

Research Article

Solving Integrated Process Planning, Dynamic Scheduling, and Due Date Assignment Using Metaheuristic Algorithms

Caner Erden ^{1,2}, Halil Ibrahim Demir ^{1,2} and Abdullah Hulusi K k am ^{1,2}

¹Sakarya University, Department of Industrial Engineering, Sakarya, Turkey

²Sakarya University Artificial Intelligence Systems Application and Research Center, Turkey

Correspondence should be addressed to Caner Erden; cerden@sakarya.edu.tr

Received 5 March 2019; Revised 2 May 2019; Accepted 13 May 2019; Published 30 May 2019

Academic Editor: Alessandro Palmeri

Copyright   2019 Caner Erden et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Because the alternative process plans have significant contributions to the production efficiency of a manufacturing system, researchers have studied the integration of manufacturing functions, which can be divided into two groups, namely, integrated process planning and scheduling (IPPS) and scheduling with due date assignment (SWDDA). Although IPPS and SWDDA are well-known and solved problems in the literature, there are limited works on integration of process planning, scheduling, and due date assignment (IPPSDDA). In this study, due date assignment function was added to IPPS in a dynamic manufacturing environment. And the studied problem was introduced as dynamic integrated process planning, scheduling, and due date assignment (DIPPSDDA). The objective function of DIPPSDDA is to minimize earliness and tardiness (E/T) and determine due dates for each job. Furthermore, four different pure metaheuristic algorithms which are genetic algorithm (GA), tabu algorithm (TA), simulated annealing (SA), and their hybrid (combination) algorithms GA/SA and GA/TA have been developed to facilitate and optimize DIPPSDDA on the 8 different sized shop floors. The performance comparisons of the algorithms for each shop floor have been given to show the efficiency and effectiveness of the algorithms used. In conclusion, computational results show that the proposed combination algorithms are competitive, give better results than pure metaheuristics, and can effectively generate good solutions for DIPPSDDA problems.

1. Introduction

Process planning, scheduling, and due date assignment functions have an important role in modern manufacturing systems. During the process planning stage, operations of jobs are sequenced and the production information is transferred to the scheduling stage. In a production facility, process planning stage comprises processes such as determination of the manufacturing method for the product and selection of appropriate machines for each part. Besides, in a job shop scheduling problem (JSSP), machines are assigned to operations of jobs with the objective of optimizing determined performance measures. Although process planning is used as an input in scheduling, process planning and scheduling functions are considered as two independent and separate functions in classic manufacturing models [1]. In recent years, it has been observed by various researchers across the world that these two functions are interrelated and needed

to be handled together [2]. Integration of process planning and scheduling (IPPS) studies aim to improve scheduling performance measures such as optimizing makespan, job flow times, earliness and tardiness (E/T) of each job, and machine utilization level using different metaheuristic algorithms. Furthermore, there are several review articles on IPPS which can be given as [3–6]. On the other hand, many studies proposed to integrate scheduling with due date assignment (SWDDA) as well. Although there are numerous works on IPPS and SWDDA, limited works on integrated process planning, scheduling, and due date assignment (IPPSDDA) have been conducted so far.

Obviously, integrating due date assignment function with IPPS problem can provide a significant improvement in the objective function and the global performance of the manufacturing system. The first study on IPPSDDA was conducted by Demir and Taskin [7]. Demir et al. [8–12] also presented genetic algorithm (GA), simulated annealing

(SA), tabu algorithm (TA), evolutionary strategies (ES), and hybrid algorithms to improve the global performance of the IPPSDDA problem. In this study, in addition to the other studies on IPPSDDA, we considered a dynamic scheduling model in which each job enters the system at a random time and we introduced the integration of process planning, dynamic scheduling, and due date assignment (DIPPSDDA) problem. The main contribution, and the point at which this study is separated from other studies, is to build a discrete event simulation model for IPPSDDA problem that can adapt to unexpected and sudden internal or external changes (dynamic events) in a manufacturing system. We studied new job arrivals that may occur in real shop floors as dynamic event for this study. Furthermore, it is also aimed at solving the DIPPSDDA problem with different efficient metaheuristic algorithms which are simulated annealing (SA), tabu algorithm (TA), genetic algorithm (GA), and the combination of GA/SA, GA/TA to improve the performance of the solutions. The proposed model has been built with the advantages of object-oriented programming (OOP) and discrete event simulation (DES) to enhance the flexibility of the model.

This paper is organized as follows: Section 1 is the introduction part of the study. A review of related works is presented in Section 2. The background and definition of DIPPSDDA are given in Section 3. Section 4 discusses solution techniques for the DIPPSDDA. Section 5 gives the computational results and analysis of the study. Conclusions and future researches are also summarized in Section 6.

2. Related Works

Minimizing costs by eliminating wastes is a core principle of a just-in-time (JIT) production philosophy. Thus, determining due dates as close as possible to their due dates is inseparable from this philosophy. If companies can complete their productions on given due dates, they can make more realistic plans and have optimum capacity usage for shop floors. In this way, resources are used efficiently, and the company increases customer satisfaction. In most modern JIT companies, a job is expected to be completed in its time. The completion of jobs after their due dates could lead to an increase in costs and a decrease in customer satisfaction and, worst, customer loss. On the other hand, the completion of jobs before their due date can cause inventory holding costs. Integrated systems were proposed by considering the process planning and scheduling or scheduling with due date assignment together to avoid ineffective production. Some of the earlier studies reported that it is possible to find the optimal due dates and optimal sequences of jobs using different heuristic algorithms [13–20]. The objective of these studies is to minimize the cost related to due date assignment and scheduling functions.

Besides, some of the recent studies on SWDDA can be given such as [21–30]. There are 3 types of due date assignment models in the literature as follows:

- (i) Common due date assignment (CON): a predefined common due date for each job.

- (ii) Due date assignment considering process times: slack due dates (SLK), total-work-content (TWK), processing plus due dates (PPW), etc.
- (iii) Assignment of due dates according to given values.

Scheduling problems with an optimized due date has been the subject of considerable works over the last several decades. Determining the due date is a matter that needs to be emphasized in terms of customer satisfaction for companies. If the due dates are not met, the companies may face various problems. Some of the studies on scheduling together with the due date covered both single machine scheduling and multimachine problems. Scheduling studies considering due dates for single machine problems can often be seen in the literature such as [17, 23, 26, 31–35]. Panwalkar and Smith [36] have studied the problem of determining the common due date for the scheduling problem consisting of n jobs and one machine in the shop floor. At the end of the study, the problem was solved using a polynomial bound scheduling algorithm. Cheng [37] has set common due dates for each job to minimize delay for a single machine scheduling problem. Biskup and Cheng [38] have attempted to solve the problem of single machine scheduling and the assignment of due dates in which job is done with cramped process times. In Gupta and Sen's [39] study, decisions about scheduling, batching, and the due date assignment were combined for a single machine scheduling problem with n groups of jobs. Bank and Werner [40] determined the duration of release and the duration of the transaction time for each job and assigned a common due date. Cheng [41] has aimed to find the optimal due dates for a scheduling problem. He has emphasized that the heuristic method finds effective solutions for the problem.

An early study on the single machine scheduling problem with early and late due date penalties was conducted by Sidney [42]. In another study, the same problem was addressed again by Seidmann et al. [43], and an algorithm was developed to improve the solution performances. Other studies in which E/T are penalized for single machine schedules can be given such as [32–34, 44].

On the other hand, there are still many studies in the literature in which there is more than one machine. All these studies have attempted to determine common due dates for each job. With the intuitive use of heuristic algorithms, many machine scheduling problems such as job floor type production can be solved. There is also a published literature review for multimachine scheduling problems conducted by Lauff and Werner [45] in which researchers used, for solving the multimachine scheduling problem, a specific due date. In another review study, Gordon et al. [46] drew a framework for the common due date and scheduling problems. In their literature study, the models in which single machine and parallel machines are included and the other studies in the literature on this subject were reviewed. A study list on SWDDA classified based on subjects or objectives of the study is given in Table 1. Other examples of studies on scheduling with due date assignment can be found in [46–48].

In a classic manufacturing model, process planning and scheduling functions are carried out separately where only fixed process plans are transferred to the scheduling stage.

TABLE 1: Studies with different approaches to SWDDA.

Studies	Subjects or objectives
[16, 18, 21–23, 25–28, 30, 32, 46, 53–56]	Scheduling and common due date assignment
[13, 17, 19, 20, 24, 26, 31, 57–59]	Scheduling and separate due date assignment
[16, 20, 23–30, 54]	Single machine scheduling with due date assignment
[53]	Two-machine flow shop scheduling with due date assignment
[18, 19, 22, 28, 46, 55, 56]	Parallel machine scheduling with due date assignment
[13, 45, 59, 60]	Multi-machine scheduling with due date assignment
[57, 58]	Job shop scheduling with due date assignment
[20]	Earliness Penalty
[31]	Maximum Lateness
[31, 58]	Total tardiness
[21, 27, 31, 54]	E/T penalties
[61]	Tardiness and due date related cost
[18, 22, 24, 26, 28, 45, 53, 55, 56, 62]	Due-date related, E/T
[29]	Due date, E/T, delivery cost
[30]	Due date assignment, earliness and weighted number of tardy jobs
[60]	Processing, E/T costs
[32]	Resource consumption, E/T costs
[17, 21, 25, 31, 32]	Weighted number of late jobs; Number of tardy jobs
[19]	Minimize maximum absolute lateness
[23]	Due date, absolute lateness and processing time reduction
[19, 57]	PPW (Process Plus Wait)
[20, 30]	SLK (Common Slack)
[57]	TWK (Total work content)
[60]	Different ready times

Fixed process plans may cause some inefficient schedules because of the inflexible nature. To overcome these problems the idea of integrating process planning with scheduling has arisen from the fact that working with alternative process plans on scheduling has improved the efficiency in manufacturing systems. Wilhelm and Shin [49] addressed the efficiency of “alternative operations” in their study in 1985 for the first time. After that, the studies on IPPS were conducted by several researchers. Chryssolouris et al. [50] tried to integrate process planning and scheduling for the first time in 1985. Other early IPPS studies were carried out by Sundaram and Fu [51].

The best way to integrate process planning and scheduling is to take two functions together. The integration of process planning and scheduling functions is NP-Hard problem, and it is difficult to find their optimal results in a reasonable time, even with small sized problems [52]. There are many studies in the literature on the integration of these two functions, which can be classified based on their solution approaches as given in Table 2.

At the beginning of 2000, a very early study on IPPSDDA was a Ph.D. thesis conducted by Demir and Taskin [7]. Integration of these three functions is useful to avoid the problems that may arise in process planning, scheduling, and due date assignment. The intent of working with integrated functions is based on more efficient flow times, processing times, and variations in machine capacities and manufacturing flexibility. From a literary perspective, the integration of

the three functions is less mentioned. In studies of Demir et al., the feasibility, advantages, solution methods, and results of the integrated operation of the three functions were mentioned. In addition to the models discussed in the IPPS problems, preparation times were also considered in these studies and the importance of considering preparation times was discussed. They integrated three functions which are process planning, scheduling, and due date assignment using genetic algorithms in their thesis [7]. It is observed that weighted due date assignment and weighted scheduling improved the overall performance of the problem substantially.

One of the current work areas related to the job shop scheduling problem is dynamic scheduling. Dynamic scheduling problems are frequently encountered in real job floors, for example, jobs arriving at the shop floor randomly over time, machine breakdowns, urgent jobs, or cancellation of existing jobs. Zandieh and Adibi [72] have used a variable neighboring search (VNS) algorithm to solve dynamic events such as random jobs and machine failures in their study. The dynamic scheduling problem can show deterministic or stochastic properties according to the arrival time of the jobs. Scheduling is deterministic if the arrival time of jobs is known in advance; conversely, if arrival times are randomly distributed according to a specific distribution, scheduling is stochastic [73].

The first literature review related to dynamic scheduling problems can be found in [98]. Dominic et al. [99] have

TABLE 2: Classification of IPPS studies based on solution approaches.

Solution Approach	IPPS Studies
Multi-Agent Systems	[63–68]
Evolutionary Algorithm	[4, 39, 69–71]
Game Theory	[31–33]
Genetic Algorithm	[7, 34–36, 40, 45–47, 49–52, 70, 72–75]
Honey Bee Optimization	[76–78]
Imperialist Competitive Algorithm	[79–81]
Particle Swarm Optimization	[82–88]
Simulated Annealing	[74, 89, 90]
Hybrid Approaches	[76, 78, 83, 91–96]
Ant Colony Optimization	[97]

developed two new dispatching rules that work in the dynamic production job floor. Dispatching rules have been revealed to be better than the existing SPT, LIFO, and LPT rules. Aydin and Oztemel [100] have found a solution for the dynamic scheduling problem using reinforced learning factors for proper dispatching rule selection. Li et al. [101] developed an artificial neural network (ANN) using a GA for the problem. Sha and Liu [102] have developed a data mining tool that adapts to the dynamic conditions of jobs. Zandieh and Adibi [72] used it to predict the appropriate parameters of scheduling methods for the shortest average processing time. Zhang et al. [103] developed a hybrid and taboo search algorithm for a dynamic flexible job shop scheduling problem.

As mentioned before, dynamic integrated process planning and scheduling problems are limited in the literature. In the study by Lin et al. [104] an integrated process planning and scheduling problem has been addressed to avoid the dynamic events that may occur in the job floors. Xia et al. [92] conducted a study using alternative process plans with the benefit of the machine breakdowns in the job floor, and the problem with the random arrival of jobs was solved using the neighbor search algorithm. Wong et al. [105] have solved the dynamic integrated process planning and scheduling (DIPPS) problem by suggesting a hybrid multifactor-based system. Recent papers, including that of Yu et al. [106], employed a discrete particle swarm optimization to solve dynamic IPPS problem. They used discrete particle swarm optimization (DPSO) algorithm to solve IPPS optimization problem. Meissner and Aurich [107] applied a cyberphysical system for IPPS. Sustainable IPPS problem was solved by Lee and Ha [108] using a standard GA. Yin et al. [109] used two competing agents for the integrated production, inventory, and batch delivery scheduling and due date assignment. Mor [110] studied common SWDDA with the focus on minmax objective functions. Teymourifar and Ozturk [111] developed a new dispatching rule for dynamic JSSP.

3. Problem Definition

Job shop scheduling problem (JSSP) mainly consists of 3 elements, which are machine configuration, process characteristics, and objective function [112]. The main objective of

JSSP is to find a feasible schedule and to optimize given performance measures such as optimizing makespan, earliness and tardiness time of each job, and machine utilization level.

Basically, scheduling models can be divided into two groups: static and dynamic models. If there are stochastic events occurring over time such as new job arrivals, machine breakdowns, and order cancellations, then this kind of scheduling problem is named “dynamic scheduling problem”. Most of the studies on scheduling focused on static job shop scheduling. Static job shop scheduling means that all scheduling conditions are static and all job information is known and ready at $t = 0$. A static JSSP deals with the process of assigning n jobs to m machines with the aim of optimizing determined performance measures simultaneously. Although working on static job shop scheduling is a well-known problem in the literature, dynamic events often occur, which needs to be handled to increase the productivity and machine balance rates in a real manufacturing system [98]. Actual manufacturing scheduling models are naturally dynamic which are harder to solve than static models [90, 91], because when there is an interruption, proposed plans and schedules should be revised to respond to such dynamic events.

In fact, JSSP belongs to the NP-Hard class without any integration. That is why the integration problems are seen as the hardest among scheduling problems [71]. It can be said that the problem of DIPPSDDA has a huge solution space which is difficult to solve in a reasonable time. Therefore, the use of metaheuristic algorithms is essential for solving DIPPSDDA. In this study, we consider that arrival of jobs to the shop floor varies over time, which makes our model dynamic [113, 114]. Jobs are stochastically arriving at the system and have different weights, due dates, routes, and precedence rules.

The assumptions of the problem to be addressed are listed below:

- (i) There is no ready job at $t = 0$, and the arrival time of the job is distributed according to the exponential distribution.
- (ii) Due dates for jobs are not assigned at $t = 0$.
- (iii) Each machine can process a single operation at the same time.

- (iv) The processing time of the jobs is distributed according to the normal distribution.
- (v) Machines are not broken down.
- (vi) The preparation times of the machines are ignored.
- (vii) Each job can only be delivered to the customer after all its operations are finished.

The notation used for the formulation of the problem is given in Table 3.

3.1. Due Date Assignment Rules. DIPPSDDA begins with the determination of due dates which is the assignment of the due dates of the jobs. In many cases, negotiations with customers can be made to arrange due dates in this process [63, 64]. Inconsistent due dates can lead to unwanted price discounts, customer dissatisfaction, or even end-customer loss. In classic manufacturing models, a job is expected to be completed before its due date. In contrast, just-in-time (JIT) manufacturing environment requires all jobs to be completed exactly on their due dates. In most cases, early due dates can lead to an increase in inventory costs. Similarly, late due dates lead to customer dissatisfaction, price disruption, and, worst, customer loss. There are many studies using quadratic penalty functions to solve tardiness and linear penalty functions to solve earliness and tardiness (E/T) problems. In this study, linear penalty functions are utilized to overcome these problems and generate more accurate process plans.

Both dynamic and static due date assignment rules were employed in this study. Dynamic due date assignment rules use information about the job, route, and operation time, as well as the shop floor and average processing times. Static due date assignment rules use the information relevant to the job such as the arrival of jobs, route information, and operations for calculating due dates. All due date assignment rules used in the study are given in Table 4 with the explanations and equations.

3.2. Dispatching Rules. Dispatching rules are widely used for job shop scheduling with simple implementation. Generally, dispatching rules are used for the selection of the operations by machines in shop floor. 8 different dispatching rules were used in the study, and the list of dispatching rules, with the priority index and the descriptions of the rules, is given in Table 5. The *slack* is calculated as in (1).

$$slack = d_i - p_i - t \quad (1)$$

3.3. Objective Function. The objective of scheduling problems has been often to minimize the makespan. Sometimes the objective function may also be to minimize the average flow time, balance the machine loads, etc. In this study, we use a penalty function for both E/T and due dates for each job. If the value of lateness is positive, this means that the job is completed lately. If the value of lateness is negative, this means that the job is completed early. The penalty for early completion is 0, as expected, and in case of the early completion time, the tardiness penalty is given as 0. Earliness,

tardiness, and due date penalties are calculated as in (2), (3), and (4), respectively.

$$T_j = \max(c_j - d_j, 0) \quad (2)$$

$$E_j = \max(d_j - c_j, 0) \quad (3)$$

$$P_D = w_j \times \left(8 \times \left(\frac{D_j}{480} \right) \right) \quad (4)$$

Weights are applied for a better objective function in the model. We considered a working day as one shift with 8 working hours and that makes 480 minutes per day. All the weighted E/T and due date related costs are punished according to (5) and (6). Meanwhile, (7) shows the sum of the total penalty.

$$P_E = w_j \times \left(5 + 4 \times \left(\frac{E_j}{480} \right) \right) \quad (5)$$

$$P_T = w_j \times \left(6 + 6 \times \left(\frac{T_j}{480} \right) \right) \quad (6)$$

$$P_{total} = P_D + P_E + P_T \quad (7)$$

The objective function of the model is to minimize the total penalties which are penalties for E/T and due dates. Then the final objective function which is a fitness value of the solution is calculated as shown in (8).

$$f_{min} = \sum_{j=1}^n P_{total} \quad (8)$$

3.4. Data Studied. The configurations for the 8 shop floors are given in Table 6. There is also a mini-shop floor for testing algorithm performances by hand. For example, the first shop floor includes 25 jobs, 10 operations, 5 different routes, and 5 machines. Also, the number of iterations of algorithms for the shop floors is given in Table 3. The processing times in the shop floor conform to the normal distribution, which has a 6 average and 12 standard deviations.

The data produced for the study were generated specifically for this study because there is no other similar data in the literature in which process planning, scheduling, and due date assignment rules are employed. The data belonging to the shop floors produced by using NumPy library in the Python programming language were separated according to the shop floors and saved as ".txt" files. Firstly, the time of job arrivals is generated according to the exponential distribution and saved in the file "arrivals_{shop_floor_number}.txt". Secondly, the machine sequences for each alternative process plan of the jobs are saved in the file "machine_numbers_{shop_floor_number}.txt" and the processing times are saved in the file "operation_durations_{shop_floor_number}.txt". Finally, weights are given in the file "weights_{shop_floor_number}.txt". The data files are provided as Supplementary Materials.

TABLE 3: The notation used for the formulation.

n	: the number of jobs	E_j	: the earliness time of j -th job
m	: the number of machines	d_i	: due date time of j -th job
r	: the number of routes	α	: the penalty coefficient of early completion, $\alpha > 0$,
o	: the number of operations	β	: the penalty coefficient of late completion, $\beta > 0$,
a_j	: the arrival time of j -th job	P_D	: the penalty for due date
w_j	: the weight of the j -th job	P_E	: the penalty for earliness
p_j	: the total processing time of j -th job	P_T	: the penalty for tardiness
p_{ij}	: the processing time of i -th operation of j -th job	P_j	: the penalty for j -th job
p_{av}	: the mean processing time of all jobs waiting	P_{total}	: the total penalty
c_j	: the departure time, the completion time of j -th job	f_{min}	: the performance function
L_j	: the lateness time of j -th job	DD_{size}	: the number of due date assignment rules
T_j	: the tardiness time of j -th job	DR_{size}	: the number of dispatching rules
I_j	: Priority index of j -th job	D_j	: the difference between due date and arrival time of job j -th job

TABLE 4: Due date assignment rules.

Rule No	Name	Explanation	Equations
0,1,2	SLK	Slack	$d_i = a_i + p_i + q_x$
3,4,5	WSLK	Weighted slack	$d_i = a_i + p_i + w_{1x}q_x$
6,7,8	TWK	Total work content	$d_i = a_i + k_x p_i$
9,10,11	WTWK	Weighted total work content	$d_i = a_i + w_{1x}k_x p_i$
12,13,14	NOPPT	Number of operations plus processing time	$d_i = a_i + p_i + 5k_x o_i$
15,16,17	WNOPPT	The weighted number of operations plus processing time	$d_i = a_i + p_i + 5w_{1x}k_x o_i$
18	RDM	Random-allowance due dates	$d_i = a_i + N(3P_{av}, P_{av})$
19,20,21,22,23,24,25,26,27	PPW	Processing-time-plus-wait	$d_i = a_i + k_x p_i + q_x$
28,29,30,31,32,33,34,35,36	WPPW	Weighted processing-time-plus-wait	$d_i = a_i + w_{2x}k_x p_i + w_{1x}q_x$

3.5. *Simulation Study.* In the beginning of the study, the shop floor is scheduled according to available jobs on hand. When a new job arrives at the shop floor according to an exponential distribution, the job list is updated, and the problem turns into dynamic JSSP. To solve the dynamic problem, a discrete event simulation has been established for a job shop configuration to validate the performance of the dispatching and due date rules. If we choose the right distributions for the job arrivals and operation durations, we can get more effective solutions for the simulations. The simulation input data which have several jobs, machines, operations, and routes were given in Table 5. Briefly, jobs have different routes and they must be processed on each machine only once. When a new job arrives at the shop floor, it is taken to the machine queue according to the selected route by the algorithm. Machines are allocated according to the dispatching rule for the jobs waiting in the machine queue. After that, processing times of each job are distributed with a normal distribution. The due dates are determined using selected due date assignment rule. Finally, 23 dispatching and 37 due date assignment rules are utilized, and simulation is run till all jobs finish on the shop floor.

The steps of the simulation are as follows:

- (i) Execute the algorithm to generate an individual solution.
- (ii) When a new job comes to the shop floor, determine weights, operation times, operation precedence and durations, and route for the job.
- (iii) Calculate the due date according to the selected due date assignment rule for the job.
- (iv) The first operation of the job enters the queue of the machine to be assigned.
- (v) The machine selects the operation from the pending operations, according to the selected dispatching rule.
- (vi) The completion time for the last operation of the job is determined as the departure time of the job.
- (vii) The earliness, tardiness (E/T), and due dates (D) for the job are calculated using the objective function which is given in (8).
- (viii) The solution is optimized by running steps of the proposed metaheuristics.

4. Solution Techniques

4.1. *Genetic Algorithm (GA).* In 1975, Holland [115] proposed GA which has a good performance in solving difficult problems by mimicking the biological evolution process. GA works based on the population of solutions which has a

TABLE 5: Dispatching rules

Rule No	Name	Explanation	Equations
0-1-2	WATC	Weighted Apparent Tardiness Cost	$I_i = \frac{w_i}{p_i} e^{(\max(\text{slack}, 0)/K\bar{p})}$
3-4-5	ATC	Apparent Tardiness Cost	$I_i = \frac{1}{p_i} e^{(\max(\text{slack}, 0)/K\bar{p})}$
6	WMS	Weighted Minimum Slack	$I_i = -(\text{slack}) w_i$
7	MS	Minimum Slack	$I_i = -(\text{slack})$
8	WSPT	Weighted shortest process time	$I_i = \frac{w_i}{p_i}$
9	SPT	Shortest process time	$I_i = \frac{1}{p_i}$
10	WLPT	Weighted longest process time	$I_i = \frac{p_i}{w_i}$
11	LPT	Longest process time	$I_i = p_i$
12	WSOT	Weighted shortest operation time	$I_i = \frac{w_i}{p_{ij}}$
13	SOT	Shortest operation time	$I_i = \frac{1}{p_{ij}}$
14	WLOT	Weighted longest operation time	$I_i = \frac{p_{ij}}{w_i}$
15	LOT	Longest operation time	$I_i = p_{ij}$
16	EDD	Earliest due date	$I_i = \frac{1}{d_i}$
17	WEDD	Weighted Earliest due date	$I_i = \frac{w_i}{d_i}$
18	ERD	Earliest release date	$I_i = \frac{1}{a_i}$
19	WERD	Weighted earliest release date	$I_i = \frac{w_i}{a_i}$
20	SIRO	Service in random order	<i>random</i>
21	FIFO	First in first out	$I_i = \frac{1}{a_i}$
22	LIFO	Last in first out	$I_i = a_i$

TABLE 6: Shop configurations.

Shop Floors	1	2	3	4	5	6	7	8
Number of jobs	25	50	75	100	125	150	175	200
Number of operations	10	10	10	10	10	10	10	10
Number of machines	5	10	15	20	25	30	35	40
Number of routes	5	5	5	5	3	3	3	3
Iteration size	150	150	100	100	75	75	50	50

common use in the field of scheduling and IPPS. Thanks to the popularity of GA and its easy-to-implement structure, researchers commonly implement GA to solve a variety of problems, including scheduling related problems. Genetic operators are the most important part of the development of GA modeling. In this study, selection, crossover, and mutation GA operators were employed. To increase performance

with GA, it is needed to consider some parameters, such as the number of initial populations, crossover rate, mutation rate, the maximum number of generations, length of chromosomes, encoding the chromosomes and decoding the individuals, and calculating and formulating fitness function in an efficient way. Algorithm 1 shows the pseudocode for GA.

```

(1) DO Random Search
(2) BEGIN to Initialize population
(3) Evaluate population
(4) Rank chromosomes
(5) for  $i \leq \text{iter\_size}$  do
(6)   Ranking selection for parents
(7)   Crossover
(8)   Mutation
(9)   Replace population
(10)  Return sum
(11) end for

```

ALGORITHM 1: Genetic algorithm.

The main steps of GA are as follow:

- (1) Initialization of population: A random search was performed in 10% of the total number of iterations before the genetic algorithm was applied. Best 10 chromosomes from random search generate an initial population for GA. Figure 1 shows a sample chromosome.
- (2) GA operators: Next step is to generate a new population from the mating pool through genetic operators which are crossover and mutation.
 - (a) Crossover: Operator selects two chromosomes as parents and creates new child individuals. In the beginning, the crossover operator determines the crossover point size, which is calculated using formula $C_{size} = \text{floor}(0.1 \times N)$ where N is the chromosome length, so C_{size} is related to the length of the chromosome. For example, if N is equal to 27 which is the value of the first shop floor, C_{size} is calculated as 2. Each gene on a chromosome has a different probability to be selected as a crossover point, where due date rule gene and dispatching rule gene have a dominant probability of 0.5 and the rest of the genes share the other 0.5 equally. An example of crossover operation is given in Figure 2.
 - (b) Mutation: Operator selects a dynamic number of points changing according to the length of the chosen chromosome and changes it to an appropriate value from the domain. Each gene of the chromosome has a different probability to be selected as a mutation point where the due date rule gene and dispatching rule gene have a dominant probability of 0.5 and the rest of the genes share the other 0.5 equally. An example of mutation operation is given in Figure 3. M_{size} is the number of mutation points which is calculated using formula $M_{size} = \text{floor}(0.5 \times N)$.
 - (c) Selection: Process of selecting parents is repeated until the stopping criteria are met which is the maximum iteration size. The average fitness value of the population decreases

in time, as the chromosomes with the best fitness values are selected for the new generation.

- (i) Ranking selection method: A linear ranking selection method is used in which the best chromosome in a population gets more probability to be selected. This method is chosen because the best and worst fitness values get closer as the iterations go on in time. In the last iteration, the fitness values are almost equal. This will cause the algorithm to use the same probabilities when selecting among the best and the worst chromosomes. Contrary to the use of fitness values alone, sorting individuals and assigning probabilities according to their positions will give better solutions to this problem. In the ranking selection method, chromosomes have the probability of being selected according to their ranking and these probabilities are fixed for all iterations. In this study, the preferred probability for 10 chromosomes is [0.3, 0.2, 0.15, 0.12, 0.10, 0.07, 0.03, 0.02, 0.006, 0.004], respectively. Table 7 shows the differences in selection probabilities between roulette wheel and ranking selection methods, for the initial and last population.
- (3) Updating the population: As the population size needs to be limited to a fixed number which is 10 for this study, 10 best chromosomes will always survive at the end of the iteration.

4.2. Tabu Algorithm (TA). Glover [116] proposed the fundamental framework of TA, which is a metaheuristic algorithm designed to find a near-optimum solution for complex problems. It starts from an initial solution generated based on local search. After it finds a feasible solution, TA tries to get better fitness values by selecting the best individual in each iteration. TA searches space for finding a good neighborhood. It stores the previous bad solutions in a tabu list. This process is repeated until the maximum iteration number is reached. The tabu list is updated and compared to the current solution at every single iteration. The size of the tabu list is kept fixed to use memory effectively. Algorithm 2 shows the pseudocode for TA, and variable neighborhood search algorithm for TA and SA is illustrated in Figure 4.

4.3. Simulated Annealing (SA). SA is used to solve large-scale optimization problems, which was developed by Kirkpatrick et al. [117] based on the cooling and recrystallization process of hot materials. SA is another neighborhood search algorithm and widely used to solve big combinatorial optimization problems. It is used in many problems in numerous disciplines; in particular, global extremum is searched within the many local extrema [118]. SA evaluates the neighborhoods with the fitness value, but it might choose the worst move

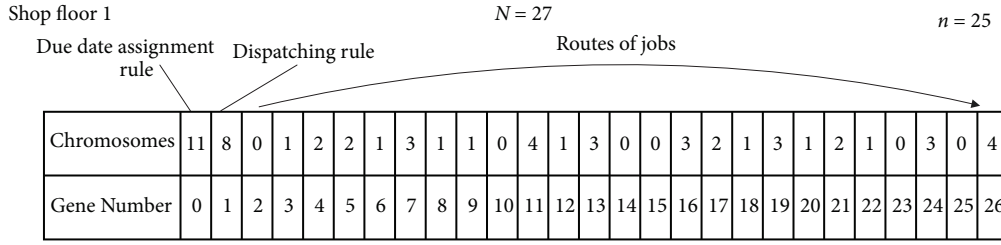


FIGURE 1: A sample chromosome.

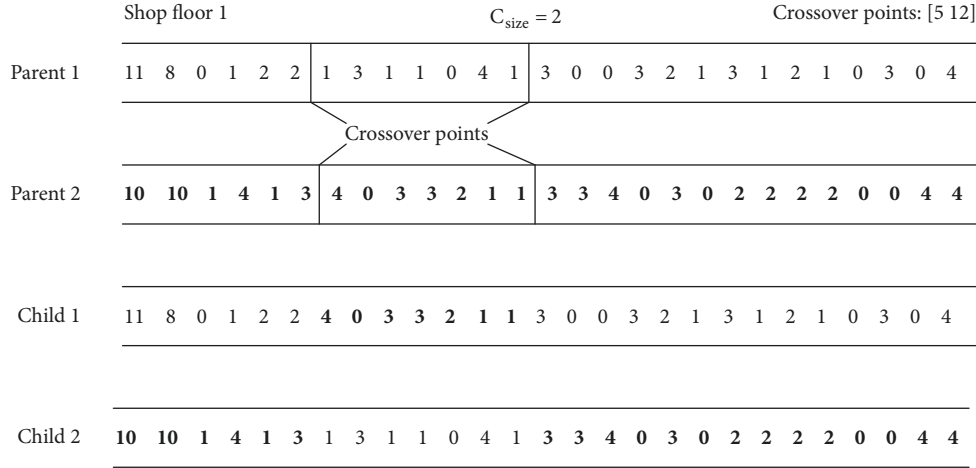


FIGURE 2: A sample crossover operation.

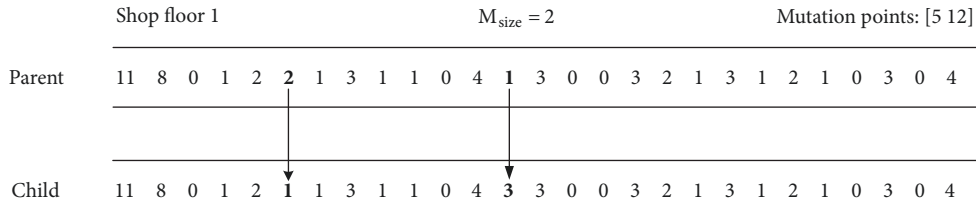


FIGURE 3: A sample mutation operation.

in each iteration to escape from a local minimum. The probability of selecting a solution is given by an exponential control function in which the parameter of the function will decrease during the execution which is given in (9). SA differs from TA as it diversifies the solutions by randomizing them, while TA diversifies them by forcing new solutions. Algorithm 3 shows the pseudocode for SA:

$$P_{accept} = e^{-(E_t - E_i)/k_B T} \quad (9)$$

where P_{accept} is the acceptance rate, T is the temperature, k_B is the Boltzmann constant, E_i is the fitness value of the current solution, and E_t is the fitness value of the next solution. P_{accept} depends on the value of the T . At the first iteration the temperature is high; when it moves to the next iterations, the temperature is decreased gradually.

5. Experimental Results

A computer program in Python programming language was developed to solve DIPPSDDA using Jupyter IDE on a computer which has Intel i5-6200U processor with 2.30 GHz speed and 8 GB of memory. The most used package list to develop the program can be given as follows: sequences, matrices, and probability calculations used in this study were coded using the NumPy library [119]. The Pandas library [120] provides important conveniences for data analysis and data science and was used to analyze data in tables. The Matplotlib library [121] was used to visualize the graphs in the study and analyze the data. A library named Salabim [122] was used for the simulation which is an important part of this study.

A discrete event simulation (DES) model was developed to represent a job shop scheduling system for this study. At $t = 0$, there is no job and the machines are ready to operate. Discrete event simulation updates the system itself when a

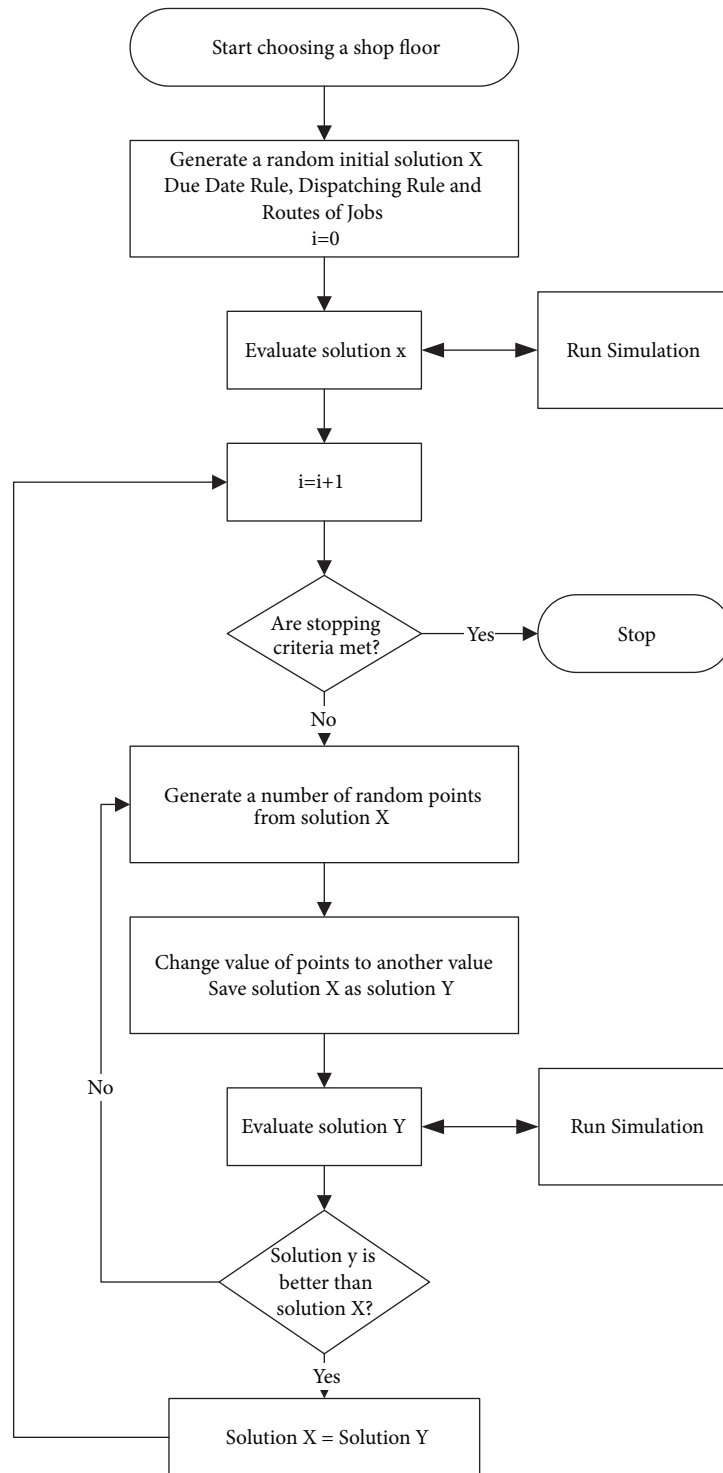


FIGURE 4: Variable neighborhood search (VNS) algorithm flow chart.

job arrives. The arrival time was assigned to the next job based on the cumulative time of the interarrivals so far. The arrival intervals of jobs and the number of jobs within these intervals in each shop floors are provided with histogram graphs in Figure 5. The first graph shows the arrival times for the first 4 shop floors, and the next 4 shop floors are shown in

the second graph, as giving all shop floors at the same chart reduces the readability of the chart. It is also assumed that the time between arrivals is an integer number. The arrival time of the next job is calculated by adding the value generated by the exponential distribution function to the arrival time of the previous job.

TABLE 7: The differences between the selection probabilities of the roulette wheel and ranking selection methods.

Chromosome	Fitness	Roulette wheel selection probability	Ranking selection probability	
Initial Population	Best	189.42	0.1277	0.300
	Average	241.53	0.1000	0.100
	Worst	380.76	0.0635	0.004
	Sum of fitness	2415.30		
Last Population	Best	159.21	0.1011	0.300
	Average	161.18	0.1000	0.100
	Worst	163.12	0.0987	0.004
	Sum of fitness	1611.80		

```

(1) BEGIN to Choose shop floor  $n, m, o$ 
(2) Generate initial solution  $solution_x$ 
(3)  $i \leftarrow 1$ 
(4) while  $iteration < iter\_size$  do
(5)   if  $solution_x$  not in tabu list then
(6)      $i \leftarrow i + 1$ 
(7)     Generate  $solution_y$  using VNS
(8)     if  $solution_y < solution_x$  then return  $solution_y$ 
(9)     end if
(10)  else
(11)    Update tabu list return  $solution_x$ 
(12)  end if
(13) end while

```

ALGORITHM 2: Tabu algorithm.

```

(1) BEGIN to Choose shop floor  $n, m, o$ 
(2) Generate  $solution_x$ 
(3)  $temperature \leftarrow 100$ 
(4)  $solutions \leftarrow \emptyset$ 
(5) while  $temperature > 0$  do
(6)   Generate  $solution_y$  using VNS
(7)    $\delta \leftarrow solution_y - solution_x$ 
(8)   if  $\delta < 0$  then return  $solution_y$ 
(9)   else
(10)    Calculate  $P_{accept}$ 
(11)    if  $P_{accept} < random$  then return  $solution_y$ 
(12)    elsereturn  $solution_x$ 
(13)    end if
(14)  end if
(15)   $temperature \leftarrow temperature - 100/iteration_{size}$ 
(16)   $solutions.append(solution_x)$ 
(17) end while

```

ALGORITHM 3: Simulated annealing.

Another purpose of this study is to determine the benefit of the integration of due date assignment and the dispatching rules with process planning by realizing 4 different levels of integration. At the first level of integration, SIRO dispatching rule and RDM due date assignment rule were tested. It can also be said that no rules are applied to this level. At the second level of integration, dispatching rules are included in the problem. However, due date assignment rule is still being

determined with RDM, and only the benefit of determining the dispatching rule is attempted. At the third level of integration, the dispatching rule was left as SIRO and the due date assignment rules were applied. At the fourth level of integration, the improvement in the solution has been observed in the case where all the dispatching rules and due date assignment rules are applied where process planning weighted scheduling and weighted due date assignment functions are fully integrated. According to these tests, results show that full integration level always gives the best result. Solutions of different integration levels for 8th shop floor are compared in Table 8. Full integration level has been observed to have the best performance as shown in Table 8.

Experimental results of VNS, SA, TA, GA, and HA algorithms are compared with each other and compared with the results of the ordinary solution, which is found without applying any algorithms. The number of iterations is determined based on the characteristics of the algorithms, as GA is population-based and SA and TA are individual-based. In each iteration of the GA, 10 individuals are generated. However, only one individual is produced in each iteration of the SA and TA algorithms. Therefore, the number of iterations given to GA is not equal to the SA and TA algorithms. Iteration numbers of SA and TA algorithms are taken as ten times of GA iteration number. Best, mean, and worst results of VNS, SA, TA, GA, and HA algorithms are recorded. The performance graphs of the algorithms are plotted separately for each shop floor and given in Figure 6.

As mentioned before, the developed model is a run on discrete event-time simulation. Simulation outputs have been examined instantaneously, proving that the system works correctly according to the different due date and dispatching rules, different shop floor types, and different types of jobs and machinery. Simulation trace records all the movements obtained over time until the last job is completed. For better reading of simulation results, Gantt charts for the first two shop floors are given in Figures 7 and 8. Figure 7 shows the Gantt chart for the first shop floor in which the best solution from TA algorithm is calculated with the chromosome of [9, 6, 4, 1, 3, 0, 4, 3, 2, 4, 2, 2, 0, 3, 3, 0, 2, 1, 4, 1, 2, 2, 1, 1, 3, 0, 1] with 163.28 fitness value. Figure 8 shows the Gantt chart for the second shop floor in which the best solution from GA/TA is calculated with the chromosome of [3, 17, 3, 2, 2, 2, 2, 2, 4, 3, 3, 1, 2, 2, 4, 4, 1, 1, 0, 0, 1, 3, 3, 3, 0, 3, 0, 0, 3, 2, 2, 2, 2, 2, 3, 2, 0, 1, 4, 2, 3, 2, 2, 4, 1, 2, 1, 3, 2, 0, 1, 3] with

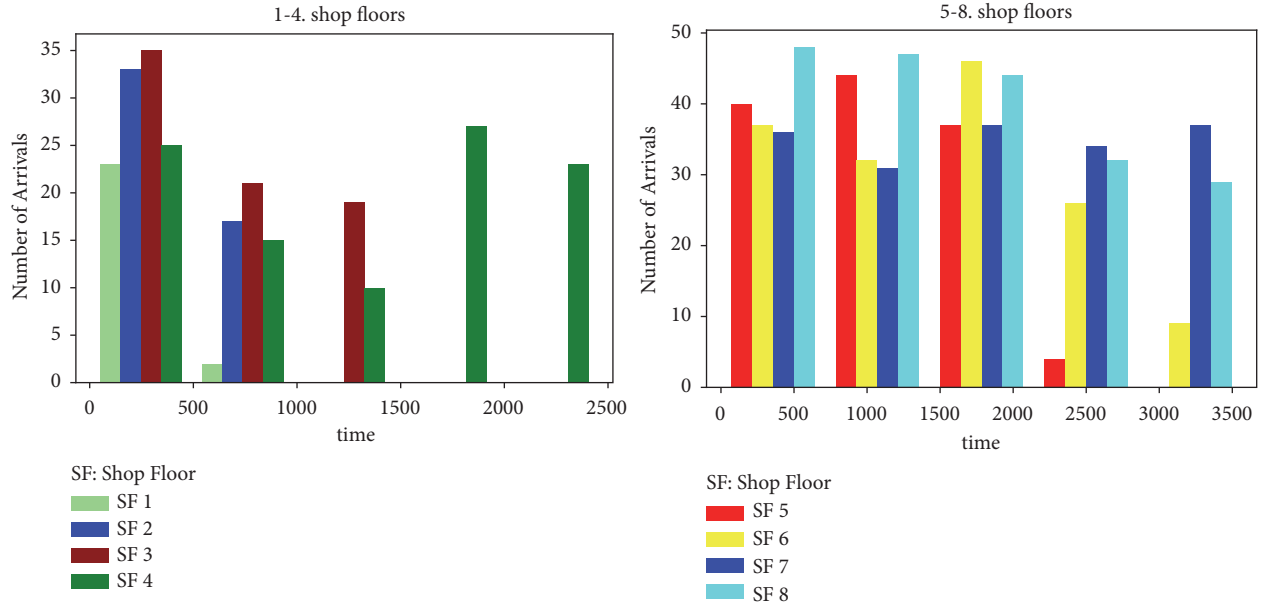


FIGURE 5: Job arrival times.

TABLE 8: Comparison of solutions at different integration levels for shop floor 8.

Level	Algorithm	Best	Average	Worst	CPU (s)
1-SIRO-RDM	VNS	1670.78	1670.78	1672.39	330
	SA	1662.96	1665.28	1667.40	336
	TA	1540.97	1583.97	1662.53	333
	GA	1576.26	1578.43	1579.66	698
	HA	1596.39	1598.75	1600.79	683
2-DSP-RDM	VNS	1524.83	1572.95	1665.59	325
	SA	1648.10	1664.70	1683.02	335
	TA	1520.66	1568.20	1654.37	332
	GA	1583.77	1585.29	1586.08	698
	HA	1592.13	1633.13	1666.77	698
3-SIRO-DD	VNS	1647.39	1647.42	1651.04	168
	SA	1641.08	1654.84	1674.16	171
	TA	1635.27	1636.65	1649.23	202
	GA	1135.05	1530.61	1623.52	350
	HA	1643.08	1654.90	1667.06	295
4-DSP-DD	VNS	1312.05	1339.74	1322.21	329
	SA	1285.0	1290.05	1300.06	328
	TA	1255.75	1260.85	1304.91	329
	GA	1265.41	1280.32	1298.51	348
	HA	1246.18	1255.60	1298.51	337

311.09 fitness value. The Gantt charts of larger shop floors are not graphically presented because they are complex and impossible to follow. In addition to the classical Gantt charts, indicators are added to the graphs, which show the arrivals of jobs. In the Gantt chart, each job is represented with a color bar and all box pieces represent the single operation of a job.

6. Discussion and Conclusions

We studied the integration of process planning, scheduling, and due date assignment manufacturing functions that are related to and affecting the performance of each other in a dynamic shop floor where jobs arrive to the system stochastically. In actual manufacturing model, these functions are

also affected by various events such as urgent orders, order cancellations, maintenance, machine breakdowns, and delays in supplies. However, studies on the integration of process planning, scheduling, and due date assignment are limited, and most of these studies are working in a static environment; our work focused on designing effective algorithm for the dynamic integrated problem. As only scheduling problem is NP-Hard, integration of scheduling with other manufacturing function is also NP-Hard. Thus, it is essential to use a heuristic solution for such a problem. We have developed 4 different pure metaheuristic algorithms and their combination algorithms to solve DIPPSDDA. Our objective for DIPPSDDA was to minimize the earliness, tardiness, and due date times of each job in 8 different shop floors. Even

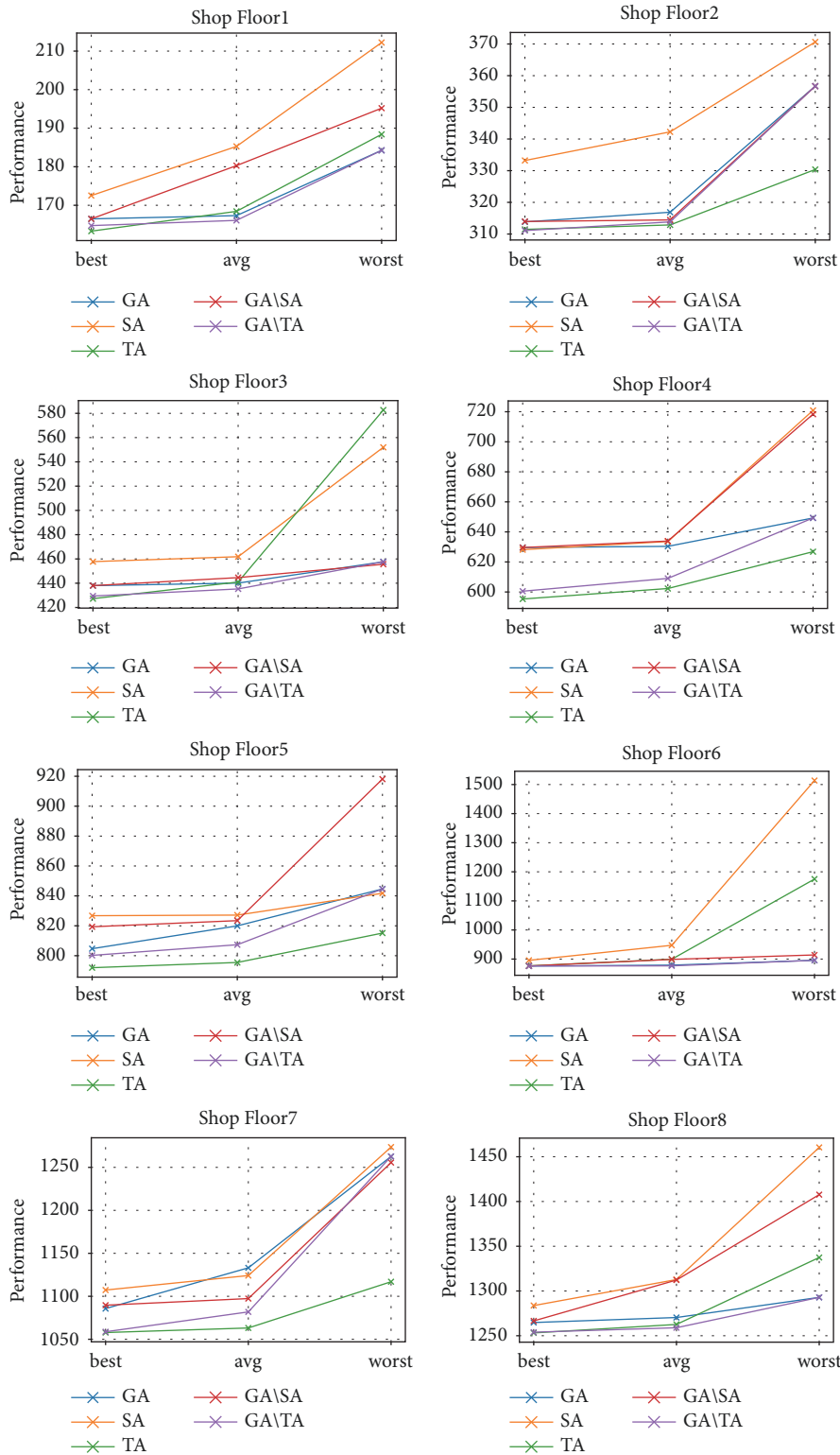


FIGURE 6: Best, average, and worst performances of proposed algorithms on different shop floors.

with integrating, these functions are not enough to model real-world problems thoroughly. Based on the tested experiments, it was shown that the proposed hybrid algorithm (GA/TA) and TA algorithms have generally a good frame

for the integration problem. Besides, GA/TA and TA show higher reliability to solve DIPPSDDA. Unfortunately, it is not possible to make a benchmark to test the performance of the proposed system as it is a new field in the literature. The main

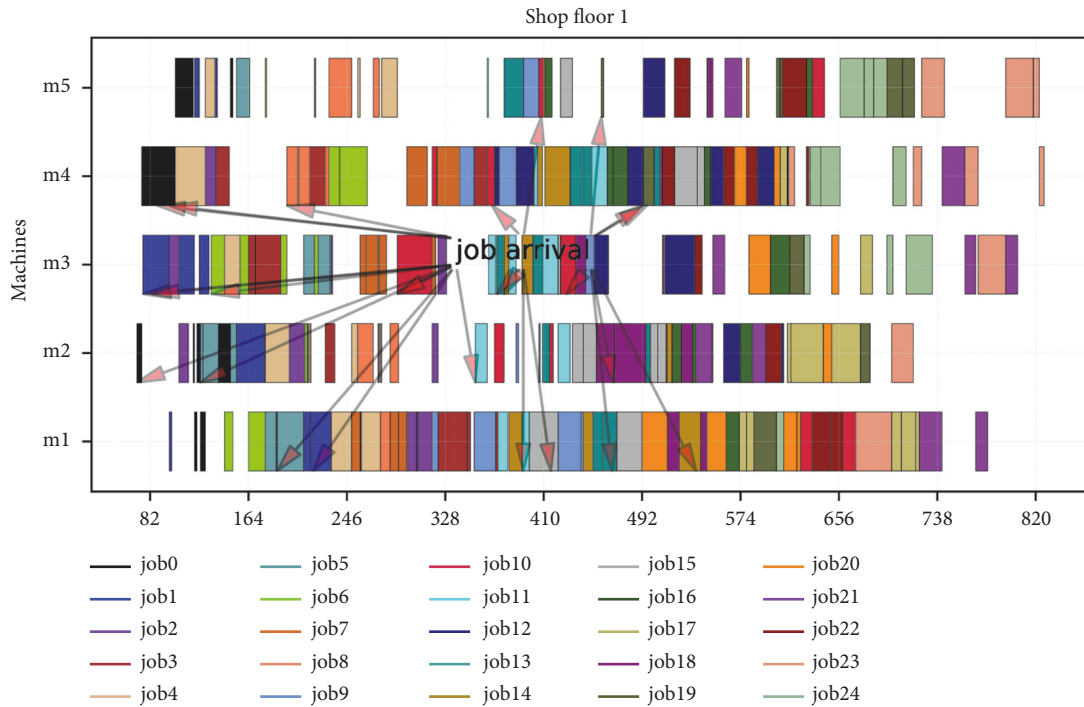


FIGURE 7: Schedule of the best solution generated from GA/TA in shop floor 1.

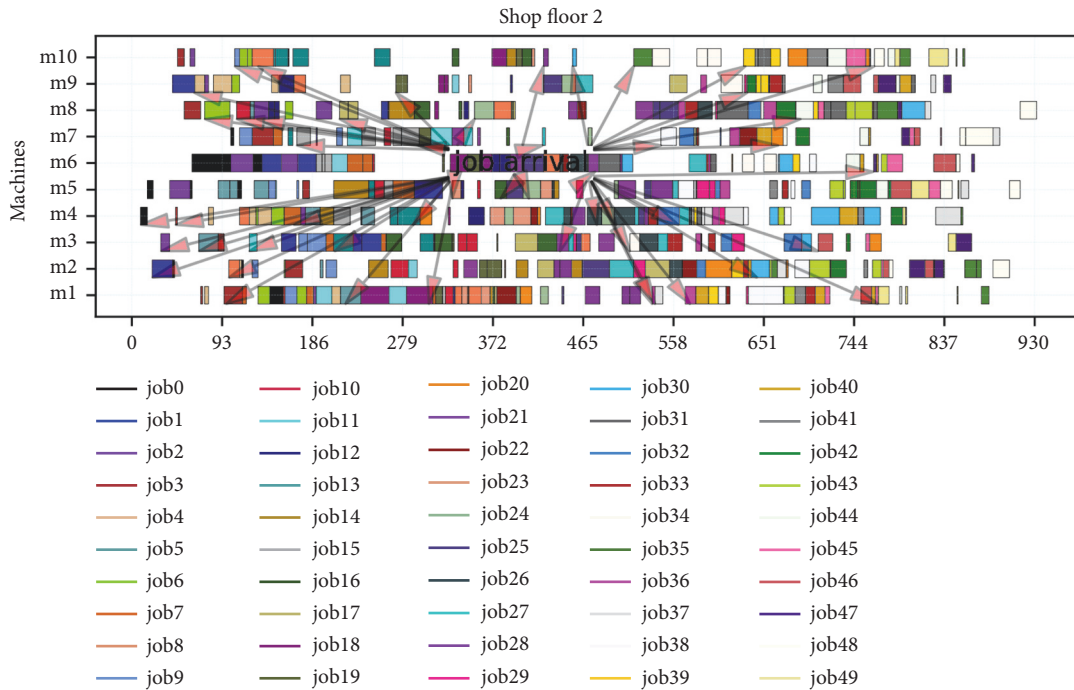


FIGURE 8: Schedule of the best solution generated from TA in the shop floor 2.

contribution of this study is to develop a new model which is called DISPPSDA over the current IPPS and DIPPS models.

The study also demonstrates the benefits of running due date assignment and dispatching rules with process plan selection together. Shop floors have been run without any due date assignment and dispatching rule; then the rules are

integrated step by step. As a result, it is observed that the best results are obtained at full integration level.

This study is one of the few studies on dynamic integrated process planning and scheduling. The following three suggestions can be the objectives of future work. Firstly, the flexibility of operations and operation numbers or the

flexibility of processes and numbers can be integrated into the structure of DIPPSDDA. Secondly, the work can be further tailored to the dynamic events such as machine breakdowns, cancellation of jobs, and the arrival of new urgent jobs. Lastly, in the objective function of the problem, earliness, tardiness, and due date times are punished. In addition, more objectives can be specified to the problem such as makespan, machine balance rates, and minimum wait times of each job. Thus, the problem can be made multiobjective.

Data Availability

The “.txt” data used to support the findings of this study are included within the Supplementary Materials.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Supplementary Materials

The data produced for the study were generated specifically for this study because there is no other similar data in the literature in which process planning, scheduling, and due date assignment rules are employed. The data belonging to the shop floors produced by using NumPy library in the Python programming language were separated according to the shop floors and saved as “.txt” files. Firstly, the time of job arrivals is generated according to the exponential distribution and saved in the file “arrivals_{shop_floor_number}.txt”. Secondly, the machine sequences for each alternative process plan of the jobs are saved in the file “machine_numbers_{shop_floor_number}.txt” and the processing times are saved in the file “operation_durations_{shop_floor_number}.txt”. Finally, weights are given in the file “weights_{shop_floor_number}.txt”. (*Supplementary Materials*)

References

- [1] Y. F. Zhang, A. N. Saravanan, and J. Y. H. Fuh, “Integration of process planning and scheduling by exploring the flexibility of process planning,” *International Journal of Production Research*, vol. 41, no. 3, pp. 611–628, 2003.
- [2] X. Li, L. Gao, X. Shao, C. Zhang, and C. Wang, “Mathematical modeling and evolutionary algorithm-based approach for integrated process planning and scheduling,” *Computers & Operations Research*, vol. 37, no. 4, pp. 656–667, 2010.
- [3] W. Tan and B. Khoshnevis, “Integration of process planning and scheduling—a review,” *Journal of Intelligent Manufacturing*, vol. 11, no. 1, pp. 51–63, 2000.
- [4] M. Gen, L. Lin, and H. Zhang, “Evolutionary techniques for optimization problems in integrated manufacturing system: State-of-the-art-survey,” *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 779–808, 2009.
- [5] X. Li, L. Gao, C. Zhang, and X. Shao, “A review on integrated process planning and scheduling,” *International Journal of Manufacturing Research*, vol. 5, no. 2, pp. 161–180, 2010.
- [6] R. K. Phanden, A. Jain, and R. Verma, “Integration of process planning and scheduling: a state-of-the-art review,” *International Journal of Computer Integrated Manufacturing*, vol. 24, no. 6, pp. 517–534, 2011.
- [7] H. I. Demir and H. Taskin, *Integrated process planning, scheduling and due-date assignment [phd thesis]*, Sakarya University, Sakarya, Turkey, 2005.
- [8] H. I. Demir and C. Erden, “Solving process planning and weighted scheduling with wnopt weighted due-date assignment problem using some pure and hybrid meta-heuristics,” *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, vol. 21, no. 2, pp. 210–222, 2017.
- [9] C. Erden, H. I. Demir, A. Göksu, and O. Uygun, “Solving process planning, ATC scheduling and due-date assignment problems concurrently using genetic algorithm for weighted customers,” *Academic Platform-Journal of Engineering and Science*, 2018.
- [10] H. I. Demir, O. Uygun, I. Cil, M. Ipek, and M. Sari, “Process planning and scheduling with slk due-date assignment where earliness and tardiness and due-dates are punished,” *Journal of Industrial and Intelligent Information*, vol. 3, no. 3, 2014.
- [11] H. I. Demir, T. Cakar, M. Ipek, B. Erdayman, and O. Canpolat, “Process planning and scheduling with PPW due-date assignment using hybrid search,” *International Journal of Science and Technology*, vol. 2, no. 1, pp. 20–37, 2016.
- [12] H. I. Demir, A. H. Kokcam, F. Simsir, and O. Uygun, “Solving process planning, weighted earliest due date scheduling and weighted due date assignment using simulated annealing and evolutionary strategies,” *World Academy of Science, Engineering and Technology International Journal of Industrial and Manufacturing Engineering*, vol. 11, no. 9, pp. 1512–1519, 2017.
- [13] H. Luss and M. B. Rosenwein, “A due date assignment algorithm for multiproduct manufacturing facilities,” *European Journal of Operational Research*, vol. 65, no. 2, pp. 187–198, 1993.
- [14] P. De, J. B. Ghosh, and C. E. Wells, “Scheduling about a common due date with earliness and tardiness penalties,” *Computers & Operations Research*, vol. 17, no. 2, pp. 231–241, 1990.
- [15] S. Li, C. T. Ng, and J. Yuan, “Group scheduling and due date assignment on a single machine,” *International Journal of Production Economics*, vol. 130, no. 2, pp. 230–235, 2011.
- [16] M. Y. Kovalyov, “Batch scheduling and common due date assignment problem: an NP-hard case,” *Discrete Applied Mathematics: The Journal of Combinatorial Algorithms, Informatics and Computational Sciences*, vol. 80, no. 2-3, pp. 251–254, 1997.
- [17] V. Gordon and W. Kubiak, “Single machine scheduling with release and due date assignment to minimize the weighted number of late jobs,” *Information Processing Letters*, vol. 68, no. 3, pp. 153–159, 1998.
- [18] G. I. Adamopoulos and C. P. Pappis, “Scheduling under a common due-date on parallel unrelated machines,” *European Journal of Operational Research*, vol. 105, no. 3, pp. 494–501, 1998.
- [19] T. C. E. Cheng and M. Y. Kovalyov, “Complexity of parallel machine scheduling with processing-plus-wait due dates to minimize maximum absolute lateness,” *European Journal of Operational Research*, vol. 114, no. 2, pp. 403–410, 1999.
- [20] V. S. Gordon and V. A. Strusevich, “Earliness penalties on a single machine subject to precedence constraints: SLK due date assignment,” *Computers & Operations Research*, vol. 26, no. 2, pp. 157–177, 1999.

- [21] D. Biskup and H. Jahnke, "Common due date assignment for scheduling on a single machine with jointly reducible processing times," *International Journal of Production Economics*, vol. 69, no. 3, pp. 317–322, 2001.
- [22] G. Mosheiov and U. Yovel, "Minimizing weighted earliness-tardiness and due-date cost with unit processing-time jobs," *European Journal of Operational Research*, vol. 172, no. 2, pp. 528–544, 2006.
- [23] J.-B. Wang, "Single machine scheduling with common due date and controllable processing times," *Applied Mathematics and Computation*, vol. 174, no. 2, pp. 1245–1254, 2006.
- [24] Y. Xia, B. Chen, and J. Yue, "Job sequencing and due date assignment in a single machine shop with uncertain processing times," *European Journal of Operational Research*, vol. 184, no. 1, pp. 63–75, 2008.
- [25] D. Shabtay, "Due date assignments and scheduling a single machine with a general earliness/tardiness cost function," *Computers & Operations Research*, vol. 35, no. 5, pp. 1539–1545, 2008.
- [26] V. S. Gordon and V. A. Strusevich, "Single machine scheduling and due date assignment with positionally dependent processing times," *European Journal of Operational Research*, vol. 198, no. 1, pp. 57–62, 2009.
- [27] H. Allaoua and I. Osmane, "Variable parameters lengths genetic algorithm for minimizing earliness-tardiness penalties of single machine scheduling with a common due date," *Electronic Notes in Discrete Mathematics*, vol. 36, no. C, pp. 471–478, 2010.
- [28] N. H. Tuong and A. Soukhal, "Due dates assignment and JIT scheduling with equal-size jobs," *European Journal of Operational Research*, vol. 205, no. 2, pp. 280–289, 2010.
- [29] Y. Q. Yin, T. C. E. Cheng, D. H. Xu, and C. C. Wu, "Common due date assignment and scheduling with a rate-modifying activity to minimize the due date, earliness, tardiness, holding, and batch delivery cost," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 223–234, 2012.
- [30] S. Li, C. T. Ng, and J. Yuan, "Scheduling deteriorating jobs with CON/SLK due date assignment on a single machine," *International Journal of Production Economics*, vol. 131, no. 2, pp. 747–751, 2011.
- [31] X. Qi, G. Yu, and J. F. Bard, "Single machine scheduling with assignable due dates," *Discrete Applied Mathematics: The Journal of Combinatorial Algorithms, Informatics and Computational Sciences*, vol. 122, no. 1-3, pp. 211–233, 2002.
- [32] C. T. D. Ng, T. C. Cheng, M. Kovalyov, and S. S. Lam, "Single machine scheduling with a variable common due date and resource-dependent processing times," *Computers & Operations Research*, vol. 30, no. 8, pp. 1173–1185, 2003.
- [33] T. C. Cheng, Z.-L. Chen, and N. V. Shakhlevich, "Common due date assignment and scheduling with ready times," *Computers & Operations Research*, vol. 29, no. 14, pp. 1957–1967, 2002.
- [34] A. C. Nearchou, "A differential evolution approach for the common due date early/tardy job scheduling problem," *Computers & Operations Research*, vol. 35, no. 4, pp. 1329–1343, 2008.
- [35] K.-C. Ying, "Minimizing earliness-tardiness penalties for common due date single-machine scheduling problems by a recovering beam search algorithm," *Computers Industrial Engineering*, vol. 55, no. 2, pp. 494–502, 2008.
- [36] S. S. Panwalkar, M. L. Smith, and A. Seidmann, "Common due date assignment to minimize total penalty for the one machine scheduling problem," *Operations Research*, vol. 30, no. 2, pp. 391–399, 1982.
- [37] T. C. Cheng, "An alternative proof of optimality for the common due-date assignment problem," *European Journal of Operational Research*, vol. 37, no. 2, pp. 250–253, 1988.
- [38] D. Biskup and T. C. E. Cheng, "Single-machine scheduling with controllable processing times and earliness, tardiness and completion time penalties," *Engineering Optimization*, vol. 31, no. 1-3, pp. 329–336, 1999.
- [39] S. K. Gupta and T. Sen, "Minimizing a quadratic function of job lateness on a single machine," *Engineering Costs and Production Economics*, vol. 7, no. 3, pp. 187–194, 1983.
- [40] J. Bank and F. Werner, "Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties," *Mathematical and Computer Modelling*, vol. 33, no. 4-5, pp. 363–383, 2001.
- [41] T. C. Cheng, "Optimal due-date determination and sequencing with random processing times," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 9, no. 8, pp. 573–576, 1987.
- [42] J. B. Sidney, "Optimal single-machine scheduling with earliness and tardiness penalties," *Operations Research*, vol. 25, no. 1, pp. 62–69, 1977.
- [43] A. Seidmann, S. S. Panwalkar, and M. L. Smith, "Optimal assignment of due-dates for a single processor scheduling problem," *International Journal of Production Research*, vol. 19, no. 4, pp. 393–399, 1981.
- [44] T. Ibaraki and Y. Nakamura, "A dynamic programming method for single machine scheduling," *European Journal of Operational Research*, vol. 76, no. 1, pp. 72–82, 1994.
- [45] V. Lauff and F. Werner, "Scheduling with common due date, earliness and tardiness penalties for multimachine problems: a survey," *Mathematical and Computer Modelling*, vol. 40, no. 5-6, pp. 637–655, 2004.
- [46] V. S. Gordon, J. M. Proth, and C. Chu, "A survey of the state-of-the-art of common due date assignment and scheduling research," *European Journal of Operational Research*, vol. 139, no. 1, pp. 1–25, 2002.
- [47] T. C. E. Cheng and M. C. Gupta, "Survey of scheduling research involving due date determination decisions," *European Journal of Operational Research*, vol. 38, no. 2, pp. 156–166, 1989.
- [48] K. R. Baker, "Sequencing rules and due-date assignments in a job shop," *Management Science*, vol. 30, no. 9, pp. 1093–1104, 1984.
- [49] W. E. Wilhelm and H.-M. Shin, "Effectiveness of alternate operations in a flexible manufacturing system," *International Journal of Production Research*, vol. 23, no. 1, pp. 65–79, 1985.
- [50] G. Chryssolouris, S. Chan, and N. P. Suh, "An integrated approach to process planning and scheduling," *CIRP Annals - Manufacturing Technology*, vol. 34, no. 1, pp. 413–417, 1985.
- [51] R. M. Sundaram and S.-s. Fu, "Process planning and scheduling Γ , a method of integration for productivity improvement," *Computers & Industrial Engineering*, vol. 15, pp. 296–301, 1988.
- [52] B. Khoshnevis and Q. M. Chen, "Integration of process planning and scheduling functions," *Journal of Intelligent Manufacturing*, vol. 2, no. 3, pp. 165–175, 1991.
- [53] M. Birman and G. Mosheiov, "A note on a due-date assignment on a two-machine flow-shop," *Computers & Operations Research*, vol. 31, no. 3, pp. 473–480, 2004.
- [54] X. Cai, V. Y. S. Lum, and J. M. T. Chan, "Scheduling about a common due date with job-dependent asymmetric earliness and tardiness penalties," *European Journal of Operational Research*, vol. 98, no. 1, pp. 154–168, 1997.

- [55] L. Min and W. Cheng, "Genetic algorithms for the optimal common due date assignment and the optimal scheduling policy in parallel machine earliness/tardiness scheduling problems," *Robotics and Computer-Integrated Manufacturing*, vol. 22, no. 4, pp. 279–287, 2006.
- [56] G. Mosheiov, "A common due-date assignment problem on parallel identical machines," *Computers & Operations Research*, vol. 28, no. 8, pp. 719–732, 2001.
- [57] V. Vinod and R. Sridharan, "Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system," *International Journal of Production Economics*, vol. 129, no. 1, pp. 127–146, 2011.
- [58] T. Yang, Z. He, and K. K. Cho, "An effective heuristic method for generalized job shop scheduling with due dates," *Computers & Industrial Engineering*, vol. 26, no. 4, pp. 647–660, 1994.
- [59] S. R. Lawrence, "Negotiating due-dates between customers and producers," *International Journal of Production Economics*, vol. 37, no. 1, pp. 127–138, 1994.
- [60] J. N. D. Gupta, K. Krüger, V. Lauff, F. Werner, and Y. N. Sotskov, "Heuristics for hybrid flow shops with controllable processing times and assignable due dates," *Computers & Operations Research*, vol. 29, no. 10, pp. 1417–1439, 2002.
- [61] D. Chen, S. Li, and G. Tang, "Single machine scheduling with common due date assignment in a group technology environment," *Mathematical and Computer Modelling*, vol. 25, no. 3, pp. 81–90, 1997.
- [62] D. Shabtay and G. Steiner, "The single-machine earliness-tardiness scheduling problem with due date assignment and resource-dependent processing times," *Annals of Operations Research*, vol. 159, pp. 25–40, 2008.
- [63] T. N. Wong, C. W. Leung, K. L. Mak, and R. Y. K. Fung, "An agent-based negotiation approach to integrate process planning and scheduling," *International Journal of Production Research*, vol. 44, no. 7, pp. 1331–1351, 2006.
- [64] M. K. Lim and Z. Zhang, "A multi-agent based manufacturing control strategy for responsive manufacturing," *Journal of Materials Processing Technology*, vol. 139, no. 1-3, pp. 379–384, 2003.
- [65] P. Gu, S. Balasubramanian, and D. H. Norrie, "Bidding-based process planning and scheduling in a multi-agent system," *Computers & Industrial Engineering*, vol. 32, no. 2, pp. 477–496, 1997.
- [66] X. Li, C. Zhang, L. Gao, W. Li, and X. Shao, "An agent-based approach for integrated process planning and scheduling," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1256–1264, 2010.
- [67] M. Lim and D. Zhang, "An integrated agent-based approach for responsive control of manufacturing resources," *Computers & Industrial Engineering*, vol. 46, no. 2, pp. 221–232, 2004.
- [68] T. N. Wong, C. W. Leung, K. L. Mak, and R. Y. Fung, "Integrated process planning and scheduling/rescheduling—an agent-based approach," *International Journal of Production Research*, vol. 44, no. 18-19, pp. 3627–3655, 2006.
- [69] P. S. Ow and T. E. Morton, "The single machine early/tardy problem," *Management Science*, vol. 35, no. 2, pp. 177–191, 1989.
- [70] C. Y. Lee and J. Y. Choi, "A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights," *Computers & Operations Research*, vol. 22, no. 8, pp. 857–869, 1995.
- [71] C. Moon and Y. Seo, "Evolutionary algorithm for advanced process planning and scheduling in a multi-plant," *Computers & Industrial Engineering*, vol. 48, no. 2, pp. 311–325, 2005.
- [72] M. Zandieh and M. A. Adibi, "Dynamic job shop scheduling using variable neighbourhood search," *International Journal of Production Research*, vol. 48, no. 8, pp. 2449–2458, 2010.
- [73] S.-C. Lin, E. D. Goodman, and W. F. Punch, "A genetic algorithm approach to dynamic job shop scheduling problems," in *Proceedings of the 7th International Conference on Genetic Algorithms*, 1997.
- [74] T. C. E. Cheng, "A heuristic for common due-date assignment and job scheduling on parallel machines," *Journal of the Operational Research Society*, vol. 40, no. 12, pp. 1129–1135, 1989.
- [75] K. R. Baker and G. D. Scudder, "Sequencing with earliness and tardiness penalties: a review," *Operations Research*, vol. 38, no. 1, pp. 22–35, 1990.
- [76] B. Yuan, C.-Y. Zhang, X.-Y. Shao, and Z.-B. Jiang, "An effective hybrid honey bee mating optimization algorithm for balancing mixed-model two-sided assembly lines," *Computers & Operations Research*, vol. 53, pp. 32–41, 2015.
- [77] X. Li, W. Li, X. Cai, and F. He, "A honey-bee mating optimization approach of collaborative process planning and scheduling for sustainable manufacturing," in *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 465–470, IEEE, Whistler, BC, Canada, June 2013.
- [78] X. X. Li, W. D. Li, X. T. Cai, and F. Z. He, "A hybrid optimization approach for sustainable process planning and scheduling," *Integrated Computer-Aided Engineering*, vol. 22, no. 4, pp. 311–326, 2015.
- [79] L. Liqing, L. Hai, L. Gongliang, and Y. Guanghong, "Integrated production planning and scheduling system design," in *Proceedings of the 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 731–734, IEEE, Beijing, China, November 2017.
- [80] K. Lian, C. Zhang, X. Shao, and L. Gao, "Optimization of process planning with various flexibilities using an imperialist competitive algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 59, no. 5-8, pp. 815–828, 2012.
- [81] S. Zhang, Y. Xu, Z. Yu, W. Zhang, and D. Yu, "Combining extended imperialist competitive algorithm with a genetic algorithm to solve the distributed integration of process planning and scheduling problem," *Mathematical Problems in Engineering*, vol. 2017, Article ID 9628935, 13 pages, 2017.
- [82] X. Li, L. Gao, and X. Wen, "Application of an efficient modified particle swarm optimization algorithm for process planning," *The International Journal of Advanced Manufacturing Technology*, vol. 67, no. 5–8, pp. 1355–1369, 2013.
- [83] F. Zhao, Y. Hong, D. Yu, Y. Yang, and Q. Zhang, "A hybrid particle swarm optimisation algorithm and fuzzy logic for process planning and production scheduling integration in holonic manufacturing systems," *International Journal of Computer Integrated Manufacturing*, vol. 23, no. 1, pp. 20–39, 2010.
- [84] Y. W. Guo, W. D. Li, A. R. Mileham, and G. W. Owen, "Applications of particle swarm optimisation in integrated process planning and scheduling," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 2, pp. 280–288, 2009.
- [85] Y.-J. Tseng, F.-Y. Yu, and F.-Y. Huang, "A green assembly sequence planning model with a closed-loop assembly and disassembly sequence planning using a particle swarm optimization method," *The International Journal of Advanced Manufacturing Technology*, vol. 57, no. 9-12, pp. 1183–1197, 2011.
- [86] H. Zhu, W. Ye, and G. Bei, "A particle swarm optimization for integrated process planning and scheduling," in *Proceedings of*

- the 2009 IEEE 10th International Conference on Computer-Aided Industrial Design Conceptual Design, pp. 1070–1074, IEEE, 2009.
- [87] Y. W. Guo, W. D. Li, A. R. Mileham, and G. W. Owen, "Optimisation of integrated process planning and scheduling using a particle swarm optimisation approach," *International Journal of Production Research*, vol. 47, no. 14, pp. 3775–3796, 2009.
- [88] L. I. Ba, Y. Li, M. Yang et al., "A mathematical model for multi-workshop IPPS problem in batch production," *Mathematical Problems in Engineering*, vol. 2018, Article ID 7948693, 16 pages, 2018.
- [89] W. D. Li and C. A. McMahon, "A simulated annealing-based optimization approach for integrated process planning and scheduling," *International Journal of Computer Integrated Manufacturing*, vol. 20, no. 1, pp. 80–95, 2007.
- [90] A. A. R. Hosseinabadi, H. Siar, S. Shamshirband, M. Shojafar, and M. H. N. Md. Nasir, "Using the gravitational emulation local search algorithm to solve the multi-objective flexible dynamic job shop scheduling problem in Small and Medium Enterprises," *Annals of Operations Research*, vol. 229, no. 1, pp. 451–474, 2014.
- [91] A. B. Farahabadi and A. Hosseinabadi, "Present a new hybrid algorithm scheduling flexible manufacturing system consideration cost maintenance," *International Journal of Scientific & Engineering Research*, vol. 4, no. 9, pp. 1870–1875, 2013.
- [92] H. Xia, X. Li, and L. Gao, "A hybrid genetic algorithm with variable neighborhood search for dynamic integrated process planning and scheduling," *Computers & Industrial Engineering*, vol. 102, pp. 99–112, 2016.
- [93] J. N. Hooker, "A hybrid method for planning and scheduling," *Constraints. An International Journal*, vol. 10, no. 4, pp. 385–401, 2005.
- [94] X. Y. Li, X. Y. Shao, L. Gao, and W. R. Qian, "An effective hybrid algorithm for integrated process planning and scheduling," *International Journal of Production Economics*, vol. 126, no. 2, pp. 289–298, 2010.
- [95] W. Huang, Y. Hu, and L. Cai, "An effective hybrid graph and genetic algorithm approach to process planning optimization for prismatic parts," *The International Journal of Advanced Manufacturing Technology*, vol. 62, no. 9–12, pp. 1219–1232, 2012.
- [96] W. Zhang, M. Gen, and J. Jo, "Hybrid sampling strategy-based multiobjective evolutionary algorithm for process planning and scheduling problem," *Journal of Intelligent Manufacturing*, vol. 25, no. 5, pp. 881–897, 2014.
- [97] C. W. Leung, T. N. Wong, K. L. Mak, and R. Y. K. Fung, "Integrated process planning and scheduling by an agent-based ant colony optimization," *Computers & Industrial Engineering*, vol. 59, no. 1, pp. 166–180, 2010.
- [98] R. Ramasesh, "Dynamic job shop scheduling: A survey of simulation research," *Omega*, vol. 18, no. 1, pp. 43–57, 1990.
- [99] P. D. Dominic, S. Kaliyamoorthy, and M. S. Kumar, "Efficient dispatching rules for dynamic job shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. -1, no. 1, pp. 1-1, 2003.
- [100] M. E. Aydin and E. Öztemel, "Dynamic job-shop scheduling using reinforcement learning agents," *Robotics and Autonomous Systems*, vol. 33, no. 2, pp. 169–178, 2000.
- [101] S. Li, Z. Wu, and X. Pang, "Job shop scheduling in real-time cases," *The International Journal of Advanced Manufacturing Technology*, vol. 26, no. 7-8, pp. 870–875, 2005.
- [102] D. Y. Sha and C.-H. Liu, "Using data mining for due date assignment in a dynamic job shop environment," *The International Journal of Advanced Manufacturing Technology*, vol. 25, no. 11-12, pp. 1164–1174, 2005.
- [103] 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*, vol. 51, no. 12, pp. 3516–3531, 2013.
- [104] C. R. Lin, H. Chen, and Q. Xiao, "Dynamic integrated process planning and scheduling," *Journal of the Chinese Institute of Engineers*, vol. 18, no. 2, pp. 21–32, 2001.
- [105] T. N. Wong, C. W. Leung, K. L. Mak, and R. Y. K. Fung, "Dynamic shopfloor scheduling in multi-agent manufacturing systems," *Expert Systems with Applications*, vol. 31, no. 3, pp. 486–494, 2006.
- [106] M. R. Yu, B. Yang, and Y. Chen, "Dynamic integration of process planning and scheduling using a discrete particle swarm optimization algorithm," *Advances in Production Engineering & Management*, vol. 13, no. 3, pp. 279–296, 2018.
- [107] H. Meissner and J. C. Aurich, "Implications of cyber-physical production systems on integrated process planning and scheduling," *Procedia Manufacturing*, vol. 28, pp. 167–173, 2019.
- [108] H. Lee and C. Ha, "Sustainable integrated process planning and scheduling optimization using a genetic algorithm with an integrated chromosome representation," *Sustainability*, vol. 11, no. 2, p. 502, 2019.
- [109] Y. Yin, Y. Yang, D. Wang, T. C. Cheng, and C.-C. Wu, "Integrated production, inventory, and batch delivery scheduling with due date assignment and two competing agents," *Naval Research Logistics (NRL)*, vol. 65, no. 5, pp. 393–409, 2018.
- [110] B. Mor, "Minmax scheduling problems with common due-date and completion time penalty," *Journal of Combinatorial Optimization*, pp. 1–22, 2018.
- [111] A. Teymourifar and G. Ozturk, "New dispatching rules and due date assignment models for dynamic job shop scheduling problems," *International Journal of Manufacturing Research*, vol. 13, no. 4, pp. 302–329, 2018.
- [112] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Springer, Cham, Switzerland, 5th edition, 2016.
- [113] L. Chunlin, Z. J. Xiu, and L. Layuan, "Resource scheduling with conflicting objectives in grid environments: Model and evaluation," *Journal of Network and Computer Applications*, vol. 32, no. 3, pp. 760–769, 2009.
- [114] H. Van Luu and X. Tang, "An efficient algorithm for scheduling sensor data collection through multi-path routing structures," *Journal of Network and Computer Applications*, vol. 38, no. 1, pp. 150–162, 2014.
- [115] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Oxford, UK, 1975.
- [116] F. Glover, "Tabu search: a tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, 1990.
- [117] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [118] S. Kirkpatrick, "Optimization by simulated annealing: quantitative studies," *Journal of Statistical Physics*, vol. 34, no. 5-6, pp. 975–986, 1984.
- [119] T. E. Oliphant, *A guide to NumPy*, vol. 1, Trelgol Publishing USA, 2006.

- [120] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, Austin, TX, USA, 2010.
- [121] J. D. Hunter, "Matplotlib: a 2D graphics environment," *Computing in Science & Engineering*, vol. 9, pp. 90–95, 2007.
- [122] R. van der Ham, "salabim: discrete event simulation and animation in Python," *Journal of Open Source Software*, vol. 3, no. 27, p. 767, 2018.

