algorithm is introduced as a cost-centric strategy for hybrid flow shop scheduling, engineered to discover solutions that not only minimize makespan and enhance productivity but also minimize energy consumption [20]. This positions it as a valuable instrument for advancing the sustainability and efficiency of manufacturing systems. The Tiki-Taka algorithm draws inspiration from a football playing style pioneered by Barcelona Football Club, emphasizing specific techniques such as short passing, strategic player positioning, movement, and possession control. In this tactical approach, a football team strategically forms triangles around opponents, engaging in short passing within the triangular formation to methodically build up their attack from a defensive position. The primary goal of the triangular formation is to attract multiple opponent players, prompting the creation of a new triangle elsewhere. Players then focus on identifying key players, those with astute game-reading abilities and swift movements that can capitalize on opportunities to score goals and maintain precise positions. The Tiki-Taka Algorithm employs a football-inspired framework where player positions represent candidate solutions, while the ball position serves as a solution vector that determines subsequent player locations. Additionally, key players function as solution leaders who influence the ball's position throughout the optimization process. The algorithm incorporates two fundamental features derived from tiki-taka football tactics: short passing and player positioning. The short passing mechanism enables information gathering from neighbouring players to determine new player positions, which are further refined by incorporating information from key players and the current ball position. The initialization process begins by defining essential parameters, including the number of players, problem dimensions, key players, and relevant coefficients. For a Hybrid Flow Shop problem involving n jobs and m machines, the solution space is represented in n × m dimensions. A population of w players, each characterized by $n \times m$ dimensions, is randomly generated to form the initial candidate solution set, denoted as P_w, with population size w. Simultaneously, a matrix representing the ball position, B, is created. During the initial stage, the ball position for each player corresponds directly to their starting position, such that $Bw = Pw$.

The evaluation process assesses each player's position in relation to the optimization objective being pursued. However, before the fitness function can be applied, the player's position must first undergo a decoding transformation to convert it into a valid HFS solution. This transformation begins by dividing the player position, denoted as pw, into segments corresponding to the number of jobs in the problem. To illustrate this decoding process, consider an HFS problem comprising 14 jobs distributed across three stages, where each stage contains three, three, and two machines, respectively. This configuration produces a solution dimension of 112. The player position can be expressed as $P_w = [P_{w,1}, P_{w,2}, P_{w,3}, \ldots, P_{w,112}]$. During decoding, this position vector is partitioned into n job-based segments, resulting in matrices labeled as X_i. Individual elements within these matrices are identified using the notation X_i, s, k. It should be noted that both X_i and X_i, s, k notations serve exclusively for the decoding and evaluation phases of the algorithm. Within the TTA framework, key players represent a subset of the highest-performing solutions from the overall population. The number of key players, designated as n_k, equals ten percent of the total population size, with a minimum threshold of three players. These key players are identified and refreshed based on the superior fitness values emerging from each evaluation cycle. In the context of HFS problems, key players correspond to those solution representations that achieve the best makespan (C_{max}) values in the generated schedules. For instance, when working with a population of 300 players, the algorithm selects 30 key players (representing the top 10%) based on their fitness performance. This key player mechanism mirrors

the tiki-taka football strategy, which relies on multiple skilled players to create varied passing patterns and ball movement throughout the team. When translated to optimization, this concept enables the algorithm to preserve solution diversity and avoid premature convergence. The key player archive, denoted as h, undergoes continuous updates with each iteration to ensure it consistently reflects the best-performing solutions discovered during the search process. In terms of optimization, this idea allows the algorithm to maintain diversity and prevent premature convergence. The key player archive, marked by h, is continuously updated after each generation, ensuring it constantly represents the best solutions found at any point in the search.

The Tiki-Taka Algorithm has a scheme for updating balls that resembles the style of tiki-taka passing in soccer. This mechanism transfers the ball from one player to another in an adjacent position in the solution space. This algorithm takes into consideration the possibility of losing ball possession during a pass (which is common in actual gameplay). The loss probability (problose) is usually set to 10%, providing a realistic estimate of uncertainty in the search. The ball position update b_i', is, according to a random number rp within the interval [0, 1], moved along one of two possible paths. When an attempted pass succeeds (r_p > problose), the ball is passed to a random neighbouring player of the target, rendering the solution more diversified. Reciprocally, when a pass fails ($r_p \leq$ problose), then the ball is blocked. Movements of the ball are modified by a coefficient c_1, which modulates how much it reflects from its destination. Following each ball update, player positions are recalculated based on both the current ball location and the positions of designated key players. Since the algorithm employs multiple solution leaders, one key player is selected randomly during each iteration to influence the movement. The position update mechanism incorporates two additional coefficients: c_2, which ranges from 1.5 to 2.5, and c_3, which ranges from 0.5 to 1.5. These coefficients work together to balance exploration of new solution regions with exploitation of promising areas already discovered. For multi-objective optimization problems, the algorithm integrates the Pareto front concept, a set of solutions that are considered optimal when evaluated against multiple conflicting objectives. No solution in the Pareto front can be improved in one objective without worsening at least one other objective. In the current study, the algorithm operates with the following parameter settings: c_1 = 1.5, c_2 = 2.5, c_3 = 1.4, and problose = 0.3.

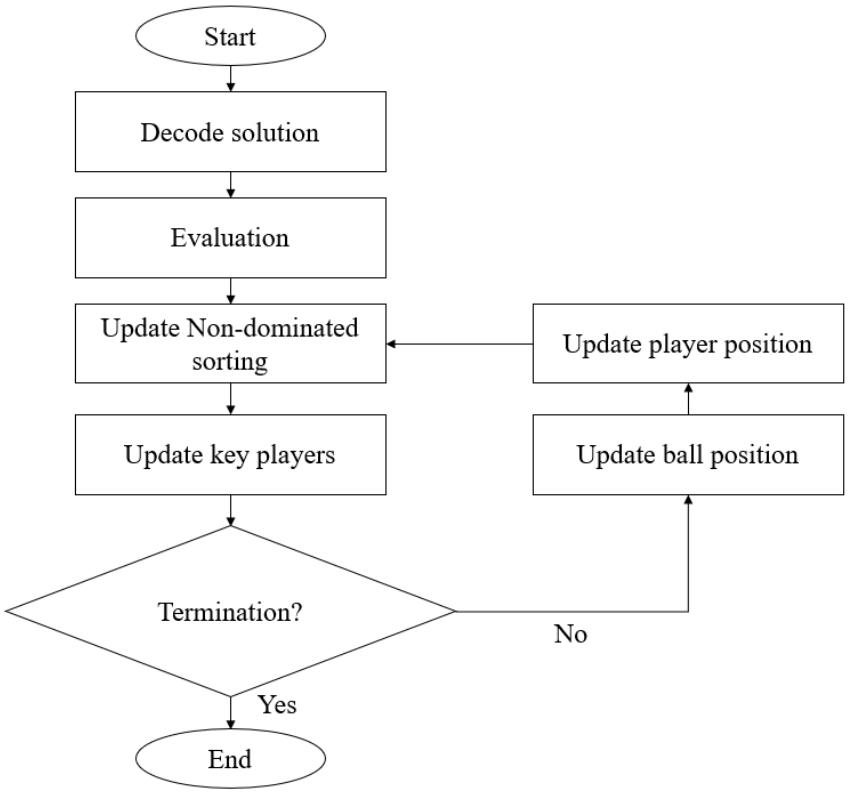

Figure 3. Flow chart MOTTA

The Pareto front embodies the idea of trade-offs and efficiency in decision-making. Typically, there are two or more objectives that need to be optimized simultaneously. The Pareto front consists of solutions where no solution in the set can be improved in one objective without sacrificing performance in another objective. In other words, it represents the set of solutions that achieve the best possible trade-offs between conflicting objectives. Finding the Pareto front involves exploring the solution space, evaluating solutions with respect to the multiple objectives, and identifying solutions that are not dominated by any other solution. Techniques such as non-dominated sorting and evolutionary algorithms are commonly used for this purpose. In the case of MOTTA, the first involvement in creating the Pareto front is to create a set of candidate solutions that covers the problem space. As for this problem, the two objectives which is the makespan and energy consumption. The solution of makespan is represented as points along the x-axis, while the energy consumption is defined along the y-axis. After decoding the solution, each population (player positions) is evaluated.

During this process, it is crucial to have data for non-dominated sorting as it captures the performance of each player across both objectives. It is also essential to identify the trade-off between different objectives and preparation for the sorting process. Once the solution is evaluated, the non-dominated sorting is applied. His sorting method classifies the players into different fronts based on Pareto dominance. The first front consists of non-dominated players, those who are not outperformed by any other player in all objectives. Players form subsequent fronts dominated only by those in preceding fronts, establishing a clear hierarchy of solutions based on their multi-objective performance. The steps of creating non-dominated players are explained in Figure 3.

3. RESULTS AND DISCUSSION

The Multi-Objective Tiki-Taka Algorithm was employed to solve the case study problem. To obtain trustworthy performances, the optimization was repeated 30 times independently. Once all runs had been performed, we merged the solutions obtained in each repetition into a unified set and removed any repeated solutions to prevent redundancy. For the remaining unique solutions, additional computation was used to identify non-dominated ones, i.e., those that represent the best trade-offs between competing objectives. To assess the performance of MOTTA, it was compared with four well-known algorithms: NSGA-II [21], MOEA/D [22], MOGWO [23], and MOPSO [24]. These comparison methods were chosen based on two main aspects: first, due to their proven ability to solve the Hybrid Flow Shop scheduling problem in machines; second, because of their recognition in the research area, as attested by several citations in the literature. Every compared algorithm underwent the same extensive testing to ensure a fair and comparable evaluation. These algorithms differ in their characteristics and approximations to the solution. Some of these algorithms use binary representation, whereas others use real-value vectors. The fitness assignments of these algorithms are also different. Some algorithms assign fitness values to individuals based on distances, while others combine dominance counts and a technique called crowding distance to determine fitness values. When storing solutions or archiving, some algorithms maintain the archive size, while others do not. However, the difference lies in archive management methods such as grid-based methods and clustering methods. In terms of reproduction or selection methods, many algorithms suggest tournament selection and fitness-based selection for their offspring. The way these algorithms maintain their best individuals, or elitism, differs among them. Some algorithms use elitist strategies to preserve the best individuals, whereas others do not. Furthermore, finally, for convergence and diversity criteria, some algorithms can preserve diversity but lack convergence to the true Pareto front. Some algorithms have strategies to maintain a satisfactory balance between the two. The overall performance of all algorithms was evaluated based on their non-dominated solutions (NDS) that are close to the Pareto front, their diversity, the spread of their results in the objective space, and their computational time to solve the problems. All comparison algorithms have similar settings: the population size is set to 100, the maximum number of iterations is 500, and the full 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). The solution for the error ratio of each NDS algorithm is then compared to the PF solution.

Table 6. Algorithms comparison

Performance Metrics	Algorithms				
	MOTTA	NSGA-II	MOGWO	MOPSO	MOEA/D
Error Ratio (ER)	0.21875	0.4375	0.5625	0.59375	0.59375
Number of ND	25	18	14	13	13
Generational Distance (GD)	8.9095	11.9220	12.4624	12.8066	12.9088
Inverted Generational Distance (IGD)	8.3692	41.0344	20.9127	43.2070	69.6660
Spacing (SP)	9.414	13.4954	22.7704	14.8159	19.9290
Maximum Spread (MS)	991.1015	967.0754	1017.1034	1020.1215	1089.2987
Hyper Volume (HV)	2.52E+06	3.49E+07	1.28E+06	2.37E+06	1.06E+07
Computational time (s)	1467	2103	1798	819	1572

The optimization results for the case study problem are summarized in Table 6, which indicates noticeable performance differences across the five algorithms tested. The authors showed that MOTTA achieved the best uniform performance, followed by NSGA-II and MOGWO from worst to best, with MOGO, MOPSO, and MOEA/D, respectively dominating: this was assessed according to eight performance metrics (i.e., Pareto percentage, error ratio, gd/ratio inverted generational distance metric (IGD), spacing, maximal spread, hypervolume, and computational time) over 30 independent repetitions. Two cardinality measures were used to test the algorithms' capacity to navigate through the solution space. The Pareto percentage indicates the relationship between the solutions obtained from each method and the overall solution, which is a measure of the feasibility of pursuing optimal solutions using different methods. A higher value means better exploration ability. The error ratio complements this measure by calculating the proportion of an algorithm's non-dominated solutions relative to the total set of non-dominated solutions identified across all algorithms. Together, these metrics reveal the breadth of solutions each algorithm can generate, with higher values indicating a more diverse solution set along the Pareto front. The convergence quality of each algorithm was evaluated using two distance-based metrics originally proposed by Van Veldhuizen and Lamont. Generational distance measures the average distance from each

algorithm's non-dominated solutions to the actual Pareto optimal front, with lower values indicating better convergence toward optimal solutions. Inverted generational distance provides a complementary perspective by measuring the average distance from points on the Pareto front to the nearest solution generated by the algorithm. This metric identifies gaps in coverage, with smaller values reflecting a more complete and accurate approximation of the Pareto front. Both metrics serve as convergence indicators, where lower values demonstrate superior algorithmic performance in approaching the optimal solution set. In terms of Pareto front distributions, the measurements used are MS, SP, and HV. The HV is the volume of the objective space of the solutions obtained. A good algorithm is characterized by a higher HV value, indicating a distinct distribution of solutions. The MS quantifies the solution by measuring its distance from the two non-dominated solutions. A higher spread value means a better distribution. Finally, Sp measures the average distance between the non-dominated solution and its nearest neighbour. The smaller value of SP signifies that the algorithm provides a well-distributed solution. The algorithm applies SP to the non-dominated solution distributions in the objective space.

Table 7. The overall ranking of all algorithms

Metrics	MOTTA	NSGA-II	MOGWO	MOPSO	MOEAD
Num ND	**1**	2	3	4	4
ER	**1**	2	3	4	4
Sp	**1**	2	5	3	4
MS	4	5	3	2	**1**
GD	**1**	2	3	4	5
IGD	**1**	3	2	4	5
HV	3	**1**	5	4	2
Average	1.7142	2.4285	3.4285	3.5714	3.5714
Rank	1	2	3	4	4

Table 8. Pareto optimal front solutions

Pareto Front Solutions	Makespan (minutes)	EE_{total} (kW-h)	Pareto Front Solutions	Makespan (minutes)	EE_{total} (kW-h)
1	2676	183.77	17	3158	163.26
2	2745	174.01	18	3193	163.18
3	2748	172.77	19	3205	162.73
4	2784	172.31	20	3213	162.71
5	2798	170.76	21	3292	162.59
6	2812	168.51	22	3315	162.32
7	2876	166.25	23	3326	161.91
8	2882	165.83	24	3390	161.20
9	2900	165.76	25	3400	161.18
10	2916	165.71	26	3403	161.04
11	2932	165.48	27	3435	160.72
12	2942	164.98	28	3453	160.71
13	3022	164.94	29	3500	160.34
14	3032	164.88	30	3528	160.31
15	3034	164.25	31	3541	158.80
16	3103	163.77	32	3691	158.26

Table 6 demonstrates that MOTTA outperformed all other algorithms in terms of ER, the number of ND on the Pareto front, GD, IGD, and Sp. This indicates that the MOTTA algorithm outperforms all other algorithms in terms of the dominance of non-dominated solutions on the Pareto front, while also achieving the best convergence near the Pareto front among different algorithms. Furthermore, MOTTA's error ratio demonstrates its ability to achieve optimal cardinality performance. Figure 4 shows that the distribution of the MOTTA non-dominated solution outperforms that of another algorithm. MOTTA, however, did not replicate the results for diversity and area covered. The results for maximum spread performance indicate that MOTTA still has room to improve its performance in spreading the solution. The presentation of the hypervolume result confirmed this similarity. The MOTTA solution did not cover the same area as another algorithm. The MOEA/D algorithm achieved the optimal maximum spread value. As for HV, NSGA-II produces the best-covered area among the algorithm's solutions. For an overall comparison, ranking is the best way to interpret the solution. Table 7 shows that the best algorithm based on the rankings is MOTTA, which ranks first, followed by NSGA-II in second place and MOGWO in third place, respectively. Despite all algorithms' solution metrics and

quality, one other comparison was computational time. The results clearly showed that MOPSO achieved the fastest computational time.

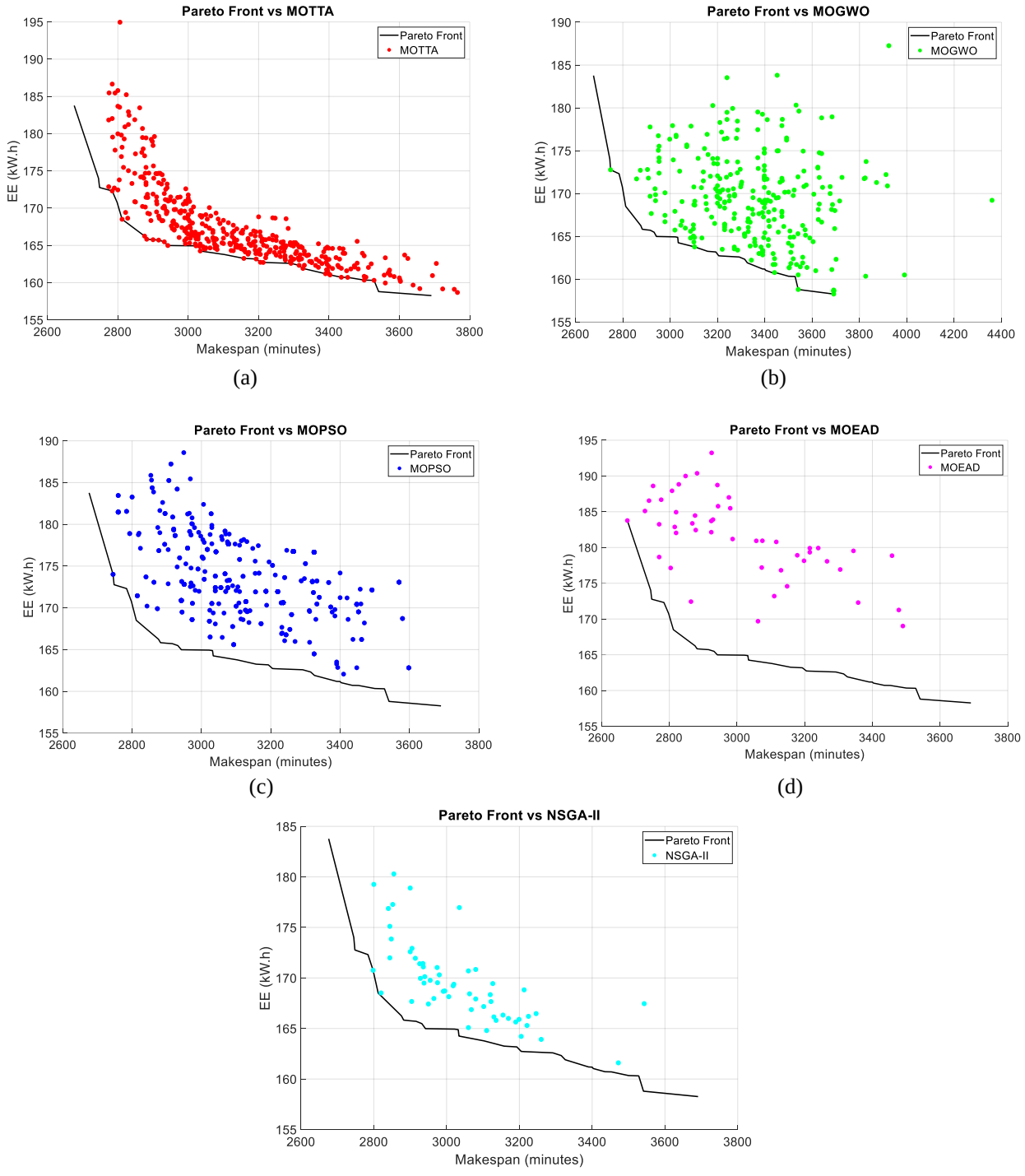

Figure 4. Solution and non-dominated solution of every algorithm: (a) MOTTA Pareto front, (b) MOGWO Pareto front, (c) MOPSO Pareto front, (d) MOEAD Pareto front, and (e) NSGA-II Pareto front

Figure 4 presents the solution distribution patterns across all evaluated algorithms, illustrating how non-dominated solutions emerge through an iterative process of elimination and selection. After conducting 30 independent runs for each algorithm, the non-dominated solutions are displayed as coloured points in Figure 4, while the final selected solutions after applying the elimination factor are marked as square points. This visualization effectively demonstrates the formation of the Pareto front, which represents the optimal trade-off boundary between competing objectives. The comparative analysis reveals notable differences in algorithm performance. MOTTA achieved the highest number of non-dominated solutions with 25 points, followed by NSGA-II with 18 points, MOGWO with 14 points, and both MOPSO and MOEAD with 13 points each. After consolidating the results from all algorithms and removing duplicate solutions, a total of 32 unique solutions were identified on the Pareto optimal front, as detailed in Table 8. This comprehensive set of solutions provides decision-makers with a range of options that represent different trade-offs between makespan and

energy consumption. For practical implementation in the case study, selecting the most appropriate solution from these 32 Pareto optimal alternatives depends on the specific priorities and constraints faced by stakeholders and policymakers. Different operational scenarios may prioritize either minimizing production time or reducing energy consumption, or seek a balanced compromise between both objectives. To demonstrate this flexibility, three representative solutions were selected from the Pareto front and are presented in Figures 5 through 7, each highlighting different strategic approaches to scheduling optimization.

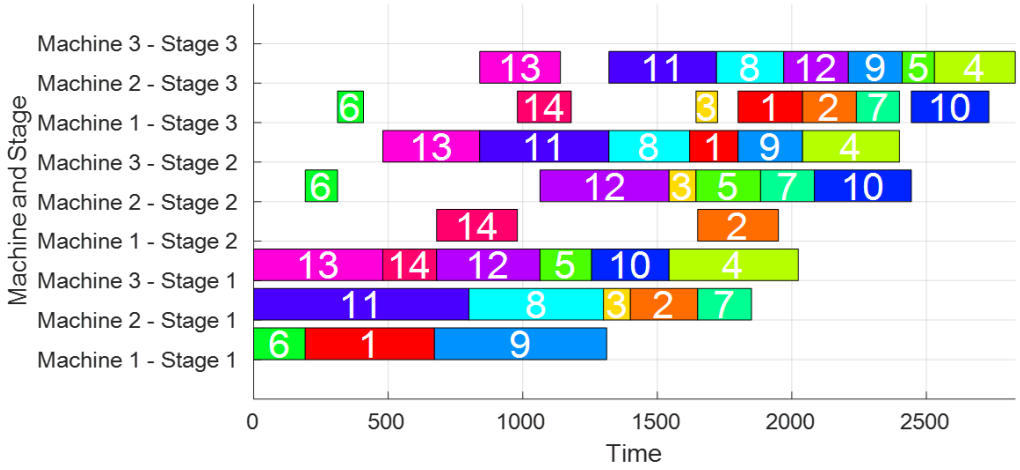

Figure 5. Scheduling for makespan = 2676 minutes and EE_{total} =183.77 kW-hr

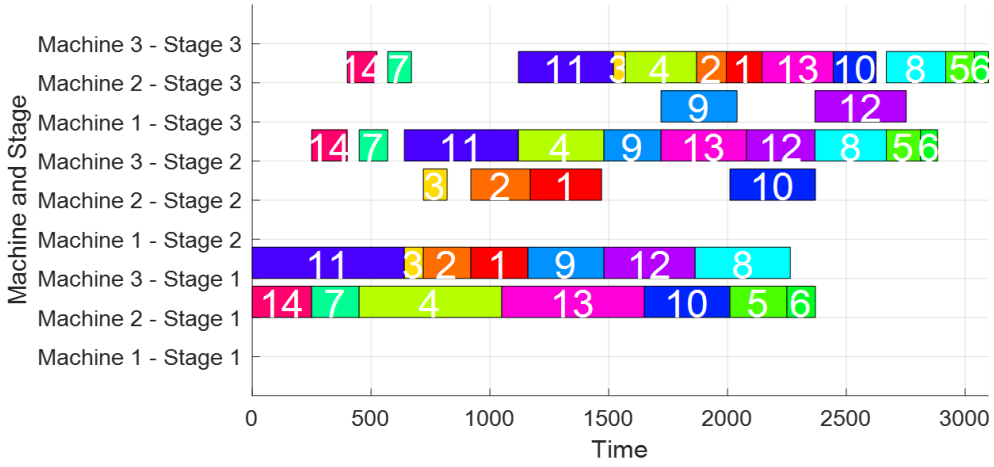

Figure 6. Scheduling for makespan = 3103 minutes and EE_{total} = 163.77 kW-hr

The optimization process generated multiple solutions that demonstrate the inherent trade-off between completion time and energy consumption in the scheduling problem. Three distinct solutions were selected to represent different operational priorities and decision-making scenarios. The first solution, presented in Figure 5, prioritizes rapid completion above all other considerations. This approach yielded a makespan (C_{max}) of 2,676 minutes with a corresponding energy expenditure of 183.77 kWh. As expected, achieving the fastest possible completion time resulted in the highest level of energy consumption among all solutions examined. This solution would be appropriate for situations where meeting tight deadlines is the paramount concern, regardless of energy costs. The second solution, illustrated in Figure 6, represents a balanced approach derived from a weighted analysis of the 32 generated solutions. This middle-ground option achieved a makespan of 3,103 minutes while reducing energy expenditure to 163.18 kWh. By accepting a moderate increase in completion time of approximately 16%, this solution delivered an 11% reduction in energy consumption compared to the first solution. This balanced configuration exhibits the least trade-off effect, making it suitable for operational contexts where both objectives hold relatively equal importance. The third solution, shown in Figure 7, prioritizes energy efficiency as the primary objective. This approach resulted in the lowest energy expenditure of 158.26 kWh, though it required extending the makespan to 3,691 minutes. Compared to the first solution, this energy-focused approach reduced power consumption by approximately 14% while increasing completion time by 38%. This solution is most appropriate when energy costs are a critical concern and scheduling flexibility allows for longer completion times. The selection among these three solutions ultimately depends on managerial priorities and operational constraints. Decision-makers must determine whether to prioritize makespan minimization followed by energy reduction, or conversely, to prioritize energy efficiency with makespan as a secondary consideration. The availability of these distinct solutions provides flexibility in addressing different business scenarios and strategic objectives. Regarding the scheduling of jobs, the schedule with the least makespan has larger idle machine gaps compared to the schedule that prioritizes energy consumption. When the primary objective is to minimize the makespan, the scheduling algorithm strives to expedite the completion of all jobs.

This frequently entails prioritizing tasks that can be promptly executed on any accessible machine, which may result in periods of machine idleness while waiting for specific tasks to be prepared for execution. The primary objective is to minimize the length of the critical path, which is the series of tasks that determines the overall duration of the project, even if it results in temporary underutilization of specific resources. When the focus shifts to minimizing energy consumption, the scheduling algorithm may prioritize keeping machines running continuously or reducing the number of machine start-ups and shutdowns. This approach can lead to a more even distribution of jobs and fewer idle periods, as the algorithm attempts to keep machines operating at a steady rate to avoid energy-intensive idle states.

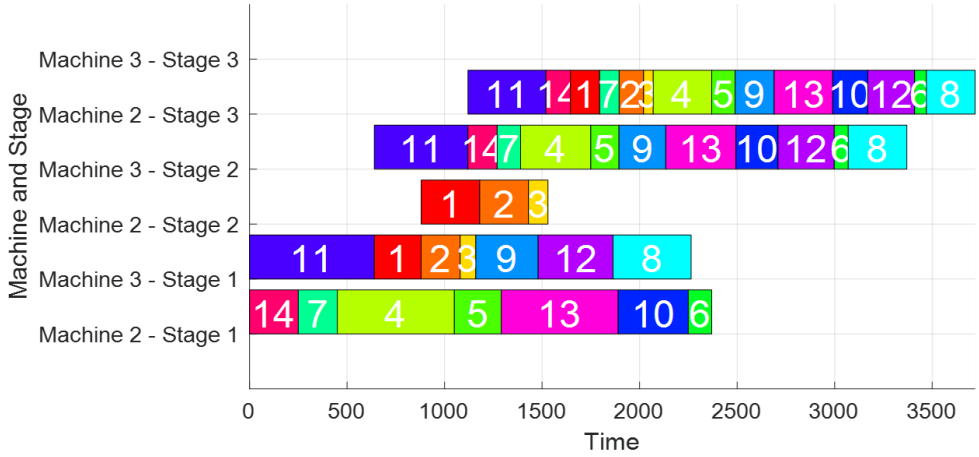

Figure 7. Scheduling for makespan = 3691 minutes and EE_{total} = 158.26 kW-hr

4. CONCLUSIONS

This study addresses the optimization of hybrid flow shop scheduling with dual objectives of total energy efficiency and makespan minimization. A practical case study was conducted involving five manufacturing operations: punching, bending, welding, pressing, and drilling. Given the heterogeneous nature of the machines and the constraint of a single drilling machine, the resulting energy efficiency and makespan values exhibited greater predictability. However, substantial variations were observed across the performance of different algorithms. The Multi-Objective Tiki-Taka Algorithm was implemented and systematically compared against alternative approaches using multiple verification criteria for multi-objective optimization. Computational results demonstrate that MOTTA outperforms competing algorithms across five performance metrics. The algorithm exhibits particularly strong performance in distribution-related measures, achieving first-place rankings in Error Ratio, Spacing, Generational Distance, and Inverted Generational Distance among the five algorithms evaluated. This capability to reach optimal solutions while maintaining solution diversity highlights MOTTA's effectiveness in managing various cost functions simultaneously. Nevertheless, MOTTA exhibits moderate performance in certain areas, specifically in terms of covered solution space (hypervolume) and diversity range (maximum spread). While MOTTA achieves the highest overall ranking among the five algorithms, these findings indicate that further refinement is necessary to establish comprehensive superiority across all evaluation dimensions. The algorithm demonstrates notable strengths in solution distribution (first rank), convergence quality (second rank), while showing adequate but improvable performance in solution cardinality (fourth rank). These results suggest that MOTTA represents a promising approach with clear potential for improvement, positioning it as a competitive method for multi-objective hybrid flow shop scheduling problems, while acknowledging opportunities for further development.

ACKNOWLEDGEMENTS

The author would like to acknowledge University Malaysia Pahang Al-Sultan Abdullah, for funding this research under research grant RDU240315.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

AUTHORS CONTRIBUTION

S. N. H. H. M. HATA (Computational experiment; Result analysis; Draft writing)

M. A. N. Mutasim (Code troubleshooting; Writing improvement; Supervision)

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

AVAILABILITY OF DATA AND MATERIALS

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

ETHICS STATEMENT

The authors would like to state that no AI or AI-assisted technologies have been claimed as authors or co-authors of this work. Authors understand that authorship has some responsibilities that humans cannot be replaced. The authors have followed these guidelines in the preparation of the present manuscript.

REFERENCES

[1] 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.

[2] R. Ruiz, J. A. Vázquez-Rodríguez, "The hybrid flow shop scheduling problem," *European Journal of Operational Research*, vol. 205, no. 1, pp. 1-18, 2010.

[3] M. S. Salvador, "A solution to a special case of flow shop scheduling problems," in *Symposium on the Theory of Scheduling and Its Applications*, vol. 86, 1973.

[4] M. A. Hassanchokami, Vital-Soto, J. Olivares-Aguila, "The Role of environmental factors in the flexible job-shop scheduling problem: A literature review," *International Federation of Automatic Control*, vol. 55, no. 10, pp. 175-180, 2022.

[5] K. V. Sagar, J. Jerald, M. A. Khan, "An energy-aware optimisation model to minimise energy consumption and carbon footprint in a flexible manufacturing system," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 18, no. 5, pp. 2869-2880. 2023.

[6] T.-L. Chen, C.-Y. Cheng, Y.-H. Chou, "Multi-objective genetic algorithm for energy-efficient hybrid flow shop scheduling with lot streaming," *Annals of Operations Research*, vol. 290, pp. 813-836, 2020.

[7] L. Cheng, Q. Tang, L. Zhang, K. Meng, "Mathematical model and enhanced cooperative co-evolutionary algorithm for scheduling energy-efficient manufacturing cell," *Journal of Cleaner Production*, vol. 326, p. 129248, 2021.

[8] Z. Zeng, M. Honh, Y. Man, J. Li, Y. Zhang, H. Liu, "Multi-object optimization of flexible flow shop scheduling with batch process—Consideration total electricity consumption and material wastage," *Journal of Cleaner Production*, vol. 183, pp. 925-939, 2018.

[9] Z. Han, S.Wang, X. Dong, X. Ma, "Improved NSGA-II algorithm for multi-objective scheduling problem in hybrid flow shop. Innovative Techniques and Applications of Modelling," *Identification and Control: Selected and Expanded Reports from ICMIC'17*, pp. 273-289. 2018.

[10] S. Mirjalili, P. Jangir, S. Saremi, "Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems," *Applied Intelligence*, vol. 46, no. 1, pp. 79-95, 2017.

[11] X. He, S. Dong, N. Zhao, "Research on rush order insertion rescheduling problem under hybrid flow shop based on NSGA-III," *International Journal of Production Research*, vol. 58, no. 4, pp. 1161-1177, 2021.

[12] D. Kumar, V. Kumar, "Impact of controlling parameters on the performance of MOPSO algorithm," *Procedia Computer Science*, vol. 167, pp. 2132-2139, 2020.

[13] Y. Zuo, Z. Fan, T. Zou, P. Wang, "A novel multi-population artificial Bee Colony Algorithm for energy-efficient hybrid flow shop scheduling problem," *Symmetry*, vol. 13, p. 2421, 2021.

[14] C. Lu, L. Gao, Q. Pan, X. Li, J. Zheng, "A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution," *Applied Soft Computing*, vol. 75, pp. 728-749, 2019.

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

[16] G. Wang, X. Li, L. Gao, P. Li, "Energy-efficient distributed heterogeneous welding flow shop scheduling problem using a modified MOEA/D," *Swarm and Evolutionary Computation*, vol. 62, p. 100858, 2021.

[17] I. Huseyinov, A. Bayrakdar, "Performance evaluation of BGSA-III and SPEA2 in solving a multi-objective single-period multi-item inventory problem," in 4*th International Conference on Computer Science and Engineering (UBMK), pp. 531-535*, 2019.

[18] M. A .N. Mutasim, M. F. F. A. Rashid, "A trade-off criterion for bi-objective problem in solving hybrid flow shop scheduling with energy-efficient (EE-HFS) using multi-objective dragonfly algorithm (MODA)," in *AIP Conference Proceedings*, pp. 1-12, 2024.

[19] M. A. N. Mu'Tasim, 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] M. F. F. Ab. Rashid, "Tiki-taka algorithm: A novel metaheuristic inspired by football playing style," *Engineering Computations*, vol. 38, no. 1, pp. 313-343, 2021.

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

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

[23] S. Mirjalili, S. M. Mirjalili, A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.

[24] C. C. Coello, M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation*, vol. 2, pp. 1051-1056, 2002.

NOMENCLATURE

n	Total number of jobs	C_i	Completion time of job i
i	Job indexes $i = 1, 2 \ldots n$	$C_{\max}$	Makespan (time)
S	Number of stages	E_{ks}	Power use of machine k at stage s in Watt
s	Stage indexes, $s = 1, 2 \ldots S$	EE	Total energy used
K_s	Number of machines at S	y_{iks}	If job is processed at machine k at stage s
k_s	Machine indexes as stage S, $k_s = 1, 2 \ldots K_s$	J	Number of jobs
t_{iks}	Processing time of job i on machine k in stage s	IT_{ijk}	Idle time between job i and job j on machine k_s
ST_{ij}	The start time of job J_i on machine k_s	FT_{ij}	The finish time of the job J_i on machine k_s