

Applying Job Shop Scheduling to SMEs Manufacturing Platform to Revitalize B2B Relationship

Yeonjee Choi¹, Hyun Suk Hwang² and Chang Soo Kim^{1,*}

¹Department of Information Systems, Pukyong National University, Busan, 608737, Korea

²Research Center, Hanyoung DS, Inc., Seoul, 07600, Korea

* Corresponding Author: Chang Soo Kim. Email: cskim@pknu.ac.kr

Received: 12 August 2022; Accepted: 06 October 2022

Abstract: A small and medium enterprises (SMEs) manufacturing platform aims to perform as a significant revenue to SMEs and vendors by providing scheduling and monitoring capabilities. The optimal job shop scheduling is generated by utilizing the scheduling system of the platform, and a minimum production time, i.e., makespan decides whether the scheduling is optimal or not. This scheduling result allows manufacturers to achieve high productivity, energy savings, and customer satisfaction. Manufacturing in Industry 4.0 requires dynamic, uncertain, complex production environments, and customer-centered services. This paper proposes a novel method for solving the difficulties of the SMEs manufacturing by applying and implementing the job shop scheduling system on a SMEs manufacturing platform. The primary purpose of the SMEs manufacturing platform is to improve the B2B relationship between manufacturing companies and vendors. The platform also serves qualified and satisfactory production opportunities for buyers and producers by meeting two key factors: early delivery date and fulfillment of processing as many orders as possible. The genetic algorithm (GA)-based scheduling method results indicated that the proposed platform enables SME manufacturers to obtain optimized schedules by solving the job shop scheduling problem (JSSP) by comparing with the real-world data from a textile weaving factory in South Korea. The proposed platform will provide producers with an optimal production schedule, introduce new producers to buyers, and eventually foster relationships and mutual economic interests.

Keywords: Manufacturing platform; job shop scheduling problem (JSSP); genetic algorithm; optimization; textile process

1 Introduction

Production scheduling can maximize customer satisfaction by increasing the efficiency of the shop floor and minimizing production time in Industry 4.0. The increased efficiency brings the cost reduction effect due to minimizing labor cost and delivery time [1]. The production scheduling and management of manufacturing systems in Industry 4.0 is based on the digital system. In addition,



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Industry 4.0 has led manufacturers to produce small-batch productions under various types of conditions. This trend requires manufacturers to stock up on different types of resources and to respond flexibly to an environment of dynamic and uncertain working conditions. Hence, large enterprises have started to invest in smart grid systems. Manufacturing plants need significant capital and a research workforce to actualize the systems. It is difficult for SMEs to qualify for the requirements that cannot be set up for real-time scheduling or monitoring. Therefore, SMEs leverage manufacturing software in their factories instead of shifting to smart manufacturing [2–4]. A typical system is enterprise resource planning (ERP) [5]. However, it is unsuitable for job shop schedules as it mainly focuses on resource planning. Resource planning in ERP focuses on when resources are brought into the plant and how to manage their lifecycle to reduce costs on the shop floor. The shop floor scheduling software generally uses manufacturing execution systems (MES), which require IoT devices or sensors to transmit real-time data [6]. Job shop scheduling is essential in manufacturing for plant efficiency and economic activation. The SMEs' limitations in manufacturing automation may also have side effects in that they can struggle to attract customers [7,8]. The current method of connecting SMEs as producers and companies as buyers are to contact existing buyers by fax, email, phone, or introduce them through acquaintances. In this sense, new or unknown producers may experience severe financial difficulties and be restricted in promoting their new products.

To be specific, in the case of textile manufacturing, this industrial transformation hugely impacts the industry. The textile industry in South Korea heavily depends on SMEs. According to the National Statistical Office in Korea, 99.97% of the textile garment industry was owned by SMEs in 2018. The large companies occupy the remaining percentage. At the same time, this trend has continued until recently. Without the manufacturing software, the scheduling tasks are entirely up to workers, that their scheduling is not as efficient as the software can schedule. In this sense, SMEs are suffering from severe economic difficulties. A solution should be proposed to overcome the limitations of the economic difficulties of SMEs mentioned above. Therefore, this paper proposes a method to solve SMEs' limitations in economic difficulties and overcome the disadvantages of digital transformation. This will bridge the gap between large companies and SMEs by minimizing the flow of orders in textile manufacturing.

In this sense, in the study, we used textile manufacturing SMEs data in Korea to test the genetic algorithm (GA) model to solve the job shop scheduling problem (JSSP) and obtain the optimal scheduling result. GA is a heuristic searching algorithm inspired by natural selection and genetic nature [9]. In addition, it is a meta-heuristic algorithm adaptable for real-world problems based on its nature-inspired features [10]. This algorithm has high performance in finding the optimal solution to various problems. Similar to GA, there are several optimization algorithms to solve a non-deterministic polynomial problem (NP)-hard. Tang et al. [11] used hybrid discrete particle swarm optimization integrated with simulated annealing (HDPSO-SA) to generate improved initial population and global searching ability. This method has a high problem-solving ability in various benchmark problems of flexible job shop scheduling (FJSP). Peng et al. [12] proposed a hybrid PSO algorithm to overcome the limitations of conventional PSO that cannot escape from local optima. Pongchairerks [13] established a two-level metaheuristic algorithm to solve the JSSP with multipurpose machines. Wang et al. [14] used a hybrid gray wolf weed algorithm (GIWO) to obtain a minimal makespan by solving the flexible job shop scheduling problem. The GIWO first initializes the population and apply an invasive weed optimization algorithm to improve the local search ability of the gray wolf algorithm. Even though the PSO, GIWO, and other optimization algorithms can solve job shop scheduling problems and achieve the minimum makespan, they are desirable for continuous optimization [15]. Since the platform utilizes real-world data from textile manufacturing, it is suitable for discrete optimization.

Although GA is optimal for solving the JSSP in the platform, it takes time to schedule with massive JSSP instances. Various methods have been studied recently to overcome the shortcomings of GA. A representative method is to apply a machine learning-based method [16]. Machine learning-based methods can solve JSSP at a fast turnaround speed with a large dataset. However, as the data we used are based on the actual production, the production proceeds on an average of 10 to 20 machines. Hence, the platform adopted GA model for solving the JSSP and compared its scheduling results with the manual scheduling of the actual manufacturing to evaluate its feasibility on the platform. Experiments for the GA-based JSSP in the platform were designed to determine the minimum makespan of a scheduling result. The proposed method was compared with field experts scheduling that they manually scheduled in a real-world weaving factory in Korea to test its feasibility. Therefore, for the digitization of manufacturing and customer satisfaction, the platform minimizes production time with an optimal schedule and allows buyers to make final decisions by selecting producers based on their criteria.

In this study, a scheduling system for the SMEs manufacturing platform is proposed and is expected to provide users with fast and efficient scheduling ability and reflect the real-world practice of the existing field. The job shop scheduling engine has an algorithm for solving the JSSP to get the optimized scheduling result. The rest of the paper is presented as follows. Section 2 introduces knowledge of the manufacturing platform and JSSP. Section 3 briefly introduces the JSSP model of the platform system. Section 4 simulates the JSSP model of the platform with the data of textile manufacturing owned by SMEs in South Korea and describes the implementation of the platform. Finally, Section 5 presents the conclusion of the paper.

2 Related Research

2.1 Manufacturing Platform

The manufacturing platforms aim to satisfy the needs of the shop floor. This manufacturing platform can reduce production costs, detect anomalies, schedule production plans, or manage production processes in real-time [17]. Platforms generally allow users to customize, extend, and easily access system services. The emerging manufacturing platform is a cloud-based platform classified as a service-oriented manufacturing model whose primary purpose is to cooperate or share resources [18]. Liu et al. [19] utilized the industrial internet of things (IoT) technologies in a cloud manufacturing system to operate the core functions of a manufacturing system. The system is designed to provide high efficiency in the decision-making processes of the manufacturing. However, the SMEs cannot adopt the cloud manufacturing system because of its limitation in launching digital transformation. Digital transformation includes physical changes on the shop floor and virtual elements of the manufacturing systems [20]. To initiate these changes, producers must be able to install sensors or IoT devices on the shop floor to collect or control real-time data of the manufacturing automatically. Regarding the limitations in the innovation of SMEs, Andalib et al. [21] proposed a blockchain-based manufacturing platform for SMEs in Bangladesh to save innovation costs. Han et al. [22] implemented a data science platform to provide collaboration opportunities for SMEs. Data science platforms are expected to achieve data mining, analysis, sharing, and pre-processing. Along with the concept of a service-oriented manufacturing platform and low-priced solutions, we design and implement the novel concept of a manufacturing service platform in this paper.

2.2 Job Shop Scheduling Problem

Numerous kinds of research have been conducted to solve the problem of shop floor scheduling, also known as the job shop scheduling problem (JSSP). The JSSP is proved to be the NP-hard problem [23]. Solving the NP-hard problem is to find a way to allocate n jobs on m machines efficiently. Each job J_i is divided into O_{ij} operations, and O_{ij} are assigned to the machine M_{ij} with processing time $Time_{ij}$, as shown in Table 1. The well-planned schedule achieves the minimum makespan compared to other schedule results. The scheduling results are represented in Gantt chart format as shown in Fig. 1. Optimal scheduling between tasks and machines contributes to efficient production by minimizing the waste of resources and production time in factories. The JSSP is based on the static environment. The process of each job has conditions that it can be performed on specific machines, and one job cannot be performed on multiple machines at the same time.

Table 1: 3×5 JSSP example

	O_{i1}	$Time_{i1}$	O_{i2}	$Time_{i2}$	O_{i3}	$Time_{i3}$	O_{i4}	$Time_{i4}$	O_{i5}	$Time_{i5}$
Job 1	1	31	0	24	2	50	3	86	4	72
Job 2	2	87	1	98	3	46	0	61	4	78
Job 3	4	43	3	75	2	20	1	32	0	53

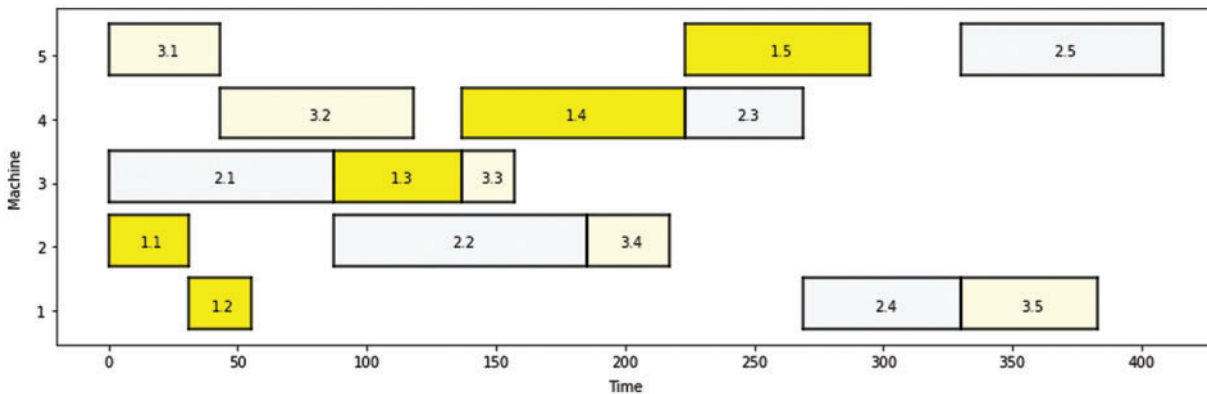


Figure 1: The corresponding solution for 3×5 JSSP solved with GA in Gantt chart representation

This aspect of the JSSP has some limitations on the existing field systems [24]. The GA can be applied to various tasks or problems to function as an optimizer. Omar et al. [25] utilized stopping criteria in GA to solve the JSSP. The proposed GA includes a critical block neighborhood and distance measuring method to stop the crossover and mutation step iteration. Although these stopping criteria are used to reduce the running time of the GA, the data used in the study are small and, therefore, unsuitable for large-scale problems. The GA first randomly generates a population of chromosomes. The generated chromosomes are evaluated for the effectiveness of the solution. The next chromosome is then generated by selecting and combining the original solution to compare the original solution with the newly generated solution. Generating the new generation requires the operation of three leading operators; reproduction (or selection), crossover, and mutation. The reproduction makes copies of improved strings for generating a new population. To make this platform optimal for textile

manufacturing, we applied the GA to solve the JSSP in the proposed system and set the deadline constraints to determine possible production schedules for each producer.

2.3 Scheduling in Textile Manufacturing

Scheduling textile manufacturing has been studied in numerous studies, and the textile processes used are different. Saydam et al. [26] focused on the dyeing process to efficiently schedule by maximizing the utilization of machines. Guo et al. [27] used genetic optimization and mixed integer programming to solve the scheduling problem for apparel manufacturing. Mourtos et al. [28] proposed the scheduling method for the weaving process in textile manufacturing by splitting the jobs and setup the resource constraints. These resource constraints limit machines from processing all of them simultaneously because of the limited number of workers on the shop floor. Wang et al. [29] utilized GA for generating an automatic scheduling method to minimize downtime and gaiting load to obtain efficiency in weaving enterprises. Perret et al. [30] proposed a two-stage algorithm using tabu search and job-wise shift operator to utilize it in the mass customized production processes and decrease the production cost. These approaches were made to minimize the production makespan and benefit the manufacturer by lowering the production cost, energy, and workforce. In this sense, scheduling the textile manufacturing process to minimize production costs and the energy consumption is vital for SMEs. This approach will be beneficial not only for textile manufacturing SMEs but it will be significantly impact general manufacturing SMEs. In this paper, the test data focuses on the weaving process of a small textile manufacturing company because of the difficulty of actual data collection.

3 SMEs Manufacturing Platform Design

The SMEs manufacturing platform has four operation modules: user interface, platform server, database, and job shop scheduling engine. The users of the platform are categorized as buyers and producers. In general, the buyers refer to companies that merchandise finalized products with a direct market connection—for example, the fashion companies that sell knit, dress, or other apparel. The producer is a manufacturer or vendor company that produces resources for producing finished products. In this sense, depending on the user type, the user interface of the platform serves two different purposes. First, the user interface gives a buyer with notification and monitoring services. The notification service provides buyers with scheduling results by utilizing the notification function of the platform. The process monitoring service allows buyers to track the production status of their orders in real time. Second, the user interface provides producers with notification, scheduling, and scheduled processing monitoring services. The notification service allows producers to notify buyers with real-time confirmed order results and production progress information. The scheduling service will connect producers to operate the job shop scheduling engine for scheduling execution. Finally, the scheduled processing monitoring service is a service that shows the history and progress of the production information.

The job shop scheduling engine operates as a module that handles scheduling-related tasks such as scheduling execution, scheduling history monitoring, and scheduling result or historical data storage. The structure of the proposed SMEs manufacturing platform is depicted in Fig. 2. The database of the platform has five tables: order, company, schedule, product, and buyer. First, the producer table collects data from producers, which are company name, location, credibility, contact number, and email. Second, the products table collects product name, density, and rpm. Third, the schedule table collects schedule id, order id, product id, company id, work start date, and work end date. Fourth, the

order table collects the order id, buyer id, expected delivery date, and producer id. Finally, the buyer table includes the buyer's name and location.

All modules are linked to each other and have a high dependency. The primary functions of the platform are order management, product management, company management, facility management, scheduling, and evaluation. The overall operation model of the platform is as follows.

Step 1: Buyers register their orders on the platform through the order module. The buyers must also provide their expected delivery date for the scheduling process. After the buyers execute the order process, producers check the order lists to activate the scheduling system.

Step 2: Producers obtained the scheduling results from the scheduling module and estimated production delivery dates. The producers notify the buyers with a notification function.

Step 3: When more than one producer suggests the possibility of processing their orders, the buyers select the production company to progress the production.

Step 4: The producers confirm the confirmed orders and the schedule to start the production process.

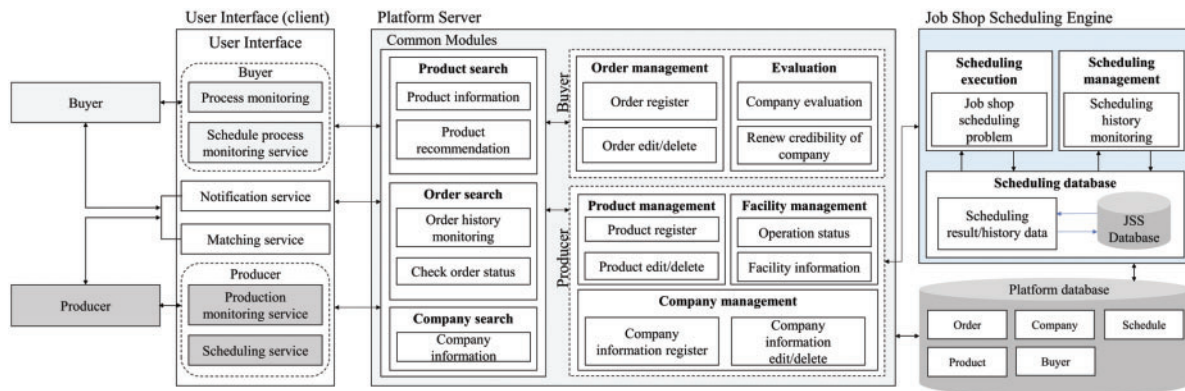


Figure 2: The structure of the proposed SMEs manufacturing platform

4 Simulation and Implementation

4.1 Applying GA to Weaving Process of Textile Manufacturing

In this paper, the weaving process data of SME-owned manufacturing are used to test the GA model. There are three main operations in textile manufacturing: weaving and knitting, dyeing, and sewing and post-processing. First, the weaving process produces simple fabrics for clothes or shoes. This simple fabric results from various patterns and designs that require a high level of technology. Depending on the type of weaving, the processing time may be longer than other designs for the finest and sheer fabric quality. Second, the dyeing process dyes the fabrics. This process requires diverse temperature settings that require more energy than other processes. Finally, the sewing and post-processing finalize the textile manufacturing. On the other hand, if the product the buyer wants to order is fabric, sewing and post-processing are not always required—only apparel goods are required to proceed with the process. Since the weaving process is longer than other processes, efficient scheduling is required. Generally, the weaving process takes more than three months on one machine. In some cases, it lasts more than six months. These practices of the weaving process can delay the production time of manufacturing. According to a condition of JSSP, the operation

should be declared before executing the scheduling. However, as the study focuses on solving real-world problems, textile manufacturing does not always cover all three processes on one production site. Thus, parallel operation of the same job on different machines is allowed in this weaving process scheduling.

The proposed approach is tested by scheduling examples from the SME-owned weaving factory. The raw data were collected for one year, from January 1, 2020, to December 31, 2020. The total data are 62,452 records, and we used 11% of the total data in this study. There are three conditions in data selection. First, the data selection was based on machine numbers from 1 to 20. Second, among the selected machines, the jobs need to have lasted more than three days. Third, the chosen machines have produced two or more products on each machine. Utilizing the inputs for the GA model, the raw data are required to be preprocessed. There are seven columns in raw data: work date Wd , machine name, work groups Wg , product name p , density Den_p , rpm Rpm , and demand. Each product has its Den_p value. In addition, depending on p , Wd , Wg , and machine, Rpm is different from each other by a difference of ± 3 . Hence, we calculate the mean value of Rpm for input data generation. Preprocessing the raw data for GA model input requires a name/information of each product $Prod_i$, machine number m , density of $Prod_i$ Den_{Prod_i} , mean of rpm of $Prod_i$ $\overline{Rpm_{Prod_i}}$, and processing time $PT_{T_{ij}}$ data. The notations of input data for JSSP are defined in Table 2.

Table 2: Notations and related definitions for JSSP instances

Notations	Definition
n	Number of jobs
m	Number of machines
pt	Processing time
$prod_i$	Product list from raw data
$Prod_i$	i^{th} product information/name
T_{ij}	j^{th} task for $Prod_i$
Abs	Rate of absenteeism (set as 5%)
Den_{Prod_i}	Density of $Prod_i$
n_e	Number of employees in manufacturing (set as 40)
Rpm_{Prod_i}	rpm of $Prod_i$
$\overline{Rpm_{Prod_i}}$	Mean value of rpm
$PT_{T_{ij}}$	Processing time of T_{ij}
St_{Prod_i}	Work start date of order $Prod_i$
Dd_{Prod_i}	Work due date for order $Prod_i$

There are three steps in raw data preprocessing. First, the number of products is divided based on the input data m set by the producer. Second, the processing time is the primary input data for GA and is generated through Algorithm 1. After dividing the J_n , assigning the processing time of each M_m is essential. The division process is defined and described in Algorithm 1 lines 19. Each divided task T_{ij} has different processing time values depending on the total demand of the orders. The processing time $PT_{T_{ij}}$ of the j^{th} task for a $Prod_i$ is calculated as Algorithm 20 line 24. Calculating the $PT_{T_{ij}}$ requires average rpm. Through the function in Algorithm 1 lines 1 to 7, the value of $\overline{Rpm_{Prod_i}}$ can be obtained. The data generation function calculates the processing time based on the history data set of each

product $prod_l$, machine list ml , demand of the order, and work start date St_{prod_l} . The number of orders can be listed in each order from one to infinite. Therefore, each order has different $prod_l$, ml , Den_{prod_l} , \overline{Rpm}_{prod_l} , demand, and St_{prod_l} . In this sense, it is important to compute the $PT_{T_{ij}}$ and divide T_{ij} based on the m . Moreover, since the weaving process has only one operation, we defined the operation in JSSP based on the real dataset by allocating jobs J_n on M_m , as shown in Table 3.

Algorithm 1: Psuedo-code for generating processing time for JSSP

```

Input  $prod_l$ : history data list of each product,  $ml$ : machine list, demand: demand of the order,
 $St_{prod_l}$ : work start date of product  $Prod_l$ 
Result  $L_i \leftarrow$  processing time and machine allocation data
1: def average_rpm( $prod_l$ ):
2:  $Rpm \leftarrow$  (empty value)
3:  $tot\_Rpm \leftarrow$  (empty value)
4: for  $i = 0$  to  $len(prod_l)$  do
5:  $tot\_Rpm += prod_l[i].rpm$ 
6:  $Rpm = tot\_Rpm // len(prod_l)$ 
7: return  $Rpm$ 
8: def generate_data ( $prod_l, ml, demand, St_{prod_l}$ ):
9: ( $L_i$ )  $\leftarrow$  (empty list)
10: ( $Dict(m, pt)$ )  $\leftarrow$  (empty dictionary)
11: ( $n_e$ )  $\leftarrow$  (40)
12: ( $Abs$ )  $\leftarrow$  (0.95)
13:  $\overline{Rpm}_{prod_l} = average\_rpm(prod_l)$ 
14:  $Den_{prod_l} = prod_l[0].density$ 
15:  $PT\_temp = demand // (\overline{Rpm}_{prod_l} * n_e * Abs * // Den_{prod_l})$ 
16:  $PT\_temp\_rest = demand \% (\overline{Rpm}_{prod_l} * n_e * Abs * // Den_{prod_l})$ 
17:  $Dict[work\_str\_date] = St_{prod_l}$ 
18: for  $i=0$  to  $len(ml)$  do
19:  $Dict[m] = ml[i]$ 
20: if  $PT\_temp\_rest \neq 0$  and  $i==0$ :
21:  $PT_{T_{ij}}\_rest = PT\_temp\_rest$ 
22:  $Dict[pt] = PT_{T_{ij}} + PT_{T_{ij}}\_rest$ 
23: else:
24:  $Dict[pt] = PT_{T_{ij}}$ 
25:  $L_i.insert(Dict)$ 
26: return  $L_i$ 
    
```

Table 3: The textile manufacturing weaving data description

Product name	Buyer		Producer				
	Demand	Delivery date	Den_{Prod_k}	\overline{Rpm}_{Prod_k}	Processing time/day	Available machine	Work start date
6CL-FJ	34,686	2020/08/01	49	555	27,360(19)	M2, M3, M6, M8, M14	2020/07/01

(Continued)

Table 3: Continued

Product name	Buyer		Producer				
	Demand	Delivery date	Den_{Prod_k}	Rpm_{Prod_k}	Processing time/day	Available machine	Work start date
6KOS-FD2	23,605	2020/07/11	67	560	86,400(60)	M2, M8, M13	2020/05/10
6NF-1493-1	364,351	2020/09/06	76	554	168,480(117)	M2, M4, M9, M10, M12, M14, M15, M16, M17, M18, M20	2020/05/10
ECO-SL157	220,455	2020/08/24	176	516	74,880(52)	M1, M2, M5, M10	2020/07/01
ES FD TSL R/S	49,406	2020/08/26	60	509	77,760(54)	M2, M4, M10, M18	2020/07/01
FBR 0505	33,447	2020/07/10	183	507	8,640(6)	M2, M18	2020/07/01
FD 210T R/S REC	136,268	2020/08/17	81	544	69,120(48)	M1, M2, M3, M5, M7, M9, M11, M14, M15, M19, M20	2020/05/10

The final step of the raw data preprocessing is to save a list of input data into the .csv file. After this process, the GA model is ready for the execution process. In the execution process of the GA model, setting up the three values is important. The three inputs are population size, iteration count, and mutation rate. The initiation of the model begins with setting up the population size to generate a set of chromosomes so that the first operation of the algorithm, selection, can be used. Iterations are initiated depending on the population size, and each iteration randomly generates a set of chromosomes within the range multiplied by m and J_n . These randomly generated chromosomes are repeated within the *iteration count* until an optimal solution is found. However, when a buyer provides the work end date in the platform, the scheduling iteration of GA terminates when the target value of the work end date has been met. Finally, the mutation operator randomly generates new information in the children's generation of the chromosome; some are inherited from the parent generation. The mutation function first swaps the two elements of the parent generation with the next generation. Then it randomly selects an element and sets it to its children's generation. This mutation operator is expected to diversify the population and is beneficial for local search. Operating the mutation function requires mutation probability, which defines the size of the parts that can be changed during a mutation function. During the selection, the calculation of makespan computes a fitness value to rank the best solution. The population stacks the next best solution during the three main processes of algorithms by calculating the makespan of each population.

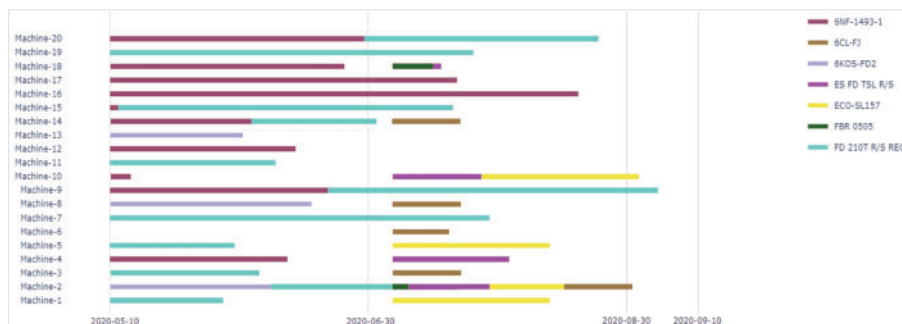
Then the producer can execute the GA model. According to the Table 3, seven products are grouped as Order 1 to generate the job order and processing timetable, which are required to execute the scheduling model of the platform. Depending on each J_n , there are restrictions on machine allocations due to beam and textile types. In this sense, not all J_n are evenly distributed to every M in this study. The job shop scheduling engine execution process requires input m , St_{Prod_i} , and $Prod_i$. In addition, Dd_{Prod_i} is required to schedule execution process to determine what job can be done within the buyer's expected Dd_{Prod_i} . As the scheduling execution returns result of makespan and Gantt chart data, the parameter of the result includes x-axis as processing time, y-axis as machine number, producer P , St_{Prod_i} , Dd_{Prod_i} , and J_n .

In this paper, the data are divided into individual orders according to the average production cycle (three months/six months) of each product and 20 machines from the actual data. We test the feasibility

of the GA model in textile manufacturing by comparing manual scheduling from field experts in the weaving process in textile manufacturing owned by SMEs, i.e., manual scheduling. We used Intel Core i-7 with 32 GB memory at a speed of 2.9 GHz and Python 3.9 to solve the JSSP and run the platform. Since the products are divided into multiple J_n , the blank area of each machine can process other products or be empty to prepare for the next J_n . This aspect is also adopted in the GA model in this paper. Table 4 indicates the processing time of each order. Fig. 3 is the scheduling result in Gantt chart format. Fig. 3a is the Order 1 data that utilizes 40 jobs and 20 machines to schedule. The work start date starts on May 10, 2020 and ends on September 6, 2020. Fig. 3b is the Order 2 data that utilizes 38 jobs and 13 machines to schedule. Fig. 3c is the Order 3 data that utilizes 27 jobs and 8 machines to schedule. Orders 1, 2, and 3 overlap each other. The scheduling assign tasks to not to affect the existing Orders by saving the former schedule in the job shop scheduling engine database and setting up the distinct work start dates. Finally, Fig. 3d is the Order 4 data that utilizes 63 jobs and 19 machines to schedule.

Table 4: The comparative results of GA and manual scheduling for JSSP resolution model

Orders	n	m	GA (days)	Manual scheduling (days)
Order 1	40	20	152	155
Order 2	38	13	174	190
Order 3	27	8	136	135
Order 4	63	19	118	136



(a) Order 1 data ($n 40 \times m 20$)



(b) Order 2 data ($n 38 \times m 13$)

Figure 3: (Continued)

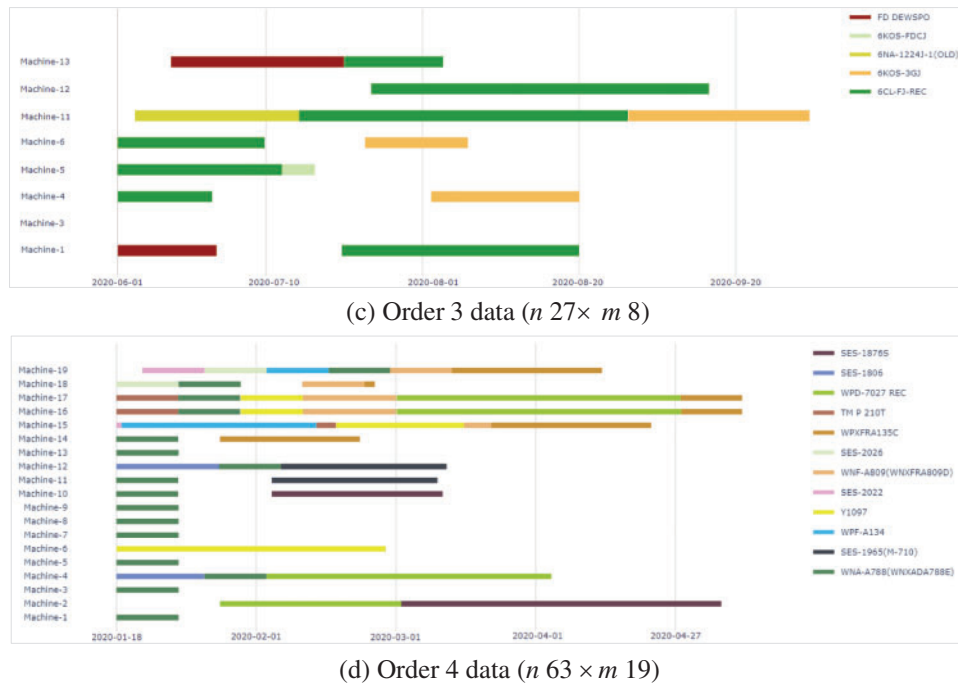


Figure 3: The scheduling result in Gantt chart using GA based JSSP model

As shown in Fig. 4, the comparative results between the GA model and manual scheduling are similar, but the values are decreased. Order 1 has a smaller makespan of 152 than 155 of manual scheduling. However, according to Table 4, GA is not always having a minimum makespan compared to the manual scheduling. The manual scheduling for Order 3 is smaller at a makespan of 135 than for the GA model of 136. This result was obtained because the jobs were evenly assigned to 8 machines for three months. Although there is a difference of 1-day, it is an acceptable period that does not significantly impact the overall end date of orders. Therefore, the use of the GA in this platform is feasible since the job shop scheduling engine aims to minimize the machine usage per order and obtain optimal scheduling. In addition, in this paper, the condition that other products do not preempt the operation sequence of each product was defined in advance, including the machine usage for the textile manufacturing-based JSSP. This feature is closely related to the use of GA. It is crucial to rearrange the population of textile manufacturing JSSP instance in order of each product through GA. Since the GA has a fitness value to rank the best solution between the randomly generated populations, the textile manufacturing JSSP instances will be well suited for obtaining the best solution regarding the precondition of this paper. Moreover, the scheduling system takes 2 s to schedule productions on average. This speed in production scheduling will save workers time and labor costs in scheduling jobs.

4.2 The Industrial Application and Implementation

The job shop scheduling execution is desired to get the optimized schedule for the production plan. Optimized schedules are based on the shortest makespan. Producers are expected to be able to run as many jobs as possible. When the scheduling engine generates the makespan, producers select the jobs that end within expected due dates. The job shop scheduling execution process requires several input data to be prepared for its algorithm. We used python language and Django Web application framework for scheduling and matching systems in this platform. Fig. 5a is the screen that shows how

producers set their input data to the job shop scheduling engine. The input data required by the JSSP algorithms is processing time and machine allocation settings. In addition, the input data includes the start date and end date of the job as the criteria by which the job can be processed. Fig. 5b shows how the scheduling results came out after executing the scheduling function with the input data for the weaving process. If the jobs are deployed before the expected end date set by the producer, it will appear on the right side of the screen, so the producer can send a notification to the user to check what producers are available.

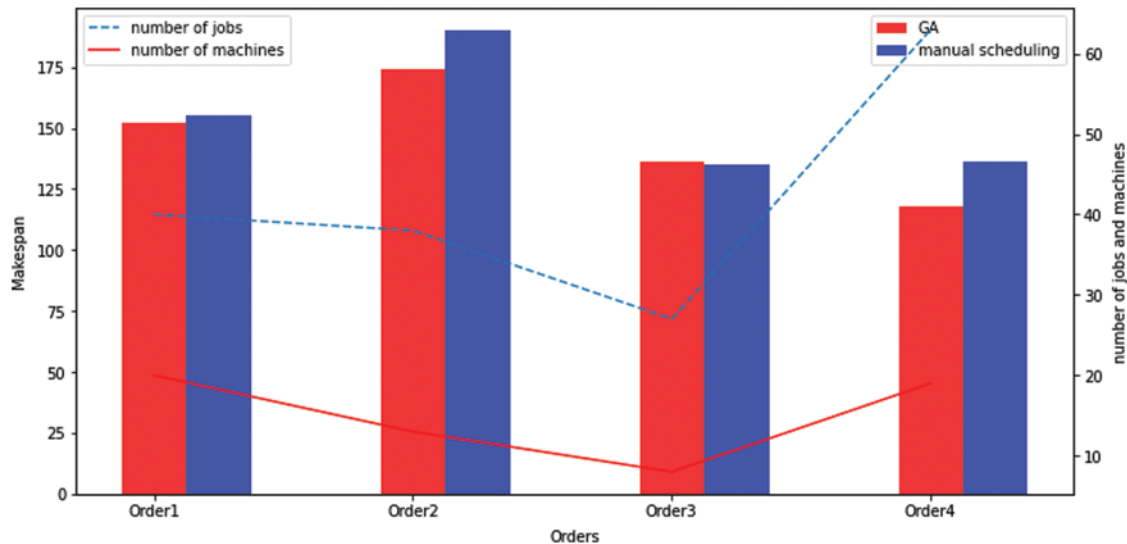
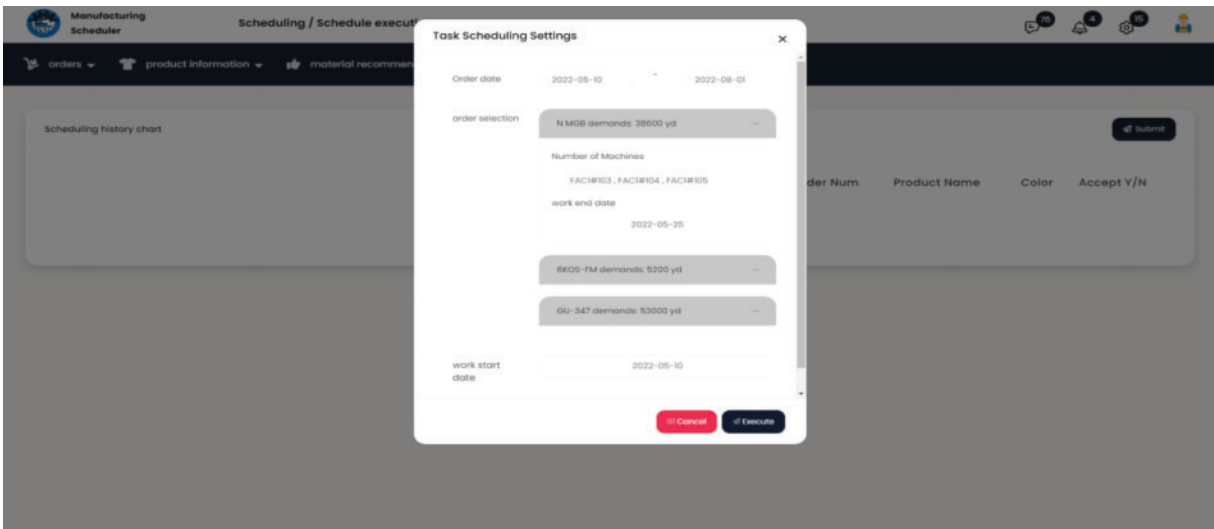
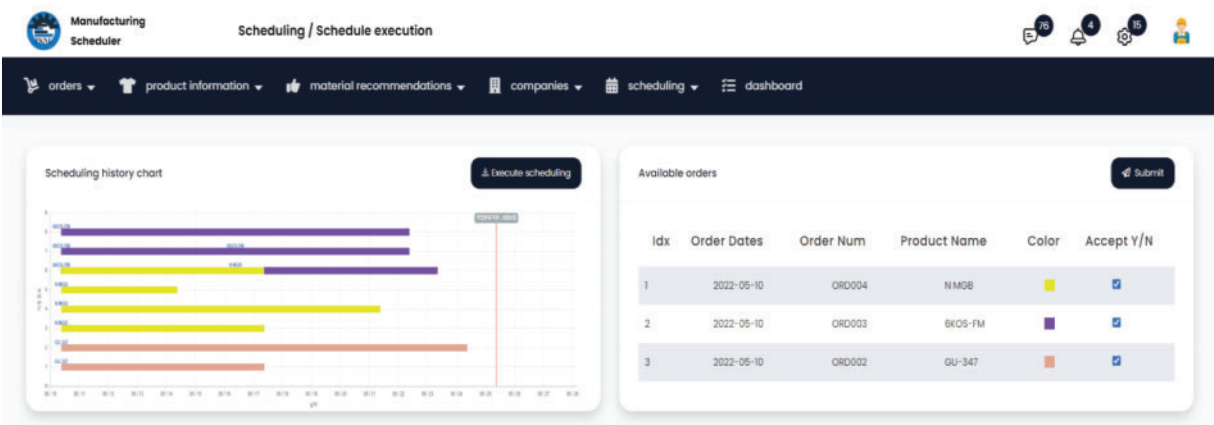


Figure 4: The comparative results of GA and manual scheduling of Order 1–4

The JSSP algorithm in the platform is based on the GA model. Shao et al. [16] proposed self-supervised long-short term memory (SS-LSTM) to minimize the makespan by predicting the following action with high feature extraction of hidden patterns. The method used GA to generate the optimized solution for SS-LSTM and test datasets. This method outperformed when it was tested with the large-scale JSSP instances. Chang et al. [31] also utilized the GA for generating a hybrid genetic algorithm (HGA) for solving the distributed and flexible job-shop scheduling problem (DFJSP) in manufacturing. This proposed method used benchmark data of various manufacturing and obtained its feasibility by comparing the scheduling results with the previous studies that used the same benchmark manufacturing data. Liu et al. [32] proposed a refined genetic algorithm to solve the problem of the traditional GA algorithm and generate an improved encoding method to reflect the real-world environment fully. This newly developed encoding method is utilized to diversify chromosomes in GA, and the scheduled results were close to real fastener manufacturing systems in Taiwan. Although these GA models have been tested for feasibility in real-world manufacturing, they are limited in practical usage. Unlike the studies mentioned above, the proposed platform in this paper tests its feasibility in the weaving process of SMEs textile manufacturing and is also implemented as a platform that buyers and producers can use. Therefore, since the platform includes real-world data from actual manufacturing and tests the feasibility of the GA model based on it, it has significance in directly applied to real-world manufacturing. Table 5 compares related studies with the scheduling and utilization methods of this paper.



(a) Setting screen for job shop scheduling execution



(b) Job shop scheduling result screen for weaving process

Figure 5: Job shop scheduling implementation

Table 5: Related research work

Types	Articles	Optimization objectives	Algorithm
JSSP	[16]	Makespan, high feature extraction of hidden pattern during scheduling using auto-encoder	Self-supervised long-short term memory with GA
FJSP	[31]	Makespan, distributed manufacturing environment	Hybrid GA algorithm
DFJSP	[32]	Makespan, real-world data encoding	Refined GA algorithm
Proposed system		Makespan, delivery date	Proposal of framework using GA algorithm

5 Conclusion

In this paper, we present a novel method of manufacturing platform by enabling the scheduling process. The advantages of our system are to help producers to make faster and optimal decisions for the production schedule by executing the job shop scheduling engine and broadening the opportunity to attract buyers. Buyers can have a chance to discover new SMEs and obtain reasonable prices from the producers. Moreover, this order management, product information management, and dashboard menu allow users to track their orders and manage their production site conditions in real time. This paper uses the GA to solve the JSSP to obtain the optimal schedule. To test the feasibility of the JSSP model, we used textile manufacturing data, which is focused on the weaving process. The raw data are preprocessed for input setup by utilizing the weaving process to the GA model of JSSP. The procedure for executing the JSSP model is as follows. First, calculate the total processing time based on the features of the product. Second, separate jobs based on the number of machines and processing time. Third, make a job order table and processing timetable to operate the GA. Fourth, design and implement the system. The platform design aims to connect buyers and producers in one place to obtain the benefit of short delivery time, efficient scheduling, increased opportunities to discover new producers and buyers, and fast response. Finally, display the result with a Gantt chart. However, in the future, the automation of the job shop scheduling engine will predict the optimal rpm value that will affect processing time, as the current way of executing the job shop scheduling engine requires producers to input the rpm value for each product manually. Therefore, we will implement advanced algorithms using deep learning methods to automate the system and improve its applicability to real-world manufacturing factories.

Funding Statement: This work was supported by the Technology Innovation Program 20004205 (the development of smart collaboration manufacturing innovation service platform in the textile industry by producer-buyer) funded by MOTIE, Korea.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] S. K. Park, "A case study on production scheduling for a job-shop-type automobile parts manufacturing system," M.S. dissertation, Hanyang University, Korea, 2016.
- [2] M. Haddara and O. Zach, "ERP systems in SMEs: A literature review," in *Proc. 2011 44th Hawaii Int. Conf. on System Sciences*, Washington, DC, USA, pp. 1–10, 2011.
- [3] P. Kilimis, W. Zou, M. Lehmann and U. Berger, "A survey on digitalization for SMEs in Brandenburg, Germany," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2140–2145, 2019.
- [4] M. Gabriel and E. Pessl, "Industry 4.0 and sustainability impacts: Critical discussion of sustainability aspects with a special focus on future of work and ecological consequences," *Annals of the Faculty of Engineering Hunedoara*, vol. 14, no. 2, pp. 131–136, 2016.
- [5] O. Zach, B. E. Munkvold and D. H. Olsen, "ERP system implementation in SMEs: Exploring the influences of the SME context," *Enterprise Information Systems*, vol. 8, no. 2, pp. 309–335, 2014.
- [6] B. S. De Ugarte, A. Artiba and R. Pellerin, "Manufacturing execution system—A literature review," *Production Planning and Control*, vol. 20, no. 6, pp. 525–539, 2009.
- [7] H. Reijonen and T. Laukkanen, "Customer relationship oriented marketing practices in SMEs," *Marketing Intelligence & Planning*, vol. 28, no. 2, pp. 115–136, 2010.

- [8] Q. Zhang, M. A. Vonderembse and M. Cao, "Product concept and prototype flexibility in manufacturing: Implications for customer satisfaction," *European Journal of Operational Research*, vol. 194, no. 1, pp. 143–154, 2009.
- [9] Z. Wu, L. Zhang, W. Wang and K. Wang, "Research on job-shop scheduling problem based on genetic ant colony algorithms," in *Proc. 2009 Int. Conf. on Computational Intelligence and Security*, Washington, DC, USA, pp. 114–118, 2009.
- [10] M. M. Eid, E. S. M. El-kenawy and A. Ibrahim, "A binary sine cosine-modified whale optimization algorithm for feature selection," in *Proc. 2021 National Computing Colleges Conf. (NCCC)*, Taif, Saudi Arabia, pp. 1–6, 2021.
- [11] H. Tang, R. Chen, Y. Li, Z. Peng, S. Guo *et al.*, "Flexible job-shop scheduling with tolerated time interval and limited starting time interval based on hybrid discrete PSO-SA: An application from a casting workshop," *Applied Soft Computing*, vol. 78, pp. 176–194, 2019.
- [12] X. Peng, X. Sun and Z. He, "A hybrid particle swarm optimizer for curriculum sequencing problem," *Discrete Dynamics in Nature and Society*, vol. 2022, no. 4, pp. 1–10, 2022.
- [13] P. Pongchairerks, "A two-level metaheuristic for the job-shop scheduling problem with multipurpose machines," *Complexity*, vol. 2022, no. 5, pp. 1–17, 2022.
- [14] Y. Wang, Y. Song, Y. Zou, Q. Lei and X. Wang, "A hybrid gray wolf weed algorithm for flexible job-shop scheduling problem," *Journal of Physics: Conference Series*, vol. 1828, no. 1, pp. 012162, 2021.
- [15] V. Kachitvichyanukul, "Comparison of three evolutionary algorithms: GA, PSO, and DE," *Industrial Engineering & Management Systems*, vol. 11, no. 3, pp. 215–223, 2012.
- [16] X. Shao and C. S. Kim, "Self-supervised long-short term memory network for solving complex job shop scheduling problem," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 8, pp. 2993–3031, 2021.
- [17] Y. C. Lin, M. H. Hung, H. C. Huang, C. C. Chen, H. C. Yang *et al.*, "Development of advanced manufacturing cloud of things (AMCoT)-a smart manufacturing platform," *IEEE Robot and Automation Letters*, vol. 2, no. 3, pp. 1809–1816, 2017.
- [18] B. Huang, C. Li, C. Yin and X. Zhao, "Cloud manufacturing service platform for small- and medium-sized enterprises," *The International Journal of Advanced Manufacturing Technology*, vol. 65, no. 9, pp. 1261–1272, 2013.
- [19] C. Liu, Z. Su, X. Xu and Y. Lu, "Service-oriented industrial internet of things gateway for cloud manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 73, pp. 102217, 2022.
- [20] T. Borangiu, D. Trentesaux, A. Thomas, P. Leitao and J. Barata, "Digital transformation of manufacturing through cloud services and resource virtualization," *Computers in Industry*, vol. 108, no. 65, pp. 150–162, 2019.
- [21] T. W. Andalib, N. A. Azizan, J. A. Turi, N. Danilla and Z. K. Kaman, "A frugal innovation model assimilating block-chain technology in the manufacturing SMEs," *Journal of Positive School Psychology*, vol. 6, no. 2, pp. 1348–1362, 2022.
- [22] H. Han and S. Trimi, "Towards a data science platform for improving SME collaboration through industry 4.0 technologies," *Technological Forecasting and Social Change*, vol. 174, pp. 121242, 2022.
- [23] J. Blazewicz, W. Domschke and E. Pesch, "The job shop scheduling problem: Conventional and new solution techniques," *European Journal of Operational Research*, vol. 93, no. 1, pp. 1–33, 1996.
- [24] H. Yamashiro and H. Nonaka, "Estimation of processing time using machine learning and real factory data for optimization of parallel machine scheduling problem," *Operations Research Perspectives*, vol. 8, no. 4, pp. 100196, 2021.
- [25] M. Omar, A. Baharum and Y. Hasan, "A job-shop scheduling problem (JSSP) using genetic algorithm (GA)," in *Proc. 2nd IMT-GT Regional Conf. on Mathematics, Statistics, and Applications*, Penang, Malaysia, pp. 13–15, 2006.
- [26] C. Saydam and W. D. Cooper, "Dye machine scheduling and roll selection," *Production and Inventory Management Journal*, vol. 36, no. 4, pp. 67–70, 1995.

- [27] Z. X. Guo, W. K. Wong, S. Y. S. Leung, J. T. Fan and S. F. Chan, "Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: A case study based on the apparel industry," *Computers & Industrial Engineering*, vol. 50, no. 3, pp. 202–219, 2006.
- [28] I. Mourtos, S. Vatikiotis and G. Zois, "Scheduling jobs on unrelated machines with job splitting and setup resource constraints for weaving in textile manufacturing," in *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*. Cham, Switzerland: Springer International Publishing, 2021. <https://link.springer.com/content/pdf/10.1007/978-3-030-85874-2.pdf>
- [29] J. A. Wang, R. Pan, W. Gao and H. Wang, "An automatic scheduling method for weaving enterprises based on genetic algorithm," *The Journal of The Textile Institute*, vol. 106, no. 12, pp. 1377–1387, 2015.
- [30] J. K. Perret, K. Schuck and C. Hitzegrad, "Production scheduling of personalized fashion goods in a mass customization environment," *Sustainability*, vol. 14, no. 1, pp. 538, 2022.
- [31] H. C. Chang and T. K. Liu, "Optimisation of distributed manufacturing flexible job shop scheduling by using hybrid genetic algorithms," *Journal of Intelligent Manufacturing*, vol. 28, no. 8, pp. 1973–1986, 2017.
- [32] T. K. Liu, Y. P. Chen and J. H. Chou, "Solving distributed and flexible job-shop scheduling problems for a real-world fastener manufacturer," *IEEE Access*, vol. 2, pp. 1598–1606, 2014.