Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 7948693, 16 pages
https://doi.org/10.1155/2018/7948693

Research Article

A Mathematical Model for Multiworkshop IPPS Problem in Batch Production

Li Ba, Yan Li, Mingshun Yang, Xueliang Wu, Yong Liu, Xinqin Gao, and Zhihong Miao

School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an, Shanxi 710048, China

Correspondence should be addressed to Li Ba; xautbali@163.com

Received 20 September 2017; Revised 1 December 2017; Accepted 1 February 2018; Published 20 March 2018

Academic Editor: Dylan F. Jones

Integrated Process Planning and Scheduling (IPPS) problem is an important issue in production scheduling. Actually, there exit many factors affecting scheduling results. Many types of workpieces are commonly manufactured in batch production. Moreover, due to differences among process methods, all processes of a workpiece may not be performed in the same workshop or even in the same factory. For making IPPS problem more in line with practical manufacturing, this paper addresses an IPPS problem with batches and limited vehicles (BV-IPPS). An equal batch splitting strategy is adopted. A model for BV-IPPS problem is established. Makespan is the objective to be minimized. For solving the complex problem, a particle swarm optimization (PSO) with a multilayer encoding structure is proposed. Each module of the algorithm is designed. Finally, case studies have been conducted to validate the model and algorithm.

1. Introduction

Process planning and production scheduling are two indispensable subsystems in manufacturing systems. In traditional manufacturing, they are performed independently in series. A process planning subsystem determines the process route for each workpiece, and a scheduling subsystem allocates manufacturing resources according to the results from the process planning subsystem [1–3]. The independent and serial running mode of the two subsystems may lead to unrealistic process routes, uneven resource utilization, and bottlenecks in scheduling [1–3]. Integration of the two subsystems is an effective method to eliminate conflicts of resources, shorten finishing times of workpieces, and improve machine utilization [1–3]. The integration of process planning with scheduling is important for the development of manufacturing systems.

Integrated process planning and scheduling (IPPS) problem is a significant issue in the field of production scheduling. Existing research focusing on IPPS problems has typically considered only process stages. Few studies have considered batches in the IPPS problem even though many workpieces are processed in batch production. Batch splitting problem has been a significant issue in real production environments [4, 5]. Furthermore, there are some cases in which all processes cannot be accomplished in the same workshop due to their different processing characteristics. Because machines may be located in different workshops, transportation between them must be considered in scheduling which will influence the finishing time of the workpieces.

Batch splitting problem involves splitting lot numbers and splitting lot sizes for each workpiece. In fixed batch splitting, the lot numbers of all workpieces are constant, and the lot sizes of all sublots of a workpiece are equal. Equal batch splitting is typically used in production scheduling problems [4, 6]. The lot size of each sublot of a workpiece is equal, but the lot numbers of different types of workpieces are not equal (the lot number of each workpiece can be changed). Therefore, equal batch splitting is more flexible than fixed batch splitting. Equal batch splitting is adopted in this paper because fixed batch splitting may cause an imbalance between machine and load.

In conclusion, batch and transportation are simultaneously considered so that the IPPS problem better aligns with the real production environment. An IPPS problem considering equal batch splitting and limited vehicles (BV-IPPS) is proposed. The mathematical model of BV-IPPS is established to minimize the makespan. Given the complexity of the problem, a particle swarm optimization (PSO) is designed with a multilayer encoding structure. Finally, the model and algorithm are proven through a case study.

This paper initially provides a brief literature review related to IPPS problem in Section 2. A mathematical model for the problem is proposed in Section 3. A PSO with a multilayer encoding structure is designed in Section 4. The computational results are analysed in Section 5. Finally, the conclusions are provided in Section 6.

2. Literature Review

The basic IPPS problem can be defined as follows [7]: "Given a set of n parts that are to be processed on m machines with operations including alternative manufacturing resources, select suitable manufacturing resources and sequences of operations to determine a schedule in which the precedence constraints among operations can be satisfied and the corresponding objectives can be achieved."

Researchers began studying IPPS problem in the 1980s [8]. Chryssolouris et al. [9] proposed an approach to integrate process planning and scheduling. An increasing number of studies have focused on the IPPS problem in recent years.

Phanden et al. [1] determined that considering process planning and scheduling separately may cause many problems and limitations. They introduced three common approaches for integrating process planning and scheduling. Potential avenues for future IPPS work are discussed at the end of the paper.

(1) Integrated Approaches. Saygin and Kilic [10] proposed a framework for integrating process planning and scheduling that consisted of machine tool selection, process plan selection, scheduling, and rescheduling modules. The framework was validated using many samples. Phanden et al. [11] established a makespan-targeted IPPS model composed of a process route selection module, scheduling module, analysis module, and process route modification module. A genetic algorithm (GA) was designed to solve the model. The availability of the model and algorithm was proven by a case study. Manupati et al. [12] proposed a mobile agent-based approach for integrating process planning and scheduling. A mathematical model for a biobjective IPPS problem with consideration of transportation was established. The approach was proven through many examples.

(2) Improvements to Algorithms. Petrović et al. [3] proposed a hybrid algorithm based on PSO and chaos theory. The advantages of the hybrid algorithm have been proven through many benchmarks by comparisons with other approaches. Xia et al. [13] proposed a dynamic IPPS problem with consideration of machine breakdown and new job arrival. A model for the problem was established. A hybrid GA with variable neighbourhood search was designed to solve the problem. Zhang and Wong [14] proposed a GA framework and integrated ant colony optimization (ACO) into the framework for solving IPPS problem. Lian et al. [15] proposed an imperialist competitive algorithm (ICA) to solve IPPS problem. Lian et al. [16] proposed a mathematical model for process planning problem with an objective of total cost minimization. An ICA was designed to solve the problem. Shao et al. [17] established an IPPS model with two objectives. Makespan and machine utilization are objectives which will be optimized. A modified genetic algorithm-based approach was designed for solving the problem. Li et al. [18] proposed an evolutionary algorithm-based approach for solving IPPS problem. Guo et al. [7] presented an advanced PSO approach to solve the IPPS problem. The advantages of PSO were proven by comparison with other intelligent algorithms. Seker et al. [19] proposed a hybrid heuristic algorithm based on a GA and artificial neural network (ANN) to solve IPPS problem.

(3) Other Research on IPPS Problem. Zhang et al. [20] established an IPPS model with the total energy consumption as the objective. A genetic algorithm-based approach was proposed to solve the problem. Haddadzade et al. [2] proposed an IPPS problem that considered stochastic processing time. The Dijkstra algorithm and Monte Carlo sampling method were used to create examples, and a hybrid algorithm based on simulated annealing and tabu search was designed to solve the problem. Kis [21] proposed a particular job-shop scheduling problem in a chemical production environment. The process routes were directed acyclic graphs and consisted of several alternative subgraphs. A tabu search and GA were proposed to solve the problem. Li and McMahon [22] proposed a multiobjective IPPS problem. The target to be optimized was obtained by combining multiple objectives through linear weighting. A simulated annealing algorithm (SAA) was proposed to solve the problem. Moon and Seo [23] considered transportation in IPPS problem with the makespan as the objective, and an evolution algorithm was proposed to solve the problem.

Many existing studies on IPPS problem only involve the process stage; few consider batching, which may cause the IPPS problem to unrealistic. In real production, many workpieces are processed in batches. Batch splitting problem is a significant issue in real production environments [4, 5]. Furthermore, all processes may not able to be processed in the same workshop due to different process types. Transportation is another factor that must be considered in real production environments. References [12, 23] considered transport in an IPPS problem but did not consider the amounts and locations of transports, which may cause the methodology to not fit real production environments.

Recently, several studies have focused on production scheduling problems considering batch splitting. For example, batching has been considered in a job-shop scheduling problem (JSP) [4, 24], flow-shop scheduling problem (FSP) [25, 26], and parallel machine scheduling problem [27]. Research on production scheduling problems considering batch splitting has mainly focused on JSP and FSP. As manufacturing technology has improved over time, normal machine tools now coexist with numerically controlled machine tools and machining centres in many workshops [28]; however, because different machines have similar functions, multiple process routes are designed for the same workpiece to fully utilize different machines [1–3]. Designing multiple process routes for a workpiece is of great significance for

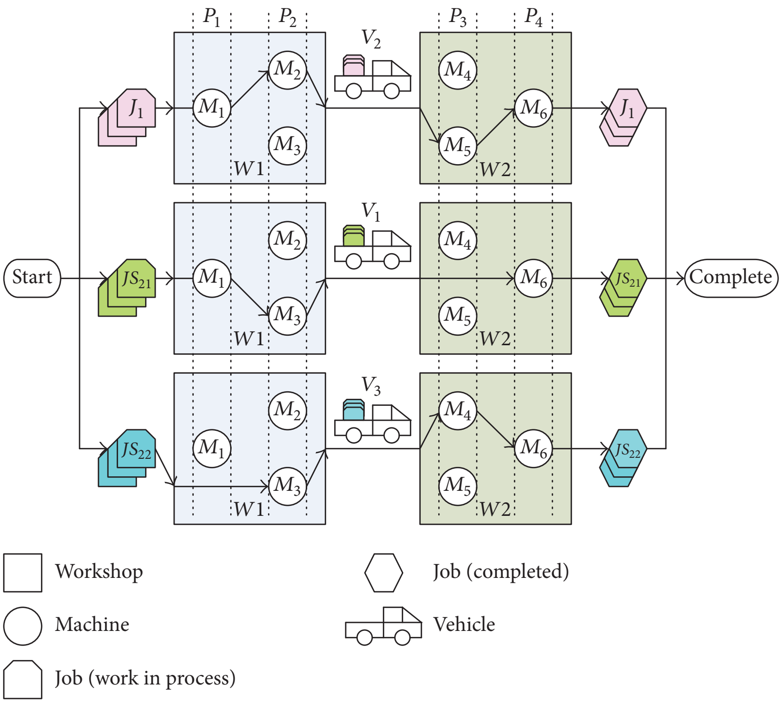


Figure 1: Processing flow of a scheduling solution.

improving the flexibility of scheduling and reducing resource conflicts [1–3]. IPPS problem involves parallel machines and alternative process routes simultaneously. Compared to JSP and FSP, IPPS problem has a larger solution space and is more complex. The scheduling result of IPPS problem is more in line with real production situations. IPPS problem has been identified as NP-hard [19].

Based on the above literature, this paper proposes an IPPS problem considering equal batch splitting and limited vehicles with the makespan as the objective to be minimized. A model for the problem is built, and a PSO with a multilayer encoding structure is proposed. Finally, the model and algorithm are validated through a case study.

3. BV-IPPS Problem Formulation

3.1. Problem Description. There are different workpieces that must be processed in a factory. The batches of different workpieces are different. Each workpiece can be processed through more than one process route, and each process can be processed by more than one machine. There are several vehicles for transporting workpieces. Under these conditions, the makespan is regarded as an objective. An optimal solution is obtained by selecting the process routes, machines, vehicles and sorting process sequences for each workpiece.

A simple example is shown to illustrate BV-IPPS problem. An order involves two types of jobs: J_1 and J_2. A lot of J_1 and J_2 will be processed. J_1 is not divided. J_2 is divided into two sublots: JS_{21} and JS_{22}. There are six machines: M_1–M_6. There are two workshops: W_1 and W_2. M_1, M_2 and M_3 belong to W_1. M_4, M_5 and M_6 belong to W_2. There are four process methods: P_1, P_2, P_3 and P_4. M_1 is able to complete P_1. M_2 and M_3 are able to complete P_2. M_4 and M_5 are able to complete P_3. M_6 is able to complete P_4. There are three process routes: R_1 (P_1-P_2-P_3-P_4), R_2 (P_1-P_2-P_4) and R_3 (P_2-P_3-P_4). R_1 is designed for J_1. R_2 and R_3 are designed for J_2. There are three vehicles for transmitting jobs: V_1, V_2 and V_3. Assuming R_1 is used for J_1, R_2 is used for JS_{21}, and R_3 is used for JS_{22}. Processing flow of a scheduling solution is shown as Figure 1.

3.2. BV-IPPS Model. Based on the above descriptions, a BV-IPPS model is proposed. The following parameters are defined to establish the model for the problem:

n: Number of workpieces;

m: Number of machines;

t: Number of vehicles;

w: Number of workshops;

p: Number of process routes;

J_i: Workpiece i, $i \in [1, n]$;

JS_{il}: lth sublot of J_i, $l \in [1, bc_i]$;

R_k: Process route k, $k \in [1, p]$;

M_j: Machine j, $j \in [1, m]$;

V_r: Vehicle r, $r \in [1, t]$;

W_g: Workshop g, $g \in [1, w]$;

ps_k: Number of processes of R_k;

b_i: Lot size of J_i;

bc_i: Lot number of J_i;

bs_{il}: Lot size of JS_{il};

WM_j: Workshop in which M_j is located;

WR_r: Current workshop in which V_r is located;

TMS_j: Earliest time at which M_j can be used;

TJS_{il}: Earliest time at which JS_{il} can be processed;

TSJ_{ilkvj}: R_k is chosen for JS_{il}. The starting time of the vth process of R_k, which is processed on M_j, $v \in [1, ps_k]$;

TMJ_{ilkvj}: R_k is chosen for JS_{il}. The process time of the vth process per one piece of J_i, which is processed on M_j;

TFJ_{ilkvj}: R_k is chosen for JS_{il}. The completion time of the vth process of R_k, which is processed on M_j;

$TFFJ_{il}$: Completion time of JS_{il};

$TRF_{rg'g''}$: Transport time of V_r, which travels from $W_{g'}$ to $W_{g''}$, $g' \in [1, w]$, $g'' \in [1, w]$;

TRS_r: Earliest time at which V_r can be used;

RJP_{ilk}: Boolean value. If R_k is chosen for JS_{il}, the value is 1; otherwise, it is 0;

RPM_{kvj}: Boolean value. If the vth process of R_k is processed on M_j, the value is 1; otherwise, it is 0;

RMW_{jg}: Boolean value. If M_j is located in W_g, the value is 1; otherwise, it is 0.

Based on the above parameters, the makespan is the target to be minimized. The objective function is established as:

$$\min \{\max \{TFFJ_{il}\}\} \quad i \in [1, n], \ l \in [1, bc_i]. \tag{1}$$

The following constraints are considered in the model:

(1) The sum of the lot sizes of each sublot of a workpiece is equal to the total lot size of the workpiece.

$$\sum_{l=1}^{bc_i} bs_{il} = b_i \tag{2}$$

(2) One machine can only process one process of a sublot of a workpiece at a time.

If from time t,

$$RJP_{ilk} \cdot RPM_{kv'j} = 1 \tag{3}$$

then during time slot $[t, t + TMJ_{ilkv'j} \cdot bs_{il}]$,

$$\sum_{v=1}^{ps_k} RJP_{ilk} \cdot RPM_{kvj} = 1 \quad v' \in [1, ps_k]. \tag{4}$$

(3) One process cannot be processed by more than one machine.

If from time t,

$$RJP_{ilk} \cdot RPM_{kvj'} = 1 \tag{5}$$

then during time slot $[t, t + TMJ_{ilkvj'} \cdot bs_{il}]$,

$$\sum_{j=1}^{m} RJP_{ilk} \cdot RPM_{kvj} = 1 \quad j' \in [1, m]. \tag{6}$$

(4) Assume that $RJP_{ilk} \cdot RPM_{kv''j'} = 1$, $RMW_{j'g'} = 1$, $RJP_{ilk} \cdot RPM_{k(v''-1)j''} = 1$, and $RMW_{j''g''} = 1$. Then, the $v'' - 1$ th process of JS_{il} is completed. $\forall r'$:

$$\begin{aligned} &\max \left\{TRS_{r'} + TRF_{r'WR_{r'}g''}, TJS_{il}\right\} + TRF_{r'g''g'} \\ &\quad = \min \left\{\max \left\{TRS_r + TRF_{rWR_r g''}, TJS_{il}\right\} + TRF_{rg''g'}\right\} \end{aligned} \tag{7}$$

Then,

$$\begin{aligned} TSJ_{ilkv''j'} = \max &\left\{\left(\max \left\{TRS_{r'} + TRF_{r'WR_{r'}g''}, TJS_{il}\right\}\right.\right. \\ &\left.\left. + TRF_{r'g''g'}\right), TMS_{j'}\right\} \end{aligned} \tag{8}$$

Based on constraint (4), $WR_{r'}$, $TRS_{r'}$, $TFJ_{ilkv''j'}$, TJS_{il} and $TMS_{j'}$ are updated by the following formulas:

$$WR_{r'} = g' \tag{9}$$

$$\begin{aligned} TRS_{r'} = \max &\left\{TRS_{r'} + TRF_{r'WR_{r'}g''}, TJS_{il}\right\} \\ &+ TRF_{r'g''g'} \end{aligned} \tag{10}$$

$$TFJ_{ilkv''j'} = TSJ_{ilkv''j'} + TMJ_{ilkv''j'} \cdot bs_{il} \tag{11}$$

$$TMS_{j'} = TSJ_{ilkv''j'} + TMJ_{ilkv''j'} \cdot bs_{il} \tag{12}$$

$$TJS_{il} = TSJ_{ilkv''j'} + TMJ_{ilkv''j'} \cdot bs_{il} \tag{13}$$

If $v'' = ps_k$, then

$$\begin{aligned} TFFJ_{il} &= TSJ_{ilkv''j'} + TMJ_{ilkv''j'} \cdot bs_{il} \\ &\qquad v'' \in [2, ps_k], \ r' \in [1, t] \end{aligned} \tag{14}$$

Formula (2) ensures that each batch of a workpiece is correct. The relationships between processes and machines are constrained through formulas (3)–(6) to ensure the rationality of the manufacturing process. If a process of a sublot of the workpiece and the front of the process are performed in different workshops, a vehicle for transporting the sublot of the workpiece is determined by formula (7). The starting time of each process is calculated by formula (8). Several parameters are calculated based on formula (8). The vehicles' locations are updated based on formula (9). The vehicles' earliest free time are updated by formula (10). The completion time for each process is calculated by formula (11). The machines' earliest free time is updated by formula (12). The earliest starting time of each sublot of the workpiece is updated by formula (13). The final completion time of each sublot of workpiece is calculated by formula (14).

Table 1: Sample coding of BV-IPPS.

J_i	Code of the workpiece							
	2	1	2	1	2	2	1	2
JS_{il}	Code of the sublot of the workpiece							
	21	11	22	11	21	22	11	21
bs_{il}	Lot size of each sublot of the workpiece							
	26	70	34	70	26	34	70	26
M_j	Code of the machine							
	4	2	1	4	3	1	1	2
W_g	Code of the workshop							
	2	1	1	2	2	1	1	1
TMJ_{ilkvj}	Process time per one piece of the workpiece							
	0.8	0.9	0.7	1.3	1	0.6	1.1	0.7
R_k	Code of the process route							
	3	2	1	2	3	1	2	3

4. PSO with a Multilayer Encoding Structure

Based on the description and modelling of the problem, the BV-IPPS problem is an NP-hard problem. PSO was proposed by Kennedy and Eberhart in 1995 [29] and was formerly used to solve continuous optimization problems. Hitherto, researchers have attempted to improve PSO to solve discrete NP-hard problems. Considering the complexity and discreteness of BV-IPPS problems, this study proposes a PSO to solve the BV-IPPS problem. Each module of the algorithm is introduced below. A flowchart of the PSO with a seven-layer encoding structure for the BV-IPPS problem is proposed at the end of this section.

4.1. Solution Representation. A seven-layer encoding structure is proposed to construct solutions. The structure consists of seven layers: the workpiece, sublot of the workpiece, batch, machine, workshop, process time and process route. An example of the encoding structure of BV-IPPS is provided in Table 1.

In Table 1, for example, the codes on the first column mean that R_3 is adopted for JS_{21}. Because code "21" first appears in the layer of the sublot of the workpiece, it represents the first process of JS_{21}. There are 26 pieces of JS_{21} that will be processed. M_4 is chosen to complete this process. The process time is 0.8 per workpiece. M_4 is located in W_2, and so on.

Vehicles are not contained in the solution representation. In this study, vehicles are scheduled dynamically by constraint (4) in Section 4. An order of vehicles is generated after scheduling. For example, order [2 1 1 3] means that V_2 is used first, then V_1 is used for transportation, and so on.

4.2. Updating. Individuals are updated with the current global optimal individual or their own best solutions of history based on a probability. A precedence preserving order-based crossover (POX) [30] operator is used for updating. Individuals are still feasible after updating by using the POX operator. Assuming that workpiece J_3 will be updated, an example of a POX operator for the BV-IPPS problem is shown in Figure 2. Steps of Figure 2 are shown as follows.

(1) The columns which contain J_3 are cleared in F1. Then, the temporary F1 is generated.

(2) Each column of global optimal individual is traversed orderly. The columns which contain J_3 are taken.

(3) Each column of J_3 is inserted into the blank columns of temporary F1. If the number of blank columns of temporary F1 is less than the number of columns of J_3, the rest of columns of J_3 are inserted in the end of temporary F1. The new F1 is generated.

4.3. Mutating. The process sequences and machines are mutated separately. Swapping is used to mutate the process sequence. Two locations are generated randomly. The two columns of data in the locations are exchanged. This may lead to an unfeasible solution after swapping. The new solution will be corrected after swapping. For machine mutating, a location is generated randomly. Then, the code of the machine in the location is changed randomly.

Assuming that the fifth and ninth columns will be swapped, swapping and correction are shown in Figure 3. Steps of Figure 3 are shown as follows.

(1) The fifth and ninth columns are swapped. Then, the temporary F1 is generated.

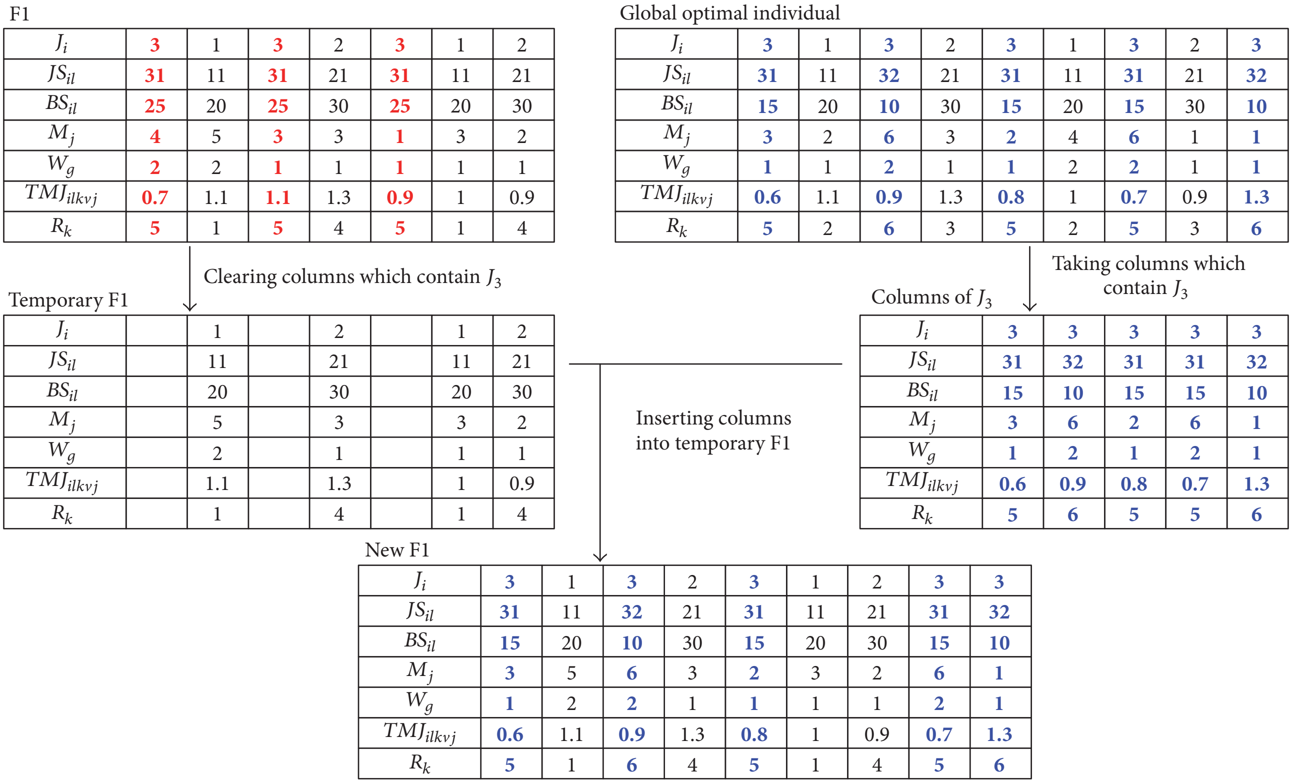
(2) Since there is an invalid column which contains JS_{31} in temporary F1, the columns which contain JS_{31} are cleared. Then, the temporary F1$'$ is generated.

(3) Each column of F1 is traversed orderly. The columns which contain JS_{31} are taken.

(4) Each column of JS_{31} is inserted into the blank columns of temporary F1$'$. The new F1 is generated.

Assuming that the machine of the third column will be mutated, M_3, M_4 and M_6 can be used to process the first process of JS_{32}. M_3 belongs to W_1. M_4 and M_6 belong to W_2. An example of changing for a machine is shown in Figure 4. Steps of Figure 4 are shown as follows.

(1) A machine is selected randomly for replacing M_6 at the third column. In Figure 4, M_6 is replaced by M_4.



F1

J_i	3	1	3	2	3	1	2
JS_{il}	31	11	31	21	31	11	21
BS_{il}	25	20	25	30	25	20	30
M_j	4	5	3	3	1	3	2
W_g	2	2	1	1	1	1	1
TMJ_{ilkvj}	0.7	1.1	1.1	1.3	0.9	1	0.9
R_k	5	1	5	4	5	1	4

Global optimal individual

J_i	3	1	3	2	3	1	3	2	3
JS_{il}	31	11	32	21	31	11	31	21	32
BS_{il}	15	20	10	30	15	20	15	30	10
M_j	3	2	6	3	2	4	6	1	1
W_g	1	1	2	1	1	2	2	1	1
TMJ_{ilkvj}	0.6	1.1	0.9	1.3	0.8	1	0.7	0.9	1.3
R_k	5	2	6	3	5	2	5	3	6

Clearing columns which contain J_3

Taking columns which contain J_3

Temporary F1

J_i		1		2		1	2
JS_{il}		11		21		11	21
BS_{il}		20		30		20	30
M_j		5		3		3	2
W_g		2		1		1	1
TMJ_{ilkvj}		1.1		1.3		1	0.9
R_k		1		4		1	4

Columns of J_3

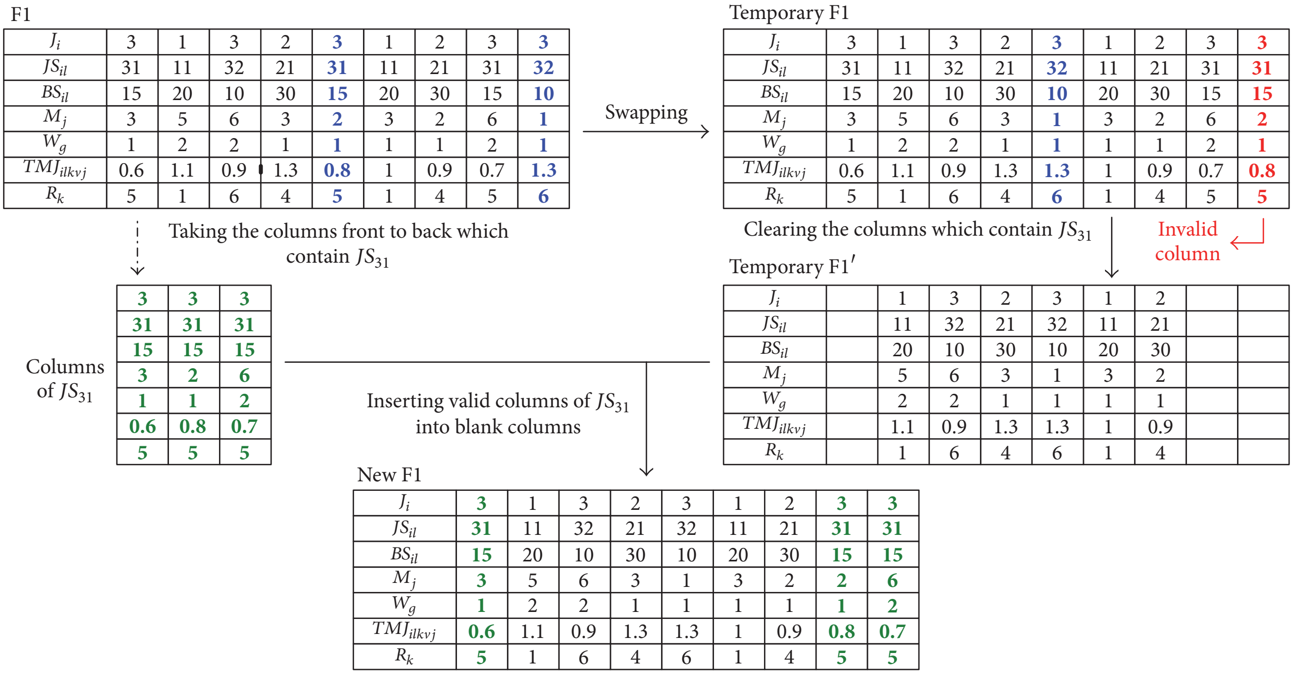
J_i	3	3	3	3	3
JS_{il}	31	32	31	31	32
BS_{il}	15	10	15	15	10
M_j	3	6	2	6	1
W_g	1	2	1	2	1
TMJ_{ilkvj}	0.6	0.9	0.8	0.7	1.3
R_k	5	6	5	5	6

Inserting columns into temporary F1

New F1

J_i	3	1	3	2	3	1	2	3	3
JS_{il}	31	11	32	21	31	11	21	31	32
BS_{il}	15	20	10	30	15	20	30	15	10
M_j	3	5	6	3	2	3	2	6	1
W_g	1	2	2	1	1	1	1	2	1
TMJ_{ilkvj}	0.6	1.1	0.9	1.3	0.8	1	0.9	0.7	1.3
R_k	5	1	6	4	5	1	4	5	6

Figure 2: Example of POX for BV-IPPS.



F1

J_i	3	1	3	2	3	1	2	3	3
JS_{il}	31	11	32	21	31	11	21	31	32
BS_{il}	15	20	10	30	15	20	30	15	10
M_j	3	5	6	3	2	3	2	6	1
W_g	1	2	2	1	1	1	1	2	1
TMJ_{ilkvj}	0.6	1.1	0.9	1.3	0.8	1	0.9	0.7	1.3
R_k	5	1	6	4	5	1	4	5	6

Swapping

Temporary F1

J_i	3	1	3	2	3	1	2	3	3
JS_{il}	31	11	32	21	32	11	21	31	31
BS_{il}	15	20	10	30	10	20	30	15	15
M_j	3	5	6	3	1	3	2	6	2
W_g	1	2	2	1	1	1	1	2	1
TMJ_{ilkvj}	0.6	1.1	0.9	1.3	1.3	1	0.9	0.7	0.8
R_k	5	1	6	4	6	1	4	5	5

Taking the columns front to back which contain JS_{31}

Clearing the columns which contain JS_{31}

Invalid column

Columns of JS_{31}

3	3	3
31	31	31
15	15	15
3	2	6
1	1	2
0.6	0.8	0.7
5	5	5

Temporary F1′

J_i		1	3	2	3	1	2		
JS_{il}		11	32	21	32	11	21		
BS_{il}		20	10	30	10	20	30		
M_j		5	6	3	1	3	2		
W_g		2	2	1	1	1	1		
TMJ_{ilkvj}		1.1	0.9	1.3	1.3	1	0.9		
R_k		1	6	4	6	1	4		

Inserting valid columns of JS_{31} into blank columns

New F1

J_i	3	1	3	2	3	1	2	3	3
JS_{il}	31	11	32	21	32	11	21	31	31
BS_{il}	15	20	10	30	10	20	30	15	15
M_j	3	5	6	3	1	3	2	2	6
W_g	1	2	2	1	1	1	1	1	2
TMJ_{ilkvj}	0.6	1.1	0.9	1.3	1.3	1	0.9	0.8	0.7
R_k	5	1	6	4	6	1	4	5	5

Figure 3: Example of swapping and correction.

F1

J_i	3	1	**3**	2	3	1	2	3	3
JS_{il}	31	11	**32**	21	31	11	21	31	32
BS_{il}	15	20	**10**	30	15	20	30	15	10
M_j	3	5	**6**	3	2	3	2	6	1
W_g	1	2	**2**	1	1	1	1	2	1
TMJ_{ilkvj}	0.6	1.1	**0.9**	1.3	0.8	1	0.9	0.7	1.3
R_k	5	1	**6**	4	5	1	4	5	6

Changing for machine →

New F1

J_i	3	1	**3**	2	3	1	2	3	3
JS_{il}	31	11	**32**	21	31	11	21	31	32
BS_{il}	15	20	**10**	30	15	20	30	15	10
M_j	3	5	**4**	3	2	3	2	6	1
W_g	1	2	**2**	1	1	1	1	2	1
TMJ_{ilkvj}	0.6	1.1	**1.2**	1.3	0.8	1	0.9	0.7	1.3
R_k	5	1	**6**	4	5	1	4	5	6

FIGURE 4: Example of changing for a machine.

(2) Process time and workshop of the third column are updated. The new F1 is generated.

4.4. Parameters and Flowchart of PSO with a Seven-Layer Encoding Structure. The parameters of PSO with a seven-layer encoding structure are M, D, Alpha, Beta, Pm, Ps and Pa, which are defined as follows:

M: Population size, controlling the number of individuals in the population.

D: Iteration, controlling the number of iterations in the algorithm.

Alpha: Probability with which controlling individuals are updated with the global optimal individual, decimals in the $(0, 1)$ interval.

Beta: Probability with which controlling individuals are updated with their best solution of history, decimals in the $(0, 1)$ interval. In this paper, Beta = 1 − Alpha.

Pm: Mutation probability of individuals, decimals in the $(0, 1)$ interval.

Ps: Mutation probability of the process sequence of an individual, decimals in the $(0, 1)$ interval.

Pa: Mutation probability of the machines of an individual, decimals in the $(0, 1)$ interval.

Based on the above description, a flowchart of PSO with a seven-layer encoding structure for BV-IPPS is shown in Figure 5.

5. Experimental Results and Analyses

5.1. Experiment 1: Batch and Transport. A lathe manufacturer receives an order. Several shaft parts must be processed. The minimum length of the shaft parts is 300 mm. The maximum length of the shaft parts is 900 mm. There are five types of shaft parts: J_1–J_5. The batch of each shaft part is 20, 10, 10, 30, and 20. There are six machines: M_1–M_6. There are three workshops: W_1–W_3. M_1 and M_2 are located in W_1, M_3 and M_4 are located in W_2, and M_5 and M_6 are located in W_3. There are six vehicles: V_1–V_6. There are six process methods: P_1–P_6. There are thirteen process routes: R_1–R_{13}. V_1, V_2, J_1 and J_2 are located in W_1 at time zero, V_3, V_4, J_3 and J_4 are located in W_2 at time zero, and V_5, V_6 and J_5 are located in W_3 at time zero. The makespan is the objective to be minimized. PSO is used to solve the BV-IPPS problem. The process routes of each workpiece are provided in Tables 2–6. The transport time between workshops is provided in Table 7. The process time and transport time are specified in minutes.

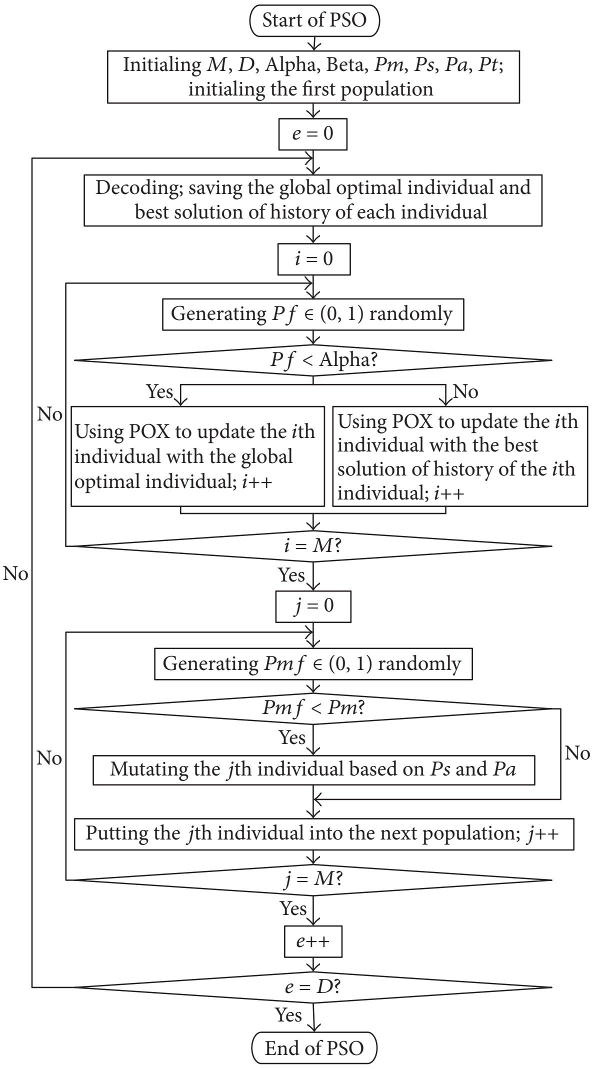

Start of PSO
Initialing M, D, Alpha, Beta, Pm, Ps, Pa, Pt; initialing the first population
e = 0
Decoding; saving the global optimal individual and best solution of history of each individual
i = 0
Generating Pf ∈ (0, 1) randomly
Pf < Alpha?
Yes
No
Using POX to update the ith individual with the global optimal individual; i++
Using POX to update the ith individual with the best solution of history of the ith individual; i++
No
i = M?
No
Yes
j = 0
Generating Pmf ∈ (0, 1) randomly
Pmf < Pm?
Yes
No
No
Mutating the jth individual based on Ps and Pa
Putting the jth individual into the next population; j++
j = M?
Yes
e++
e = D?
Yes
End of PSO

FIGURE 5: Flowchart of PSO with a seven-layer encoding structure for BV-IPPS.

5.1.1. Assumptions

(1) All machines are idle at time zero;

(2) Workpieces are processed according to the process sequence;

(3) Waiting periods between processes are allowed;

(4) Machines will never break down.

Table 2: Process routes of $J1$.

Process route	1								2							3								
Process method	1			4			5		1			3		6		2			3		5		6	
Machine	3	4	6	2	5	6	3	5	3	4	6	1	4	2	3	1	3	5	1	4	3	5	2	3
Process time	1.2	1	1.4	0.7	0.8	1.1	0.9	0.7	1.2	1	1.4	1.2	1	1	1.1	1.2	1.3	1.4	0.8	0.7	1	0.8	0.7	0.8

Table 3: Process routes of $J2$.

Process route	4							5					
Process method	2			5		6		1			4		
Machine	1	3	5	3	5	2	3	3	4	6	2	5	6
Process time	1.2	0.9	1	1	0.8	0.7	1.1	1.2	1.4	0.9	1.1	0.8	0.7

Table 4: Process routes of $J3$.

Process route	6										7							
Process method	1			3			4			5	2				4			6
Machine	3	4	6	1	4	2	5	6	3	5	1	3	5	2	5	6	2	3
Process time	0.8	1.2	1.1	1.1	0.8	1	0.8	1.2	0.6	0.9	1	1.3	0.8	1	0.8	0.9	0.9	1.2

Table 5: Process routes of $J4$.

Process route	8								9									10						
Process method	1				5	6			2		4				5	6		3			4		5	
Machine	3	4	6	3	5	2	3	1	3	5	2	5	6	3	5	2	3	1	4	2	5	6	3	5
Process time	1.1	0.8	1.1	0.7	0.9	0.8	0.6	1.2	1.1	1.3	1	0.9	1.1	1.2	0.8	0.6	0.8	1.2	0.9	0.7	1	1	0.6	0.9

Table 6: Process routes of $J5$.

Process route	11											12									13									
Process method	1				2		3			4		1				3	5		6		3			4		5		6		
Machine	3	4	6	1	3	5	1	4	2	5	6	3	4	6	1	4	3	5	2	3	1	4	2	5	6	3	5	2	3	
Process time	0.9	1	0.8	1	1	1.2	0.8	0.7	0.9	1	0.7	0.9	1	0.8	1	0.8	1.1	0.8	0.7	1	1	1.3	1.2	1	0.7	1.1	0.8	1	0.9	

Table 7: Transport time between workshops.

	W_1	W_2	W_3
W_1	0	7	13
W_2	7	0	9
W_3	13	9	0

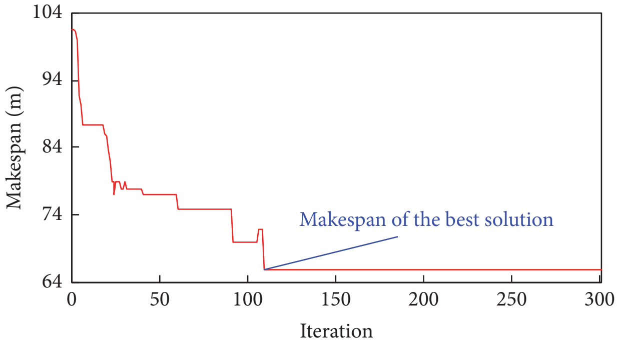


Figure 6: Curve of the makespan of the optimal solution in each iteration.

5.1.2. Results and Analyses. In this study, parameters of PSO are set: $D = 300$, $M = 500$, Alpha = 0.5, Beta = 0.5, $Pm = 0.1$, $Ps = 0.5$, and $Pa = 0.5$. Equal batch splitting is adopted, and the maximum lot number is four. The C# programming language is used to create the algorithm. The runtime is 75 seconds (CPU: 2.6 GHz, dual-core, RAM: 3 GB). The best solution is shown in Table 8. The makespan of the best solution is 66 min. The curve of the makespan of the optimal solution in each iteration is shown in Figure 6. The Gantt chart of the best solution is shown in Figure 7.

The best solution within the vehicles is shown in Table 8; for example, the first and second processes of JS_{43} are not carried out in the same workshop. V_1 is selected to transfer JS_{43} from W_2 to W_1 according to formula (7). In another example, the third and fourth processes of JS_{52} are both performed in W_1. Thus, a vehicle is not needed between these two processes.

(1) Batch. For the above problem, if workpieces are processed without considering batch splitting, the makespan is 82 min, which is longer than 66 min. The Gantt chart of the best solution without considering batch splitting is shown in Figure 8. Machines will work for prolonged periods of time. If batch splitting is adopted, each sublot of the workpieces can be processed concurrently. Therefore, the completion time of each sublot of the workpieces is shorter.

(2) Limited Vehicles. Figure 7 shows that vehicles and machines are scheduled simultaneously. With regard to the transport time of an empty vehicle, V_4 is used to transfer JS_{12} when the first process of JS_{12} has been completed. In the twentieth minute, V_4 is not in workshop W_2. However, M_4 is located in workshop W_2, which is used to process the first process of JS_{12}. Therefore, V_4 must travel to W_2 first. Thus, the transport time of empty vehicle V_4 is generated. Then, JS_{12} is transferred by V_4 from workshop W_2 to W_1. The amounts and locations of vehicles are considered in the IPPS problem simultaneously in this paper. The result is more realistic for determining the real scheduling solution than the best solution obtained by solving the basic IPPS problem.

5.2. Experiment 2: Equal Batch Splitting and Fixed Batch Splitting. To make the comparisons between the two batch splitting strategies more distinct, transport factors will be ignored in experiment 2, which involves two different scale samples. The assumptions are the same as those in experiment 1. As space is limited, several important parameters of the samples are provided in Table 9. The algorithm parameters are provided in Table 10.

In Table 9, "VRN" represents the value range of the number of procedures of each process route and "VRT" represents the process time range of each process.

Two samples are calculated 100 times for each batch splitting strategy. The results of the different batch splitting strategies are provided in Table 11.

(1) Sample 1. The makespan obtained by adopting equal batch splitting and setting the maximum lot number equal to four is better than other results obtained by adopting fixed batch splitting in sample 1. The lot number and batch of each workpiece are constant values in fixed batch splitting. This may lead to an imbalance between the machine and load. The lot number of each workpiece can be changed in equal batch splitting. Thus, equal batch splitting is more flexible than fixed batch splitting. The result obtained by adopting equal batch splitting is superior to that obtained with fixed batch splitting.

(2) Sample 2. The makespan obtained by adopting equal batch splitting and setting the maximum lot number equal to five is not the best makespan of all batch splitting strategies in sample 2. Sample 2 is a large-scale problem. Each workpiece can be divided into five types of lot numbers. The number of batch division results obtained by adopting equal batch splitting in sample 2 is 9536 ($5^{10}/4^5$) times the number of batch division results obtained by adopting equal batch splitting in sample 1. Therefore, it is more difficult to achieve a smaller makespan by adopting the equal batch splitting strategy due to large-scale problems.

6. Conclusions

In this paper, main works are summarised as follows.

(1) This study considered batches and transportation in the IPPS problem simultaneously. The equal batch splitting strategy is used to split the batch of each workpiece. A BV-IPPS problem considering equal batch splitting and a limited number of vehicles is proposed. The makespan is taken as the objective to be minimized, and a mathematical model for the BV-IPPS problem is established.

(2) Due to the complexity and discreteness of the BV-IPPS problem, a PSO within a multilayer encoding structure is proposed to solve the problem. Each module of the algorithm is designed, and a flowchart of the algorithm is given.

(3) The above model and algorithm are validated experimentally. The results show that the makespan obtained

Table 8: Best solution.

J_i	4	1	4	1	5	4	3	5	1	5	4	5	2	2
JS_{il}	*43*	11	42	***12***	51	*43*	31	51	11	51	41	52	21	21
BS_{il}	10	10	10	10	10	10	10	10	10	10	10	10	10	10
M_j	4	4	3	4	3	2	5	3	2	4	1	6	6	6
W_g	*2*	2	2	***2***	2	*1*	3	2	1	2	1	3	3	3
TMJ_{ilkvj}	0.9	1	1.1	1	0.9	0.7	0.8	1	0.7	0.7	1.2	0.8	0.9	0.7
R_k	10	1	8	2	11	10	7	11	1	11	10	11	5	5
V_r	–	1	–	***2***	5	*1*	3	–	2	–	4	–	4	–
J_i	3	5	4	1	4	5	3	1	5	1	4	5	4	4
JS_{il}	31	52	43	11	42	$\underline{52}$	31	***12***	$\underline{52}$	12	41	51	41	42
BS_{il}	10	10	10	10	10	10	10	10	10	10	10	10	10	10
M_j	5	1	3	3	5	1	2	1	2	2	5	6	3	3
W_g	3	1	2	2	3	$\underline{1}$	1	***1***	$\underline{1}$	1	3	3	2	2
TMJ_{ilkvj}	0.8	1	0.6	0.9	0.9	0.8	0.9	1.2	0.9	1	1	0.7	0.6	0.6
R_k	7	11	10	1	8	11	7	2	11	2	10	11	10	8
V_r	–	6	1	2	5	–	3	***4***	$\underline{–}$	–	6	1	5	6

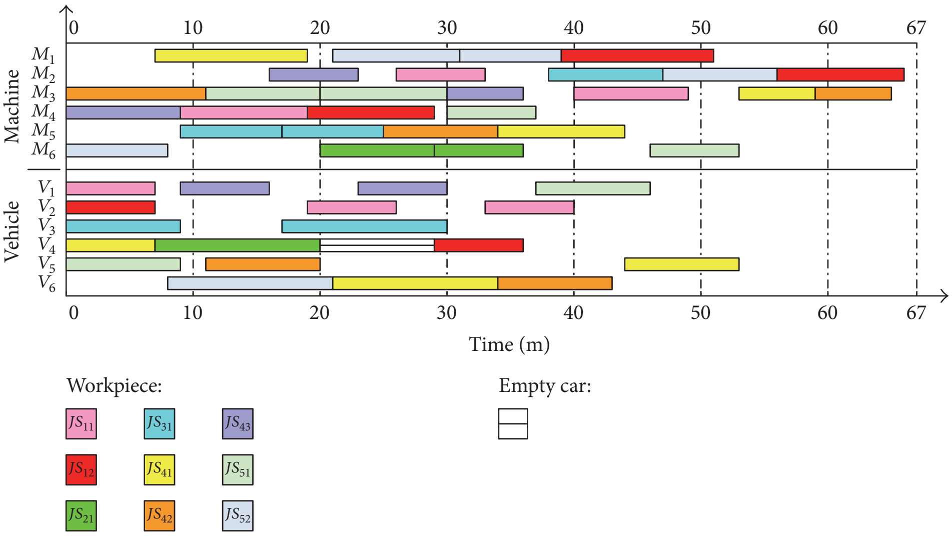


Figure 7: Gantt chart of the best solution by adopting equal batch splitting and setting the maximum lot number equal to four.

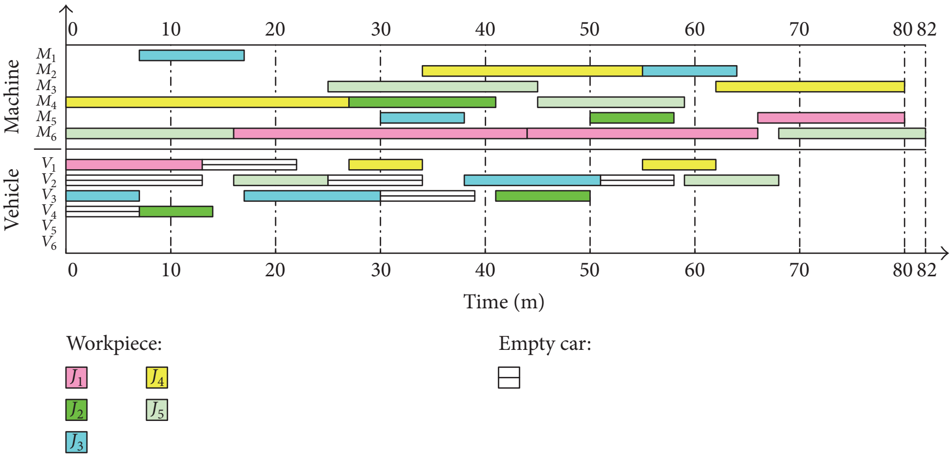


Figure 8: Gantt chart of the best solution without considering batch splitting.

Table 9: Two samples of Experiment 2.

(a)

Parameters of Sample 1											
b_1	b_2	b_3	b_4	b_5	n	m	t	w	p	VRN	VRT
20	10	10	30	20	5	6	6	3	13	2–4	0.6–1.4

(b)

Parameters of Sample 2																
b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	n	m	t	w	p	VRN	VRT
40	10	10	30	40	20	10	30	40	20	10	10	10	5	20	3–5	0.5–2.0

Table 10: Algorithm parameters of the two samples.

Algorithm parameters for Sample 1							Algorithm parameters for Sample 2						
D	M	Alpha	Beta	Pm	Ps	Pa	D	M	Alpha	Beta	Pm	Ps	Pa
300	500	0.5	0.5	0.1	0.5	0.5	400	800	0.5	0.5	0.1	0.5	0.5

Table 11: Results of the different batch splitting strategies.

(a)

Results of Sample 1					
Batch splitting strategy	Equal	Fixed	Fixed	Fixed	Not splitting
Number of lots	Max = 4	4	3	2	-
Makespan of the best solution (m)	47.7	48.2	48.6	50	64

(b)

Results of Sample 2						
Batch splitting strategy	Equal	Fixed	Fixed	Fixed	Fixed	Not splitting
Number of lots	Max = 5	5	4	3	2	-
Makespan of the best solution (m)	143.1	165.2	153.9	143.3	142	164

by adopting an equal batch splitting strategy in the BV-IPPS problem is superior to that obtained with no splitting. Additionally, transportation is considered in the IPPS problem. The best solution obtained by scheduling vehicles and machines simultaneously is more feasible for determining the real scheduling solution than the best solution obtained by scheduling machines only.

Additionally, some future works will be carried on.

(1) The result obtained using the equal splitting batch strategy in the BV-IPPS problem is better than that obtained with the fix splitting batch strategy. However, according to the experimental results, due to the larger search space in the larger-scale samples, it may be more difficult to find a better solution with the equal splitting batch strategy. In future research, we will focus on improving the algorithm. Specifically, more advanced algorithms will be developed to address large-scale BV-IPPS problems.

(2) BV-IPPS problem is a static scheduling problem in this paper. In order to make IPPS problem more in line with practical manufacturing environment, some dynamic factors will be considered in IPPS problem. For example, new job inserting, job cancelling or machine breakdown will be considered.

Conflicts of Interest

The authors declare that they have no conflicts of financial interest.

Acknowledgments

The authors acknowledge the financial support from the Xi'an University of Technology Initial Foundation for the PhDs [102-451117013] and the National Natural Science Foundation of China (NSFC) project [61402361, 51575443].

References

[1] 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.

[2] M. Haddadzade, M. R. Razfar, and M. H. F. Zarandi, "Integration of process planning and job shop scheduling with stochastic processing time," *The International Journal of Advanced Manufacturing Technology*, vol. 71, no. 1-4, pp. 241–252, 2014.

[3] M. Petrović, N. Vuković, M. Mitić, and Z. Miljković, "Integration of process planning and scheduling using chaotic

particle swarm optimization algorithm," *Expert Systems with Applications*, vol. 64, pp. 569–588, 2016.

[4] R.-H. Huang and T.-H. Yu, "An effective ant colony optimization algorithm for multi-objective job-shop scheduling with equal-size lot-splitting," *Applied Soft Computing*, vol. 57, pp. 642–656, 2017.

[5] C. H. Martin, "A hybrid genetic algorithm/mathematical programming approach to the multi-family flowshop scheduling problem with lot streaming," *Omega-International Journal of Management Science*, vol. 37, no. 1, pp. 126–137, 2009.

[6] C. Low, C.-M. Hsu, and K.-I. Huang, "Benefits of lot splitting in job-shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 24, no. 9-10, pp. 773–780, 2004.

[7] 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.

[8] 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.

[9] G. Chryssolouris, S. Chan, and W. Cobb, "Decision making on the factory floor: an integrated approach to process planning and scheduling," *Robotics and Computer-Integrated Manufacturing*, vol. 1, no. 3-4, pp. 315–319, 1984.

[10] C. Saygin and S. E. Kilic, "Integrating flexible process plans with scheduling in flexible manufacturing systems," *The International Journal of Advanced Manufacturing Technology*, vol. 15, no. 4, pp. 268–280, 1999.

[11] R. K. Phanden, A. Jain, and R. Verma, "An approach for integration of process planning and scheduling," *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 4, pp. 284–302, 2013.

[12] V. K. Manupati, G. D. Putnik, M. K. Tiwari, P. Ávila, and M. M. Cruz-Cunha, "Integration of process planning and scheduling using mobile-agent based approach in a networked manufacturing environment," *Computers & Industrial Engineering*, vol. 94, pp. 63–73, 2016.

[13] 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.

[14] L. P. Zhang and T. N. Wong, "Solving integrated process planning and scheduling problem with constructive meta-heuristics," *Information Sciences*, vol. 340/341, pp. 1–16, 2016.

[15] K. L. Lian, C. Y. Zhang, L. Gao, and X. Y. Li, "Integrated process planning and scheduling using an imperialist competitive algorithm," *International Journal of Production Research*, vol. 50, no. 15, pp. 4326–4343, 2012.

[16] K. L. Lian, C. Y. Zhang, X. Y. 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.

[17] X. Y. Shao, X. Y. Li, L. Gao, and C. Y. Zhang, "Integration of process planning and scheduling—a modified genetic algorithm-based approach," *Computers & Operations Research*, vol. 36, no. 6, pp. 2082–2096, 2009.

[18] X. Y. Li, L. Gao, and X. Y. Shao, "Mathematical modeling and evolutionary algorithm-based approach for integrated process planning and scheduling," *Computers & Operations Research*, vol. 37, no. 4, pp. 656–667, 2010.

[19] A. Seker, S. Erol, and R. Botsali, "A neuro-fuzzy model for a new hybrid integrated process planning and scheduling system," *Expert Systems with Applications*, vol. 40, no. 13, pp. 5341–5351, 2013.

[20] Z. W. Zhang, R. Z. Tang, T. Peng, L. Tao, and S. Jia, "A method for minimizing the energy consumption of machining system: integration of process planning and scheduling," *Journal of Cleaner Production*, vol. 137, pp. 1647–1662, 2016.

[21] T. Kis, "Job-shop scheduling with processing alternatives," *European Journal of Operational Research*, vol. 151, no. 2, pp. 307–332, 2003.

[22] 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.

[23] 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.

[24] D. Giglio, M. Paolucci, and A. Roshani, "Integrated lot sizing and energy-efficient job shop scheduling problem in manufacturing/remanufacturing systems," *Journal of Cleaner Production*, vol. 148, pp. 624–641, 2017.

[25] O. Shahvari and R. Logendran, "Hybrid flow shop batching and scheduling with a bi-criteria objective," *International Journal of Production Economics*, vol. 179, pp. 239–258, 2016.

[26] E. Gerstl and G. Mosheiov, "A two-stage flow shop batch-scheduling problem with the option of using Not-All-Machines," *International Journal of Production Economics*, vol. 146, no. 1, pp. 161–166, 2013.

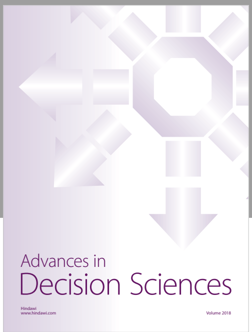
[27] J. E. C. Arroyo and J. Y.-T. Leung, "An effective iterated greedy algorithm for scheduling unrelated parallel batch machines with non-identical capacities and unequal ready times," *Computers & Industrial Engineering*, vol. 105, pp. 84–100, 2017.

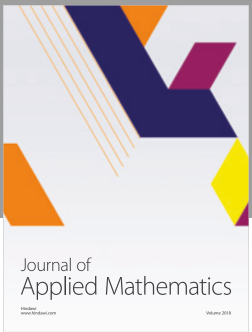
[28] Q. Zeng, Y. Yang, Y. Wang, and B. Cheng, "Multi-objective optimization scheduling for Job shop of batch production with multiple process flows," *China Mechanical Engineering*, vol. 22, no. 2, pp. 190–196, 2011.

[29] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Western Australia, November-December 1995.

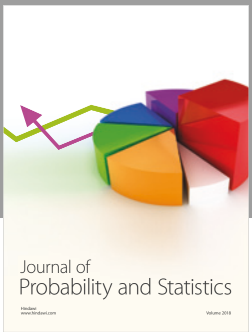
[30] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," *Computers & Operations Research*, vol. 35, no. 10, pp. 3202–3212, 2008.

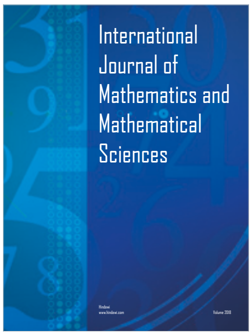
Advances in
Operations Research


Advances in
Decision Sciences

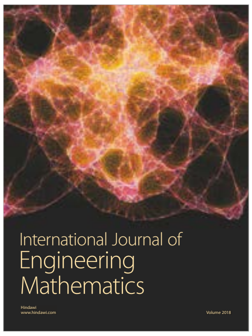

Journal of
Applied Mathematics

The Scientific
World Journal

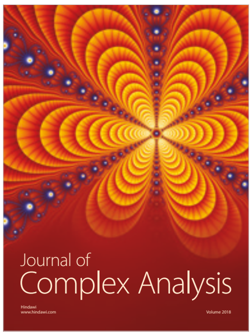

Journal of
Probability and Statistics

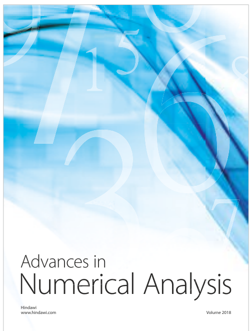

International
Journal of
Mathematics and
Mathematical
Sciences

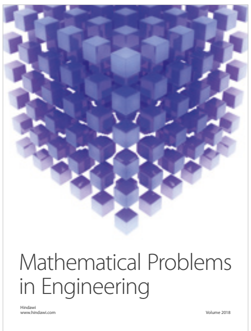
Journal of
Optimization

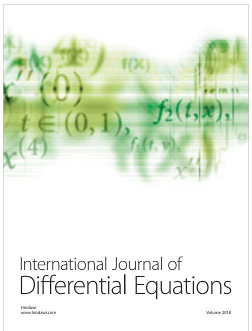

International Journal of
Engineering
Mathematics

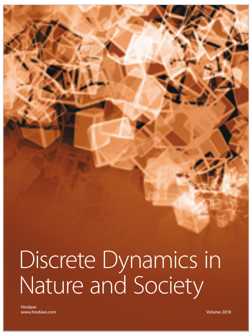
International Journal of
Analysis

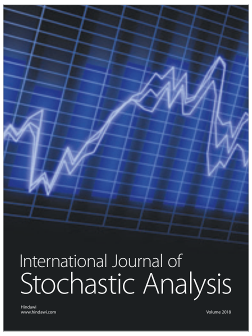

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

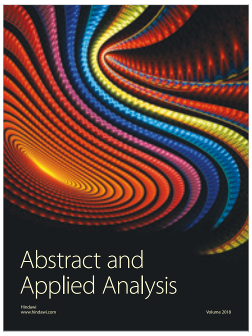

International Journal of
Differential Equations

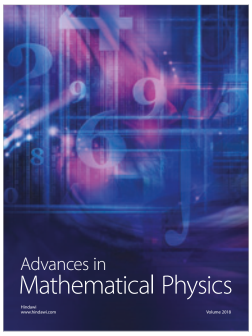

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics