APPLICATIONS OF
MODELLING AND SIMULATION
http://arqiipubl.com/ams eISSN 2600-8084 VOL 8, 2024, 225-238

Solving an Integrated Job-Shop – Mobile Robot Scheduling Problem in Flexible Manufacturing System using Enhanced Genetic Algorithm Structure with Local Search Method

Erlianasha Samsuria[1], Mohd Saiful Azimi Mahmud[1*], Norhaliza Abdul Wahab[1], Muhammad Zakiyullah Romdlony[2], Mohamad Shukri Zainal Abidin[1] and Salinda Buyamin[1]

[1]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia
[2]School of Electrical Engineering, Telkom University, Bandung, Indonesia

[*]Corresponding author: azimi@utm.my

Submitted 22 May 2024, Revised 18 July 2024, Accepted 27 July 2024, Available online 04 August 2024.

Abstract: In a highly automated Flexible Manufacturing System (FMS), optimal utilization and scheduling of resources and equipment are paramount. This necessity underpins the full utilization of automation capabilities, leading to increased productivity and minimized downtime. Efficient resource allocation and scheduling also contribute to better overall performance, allowing the FMS to meet production demands effectively while maintaining a high level of operational efficiency. In this paper, the job-shop production scheduling problem is studied which involve with the concurrent scheduling of jobs processing and mobile robot assignment for job transportation within FMS environment. The hybrid Genetic Algorithm and Tabu Search algorithm is proposed to solve the combinatorial NP-hard job-shop and mobile robot scheduling problem. The primary objective is to search for the best scheduling plan that includes job allocation and mobile robot assignment, aiming to minimize the overall time necessary to complete all tasks, also known as makespan, to the minimum as possible. The developed algorithm has been evaluated and compared with the classical (or standard) genetic algorithm and other hybrid GA (with Simulated Annealing (SA) algorithm) using two job datasets spanning from small to large-scale problems adopted from renowned benchmark job instances. The results of computer experiments substantiate the effectiveness of the proposed hybrid algorithm, showcasing superior-quality solutions with an approximate up to 2.86% and 3.55% improvements compared to the hybrid GA – SA and standard algorithm, respectively. The developed algorithm has been run and tested in the Matlab software environment.

Keywords: Flexible manufacturing system; Genetic algorithm; Hybrid optimization; Job-shop; Scheduling; Tabu search.

1. INTRODUCTION

The expeditious advancement of information technology has led to boost level of complexity and intensification of competition within the manufacturing industry. Indeed, the competitiveness of manufacturing in advanced economies is heavily contingent upon technology, with technology-intensive sectors prominently leading the field [1], [2]. Recently, the manufacturing industry has experienced a transition towards more adaptable and efficient production systems, with the flexible manufacturing system (FMS) emerging as the foremost solution. An FMS is characterized as a set of numerically controlled machine, interconnected via loading and unloading machine stations, by an automated material handling or transport system [3], [4]. In light of the pivotal role played by flexibility, cost-efficiency, and productivity as key drivers of manufacturing competitiveness, manufacturing companies must ensure their competitive edge by strategically aligning and coordinating organizational, decision-making, and operational tools [2], [5]. This strategic approach is essential for facilitating the establishment of an efficient and highly flexible manufacturing system. Insufficient support for efficient resource assignment in an FMS can greatly limit its performance, potentially reducing the benefits of flexibility, especially in terms of production costs. In such a scenario, an efficient scheduling system is essential, as it requires optimized decision-making processes to achieve savings in time, effort, and financial resources. Primarily, FMSs are favored in job shop environments for their efficiency and productivity akin to mass production, particularly in small batch manufacturing [6]. Job shop systems, known for their flexibility, are the most common form of FMS. The system processes each job on the available machines within a specified time frame, with each machine handling only one operation per job. Each job experiences multiple distinct manufacturing operations to be completed. Therefore, it requires more sophisticated scheduling algorithms to manage the flexibility and diversity of jobs involved, as producing a schedule can merely become challenging with the increasing number and complexity of operations and stages [7]. On this basis, the pursuit of generating an efficient schedule for general job shop scheduling problems has been a subject of many research since decades.

The performance of a manufacturing plant is closely tied to the implementation of a material handling system, which includes the movement and control of materials or jobs between different workstations on the production floor. The mobile robot system in well-known in automated material handling system as transportation medium broadly used in FMS. Typically, the transportation time of the jobs between different machines or workstations has been omitted in order to simplify the model and facilitate resolution. Even so, disregarding transportation is not aligned with realistic production practices, particularly when the movement of workpieces heavily relies on mobile robots. Transport tasks handled by mobile robots consume a significant amount of time and can impact the time needed to complete production operations. Idle machines may have to wait for current jobs to be delivered, causing delays. By that, scheduling mobile robots or transportation is a practical assessment of cycle time, which is as crucial as production or machine scheduling. Optimizing both aspects becomes a central task in flexible manufacturing and modern management, as it not only allocates tasks but also impacts resource utilization and energy efficiency [8], [9]. It is widely acknowledged that solving the job-shop scheduling problem is a time-consuming an falls into the category of NP-hard problems, computation time exponentially grows in respect of the size of the problem [10]–[12]. Since the job-shop operations and transport task need to be scheduled simultaneously due to the high interconnection between these two problems, they are considered as an extensive combinatorial NP-hard problems [13]. Hence, solving such complexity necessitates the use of efficient optimization techniques.

The purpose of optimization is to obtain an optimal solution by searching the best solution from all those possible ones based on a set of criteria [14]. By applying optimization techniques, the solution to the requisites in dealing with the complexities of scheduling problems in FMS environment can be accessible. From the literature, the generation of scheduling solution for production and transportation has been literarily addressed using exact approaches such as branch and bound [15] and mathematical programming model [16], [17], as well as optimization approaches such as metaheuristics [18]–[24]. In view of the NP-hardness of the FMS scheduling problem, the exact approaches ensure an optimal solution but are feasible only for small problem instance. As the scale of the problem increases, the computing space and time required also grow exponentially, making exact approaches impractical for finding an optimal solution within reasonable time limits.

In contrast, metaheuristic-based approaches frequently yield solutions that are ‘near to optimal’ and of high quality for problem instances of realistic size, all within a relatively short running time. The integrated scheduling of machines and mobile robots (i.e., AGVs) was first reported by Bilge and Ulusoy [25], [26]. Following this, numerous researchers conducted further research on the issue. In the context of the flexible job-shop scheduling problem, Deroussi and Norre [27] were among the pioneers to conduct a study on the joint scheduling of machine operations and AGVs. They proposed an iterative local search algorithm based on classical exchange, insertion, and perturbation movements. Authors in [28] developed two stages approaches to integrate a transportation tasks of a single robot into procedures for machine schedules in job-shop environment. Zhang et al. [29] provided a detailed description of the job shop scheduling problem with transport constraints, utilizing a modified analytic graph. Subsequently, they devised an enhanced moving bottleneck heuristic to effectively solve this problem.

Over the years, there has been a rising interest in utilizing population-based metaheuristic, specifically genetic algorithms (GAs), to solve engineering optimization problems, particularly in the realm of scheduling. Sharma et al., [30] demonstrated the application of GA for solving the job shop scheduling (JSS) problem, with a specific focus on providing a scheduling plan for a part involving 5 machines and 4 operations. With the primary goal of minimizing the total travel time of the robot, the authors proposed a heuristic algorithm based on GA to solve a single mobile robot scheduling problem [31], [32]. The studies from [11], [20], [21] have introduced a GA-based scheduling optimization approach to identify the optimal solution for two interconnected decision problems involving simultaneous scheduling of machines and mobile robots. Guohi et al., [33] addressed the flexible job shop scheduling problem, taking into account transportation time, by developing a modified GA. Their designed algorithm includes an operation left shift insertion method to decode the operation sequencing component. On top of that, these studies have enriched the methodology of using GA in solving the scheduling problems, particularly those with constraints related to mobile robots.

In the standard GA approach, knowledge is applied across four distinct stages, i.e., initialization, selection, crossover and mutation. The algorithm is well-known for its global optimal self-adaptive probability search, which profits from biological selection and genetic evolution mechanism [34]. Yet, the classical GA often struggles to maintain diversity in its population and may consequently converge to local optima when confronted with more complex optimization problems [35], [36]. In this research, we tackled the combinatorial NP-hard job-shop scheduling problem with mobile robot assignment for transportation, while ensuring the preservation of precedence constraints within the FMS environment. To address this problem, we proposed to enhance the performance of the standard GA structure by incorporating local search method called as Tabu Search (TS) algorithm. TS algorithm stands out as the foremost local search strategy for scheduling problems that leverage neighborhood search methodologies while also preventing cycles repetitions by the use of a memory mechanism known as the tabu list [37], [38]. This integration aims to diversify the population across generations of GA, thus helping to avoid convergence to local optima. Indeed, integrating TS into GA will creates a strong hybrid algorithm that benefits from the global search capabilities of GA and the local search strengths of TS, which helps address dominant optimization issues such as avoiding entrapment in local optima and preventing premature convergence [37]–[41]. The global search capability of GA allows it to explore a wide range of solutions across the search space, while the local search ability of TS enables it to exploit promising areas more effectively. Researchers have shown considerable interest in combining GA and TS within a unified framework to address various scheduling problems, such as job-shop scheduling [42]–[46], flow-shop scheduling [38], nurse scheduling [38] and grid scheduling [47]. These studies have utilized GA as the foundational search mechanism, integrating TS to improve the search process. In most instances, TS has been applied on each chromosome within the GA to search for neighborhood solutions. These studies highlight the notable efficacy of merging GA and TS in addressing a distinct nature of scheduling problems, outperforming the stand-alone algorithm. To a certain extent, GA excels in global search capabilities but is prone to falling into local optima, while TS provides solid local search ability, however, its optimization outcome is highly sensitive to the initial solution [40], [48]. By combining these two algorithms, their respective strengths are utilized, and their weaknesses mitigated, achieving a proper balance between diversification (exploration) and intensification (exploitation) through genetic operators and the TS procedure.

This remarkable hybrid performance has motivated the current research to utilize benefits of global and local search scheme of GA and TS in solving the complex scheduling problem. This research addresses the complex nature of the flexible scheduling problem, particularly with the inclusion of mobile robot transportation considerations. Integrating job scheduling on machines and routing robots adds complexity to the problem, leading to a larger search space and increased difficulty in finding optimal solutions. In this context, the TS procedure can be seen as a specialized operator within the GA framework to enhance the searching behavior and exploration in search space across generations. This prompts the retention or refinement of feasible solutions, which might otherwise be overlooked by the global search nature of GA alone. Nonetheless, as far as we are aware, there are limited studies in the literature that combine the advantages of GA and TS to solve complex combinatorial scheduling problems. Previous works have focused primarily on classical job-shop scheduling without considering additional resources or constraints such as transportation, which makes them less applicable to actual production, particularly in an FMS environment. Thus, this research aims to address this gap by developing an effective hybrid algorithm that leverages the strength of local search capabilities of TS to enhance the conventional GA's searching performance and create optimal or high-quality solutions for the integrated scheduling of job-shop operations on machines with the transportation of jobs on mobile robots in the practical environment of FMS. In brief, the main objective of this study is to generate the best schedule plans that includes the sequences of job operations on specified machines as well as task assignment for mobile robot, with the aim of reducing the completion time of all tasks (or called as makespan). The efficacy of the developed scheduling algorithm is evaluated using dual mobile robot with some numerical experiments of the common problem instances used from the previous researchers.

The remainder of the paper is structured as follows. Section 2 delineates the formulation of the problem under consideration and outlines the process of developing the proposed hybrid algorithm designed to address respective problem effectively. Section 3 provides an overview of the experimental scenario and delves into a discussion of the results obtained. Finally, Section 4 concludes the paper.

2. MATERIALS AND METHOD

This section describes the formulation of the respective production scheduling problem, as well as the development of the proposed hybrid algorithm designed to address the scheduling problem under study. The details are presented in the following subsections.

2.1 Problem Formulation

In this paper, the production scheduling problem under study deals with concurrent scheduling of job-shop operations and mobile robots within FMS environment, where the material transport system is based on mobile robot platform. At every workstation, an array of parallel and identical machinery is strategically placed to optimize flexibility, with cautious planning and scheduling of operations for each job. To facilitate their movement, mobile robots are prepared and ready to swiftly retrieve or deliver jobs, ensuring rapid movement to workstations in the shortest time frame possible. The work centre in the FMS is versatile, capable of performing multiple identical operations, allowing for the processing of a group of parts or tasks. Mobile robots will be responsible for transporting jobs between the machines. Loading and unloading (L/U) stations are the main entry point and exit point for all jobs entering and exiting the system. It serves as the distribution/collection centre for the components where all jobs enter and leave the system through the L/U stations. Sufficient input/output buffer space is available at each machine. Loading of machines, involving the allocation of tools and assignment of operations, has been finalized. Operations are non-preemptive, and the ready times for all jobs are directly known.

The problem of integrated job-shop and mobile robot scheduling corresponds to a number of operations of assorted jobs processed on a set of machines while additionally considering transportation times. This may be formulated as follows: We are given a set of M machines,

$$M = [M_1, M_2, \dots \dots M_m] \tag{1}$$

and a set of n jobs which represented as,

$$J = [J_1, J_2, \dots \dots J_n] \tag{2}$$

which each job consists of a set of n_i operations,

$$O_{ij} = [O_{i1}, O_{i2}, \dots \dots O_{i,n_i}] \qquad ; \quad i \in J \tag{3}$$

such that $O_{ij} \in O_i^N$ where O is the set of all operations. Each operation denoted as O_{ij} requires uninterrupted execution on one of the M available machines. Each machine has a known processing time PT_{ij}, while complying to the constraint that only one operation can be processed on each machine at any given time. Additionally, every job may require a maximum of n_i transport tasks, as each job needs to be transported to the machine processing each of its n_i operations.

To enable the construction of the mobile robot scheduling system, some reasonable assumptions, parameters and decision variables are proposed in this study:

- Operational processing and travelling times are available and deterministic
- Each MR can only transport one component or part at a time
- Mobile robots are identical, i.e., they are the same in carrying and running performance so that any transportation task which needs a MR can be accomplished by any of them
- Operation of one job cannot be manufactured before the previous operation of the same job is completed
- There is sufficient input and output buffer at each machine and L/U stations

The goal of the respective scheduling problem described for a given FMS environment is to concurrently optimize efficient job processing and the transportation cost of mobile robots, with the goal of minimizing the makespan, which represents the total time to complete all tasks. Here, the model for optimizing the fitness function, specifically the makespan, can be demonstrated mathematically by the following equations,

$$f = min\, C_{max} \tag{4}$$

The objective is to determine a feasible schedule that minimize the makespan, C_{max}, which is denoted as the maximum value among the completion times of each job, C_i, for i is ranges from 1 to n where n is the maximum jobs that are going to be scheduled. This can be described as follows,

$$C_{max} = Max\,(C_1,\ C_2, C_3, \dots .\ C_n) \tag{5}$$

which based on the total operations completion time as in the following equations,
Operation completion time,

$$T_{ij} = t_{mm'} + t'_{mm'} \tag{6}$$

$$O_{ij} = T_{ij} + PT_{ij} \tag{7}$$

Total completion time,

$$C_i = \sum O_{ij} \tag{8}$$

where i and j represent number of jobs and operations, respectively, T_{ij} denotes as mobile robot's transportation/traveling times and PT_{ij} represents processing times of each job-operation. Given that integrated scheduling is a combinatorial problem, it is imperative to select an effective method for optimization. Therefore, this research attempts to develop an efficient hybrid metaheuristic that merges a global population-based search algorithm with a local search method in an effort to achieve a feasible schedule with the best minimal makespan.

2.2 Proposed Algorithm

The interest of this study lies in procedure of investigating the application of metaheuristics in solving the integrated scheduling problem within a framework of FMS considering the precedence constraints. In brief, this study proposed a hybrid metaheuristic that specifically combines Genetic Algorithm with Tabu Seach as an alternative to enhance the performance of the existing algorithm. It is anticipated that this proposed hybrid can leverage the strengths of GA as a population-based global search method and TS as a local search method, thereby yielding superior results for the scheduling problem under consideration. Within this hybrid framework, the GA will generate the initial solutions or population utilizing its operators, including selection, crossover, and mutation rates. In the meantime, the TS is devoted to conducting neighbourhood search processes to further explore promising areas within search space of the generated individuals in each generation of GA. The detailed procedure of the proposed hybrid structure is presented in the following subsections.

2.3.1 Genetic Algorithm Design

An evolutionary algorithm (EA) is a versatile population-based metaheuristic optimization algorithm that incorporates mechanisms inspired by biological evolution. These mechanisms include reproduction, mutation, recombination, and selection. It mimic species evolution based on Darwin theory and form a cluster of algorithm by which Genetic Algorithm was the first proposed one [49]. Genetic algorithm is a kind of global optimal self-adaptive probability search algorithm that profits from biological selection and genetic evolution mechanism [34].

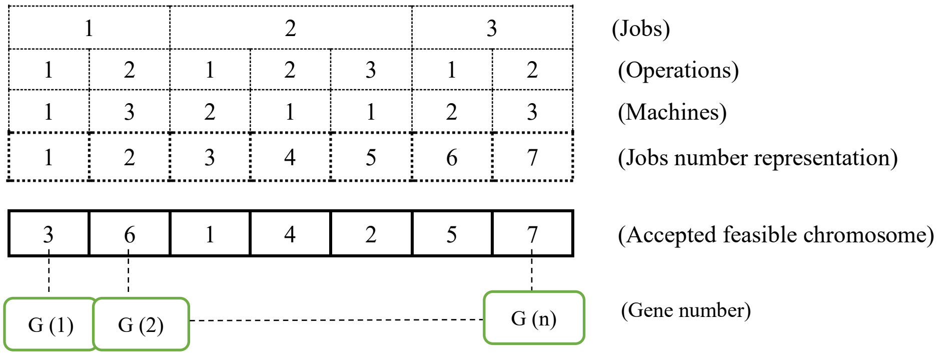


Figure 1. A structure of feasible chromosome representation

In this paper, the developed GA utilizes an operations-based coding approach, which is akin in principle to that found in flexible job shop scheduling [50],[51]. This approach helps to simplify the chromosome representation without necessitating significant additions of new GA operators. The other part which in addressing the scheduling of mobile robots for transportation task, a heuristic is integrated into the GA structure and invoked during the fitness evaluation module. The steps involved in designing the GA structure to search for the optimal solution with minimal makespan for the studied scheduling problem can be outlined as follows:

Step 1: Create an initial population of chromosomes that indicates the sequence of operations that is expressed by an integer number. Each operation represents the processing of particular task on a particular machine. In addressing the problem at hand, a permutation-based representation is employed for encoding the solution structure. This form is considered natural for problems related to operation sequences [52]. An illustration of a feasible chromosome in a scheduling scenario with a total of 7 operations comprising 3 tasks and 3 machines is provided in Figure 1.

Figure 1 illustrates that a feasible chromosome is a straightforward representation or permutation of job sequences depicted by integer numbers starting from 1. Each integer represents the order of operations, with each operation corresponding to the processing of a particular task on a specific machine. It is important to note that the generation of feasible chromosomes considers precedence constraints. Each chromosome within this genetic representation must adhere to the precedence relations governing operation sequences of each task (job). Otherwise, the unfeasible chromosome will be ignored. For instance, let's take job number 5 as an example. If job number 5 represents the second operation of the second task, it cannot begin or appear before the first operation of the second task (job number 4), which itself can only start after the first operation of its respective task (job number 3). That means, job number 5 cannot be done unless job number 4 has finished and so forth.

Step 2: Evaluate the fitness function for each chromosome in population in corresponds to the makespan minimization. To calculate fitness value of an individual, C_{max}, the chromosome is then decoded from gene $G(1)$ to $G(n)$. Note that, the information of transportation is included in the fitness evaluation module, where the mobile robot will be assigned to handle the job transportation using the heuristic function within the module.

In this context, the heuristic is responsible for decoding the GA operation sequence and assigning tasks to mobile robots for transporting jobs according to the feasible operations-based sequence. This is particularly for the case of using two mobile robots, the assignment of mobile robots is computed based on available heuristics such as the earliest/nearest rule [53],[54]. It constantly monitors the status of the current job and mobile robot, calculates which robot is available and nearest to the demand point, and then allocates tasks in accordance with the scheduling list in each chromosome that contains a sequence of locations. Following the chromosome scheduling string and mobile robot assignment, each individual in each generation will be evaluated for the selected fitness value of makespan as described in equations (4 – 8). Each job involves transportation from an origin to a destination, resulting in two types of trips for each job. The trips of the mobile robot can be categorized into two types: deadheading trips which involve the robot being unloaded and starting delivery from its current location to workstations requiring job loading after the current operation is completed; and load trips, where the robot transports jobs from one machine to another machine that carries out the next operation.

Step 3: Select the individuals with relatively higher fitness from current population that is conducted using roulette-wheel selection method [30]. Those high-fit individuals will have bigger chance of getting selected to participate in the reproduction as parent chromosome.

Step 4: To create a new offspring in crossover operation, copy and exchange the selected job-operation numbers on the same parent chromosomes. Initially, a job (or task) is chosen at random from each parent chromosome, referred to as P1 and P2. The matched operations are subsequently identified and directly copied into the corresponding positions of their respective offspring, denoted as O1 and O2. The remaining positions in the offspring are then filled with the unselected job-operation numbers according to their order in the opposite parent chromosomes. Taking the example of the job chromosome from Figure 1, let's consider job number 2 being selected, which includes a total of 3 operation numbers represented by job numbers 3, 4, and 5. The procedure of the operations in crossover process can be illustrated as shown in Figure 2. We can see that in the case of direct copying, the positions are not altered during the crossover process, so that the precedence relationship will remain intact.

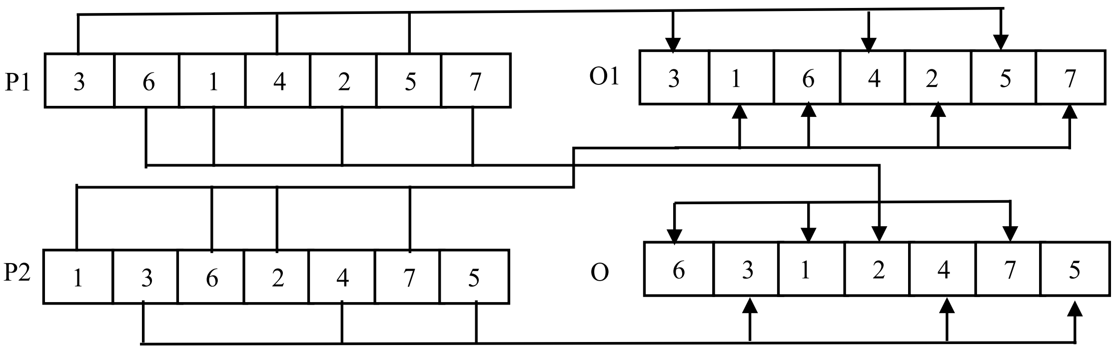


Figure 2. An example of the procedure of the operations in crossover.

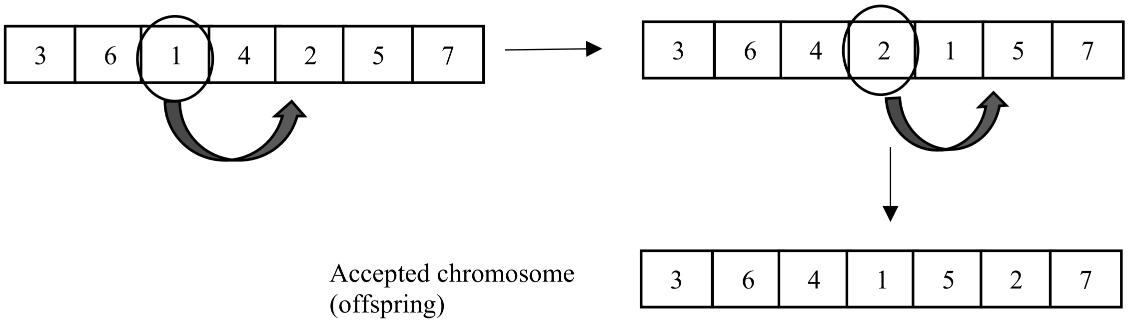


Figure 3. Position displacement procedure in mutation.

Step 5: Upon crossover, some offspring undergo operation shift mutation at each locus, which involves inverting the substring between two randomly selected positions within a chromosome. Here, the operation displacement mutation is applied in a manner that ensures the involved steps never violate the precedence constraint while altering the position of selected genes. The procedure of the displacement method is stated as follows: The process starts by randomly selecting a job number from the initial solution, for example, job number 1. Next, the job numbers corresponding to the selected task number (in this case., job numbers 1 and 2) are marked. The current positions of these selected job numbers are then moved two positions to the right accordingly. Figure 3 shows the illustration of the described process.

Step 6: New created offsprings will be placed in the population.

Step 7: After the GA generates a newly generated population through its search process, the TS algorithm takes over to conduct a neighbourhood search process to the current population. Briefly, the solutions are added to a tabu list to prevent revisiting them in subsequent iterations. From this tabu list, the best chromosome is selected based on its superior fitness value, representing the optimal solution obtained from the GA search process. The implementation procedure for TS is detailed in the following subsection.

Step 8: Rerun the algorithm with the newly generated population and return the best solution in the current population. This iterative process continues until a termination criterion is met.

2.3.2 Tabu Search Implementation

In this hybrid algorithm, the process begins with GA generating a new population of solutions in each generation. After the completion of each GA generation, the solutions in the population are added to a tabu list. Following this, the TS algorithm takes over, utilizing the tabu list to prevent revisiting previously explored solutions. TS conducts a neighbourhood search on the current population, evaluating the fitness of the solutions found during this phase. From these solutions, the best chromosome is selected based on its superior fitness value, representing the optimal solution obtained from the GA search process. This iterative process continues with GA generating new populations and TS refining these populations until a satisfactory solution is achieved or a termination criterion is met. The fundamental structure of the TS algorithm integrated into the GA structure, can be written in pseudocode form as follows:

Procedure Tabu Algorithm in GA

1. Begin
2. Initialize tabu parameters, GA population
3. Determine the aspiration criteria; in this study, makespan minimization
4. For each chromosome in current population do
   4.1 Perform a neighborhood search with position displacement method (as in Figure 3)
   4.2 Evaluate the solution
   4.3 Get all neighborhood solutions
   4.4 Update the solution with the best solution that is non-tabu of the search results
5. If the iteration has reached maximum, finish; otherwise go to Step 3
6. End

3. RESULTS AND DISCUSSION

In this section, the performance of the developed algorithm is examined using a well-known benchmark dataset of a a typical job shop scheduling problem with an added constraint of job transportations using mobile robots in FMS. This paper adopts two different sample test problems derived from the known benchmark instances to test the efficacy of the proposed algorithm. The used sample data sets are explained as follows:

- ***Dataset 1:*** The first data set is one of a small-scale benchmarking instances suggested by Bilge and Ulusoy (instance EX44) [25], [13]. It is comprised of 4 machines processing a total of 19 operations across 5 jobs. There is an available machine layout that consists of travelling times of mobile robot from one machine to another that include loading and unloading time of the jobs. The assignment problem only pertains to transport tasks in this dataset.
- ***Dataset 2:*** The second data set is obtained from the benchmark instances by [55], specifically utilizing job set '10x5 (la01)', one of the classical benchmark instances for medium or large-scale flexible job-shop problems. The dataset consists of 10 jobs, each with 5 operations, amounting to a total of 50 operations that need to be processed across 5 specified machines. In this dataset, the travel time between the machines is taken from [53] which is the layout TTM 2.

These data sets are characterized by the availability of a single machine for each processing task, multiple transport resources, and fixed processing times. In this study, these data sets are dedicated to dual mobile robot job-shop problem within FMS environment. The job descriptions, including the processing times of their operations, are provided in Table 1 and Table 2 for each example data set. Meanwhile, Table 3 display the corresponding travel time matrices for transporting jobs between machines in the small and large-scale layouts.

Table 1. Job reference details (instance EX44) – Dataset 1.

Job no.	Operation no.	Machine (Processing Times)	Job no. Representation
1	1 2 3	4(11) 1(10) 2(7)	1 2 3
2	1 2 3	3(12) 2(10) 4(8)	4 5 6
3	1 2 3 4	2(7) 3(10) 1(9) 3(8)	7 8 9 10
4	1 2 3 4	2(7) 4(8) 1(12) 2(6)	11 12 13 14
5	1 2 3 4 5	1(9) 2(7) 4(8) 2(10) 3(8)	15 16 17 18 19

Table 2. Job reference details (10x5 la01) – Dataset 2.

Job no.	Operation no.	Machine (Processing Times)	Job no. Representation
1	1 2 3 4 5	2(21) 1(53) 5(95) 4(55) 3(34)	1 2 3 4 5
2	1 2 3 4 5	1(21) 4(52) 5(16) 3(26) 2(71)	6 7 8 9 10
3	1 2 3 4 5	4(39) 5(98) 2(42) 3(31) 1(12)	11 12 13 14 15
4	1 2 3 4 5	2(77) 1(55) 5(79) 3(66) 4(77)	16 17 18 19 20
5	1 2 3 4 5	1(83) 4(34) 3(64) 2(19) 5(37)	21 22 23 24 25
6	1 2 3 4 5	2(54) 3(43) 5(79) 1(92) 4(62)	26 27 28 29 30
7	1 2 3 4 5	4(69) 5(77) 2(87) 3(87) 1(93)	31 32 33 34 35
8	1 2 3 4 5	2(38) 1(60) 2(41) 4(24) 5(83)	36 37 38 39 40
9	1 2 3 4 5	4(17) 2(49) 5(25) 1(44) 3(98)	41 42 43 44 45
10	1 2 3 4 5	5(77) 4(79) 3(43) 2(75) 1(96)	46 47 48 49 50

Table 3. Travel time matrix (time unit).

		L/U	M1	M2	M3	M4	M5
Dataset 1	L/U	0	4	8	10	14	-
	M1	18	0	4	6	10	-
	M2	20	14	0	8	6	-
	M3	12	8	6	0	6	-
	M4	14	14	12	6	0	-
Dataset 2	L/U	0	3	5	7	9	14
	M1	15	0	3	5	8	12
	M2	13	15	0	3	6	10
	M3	11	13	15	0	4	8
	M4	7	9	11	13	0	5
	M5	4	6	8	10	13	0

To assess the efficacy of the Hybrid Genetic Algorithm and Tabu Search (HGATS) in addressing the integrated scheduling problem, this paper conducted a comparative analysis between the performance of the developed algorithm, a standard Genetic Algorithm (SGA) as well as with other hybrid algorithm of GA and Simulated Annealing (HGASA). The comparison study aimed primarily to demonstrate the performance enhancements achieved by the HGATS approach over the traditional GA structure. In addition, the proposed HGATS is also compared against another hybrid GA to verify its effectiveness. The concept of hybridizing GA and SA developed in this study has a structure similar to that of the proposed HGATS. These algorithms have been programmed in MATLAB software (2022b) and run on a pc having an Intel Core i7, 16GB RAM and Windows 11 operating system.

The best selected GA parameters for all comparative algorithms in the computational experiments are set as follows a population size of 80 individuals, with crossover and mutation probabilities set at 0.6 and 0.1, respectively. The length of tabu list and iteration in HGATS are set to 5. For the parameters of Simulated Annealing (SA) in HGASA, the maximum and minimum temperatures are set to 800 and 0.001, respectively, the cooling rate is set to 0.95 and 20 iterations are done at each temperature. The maximum number of generations for all algorithms is set to 200. In the experiment, the developed algorithms will terminate the generation process once this maximum iteration (or generation) limit is reached. It should be noted that the related parameters used in these algorithms are typically set at through iterative experimentation (trial and error process). This process involves running the algorithm multiple times with different parameter settings, evaluating the performance, and adjusting the values based on the observed outcomes until an effective combination of parameters is identified.

The developed algorithms are run with the aim of minimizing the makespan, which represents the time required to complete all jobs or tasks. The fitness performance comparison between the enhanced algorithm (HGATS) and the standard or standalone algorithm (SGA) along with its comparison against the HGASA is illustrated in Figure 4 and 5 for the respective Dataset 1 and Dataset 2. In the meantime, Figures 6-8 depicts the relationship curves of the best individual solution (minimum fitness) of HGATS against HGASA and SGA across all generations for job scheduling with a dual mobile robot based on Dataset 1 and Dataset 2, respectively.

From these figures, it is clear that HGATS, HGASA and SGA reach nearly optimal points at slightly different times. For Dataset 1, SGA achieves the best makespan value of 129 at the 15th iteration, whereas for Dataset 2, it reaches 705 at the 54th iteration. The HGASA algorithm converges to makespan values of 129 and 700 at the 51 and 111 iterations for Dataset 1 and Dataset 2, respectively. On the other hand, HGATS converges after a longer period as compared to the SGA, specifically at the 62nd iteration for Dataset 1 and the 92nd iteration for Dataset 2. However, HGATS significantly improves the makespan solution, reaching the best values of 126 for Dataset 1 and 680 for Dataset 2.

As depicted in Figures 6-8, the graphs for SGA show a strong correlation between the average fitness (mean of individuals) and the minimum fitness (best individual) across the generations. In contrast, HGASA and HGATS displays more variability within the population between these two metrics. However, HGASA shows excessively high average fitness values, accompanied by a notably large gap between average and minimum fitness levels. Based on the observation, the relationship curve between the average fitness and minimum fitness in SGA signifies a more exploitation-focused approach, whereas HGASA appears overly oriented toward exploration. In contrast, the variations between these metrics in HGATS suggest a balance between exploration and exploitation of the search space.

Based on the resulting figures, it can be seen that SGA tends to converge earlier in the search process once it finds good fitness levels, which can sometimes lead to premature convergence to suboptimal solutions. On the other hand, the proposed HGATS takes more time to refining and optimizing solutions, resulting in better fitness levels at later iterations. This distinction arises from SGA's emphasis on a quick identification of good fitness solutions but may miss out on further improvement as population diversity decreases over generations, while the HGATS algorithm balances exploration and exploitation more effectively throughout the searching process. In terms of parameter settings, the genetic operators in the developed GAs, such as crossover and mutation rates are fine-tuned to promote both exploration and exploitation, aiming for diverse regions of the search space and rapid convergence. In the meantime, the inclusion of Tabu Search in the proposed hybrid algorithm focuses on local search for further refining the GA population. This local search entails additional iterations to exploit known good solutions by efficiently exploring neighborhoods and avoiding redundant search paths within the GA population across each generation. This combined effort thereby impacting both the speed of convergence and the quality of solutions achieved which may potentially prolonging the convergence rates to reach higher-quality solutions compared to SGA.

In comparison with another extended hybrid GA algorithm, namely HGASA, the findings demonstrate that HGATS exhibits better convergence performance and quality of makespan in each type of problem instances (Dataset 1 and 2). It is observed that HGASA shows no improvement in makespan compared to SGA for Dataset 1. While the improvement in makespan achieved by using HGASA over SGA is observable in Dataset 2, it is not as substantial as the enhancement seen with HGATS. This performance discrepancy between a hybrid GA with SA algorithm and a hybrid GA with TS algorithm can be attributed to their distinct operational mechanisms. Simulated Annealing initially explores widely by probabilistically accepting worse solutions, but it can struggle to make the best use of good solutions later as temperature cools down, potentially leading to less proficient at exploiting promising solutions. This characteristic of SA, which aims to avid local optima by accepting worse solutions with a certain probability, can cause the overall fitness of solutions to appear worse or not improve initially. In contrast, Tabu Search excels at intensifying search efforts around promising solutions while avoiding revisiting recent moves, thereby navigating complex problem spaces more efficiently. The sensitivity of SA for having more setup parameters i.e., temperature schedules, cooling rates and acceptance criteria, which can be difficult to adjust these parameters across diverse problem domains, may contributes to its comparatively inconsistent performance relative to TS.

Based on these results, it is evident that HGATS effectively solves the integrated job-shop and mobile robot scheduling problem, achieving the best solution of makespan among the other algorithms under comparison. This highlights the ability of the developed HGATS to address the complex coordination of job-shop and mobile robot task assignments.

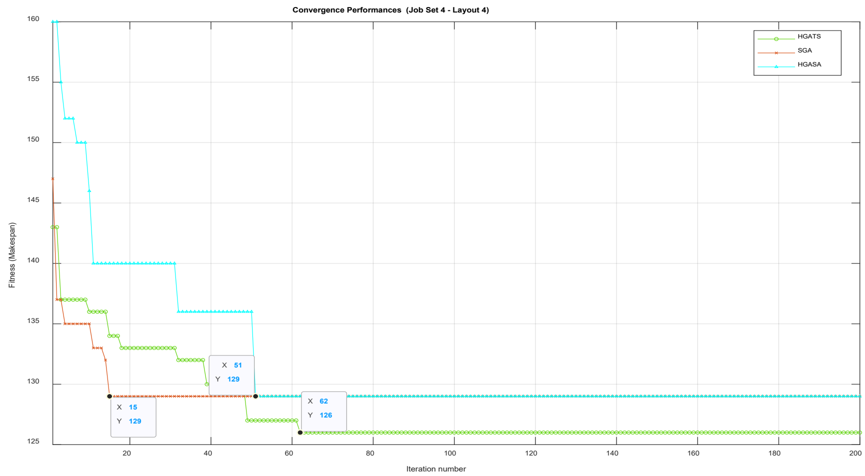


Figure 4. The convergence performances of the comparative algorithms for Dataset 1 (Smale-scale instance).

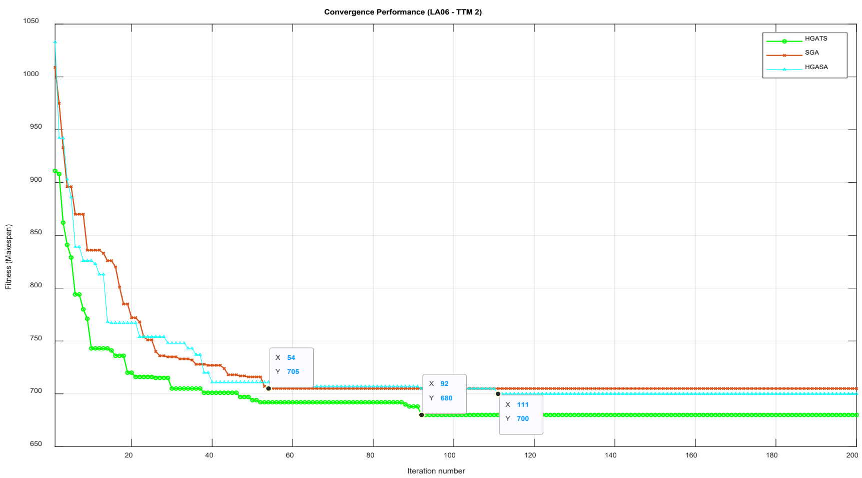
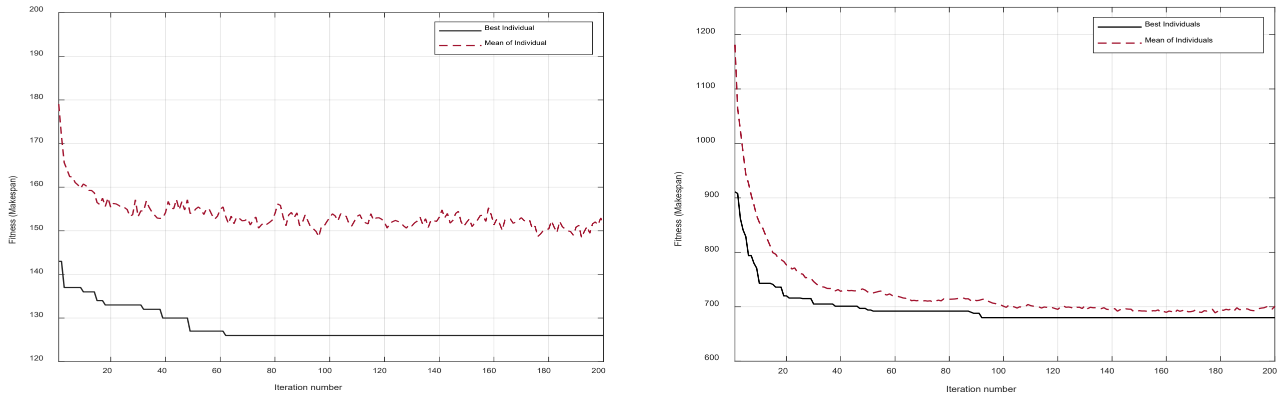
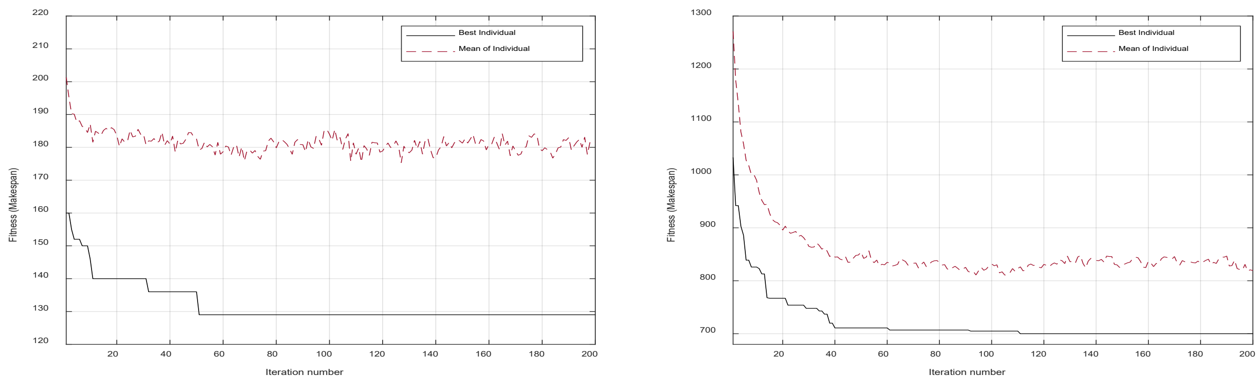


Figure 5. The convergence performances of the comparative algorithms for Dataset 2 (Large-scale instance).



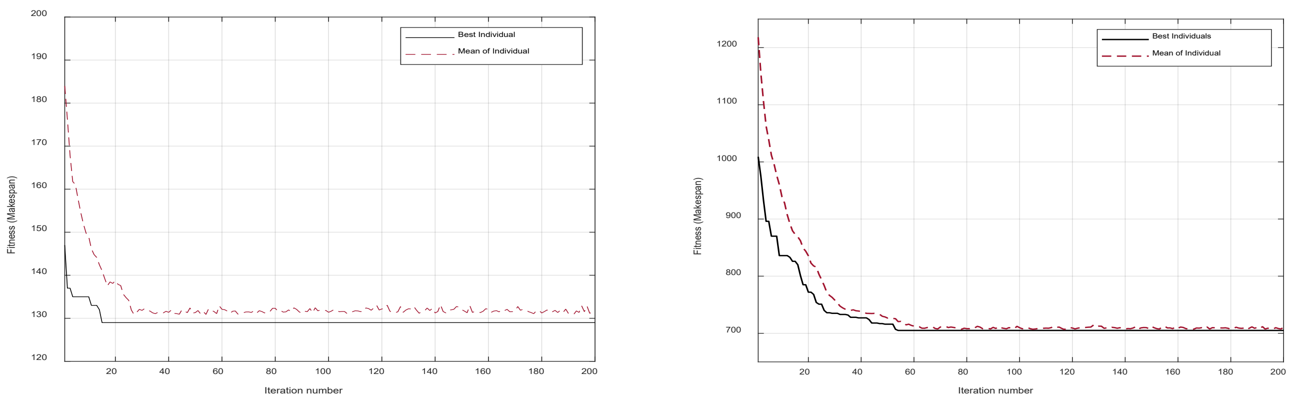
(a) (b)

Figure 6. The fitness cost over the number of generations of HGATS algorithm; (a) Dataset 1, (b) Dataset 2.



(a) (b)

Figure 7. The fitness cost over the number of generations of HGASA algorithm; (a) Dataset 1, (b) Dataset 2.



(a) (b)

Figure 8. The fitness cost over the number of generations of SGA algorithm; (a) Dataset 1, (b) Dataset 2.

The optimization results have been compiled and reported in Table 4 to provide a comprehensive overview of the scheduling outcomes. This table elaborates on the processing sequence of jobs and the assignment of mobile robots for job transportation according to the best schedule plan attained. Also, it includes the respective fitness values corresponding to the best (minimum) and mean solutions of makespan (in a time unit) obtained by the comparative algorithms. The corresponding table demonstrates the superior capability of the proposed HGATS in achieving better solution quality compared to the HGASA and SGA. in optimized minimal (best) makespan values by 2.33% for Dataset 1 compared to HGASA and SGA, and by 2.86% and 3.55% over HGASA and SGA for Dataset 2, respectively. The percentage improvement of makespan is calculated as follows,

$$\%\ Improvement = \frac{|C_{max}^{Standard} - C_{max}^{Proposed}|}{C_{max}^{Standard}} \times 100\% \tag{9}$$

Table 4. The scheduling results.

Data set	Algo.	Job sequence	MR Assignment	Fitness (C_{max})	
				Best	**Mean**
1	**HGATS**	1 – 7 – 4 – 15 – 16 – 11 – 8 – 5 – 17 – 12 – 2 – 18 – 9 – 6 – 13 – 10 – 3 – 14 – 19	1 – 2 – 1 – 2 – 2 – 1 – 2 – 2 – 1 – 2 – 1 – 2 – 1 – 2 – 2 – 1 – 2 – 1 – 2	126	149.61
	HGASA	15 – 1 – 16 – 11 – 4 – 17 – 2 – 7 – 12 – 18 – 13 – 8 – 5 – 19 – 3 – 14 – 9 – 6 – 10	1 – 2 – 1 – 1 – 2 – 1 – 1 – 2 – 2 – 2 – 1 – 2 – 2 – 2 – 1 – 1 – 2 – 1 – 2	129	181.45
	SGA	4 – 15 – 16 – 1 – 7 – 11 – 17 – 8 – 2 – 5 – 18 – 12 – 6 – 9 – 3 – 13 – 19 – 10 – 14	1 – 2 – 2 – 1 – 2 – 1 – 2 – 1 – 2 – 1 – 2 – 1 – 2 – 1 – 1 – 2 – 1 – 2 – 1	129	133.78
2	**HGATS**	31 – 46 – 26 – 21 – 6 – 41 – 16 – 32 – 27 – 47 – 36 – 28 – 1 – 22 – 42 – 17 – 33 – 7 – 48 – 11 – 2 – 29 – 18 – 49 – 23 – 43 – 19 – 37 – 8 – 12 – 30 – 34 – 24 – 44 – 50 – 38 – 9 – 20 – 3 – 13 – 35 – 39 – 45 – 40 – 4 – 25 – 10 – 14 – 15 – 5	1 – 2 – 1 – 2 – 1 – 2 – 1 – 2 – 1 – 2 – 1 – 1 – 2 – 2 – 2 – 1 – 2 – 1 – 1 – 2 – 2 – 1 – 2 – 2 – 1 – 2 – 2 – 1 – 1 – 1 – 1 – 2 – 2 – 2 – 1 – 2 – 1 – 1 – 1 – 2 – 1 – 2 – 2 – 2 – 1 – 2 – 2 – 1 – 1 – 2	680	726.13
	HGASA	46 – 26 – 31 – 16 – 6 – 36 – 1 – 27 – 21 – 32 – 7 – 41 – 17 – 28 – 22 – 33 – 2 – 18 – 37 – 23 – 11 – 42 – 8 – 29 – 12 – 34 – 19 – 35 – 38 – 47 – 43 – 24 – 48 – 9 – 3 – 39 – 44 – 20 – 13 – 49 – 25 – 14 – 40 – 15 – 45 – 4 – 10 – 5 – 50 – 30	1 – 2 – 1 – 2 – 1 – 2 – 2 – 2 – 1 – 2 – 1 – 1 – 2 – 2 – 1 – 2 – 2 – 2 – 2 – 1 – 1 – 1 – 1 – 2 – 1 – 2 – 2 – 2 – 2 – 1 – 2 – 2 – 1 – 1 – 1 – 2 – 1 – 2 – 2 – 2 – 2 – 1 – 2 – 1 – 1 – 2 – 1 – 2 – 1 – 1	700	819.32
	SGA	31 – 46 – 41 – 16 – 1 – 21 – 32 – 6 – 11 – 36 – 26 – 12 – 2 – 47 – 37 – 42 – 27 – 22 – 48 – 7 – 33 – 17 – 28 – 13 – 43 – 23 – 29 – 18 – 8 – 3 – 38 – 14 – 44 – 24 – 34 – 30 – 25 – 49 – 19 – 35 – 39 – 15 – 9 – 40 – 45 – 4 – 50 – 20 – 5 – 10	1 – 2 – 1 – 2 – 1 – 2 – 1 – 2 – 2 – 1 – 2 – 1 – 2 – 1 – 2 – 1 – 2 – 1 – 2 – 1 – 1 – 1 – 1 – 1 – 2 – 1 – 2 – 1 – 1 – 1 – 2 – 2 – 1 – 2 – 2 – 1 – 2 – 2 – 1 – 2 – 1 – 2 – 1 – 1 – 2 – 2 – 1 – 2 – 2 – 1	705	741.85

In addition, a Gantt chart representation is presented to visually depict the complete scheduling results derived from the optimization process. These Gantt charts serve as illustrations of scheduling plans for the integrated scheduling problem, specifically for a small-scale problem (Dataset 1), based on the scheduling results obtained from both the proposed and standard algorithms (see Table 4), as demonstrated in the respective Figures 9 and 10.

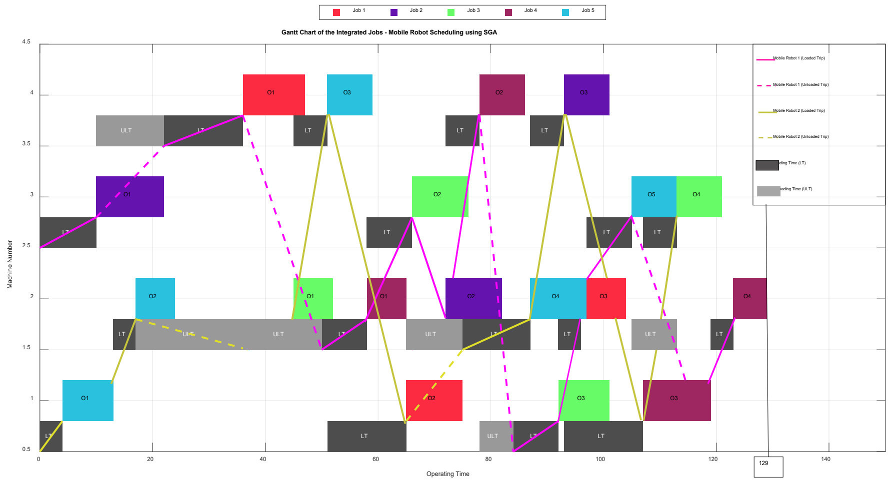


Figure 9. A Gantt Chart on the best scheduling plan obtained through the SGA algorithm.

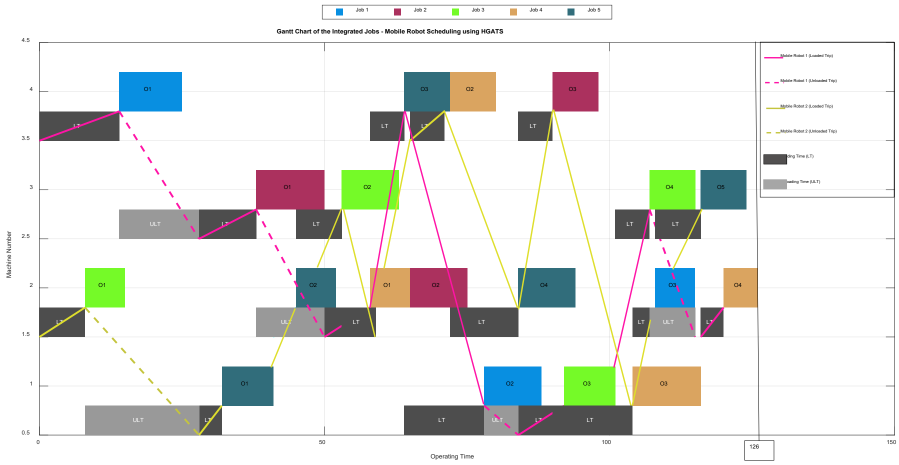


Figure 10. A Gantt Chart on the best scheduling plan obtained through the HGATS algorithm.

4. CONCLUSION

This paper addresses the production scheduling, which deals with the allocation of job-shop operations and the assignment of the transportation tasks using dual mobile robots within an FMS environment. To tackle this complex problem, the classical genetic algorithm has been augmented with a local search method, specifically the tabu search algorithm. The primary objective is to minimize the completion time of all processing jobs, including transportation tasks, constituting a combinatorial NP-hard scheduling challenge. On this basis, the paper thoroughly outlines and discusses the methodology employed in developing the algorithm, along with the nearly optimal solutions attained through computer experiments. These experiments utilize two distinct problem instances sourced from well-established benchmark job sets to assess the efficacy of the developed algorithm. The computational results demonstrated that the hybrid GA-TS offers a commendable solution, exhibiting minimal makespan and a notable performance improvement when compared against the classical GA and the hybrid GA algorithm incorporating SA algorithm. This underscores the algorithm's proficiency in dealing integrated scheduling across a range of small to large-scale problems encountered in complex FMS environments.

ACKNOWLEDGMENT AND FUNDING

The authors would like to acknowledge the financial support from the Ministry of Higher Education (MOHE) Malaysia and Universiti Teknologi Malaysia under Matching Grant between Universiti Teknologi Malaysia and Telkom University and UTM Encouragement Research Grant (Q.J130000.3023.04M12, R.J130000.7623.4B796 and Q.J130000.3823.31J58).

DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article.

REFERENCES

[1] M. H. F. Bin Md Fauadi and T. Murata, Makespan minimization of machines and automated guided vehicles schedule

using binary particle swarm optimization, *Proceedings of the International MultiConference of Engineers and Computer Scientists.* Hong Kong, 2010, 1897-1902.
[2] S. M. Homayouni and D. B. M. M. Fontes, Production and transport scheduling in flexible job shop manufacturing systems, *Journal of Global Optimization*, 79 (2), 2021, 463-502.
[3] G. A. Chang, Modeling and analysis of flexible manufacturing systems: A simulation study, *ASEE Annual Conference & Exposition*, Seattle, Washington, 2015, 12726.
[4] Y. M. Mejthab and K. M. Al-aubidy, Design and implementation of real-time scheduling algorithms for flexible manufacturing systems, *International Journal of Advanced Mechatronic Systems*, 7(4), 2017, 202-212.
[5] L. Zaidi, B. Bettayeb, and M. Sahnoun, Optimisation and simulation of transportation tasks in flexible job shop with muti-robot systems, *1st International Conference on Cyber Management and Engineering (CyMaEn),* Tunisia, 2021, 1-6.
[6] P. Kostal and K. Velisek, Flexible manufacturing system, *World Academy of Science, Engineering and Technology*, 77, 2011, 825-829.
[7] S. Gopi and S. A. Chandra, Solving distributed FMS scheduling problems with/ without breakdowns: Simulation optimization approach, *Materials Today: Proceedings*, 47(14), 2021, 4879-4884.
[8] X. Wen, Y. Fu, W. Yang, H. Wang, Y. Zhang and C. Sun, An effective hybrid algorithm for joint scheduling of machines and AGVs in flexible job shop, *Measurement and Control,* 56(9-10), 2023, 1582-1598.
[9] W. Li, D. Han, L. Gao, X. Li, and Y. Li, Integrated production and transportation scheduling method in hybrid flow shop, *Chinese Journal of Mechanical Engineering,* 35(1), 2022, 12.
[10] P. Bharti and S. Jain, Hybrid frameworks for flexible job shop scheduling, *The International Journal of Advanced Manufacturing Technology*, 108(5-6), 2020, 1563-1585.
[11] Q. V. Dang and L. Nguyen, A Heuristic approach to schedule mobile robots in flexible manufacturing environments, *Procedia CIRP,* 40, 2016, 390-395.
[12] S. Kumar, V. Manjrekar, V. Singh and B. Kumar, Integrated yet distributed operations planning approach: A next generation manufacturing planning system, *Journal of Manufacturing Systems*, 54, 2020, 103-122.
[13] S. Chakrabarti and H. N. Saha, Simultaneous Scheduling of machines and automated guided vehicles utilizing heuristic search algorithm, *IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, USA, 2018, 54-59.
[14] B. Vaisi, A review of optimization models and applications in robotic manufacturing systems: Industry 4.0 and beyond, *Decision Analytics Journal*, 2, 2022, 100031.
[15] P. Brucker, E. K. Burke and S. Groenemeyer, A branch and bound algorithm for the cyclic job-shop problem with transportation, *Computers & Operations Research,* 39(12), 2012, 3200-3214.
[16] A. Caumond, P. Lacomme, A. Moukrim and N. Tchernev, An MILP for scheduling problems in an FMS with one vehicle, *European Journal of Operational Research*, 199(3), 2009, 706-722.
[17] L. L. Fu, M. A. Aloulou and C. Triki, Integrated production scheduling and vehicle routing problem with job splitting and delivery time windows, *International Journal of Production Research*, 55(20), 2017, 5942-5957.
[18] Q. Zhang, H. Manier and M. -A. Manier, Metaheuristics for job shop scheduling with transportation, *Metaheuristics for Production Scheduling (eds B. Jarboui, P. Siarry and J. Teghem)*, 2013, 465-493.
[19] Z. Zhang, B. Pi and Z. Wang, Scheduling mobile robots in flexible manufacturing system by an adaptive large neighborhood search, *Proceedings of 6th International Conference on Computing and Data Engineering*, 2020, 232-236.
[20] Q. V. Dang and I. Nielsen, Simultaneous scheduling of machines and mobile robots, *Communications in Computer and Information Science*, 365, 2013, 118-128.
[21] D. S. Sanches, J. da Silva Rocha, M. F. Castoldi, O. Morandin and E. R. R. Kato, An adaptive genetic algorithm for production scheduling on manufacturing systems with simultaneous use of machines and AGVs, *Journal of Control Automation and Electrical Systems*, 26(3), 2015, 225-234.
[22] A. Elmi and S. Topaloglu, Cyclic job shop robotic cell scheduling problem: Ant colony optimization, *Computers & Industrial Engineering*, 111, 2017, 417-432.
[23] M. Petrović, N. Vuković, M. Mitić and Z. Miljković, Integration of process planning and scheduling using chaotic particle swarm optimization algorithm, *Expert Systems with Applications*, 64, 2016, 569-588.
[24] W. Q. Zou, Q. K. Pan, and M. F. Tasgetiren, An effective discrete artificial bee colony algorithm for scheduling an automatic-guided-vehicle in a linear manufacturing workshop, *IEEE Access*, 8, 2020, 35063-35076.
[25] U. Bilge and G. Ulusoy, Time window approach to simultaneous scheduling of machines and material handling system in an FMS, *Operations Research*, 43(6), 1995, 1058-1070.
[26] G. Ulusoy and U. Bilge, Simultaneous scheduling of machines and automated guided vehicles, *International Journal of Production Research*, 31(12), 1993, 2857-2873.
[27] L. Deroussi and S. Norre, Simultaneous scheduling of machines and vehicles for the flexible job shop problem solution approach: Basic ideas, *International Conference on Metaheuristics and Nature Inspired Computing*, 2010, 2-3.
[28] J. Hurink and S. Knust, Tabu search algorithms for job-shop problems with a single transport robot, *European Journal of Operational Research*, 162, 2005, 99-111.
[29] Q. Zhang, H. Manier and M. Manier, A modified shifting bottleneck heuristic and disjunctive graph for job shop scheduling problems with transportation constraints, *International Journal of Production Research*, 52(4), 2014, 37-41.
[30] D. Sharma, V. Singh and C. Sharma, GA based scheduling of FMS using roulette wheel selection process, *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS)*, Springer, 2011, 931-940.

[31] I. Nielsen, Q. D. Grzegorz, and B. Zbigniew, A methodology for implementation of mobile robot in adaptive manufacturing environments, *Journal of Intelligent Manufacturing*, 28, 2017, 1171-1188.
[32] I. Nielsen, N. A. Dung Do, Z. A. Banaszak and M. N. Janardhanan, Material supply scheduling in a ubiquitous manufacturing system, *Robotics and Computer-Integrated Manufacturing*, 45, 2017, 21-33.
[33] G. Zhang, J. Sun, X. Liu, G. Wang and Y. Yang, Solving flexible job shop scheduling problems with transportation time based on improved genetic algorithm, *Mathematical Biosciences and Engineering*, 16(3), 2019, 1334-1347.
[34] H. Qiu, W. Zhou and H. Wang, A genetic algorithm-based approach to flexible job-shop scheduling problem, *Fifth International Conference on Natural Computation (ICNC)*, 4, 2009, 81-85.
[35] L. Gao, F. Lu, Y. Ge and D. Feng, A fuzzy adaptive genetic algorithms for global optimization problems, *Chinese Control and Decision Conference (CCDC)*, 2010, 914-919.
[36] A. P. Rifai, S. Z. M. Dawal, A. Zuhdi, H. Aoyama and K. Case, Reentrant FMS scheduling in loop layout with consideration of multi loading-unloading stations and shortcuts, *The International Journal of Advanced Manufacturing Technology,* 82(9-12), 2016, 1527-1545.
[37] N. F. Fauziah and Y. H. Putra, Scheduling regular classrooms using heuristic genetic and tabu search algorithms, *IOP Conference Series: Materials Science and Engineering*, 407, 2018, 012116.
[38] M. S. Umam, M. Mustafid and S. Suryono, A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem, *Journal of King Saud University - Computer and Information Sciences*, 34(9), 2021, 7459-7467.
[39] M. P. A. Sonawane, Hybrid genetic algorithm and tabu search algorithm to solve class time table scheduling problem, *International Journal of Research Studies in Computer Science and Engineering*, 4840(4), 2014, 19-26.
[40] Y. Ge, A. Wang, Z. Zhao, and J. Ye, A tabu-genetic hybrid search algorithm for job-shop scheduling problem, *E3S Web of Conferences*, 95, 2019, 3-6.
[41] A. A. Abayomi-alli, F. O. Uzedu, S. Misra, O. O. Abayomi-alli and O. T. Arogundade, Hybrid Model of genetic algorithms and tabu search memory for nurse scheduling systems, *International Journal of Service Science, Management, Engineering, and Technology*, 13(1), 2022, 1-20.
[42] S. Meeran and M. S. Morshed, A hybrid genetic tabu search algorithm for solving job shop scheduling problems: A case study, *Journal of Intelligent Manufacturing*, 23(4), 2012, 1063-1078.
[43] J. J. Palacios, M. A. González, C. R. Vela, I. González-Rodríguez and J. Puente, Genetic tabu search for the fuzzy flexible job shop problem, *Computers & Operations Research*, 54, 2015, 74-89.
[44] L. Zhang, L. Gao and X. Li, A hybrid genetic algorithm and tabu search for a multi-objective dynamic job shop scheduling problem, *International Journal of Production Research,* 51(12), 2013, 3516-3531.
[45] X. Li and L. Gao, An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem, *International Journal of Production Economics*, 174, 2016, 93-110.
[46] J. Xie, X. Li, L. Gao and L. Gui, A hybrid genetic tabu search algorithm for distributed flexible job shop scheduling problems, *Journal of Manufacturing Systems*, 71, 2023, pp. 82-94.
[47] F. Xhafa, J. A. Gonzalez, K. P. Dahal and A. Abraham, A GA(TS) hybrid algorithm for scheduling in computational grids, *Hybrid Artificial Intelligence Systems,* Springer, Berlin, 2009.
[48] M. Qiu, Z. Fu, R. Eglese and Q. Tang, A tabu search algorithm for the vehicle routing problem with discrete split deliveries and pickups, *Computers & Operations Research*, 100, 2018, 102-116.
[49] Y. Qawqzeh, M. T. Alharbi, A. Jaradat and K. N. A. Sattar, A review of swarm intelligence algorithms deployment for scheduling and optimization in cloud computing environments, *PeerJ Computer Science,* 7, 2021, 1-17.
[50] M. Wan, X. Xu and J. Nan, Flexible job-shop scheduling with integrated genetic algorithm, *Proceedings of 4th International Workshop on Advanced Computational Intelligence (IWACI)*, Wuhan, China, 2011, 13-16.
[51] M. V. S. Kumar, R. Janardhana and C. S. P. Rao, Simultaneous scheduling of machines and vehicles in an FMS environment with alternative routing, *The International Journal of Advanced Manufacturing Technology*, 53, 2011, 339-351.
[52] M. Gen and L. Lin, Multiobjective evolutionary algorithm for manufacturing scheduling problems: State-of-the-art survey, *Journal of Intellingent Manufacturing*, 25(5), 2014, 849-866.
[53] B. S. P. Reddy and C. S. P. Rao, A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS, *The International Journal of Advanced Manufacturing Technology*, 31(5-6), 2006, 602-613.
[54] B. S. P. Reddy and B. S. P. Rao, Flexible manufacturing systems modelling and performance evaluation using automod, *International Journal of Simulation model*, 10(2), 2011, 78-90.
[55] S. Lawrence, *Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques*, Technical Report, Graduate School of Industrial Administration. Carnegie-Mellon University, 1984.