Int J Adv Manuf Technol (2012) 58:727–740
DOI 10.1007/s00170-011-3409-0

ORIGINAL ARTICLE

An improved genetic algorithm for integrated process planning and scheduling

Qiao Lihong · Lv Shengping

Received: 18 March 2011 / Accepted: 26 May 2011 / Published online: 9 June 2011

Abstract Process planning and scheduling are two of the most important functions involved in manufacturing process and they are actually interrelated; integration of the two is essential to improve the flexibility of scheduling and achieve a global improvement for the performance of a manufacturing system. In order to facilitate the optimization of process planning and scheduling simultaneously, a mathematical model for the integrated process planning and scheduling (IPPS) is established, and an improved genetic algorithm (IGA) is proposed for the problem. For the performance improvement of the algorithm, new initial selection method for process plans, new genetic representations for the scheduling plan combined with process plans and genetic operator method are developed. To verify the feasibility and performance of the proposed approach, experimental studies are conducted and comparisons are made between this approach and others with the makespan and mean flow time performance measures. The results show that the proposed approach on IPPS has achieved significant improvement in minimizing makespan and obtained good results for the mean flow time performance measure with high efficiency.

Keywords Process planning · Scheduling · Integrated process planning and scheduling · Improved genetic algorithm

Q. Lihong (✉) · L. Shengping
Advanced Manufacturing Technology and Systems Research Center, Department of Industrial and Manufacturing Systems Engineering, Beihang University,
Division 720, 37 Xueyuan Road, Haidian District,
100191 Beijing, People's Republic of China
e-mail: lhqiao@buaa.edu.cn

1 Introduction

Process planning and scheduling are two important manufacturing planning activities which can greatly affect the performance of manufacturing systems. These two functions are interrelated for both process planning and scheduling involve assignment of resources. However, conventionally, these functions are performed separately and sequentially. The process plan has to be prepared first; scheduling will then be performed to allocate manufacturing resources according to the process plan [1]. This sequence methodology does not take into account the flexibility of process plan which is essential for manufacturing system and always directly influence the performance of scheduling. Meanwhile, it may also bring some other problems, such as objective conflicting between the process planning and scheduling, unbalance for production machines and the infeasibility for process plan after it has been sent to productions systems for the manufacturing is dynamic in nature.

In response to these problems, Chryssolouris et al. [2] first proposed the idea to integrate the process planning and scheduling. In recent years, the integrated process planning and scheduling (IPPS) has been extensively researched over the viewpoint of framework and model [3–7], system building [8, 9], optimization approaches [10–16], and so on. The results of these researches show that the integration of the two functions may bring significant improvements to the efficiency of the manufacturing facilities through elimination or reduction in scheduling conflicts, reduction of flow time and work-in-process, improvement of production resources utilization, and adaptation to irregular shop floor disturbances [10, 11]. Therefore, the integration of process planning and scheduling is essential

to achieve a global improvement for the performance of a manufacturing system.

The IPPS problem discussed in most of these papers and will be discussed in this paper can be defined as: given a set of N jobs which are to be processed on M machines with alternative operations sequences and alternative machines for operations, find an operations sequence and corresponding machines sequence for each job and a schedule in which operations on the same machines are processed such that it satisfies the precedence constraints and it is optimal with respect to some relevant criteria, e.g., minimum makespan and minimum mean flow time.

These optimization criteria are the same as those for conventional scheduling; however, the process of integrated optimization is to determine the scheduling plan for all jobs and process plan for each job collaboratively from the view of manufacturing systems, which is different from the conventional optimization method that optimize process planning and scheduling separately and locally. The result of the integration is to help the schedule planners to determine more optimal scheduling plans and assist the process planers to determine the final process plan (including chosen operations, machine for each operation, and the sequence of operations) for each job that will be passed on to the production systems from some alternative process plans simultaneously. The final chosen process plan is the optimized one for the manufacturing systems but it may not be the best one for the process planning.

The remainder of the paper is organized as follows. Section 2 summarizes the related work of the problem. The problem of IPPS is discussed in Section 3. Section 4 gives the improved genetic algorithm (IGA) for the IPPS. In Section 5 the experiments are conducted and results are discussed. The conclusions are given in Section 6.

2 Related work

The IPPS defined in this paper is a nondeterministic polynomial time-hard problem and it is always difficult to find perfect solutions in reasonable time. Therefore, it inspires a lot of scholars to create new approaches for the problem. Agent- and algorithm-based methods are the two main approaches for it.

Agent-based approach has been widely used for the problem. Lim and Zhang [12, 13] constructed a multi-agent-based framework for the IPPS problem. This framework can also be used to optimize the utilization of manufacturing resources dynamically as well as provide a platform on which alternative configuration of manufacturing systems can be assessed. Wong et al. [14–16] proposed an online hybrid agent and an online multi-agent method in their serial articles and they also incorporated an ant colony algorithm to optimize the problem [17]. Shukla et al. [18] conceptualized a bidding-based multi-agent system for the problem, the proposed architecture consists of various autonomous agents capable of communicating (bidding) with each other and making decisions based on their knowledge; meanwhile, they gave a new paradigm by taking the tool cost as a dynamic quantity for the integration optimization. Li et al. [19] utilized the communication mechanism of agent and the optimization function of genetic algorithm to solve the problem. The agent-based approach can always optimize the process planning and scheduling simultaneously for the communication and cooperation essence of agent, but it is always with low efficiency; for the communication and negotiation between agents, it should take long time especially for large-scale problems.

Algorithm-based method is another important mechanism for the integrated problem. Weintraub et al. [20] presented an algorithm incorporated with tabu search procedure to reduce maximum lateness for the problem. Tan et al. [21] developed a linearized polynomial mixed-integer programming method for IPPS. Li and McMahon [22] proposed a simulated annealing-based algorithm to determine the process plans and scheduling simultaneously. Li et al. [23] used a hybrid method for the IPPS problem. Except for these algorithms, the evolutionary algorithm is a very popular algorithm-based approach that has been used for the problem by different scholars such as Shao et al. [10, 24], who proposed an improved framework and a modified genetic algorithm for the integration problem. However, the algorithm seldom deliberate the sequencing flexibility of process plan, which is essential for a manufacturing system, while generating the process plan from the flexible process plan network. Moon et al. [25–27] presented an advanced process planning and scheduling model for multiplant and used an evolutionary algorithm to solve the model with different objectives. Kim [28] used a distributed cooperation evolution method referred to as symbiotic evolutionary algorithm (SEA) to optimize the process planning and scheduling problem simultaneously.

The main focus of our work is to formulate a mathematical model that incorporates the machine selection and operation sequencing in process planning and the determination of their schedule and to develop an IGA with new initial selection method for process plans, new genetic representations for the scheduling plan combined

with process plans, and genetic operator method to optimize the IPPS with the makespan and mean flow time performance measures.

3 Integration of process planning and scheduling

3.1 Representation of flexible process plans

The flexible process plans are normally generated by machine substitutability, operation substitutability, and sequencing substitutability [29]. Machine substitutability relates to the possibility of performing an operation on alternative machines with possibly distinct processing times and costs. Operation substitutability is determined by the possibility of producing the same manufacturing feature with alternative operations or sequences of operations. Sequencing substitutability corresponds to the possibility of interchanging the sequence in which manufacturing operations required are performed.

Flexible process plans can be described by Petri network [30], operation relationship graph ORG [31], AND/OR network graph and so on [32]. The AND/OR network graph is adopted in this paper. There are six node types in the network: starting node, operation node, ending node, OR node, AND node, and JOIN node (see Fig. 1). A starting node and an ending node are dummy ones and, respectively, indicate the start and the completion of the manufacturing process of a job. An operation node represents an operation which contains the alternative machines that can perform the operation and corresponding processing times required for the operation. The arrows connecting the nodes represent the precedence relation between them. OR nodes are used in order to describe operation substitutability. Only one of the OR-links (the links connected by an OR node are called OR-links) needed to be traversed. An operation path that begins at an OR-link and ends as it merges with the other paths is called an OR-link path. The end of the OR-link path is denoted by a JOIN node. AND nodes are used in order to describe sequencing substitutability that the sequence of operations in different AND-link path can be interchanged while satisfying the precedence constraints in the same AND-link path. An AND-link path is an operation path that begins at an AND-link (the links connected by an AND node are called AND-links) and ends as it merges with the other paths to a JOIN node.

Figure 1 shows two jobs alternative process plan networks (jobs 1 and 2), in the network of Fig. 1b, paths {6, 7, 8, 9}, {6, 7, 10, 9} and {11, 12, 13} are three OR-link paths of OR node 1. An OR-link path can definitely contain the other OR-link paths, e.g., paths {8} and {10} in Fig. 1b {1, 2, 5, 3, 4, 6, 7, 8, 9, 15, 17, 16, 18, 19, 20} can be one of the feasible process plans while taking AND-link path and OR-link path into account.

3.2 The mathematical model

The scheduling in the integrated model is often assumed as job shop scheduling, and the following assumptions are made [24, 28].

1. Job preemption is not allowed and each machine can handle only one job at a time.
2. All jobs are simultaneously available at time zero.
3. The different operations of one job cannot be processed simultaneously.
4. After a job is processed on a machine, it is immediately transported to the next machine on its routing, and the transportation time is ignored.
5. Setup times for the operations on the machines are independent of operation sequence and are included in the processing time.

In order to ease the description of the integrated model and related algorithm in this paper, the notations and formations in this paper are defined as follows.

Notations

M	The total number of machines
N	The total number of jobs
i	The index for the job
j	The index for the process plan
k	The index for the operation
m	The index for the machine
N_i	The total number of process plans of job i
N_{ij}	The total number of operations in the jth alternative process plan of job i
a_i	The index for AND node of job i
or_i	The index for OR node of job i
s	The number of near optimal process plans selected for integrated optimization
o_{ijk}	The kth operation in the jth alternative process plan of job i
o_{ik}	The kth operation of job i
o^m_{ijk}	The kth operation in the jth alternative process plan of job i performed on machine m
p^m_{ijk}	The processing time of o^m_{ijk}
OA_{ai}	The operation set in the AND link paths split from a_i
x^m_{ijk}	0,1variable, equals to 1 if O_{ijk} is manufactured on machine m, otherwise equals to 0;

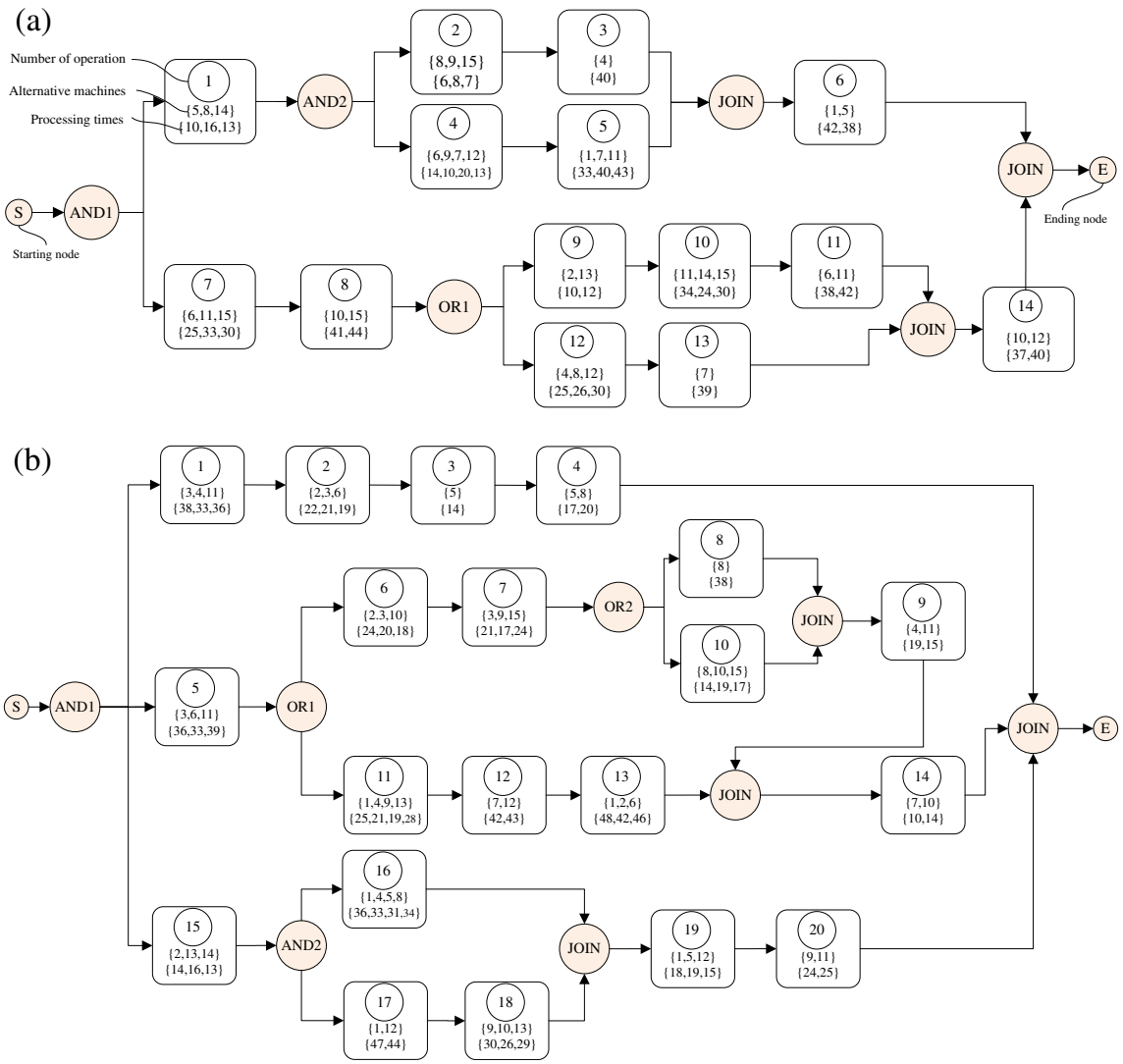

Fig. 1 Flexible process plan networks. **a** Flexible process plan network of job 1. **b** Flexible process plan network of job 2

x_{ij} Equals to 1 if the jth process plan of job i is selected, otherwise equals to 0

M_{ijk} The alternative machine set for O_{ijk}

M_{ik} The alternative machine set for O_{ik}

m_{ijk} The chosen machine of o_{ijk} in a scheduling plan

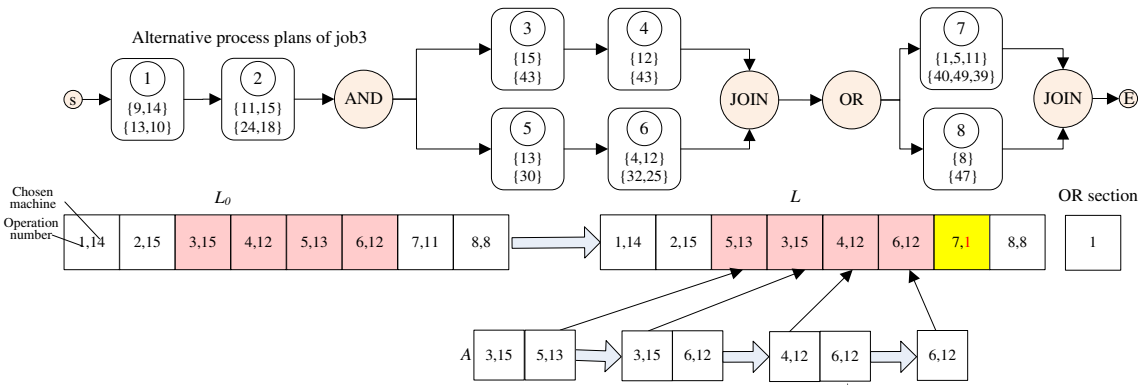

Fig. 2 Instance of generating an near-optimal process plan

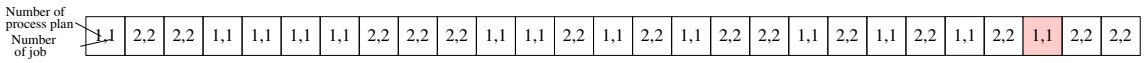

Fig. 3 Individual of scheduling plan integrated with process plan

u_i The total number of operations in the longest of process plan(s) of job i

f_i A middle integer number equals to $f_i = \text{Round}(\frac{u_i}{10}) + 1$

es_{ijk}^m The earliest start time of o_{ijk}^m

ec_{ijk}^m The earliest complete time of o_{ijk}^m,$\text{ec}_{ijk}^m = \text{es}_{ijk}^m + p_{ijk}^m$

c_i The earliest complete time of job i

d_i The due date of job i

ac_{ijk}^m The allowed start time of o_{ijk}^m

$[i\_s, i\_\text{e}]^m$ an idle area of machine m, i_s is the start time of the idle area, $i_$e is the end time of the idle area

O_m The operation set that manufactured on machine m (the sequence of different operations is determined) and defined as $O_m = \{o_{ijk}^m\}, 1 \le m \le M, 1 \le i \le N, 1 \le j \le N_i$;

N_{or_i} The total number of OR node in the process plan network of job i

N_{a_i} The total number of AND node in the process plan network of job i

W An arbitrary large positive number

Define "=>" as the precedence relation notation, it means operation $o_{i_1j_1k_1}$ precedes $o_{i_2j_2k_2}$ if exist $o_{i_1j_1k_1} => o_{i_2j_2k_2}, \forall i_1, i_2 \in [1,N], \forall j_1 \in [1,N_{i_1}], \forall j_2 \in [1,N_{i_2}], \forall k_1 \in [1, N_{i_1j_1}], \forall k_2 \in [1,N_{i_2j_2}]$.

This paper summarizes the main objectives as follows.

The objective of minimizing makespan can be defined as:

$$f_1 = \max_{\forall i \in [1,N]} (c_i) \tag{1}$$

The objective of minimizing mean flow time can be defined as:

$$f_2 = \min \frac{1}{N} \sum_{i=1}^{N} c_i \tag{2}$$

The objective of minimizing the maximum of lead time can be defined as:

$$f_3 = \min \left(\max_{\forall i \in [1,N]} (\max(d_i - c_i, 0)) \right) \tag{3}$$

The objective of minimizing the maximum tardiness time can be defined as:

$$f_4 = \min \left(\max_{\forall i \in [1,N]} (\max(c_i - d_i, 0)) \right) \tag{4}$$

The objective of minimizing the mean tardiness can be defined as:

$$f_5 = \min \frac{1}{N} \sum_{i=1}^{N} \max((c_i - d_i), 0) \tag{5}$$

The objective of maximizing resource utilization can be defined as:

$$f_6 = \max \left(\frac{\sum_{m=1}^{M} \sum_{o_{ijk} \in O_m} p_{ijk}^m}{\max_{i \in [1,N]} (c_i)} \right) \tag{6}$$

The objective of minimizing the total load of machines can be defined as:

$$f_7 = \min \left(\sum_{m=1}^{M} \sum_{o_{ijk} \in O_m} p_{ijk}^m \right) \tag{7}$$

The objective of maximize the balance of machines can be defined as:

$$f_8 = \min \sum_{m=1}^{M} \left| \sum_{o_{ijk} \in O_m} p_{ijk}^m - \frac{1}{M} \sum_{m=1}^{M} \sum_{o_{ijk} \in O_m} p_{ijk}^m \right| \tag{8}$$

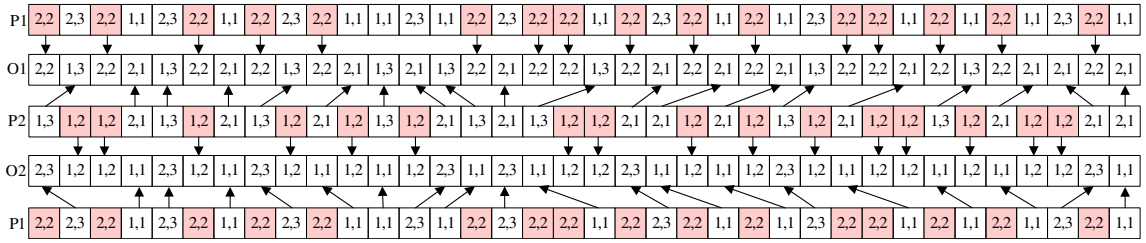

Fig. 4 Crossover for two chromosomes

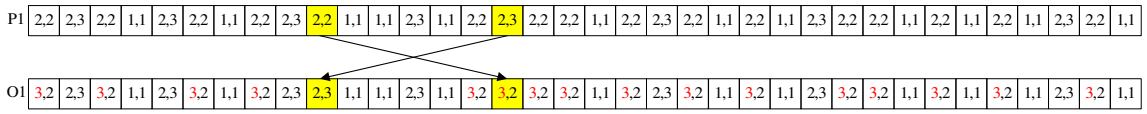

Fig. 5 Mutation for chromosome

These objectives are subjected to the following constraints.

$$\mathrm{es}^{m'}_{ijk'} \times x_{ij} \times x^{m'}_{ijk'} - \mathrm{es}^{m}_{ijk} \times x_{ij} \times x^{m}_{ijk} + W \times (1 - x_{ij}) + W \times (1 - x^{m'}_{ijk'}) \geq p^{m}_{ijk} \times x_{ij} \times x^{m}_{ijk},$$
$$\forall i \in [1, N], \forall j \in [1, N_i], \forall k', k \in [1, N_{ij}], \forall m' \in M_{ijk'}, \forall m \in M_{ijk}, o_{ijk} => o_{ijk'}. \tag{9}$$

$$\mathrm{es}^{m}_{i_2 j_2 k_2} \times x_{i_2 j_2} - \mathrm{es}^{m}_{i_1 j_1 k_1} \times x_{i_1 j_1} + W \times (1 - x_{i_2 j_2}) \geq p^{m}_{i_1 j_1 k_1} \times x_{i_1 j_1}, \forall i_1, i_2 \in [1, N], \forall j_1 \in [1, N_{i_1}], \forall j_2 \in [1, N_{i_2}],$$
$$\forall k_1 \in [1, N_{i_1 j_1}], \forall k_2 \in [1, N_{i_2 j_2}], \forall o_{i_1 j_1 k_1}, o_{i_2 j_2 k_2} \in O_m, m \in [1, M], o^{m}_{i_1 j_1 k_1} => o^{m}_{i_2 j_2 k_2} \tag{10}$$

$$\sum_{j=1}^{N_i} x_{ij} = 1, \forall i \in [1, N] \tag{11}$$

$$\sum_{m \in M_{ijk}} x^{m}_{ijk} = 1, \forall i \in [1, N], \forall j \in [1, N_i], \forall k \in [1, N_{ij}] \tag{12}$$

$$es^{m}_{ijk} \times x_{ij} \times x^{m}_{ijk} \geq 0, \forall i \in [1, N], \forall j \in [1, N_i], \forall k \in [1, N_{ij}], \forall m \in M_{ijk} \tag{13}$$

$$(o_{i_2 j_2 k_2} => o_{i_1 j_1 k_1}) \wedge (o_{i_1 j_1 k_1} => o_{i_2 j_2 k_2}) \rightarrow false$$
$$\forall i_1, i_2 \in [1, N], \forall j_1 \in [1, N_{i_1}], \forall j_2 \in [1, N_{i_2}], \forall k_1 \in [1, N_{i_1 j_1}], \forall k_2 \in [1, N_{i_2 j_2}]. \tag{14}$$

$$c_i \leq d_i, \forall i = [1, N] \tag{15}$$

Formula (9) is the operation constraints indicating two operations of a job cannot be manufactured simultaneously, and the constraints (10) means a machine cannot manufacture two operations at the same time. Constraint (11) expresses only one alternative process plan can be selected for job i. Constraints (12) means only one machine for each operation should be selected. Constraint (13) ensures that the start time of each operation should be either positive or zero. Constraint (14) indicates that there is only one precedence relation between two operations in a scheduling plan. Constraint (15) is the due date constraints for each job. This paper tends to minimize the performance measure of makespan and mean flow time.

4 Improved genetic algorithm

The genetic algorithm has been widely used for the combination optimization problem and also been used for the IPPS problem [10, 23, 25–27] for it is easy to realize and can obtain good result with high efficiency. In order to improve the performance of the genetic algorithm, an improved genetic algorithm is proposed for the IPPS based on the mathematical model. The main flow of the algorithm for IPPS in this paper mainly includes three steps as follows.

1. Generating alternative process plans and constructing the process plan network for each job in CAPP system according to the ideal resource information of job shop.
2. Generating s process plans, in which 1 process plan is the shortest one and the s-1 process plans are the near optimal ones, by the initial selection method for process plans and sent them to scheduling system. The shortest process plan is the one that every operation in the process plan chooses the machine with shortest processing time and every OR-link path with minimum operation nodes.
3. Applying the IGA to determine the process plan (selected from the s near optimal process plan) for each job and the scheduling scheme simultaneously for the production system.

Table 1 Experimental results of experiment 1

	Lee [30]	Leung [34]	Wong [17]	IGA
Makespan	439	390	380	360

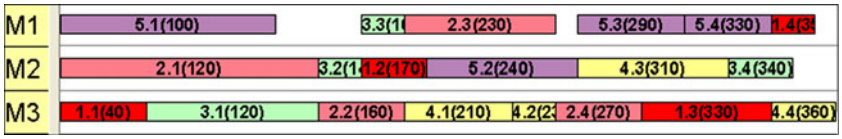

Fig. 6 Gantt chart for experiment 1

The initial selection method for process plans, genetic representations for the scheduling plan combined with process plans, genetic operator, and decoding of IGA for IPPS are described in the following.

4.1 Initial selection method for process plans

In order to generate a feasible and near-optimal initial individual for process plan from the process plan network that can include all the possible combination, this paper proposes an initial process plan selection method to generate the s-1 near optimal individuals from all existing feasible process plans as the following algorithm.

Step 1: Parse the OR-link and AND-link path from the top down based on the process plan network and generates an initial string L_0 with each operation choosing a machine with the shortest processing time. Then the cell in L_0 is made up of operation number and the chosen machine number.

Step 2: Adjust positions of nodes in AND-link paths.

Step 2.1: Find the positions of operations in the link paths of $a_i, 1 \le a_i \le N_{a_i}$ AND node to construct an operation set as $\mathrm{PS}_{a_i} := \{\mathrm{ps}_1, \mathrm{ps}_2, \ldots\}$, and record the first node for all its links as $A := \{o_{ik}, o_{i(k-1)} \notin OA_{a_i}, o_{i(k-1)} => o_{ik}\}$. Meanwhile, generate a new empty string L with the same length as L_0, and copy all the cells of L_0 except the cells at the positions in PS_{a_i} to L.

Step 2.2: Choose a node o_{ik} having no predecessors randomly from A and locate it at the first empty position in L, then update $A{:=}A/\{o_{ik}\}$. If o_{ik} exists direct immediate successor operation $o_{i(\mathrm{k}+1)}$, $o_{i(k+1)} \in OA_{a_i}$, update $A{:=}A \cup \{o_{i(k+1)}\}$. Iterate this step until all the operations in the AND-link paths already have been located in the string L.

Step 2.3: Set $L_0{:=}L$, $a_i{=}a_i{+}1$. If $a_i = N_{a_i}$, go to Step 3; otherwise, return to Step 2.1.

Step 3: Choose a discrimination integer number randomly for each OR node to form a N_{or_i} substring named as OR section and append it to L.

Step 4: Choose an operation o_{ik} randomly from L and alter its machine to another one from M_{ik}, if$|M_{ik}| \ge$. 1. Set $f_i{=}f_i{-}1$, if $f_i{\ge}1$ return to Step 4, otherwise terminate.

The 1 shortest individual process plan can also be generated by the algorithm, the difference is that the choosing of OR-link path in Step 3 can only be restricted to the path with minimum operation nodes, and the fluctuation of machine in Step 4 is not needed.

The initial selection method takes all the possible flexibility of process plan into consideration and it can easily obtain different near-optimal process plan by the fluctuation of OR-link path or operation machines. This is different from other initial algorithm that seldom considers sequencing substitutability [19, 23] or to repeat the generating process for each initial process plan [25–27]. Figure 2 is an example of generating an initial individual L for job 3. First generating a string L_0 and get the positions of operations (3, 4, 5, 6; marked with red) in the AND-link paths. Then update the first node set A repeatedly. While the OR nodes has two links, and the discrimination integer number can be chosen from [1, 2], the only gene in substring of OR section is chosen 1 that means the 1-link path including operation 7 is chosen. The cell marked with yellow in L is the operation with changed machine from 11 to 1 (marked with red) and its processing time is changed from 39 to 40 correspondingly.

4.2 Encoding and decoding of integrated schedule plan

The individual chromosome in the population is encoded by operation-based method, which is represented by the permutation of job number and its related chosen process plan. Therefore, each cell in individual constructs a gene as a data structure which is made up of two positive integer numbers. The first one, which can be selected from [1, s], is

Table 2 The comparison of makespan and mean flow time

IGA		Jain [36]		Wong [15]	
Makespan	Mean flow time	Makespan	Mean flow time	Makespan	Mean flow time
5,998	3,992	6,456	4,216	6,574	4,240

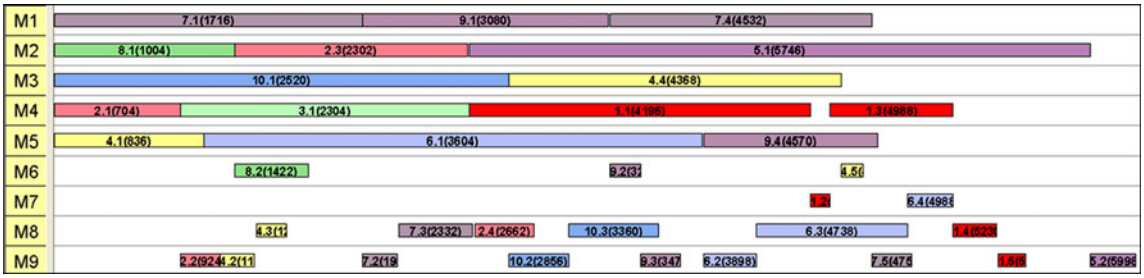

Fig. 7 Gantt chart of experiment 2 (makespan=5,998, mean flow time=3,992)

the number of the chosen process plan for the job represented by the second number in the same gene, and the second one is the number of a job. The first number should be the same for these genes with the same job number. Each individual in the population is an integrated plan that combines the process plan and scheduling plan information together by the above representation method. The appearances frequency of a job number is determined by the corresponding chosen process plan encoded by the first number in each gene. The length of the individual may be inconsistent for the number of the valid operations (all the operations except the operations in the non-chosen OR-link paths for each job) in different process plans may be different (the length of OR-link path may be different for the OR node). In order to ensure chromosomes in the population with the same length, allotting u_i cells for each job i, $1 \le i \le N$ previously, therefore, the length of each chromosome is $\text{len} = \sum_{i \in [1,N]} u_i$ for the IPPS with N jobs.

Figure 3 is an example of the individual for jobs 1 and 2 in Fig. 1, and the chosen process plan for jobs 1 and 2 is 1 and 2, respectively. For the number of operation in the longest process plan(s) of job 1 is 12 and job 2 is 16; therefore, the length of the integrated plan is equals to 28.

Bierwirth et al. [33] presented the procedures of decoding the permutations into semi-active, active, non-delay, and hybrid schedules. The active decoded is adopted in this paper. Assumed that the j th,, $1 \le j \le N_i$ process plan has been chosen for job i, $1 \le i \le N$, and the machine for each operation has also been determined, then the decoding can be described as the following algorithm.

Step 1: Determine the operation, machine and the corresponding time for each gene in turn for the selected chromosome based on these chosen process plans and the job number. If the number of valid operations for the chosen process plan of job i is less than u_i, then decode the operation number, machine number, and the corresponding time as 0 for the frequency of the job number that exceed the number of valid operations in the chosen process plan. Figure 3 shows an individual of this example, in this example, supposed the process plan 1 of job 1 choosing the second OR-link path (including operation 12 and 13) and the process plan only includes 11 operations which is less than its maximum operations 12, then the operation number, machine number, and the corresponding time are decoded as 0 for the gene marked with red.

Step 2: Construct the set of all valid operations $Z =: \left\{ o_{ijk}^m | 1 \le i \le N, 1 \le j \le N_i, \quad 1 \le k \le N_{ij}, 1 \le m \le M, p_{ijk}^m \ne 0 \right\}$ based on the permutation sequence of the selected chromosome.

Step 3: Select the first operation o_{ijk}^m from Z and compute the allowable starting time for the operation: as $_{ijk}^m = \text{ec}_{ij(k-1)}$, $\text{ec}_{ij(k-1)}$ is the completion time of the pre-operation of o_{ijk}^m for the same job, it is initialized as 0 if it is the first operation of a job.

Step 4: Scan and check the idle area $[i\_s, i\_e]^m$ for o_{ijk}^m in turn, if there is $\max(\text{as}_{ijk}^m, t\_s) + p_{ijk}^m) \le i\_e$, then the earliest starting time of o_{ijk}^m can be set as $\text{es}_{ijk}^m = \max(as_{ijk}^m, t\_s)$, $\text{ec}_{ijk}^m = \text{es}_{ijk}^m + p_{ijk}^m$. If there

Table 3 The comparison of the results of operations sequences

Job	Moon [27] (makespan=16)	MGA [10] (makespan=14)	IGA (makespan=14)
1	1 (M2)–2 (M2)	1 (M1)–2 (M2)	1 (M2)–2 (M2)
2	3 (M4)–4 (M5)	3 (M4)–4 (M5)	3 (M4)–4 (M5)
3	5 (M1)–6 (M3)–7 (M2)	7 (M2)–5 (M2)–6 (M3)	5 (M1)–6 (M4)–7 (M2)
4	8 (M3)–9 (M4)	8 (M3)–9 (M4)	8 (M3)–9 (M4)
5	10 (M1)–11 (M3)–12 (M1)–13 (M5)	12 (M3)–10 (M1)–11 (M1)–13 (M5)	12 (M3)–10 (M1)–11 (M1)–13 (M5)

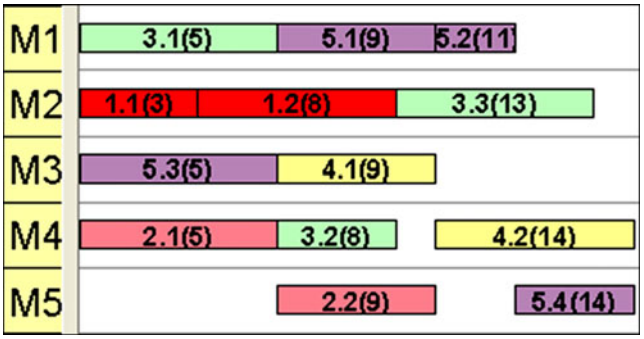

Fig. 8 Gantt chart of experiment 3 (makespan=14)

is no idle area satisfying this condition: $es^m_{ijk} = \max(as^m_{ijk}, t_m(o^m_{i_1j_1k_1}))$ (assumed $o^m_{i_1j_1k_1}$ is the pre-operation of o^m_{ijk} performed on the same machine m, and $t_m(o^m_{i_1j_1k_1})$is the available time of machine m which equals to the completion time of $o^m_{i_1j_1k_1}$). If $as^m_{ijk} \geq t_m(o^m_{i_1j_1k_1})$, update $ec^m_{ijk} = as^m_{ijk} + p^m_{ijk}$, $t_m = ec^m_{ijk}$ and $as_{ij(k+1)}$ applying the formula in step 3, meanwhile, add a new idle area $[i\_s, i\_e] = [t_m(o^m_{i_1j_1k_1}), as^m_{ijk}]$ for m. Otherwise, for $as^m_{ijk} < t_m(o^m_{i_1j_1k_1})$, update $ec^m_{ijk} = t_m(o^m_{i_1j_1k_1}) + p^m_{ijk}$, $t_m = ec^m_{ijk}$, and $as_{ij(k+1)}$ applying the formula in step 3.

Step 5: Update $Z := Z/\{o^m_{ijk}\}$. If $Z \neq \varnothing$, go to step 3. Otherwise, terminate.

The above decoding method is to decode the chromosome from the first gene to the last one in turn and try to utilize the idle area of each machine. In the above procedure, the starting time and completion time of all the operations in the chromosome are determined and it is a scheduling plan for the shop.

4.3 Initial population and fitness evaluation

The initial population is generated based on the encoding principle as described in the generic representation method. The optimization objectives in this study are makespan and mean flow time, which have been defined in formula (1) and (2), respectively, and they are subjected to (9)–(15).

4.4 Operators of improved genetic algorithm

It is vital to develop good operators that can effectively match with the problem and efficiently lead to excellent individuals in the population. Reproduction, crossover, and mutation are the main three type operators of GA algorithm

Reproduction The tournament selection method is used for reproduction operation. Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population. The tournament selection approach allows a tradeoff to be made between exploration and exploitation of the gene pool [10]. Selection pressure can be easily adjusted by changing the tournament size.

Crossover The procedure of crossover is as follows. Two empty chromosome O1, O2 are initialized first, and an integer number in [0, 1] is chosen randomly. If the chosen number is 0, then the genes with even job number in one of parent (P1) are passed on to the same positions as in the offspring (O1). These elements are removed from the other parent (P2) and the remaining elements are copied into the undetermined positions in the offspring in the same order as they appear in P2. The other offspring (O2) is made up of genes with even job numbers in one of parent (P2), these elements are removed from the other parent (P1), and the remaining elements are passed on to the undetermined positions in the offspring in the same order as they appear in P1. If the randomly chosen number is 1, only has to exchange even and odd for generating O1 and O2. Figure 4 is an example of the crossover and the randomly generated number is 0, where in order to express clearer, two P1s and P2s are figured out.

Mutation The mutation includes job position mutation and process plan mutation. Two-point mutation is used as the mutation operator for job position mutation and one-point mutation is used as the mutation operator for process plan mutation in this study. The mutation operator of the selected chromosome is operated as follows. Two genes are chosen randomly from the selected chromosome and then mutate the two genes by exchange the two positions. The mutation of process plan is carried out by choosing a mutation point randomly in the chromosome and then alter the process plan number to another new one in the range of [1, s], after this, altering each gene with the same job number as the mutated gene to the new chosen number. The mutation example is indicated in Fig. 5. The two genes marked with yellow are mutated by exchange their positions and the

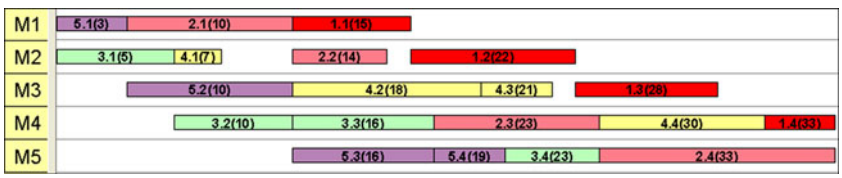

Fig. 9 Gantt chart of experiment 4 (makespan=33)

Table 4 IGA parameter

Parameters	Values
Population for makespan	600 (22, 23, 24 is 500)
Population for mean flow time	800 (22, 23, 24 is 600)
Tournament size	2
Chosen process plans *s*	4
Probability of crossover operation	0.8
Probability of mutation operation	0.15
Probability of reproduction	0.1
Termination number	100

process plan number of job 2 (marked with red) is mutated from 2 to 3 to generate a new offspring O1.

5 Experimental studies and discussions

The proposed IGA algorithm procedure is coded in JAVA and implemented on a Dell Precision T7400. In order to illustrate the effectiveness and performance of the method, five problems instances are carried out. The parameters of IGA for the experiments 1–4 are set: population for makespan and mean flow time is 200, tournament size is 2, probability of crossover operation is 0.8, probability of mutation operation is 0.15, probability of reproduction is 0.1, and termination number is 50 and the *s* is chosen as four in this paper based on some initial experiments.

5.1 Experiment 1

Experiment 1 has been adopted from Lee et al. [30], is constructed with five jobs and three machines. Each job has four operations to be performed and each operation can be done on one or more machines with respective processing times. The makespan is used as the objective. The data has also been used by Wong [17] and Leung et al. [34] in which the IPPS has been optimized by agent-based ant colony optimization algorithm. Table 1 shows the experimental results and the comparisons with other methods, the data in the column Lee, Leung and Wong are from [30, 34] and [17], respectively. Figure 6 is the Gantt chart (always used to express the scheduling result) for experiment 1 obtained by IGA.

Table 5 Results and comparisons of best makespan

Problem	Number of jobs	Job number	SEA	HA	IGA	Low bound
1	6	1, 2, 3, 10, 11, 12	428	427	427	427
2	6	4, 5, 6, 13, 14, 15	343	343	343	343
3	6	7, 8, 9, 16, 17, 18	347	345	344	344
4	6	1, 4, 7, 10, 13, 16	306	306	306	306
5	6	2, 5, 8, 11, 14, 17	319	322	304	304
6	6	3, 6, 9, 12, 15, 18	438	429	427	427
7	6	1, 4, 8, 12, 15, 17	372	372	372	372
8	6	2, 6, 7, 10, 14, 18	343	343	342	342
9	6	3, 5, 9, 11, 13, 16	428	427	427	427
10	9	1, 2, 3, 5, 6, 10, 11, 12, 15	443	430	427	427
11	9	4, 7, 8, 9, 13, 14, 16, 17, 18	369	369	368	344
12	9	1, 4, 5, 7, 8, 10, 13, 14, 16	328	327	312	306
13	9	2, 3, 6, 9, 11, 12, 15, 17, 18	452	436	429	427
14	9	1, 2, 4, 7, 8, 12, 15, 17, 18	381	380	386	372
15	9	3, 5, 6, 9, 10, 11, 13, 14, 16	434	427	427	427
16	12	1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 14, 15	454	446	433	427
17	12	4, 5, 6, 7, 8, 9, 13, 14, 15, 16, 17, 18	431	423	415	344
18	12	1, 2, 4, 5, 7, 9, 10, 11, 13, 14, 16, 17	379	377	364	306
19	12	2, 3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18	490	476	450	427
20	12	1, 2, 4, 6, 7, 8, 10, 12, 14, 15, 17, 18	447	432	429	372
21	12	2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 16, 18	477	446	433	427
22	15	2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18	534	518	491	427
23	15	1, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18	498	470	465	372
24	18	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18	587	544	532	427

5.2 Experiment 2

The data of experiment 2 are generated by Chryssolouris et al. [35] and the data constructed with 10 jobs and nine machines. The data has also been used by Wong et al. [15] and Jain et al. [36]. Table 2 is the comparison of method in Wong et al. [15] and Jain et al. [36] with makespan and mean flow time when taking mean flow time as its objective. The results of IGA win both for mean flow time and makespan. Figure 7 is the Gantt chart for experiment 2.

5.3 Experiment 3

Experiment 3 has been adopted from Moon et al. [27], and the data constructed with five jobs and five machines, and the number of operations in each jobs ranges from two to four. The best makespan 14 is obtained by MGA [10]. The comparison of the results of operations sequences with EA in Moon [27] and MGA in [10] are given in Table 3; the data in the column Moon and MGA are from [27] and [10], respectively. Figure 8 is the Gantt chart for the problem obtained by IGA.

5.4 Experiment 4

Experiment 4 use the data in Sundaram and Fu [37], which consists of five jobs and five machines. The obtained result by IGA is 33 as shown in Fig. 9. This compares with the value of 38 generated by Sundaram and Fu [37] with heuristic approach, 33 obtained by Palmer [38] with simulated annealing algorithm

5.5 Experiment 5

Large-scale IPPS problems are considered in the simulation experiment 4. The benchmark problem set used for experiment 5 have been reported in [39]. These problems are more complex as they involved 18 parts with various combinations of flexibility levels and 15 machines. Kim et al. [28] constructed 24 testbed problems with the 18 parts; meanwhile, they proposed a SEA to solve the problems efficiently. Since the 18 parts 24 testbed problems are able to reflect practical manufacturing situations of various complexities, therefore, the 24 testbed problems are chosen as benchmark cases and 10 trials of simulation are carried out for each problem that is the same as SEA. The results

Table 6 Results and comparisons of average makespan

Problems	SEA (CPU time (s))	SGA (CPU time (s))	IGA (CPU time (s))	Best	Improved rate (%)
1	437.6 (61)	436.7 (11)	427 (11)	IGA	2.4
2	349.7 (69)	361.2 (11)	344.5 (11)	IGA	1.4
3	355.2 (81)	373.1 (10)	351.0 (11)	IGA	1.2
4	306.2 (66)	322.7 (8)	307.4 (8)	SEA	−0.4
5	323.7 (64)	324.0 (8)	309.8 (8)	IGA	4.3
6	443.8 (73)	448.1 (13)	427 (13)	IGA	3.8
7	372.4 (69)	397.6 (9)	372.7 (9)	SEA	−0.1
8	348.3 (67)	384.5 (16)	357 (17)	IGA	−2.5
9	434.9 (73)	429.0 (9)	427 (9)	IGA	1.8
10	456.5 (136)	453.5 (17)	431.6 (17)	IGA	5.6
11	378.9 (166)	400.2 (17)	379.7 (16)	IGA	−0.2
12	332.8 (143)	347.5 (13)	323.7 (13)	IGA	2.7
13	469.0 (161)	463.5 (19)	442.8 (19)	IGA	5.6
14	402.4 (151)	434.7 (15)	415.3 (16)	SEA	−3.2
15	445.2 (156)	431.3 (14)	427.4 (14)	IGA	3.9
16	478.8 (334)	465.0 (23)	449.4 (23)	IGA	6.1
17	448.9 (435)	452.2 (22)	426.0 (23)	IGA	5.1
18	389.6 (357)	390.8 (20)	373.6 (20)	IGA	4.1
19	508.1 (418)	497.0 (28)	471.3 (28)	IGA	7.2
20	453.8 (384)	465.5 (26)	446.6 (26)	IGA	1.6
21	483.2 (392)	476.3 (25)	447.8 (24)	IGA	7.3
22	548.3 (1,033)	534.2 (27)	508.1 (27)	IGA	7.3
23	507.5 (1,017)	497.8 (26)	477.8 (26)	IGA	5.8
24	602.2 (1,623)	587.8 (40)	548.5 (39)	IGA	8.9

Table 7 Results and comparison of mean flow time

Problems	SEA (CPU time (s))	SGA (CPU time (s))	IGA (CPU time (s))	Best
1	318.9 (57.5)	325.4 (16)	313.3 (15)	IGA
2	287.7 (68.0)	300.3 (14)	292.2 (14)	SEA
3	304.8 (81.3)	312 (14)	307.8 (18)	SEA
4	251.3 (65.7)	263.5 (12)	258.7 (13)	SEA
5	280.3 (66.8)	270.5 (10)	263.5 (12)	IGA
6	384.7 (75.4)	389.0 (18)	382.3 (20)	IGA
7	314.1 (68.1)	323.9 (14)	317.4 (17)	SEA
8	295.2 (67.0)	309.2 (13)	305.3 (16)	SEA
9	298.9 7 (71.8)	300.6 (13)	292.7 (18)	IGA
10	349.2 (133.8)	361.3 (24)	356.4 (27)	SEA
11	312.9 (163.3)	329.4 (23)	318.6 (29)	SEA
12	279.6 (150.5)	283.1 (18)	278.6 (25)	IGA
13	387.0 (156.0)	386.9 (35)	383.2 (32)	IGA
14	346.9 (149.5)	355.9 (23)	351.6 (27)	SEA
15	316.1 (155.9)	327.8 (25)	319.7 (27)	SEA
16	359.7 (339.)	369.4 (51)	364.5 (47)	SEA
17	364.7 (438.7)	370.0 (53)	368.1 (61)	SEA
18	322.5 (355.6)	323.3 (32)	316.3 (40)	IGA
19	406.4 (416.6)	400.5 (73)	396.5 (73)	IGA
20	372.0 (337.7)	383.7 (51)	371.7 (75)	IGA
21	365.4 (364.7)	373.4 (51)	365.3 (57)	IGA
22	417.8 (1,007.6)	427.4 (43)	415.3 (34)	IGA
23	404.7 (999.3)	408.7 (37)	399.7 (35)	IGA
24	452.9 (1,597.2)	456.9 (79)	447.6 (78)	IGA

SEA symbiotic evolutionary algorithm, *IGA* improved genetic algorithm

are compared with the performance of SEA and the hybrid algorithm (HA) [23]. The IGA parameters for these problems are given in Table 4.

Table 5 shows the best result of makespan, and the comparisons between the proposed IGA, SEA in [28], and HA (tabu search combined with genetic algorithm) in [23]. The data in the column of SEA and HA are adopted from [28] and [23], respectively. In order to give a clearer comparison result for the algorithm, taking the maximum of the shortest process plans in each case as the low bound of the makespan. The results in Table 5 shows that only one solution (problem 14) of IGA is worse than SEA and HA and a few solutions of GA are equal to SEA, almost solutions (17 problems) of IGA are better than the other methods. In addition, the solutions of problems 1–10 and 15 are equals to its low bound.

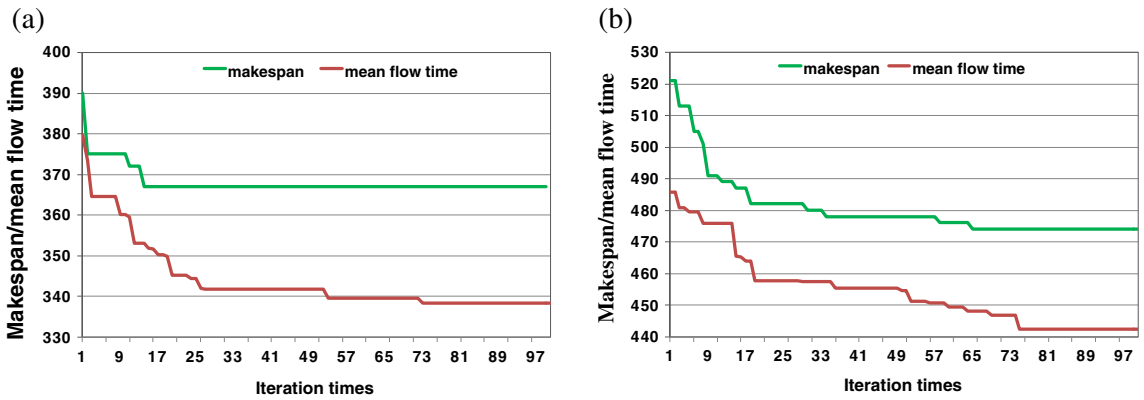

Fig. 10 Convergence curves for the problems 10 and 24. **a** Convergence curves for problem 10. **b** Convergence curves for problem 24

For the purpose of comparison, this paper also carries out the experiment for the no integrated situation for which only one shortest process plans for each job has been passed on to IGA, named as SGA. Tables 6 and 7 shows the average result of makespan and mean flow time, respectively. The comparisons between the proposed IGA, SEA in [28], SGA, and the CPU time for each problem are also included in the two tables.

From the result of Table 6, we can see that the average of makespan obtained by IGA are superior to SEA in most of the case except for the problems 4, 7, and 14. The improved rate indicates that the IGA can obtain greater scope of improvement while the scheduling scale increased. The average result of 1, 2, 4, 6, 7, and 15 reach or near to its low bound; it proves that the algorithm can achieve good average result. Additionally, it indicates that the algorithm has good robustness. The results of SGA show that it is difficult to obtain better result for the no integrated model by comparing to the integrated IGA.

Table 7 shows the IGA has no obvious superiority from the comparison results but still there 13 cases superior to SEA for the 24 cases. Meanwhile, it has obvious advantage to the no integrated situation SGA. For the algorithm are implemented on different computers, it is difficult to compare the efficiency of different algorithms. However, the CPU time of IGA in Tables 6 and 7 indicates that the algorithm is efficient even for the problem of 24 with 18 jobs and 300 operations taking only 39 and 78 s, respectively, to obtain a near-optimal result.

Figure 10 gives the convergence curves of a run for problems 10 and 24 with the objective of makespan and mean flow time. The curves of makespan in Fig. 10a, b are generated by the value of makespans 60 80. The convergence curves show that the proposed IGA obtains 427 (low bound) makespan for the problem 10 with 15 iterations and 544 for the problem 24 with 62 iterations. Meanwhile, it only takes 72 iterations to obtain the 338.3 mean flow time for the problem 10, and 75 iterations to obtain the 442.2 mean flow time for the problem 24. It indicates that the algorithm can converge to a relative optimal solution with little iteration especially for makespan.

6 Conclusions

In order to improve the flexibility of scheduling and achieve a global improvement for the performance of a manufacturing system, a mathematical model and an improved genetic algorithm have been developed in this study to facilitate the integration of process planning and scheduling and to optimize and determine both process plan and scheduling plan simultaneously. To improve the optimization performance of the proposed approach, new initial selection method for process plans; new genetic representations for the scheduling plan combined with process plans and genetic operator schemes have been developed. The initial selection method keeps the shortest process plan for each job to ensure the scheduling has the possibility to reach the low bound. Meanwhile, the integrated scheduling algorithm taking multi-optimal process plans but not all process plans as input to improve the flexibility of scheduling with smaller search space. Five experimental studies have been carried out to compare this approach with other previous methods and the no integrated model. The experiments and comparison results show that the algorithm has achieved significant improvement in minimizing makespan and obtained good results for the mean flow time performance measure with high efficiency. It verifies that this research is necessary and effective for the objective of makespan and mean flow time.

Further work can be conducted through the following three aspects:

1. The scheduling problem may be interrupted by new added orders, breakdowns, shortage of material and tooling, quality problems, machine unavailability, over/under estimation of machining time, and so on; an important further research issue is to extend the mathematical models and algorithm to these situations and make it more practical to the manufacturing workshop.
2. This paper only tried to minimize the makespan and mean flow time; additional performance measure such as the objectives defined by formula (3)–(8) should also be carried out.
3. The coordination and cooperation mechanisms and framework for the integrated problem should also be conducted in the further research.

Acknowledgments This work is supported by the National Natural Science Foundation of China (grant 51075022), Significant Science and Technology Special Program of Advanced Machine tools and Elementary Manufacturing Equipment (2010ZX04016-011, 2010ZX04015-011), and Beijing Municipal Education Commission (Build a Project). The authors would like to express their appreciation to the agency.

References

1. Jain A, Jain PK, Singh IP (2006) An integrated scheme for process planning and scheduling in FMS. Int J Adv Manuf Technol 30:1111–1118
2. Chryssolouris G, Chan S, Cobb W (1984) Decision making on the factory floor: an integrated approach to process planning and scheduling. Robot Comput Integr Manuf 3/4:315–319
3. Mamalis AG, Malagardis I, Kambouris K (1996) On-line integration of a process planning module with production scheduling. Int J Adv Manuf Technol 12:330–338

4. Zhang J, Gao L, Chan F (2003) A holonic architecture of the concurrent integrated process planning system. J Mater Process Technol 139:267–272
5. Liu M, Bai L, Zhang SS (2004) Modeling integrated CAPP/PPS systems. Comput Ind Eng 46:275–283
6. Wang LH, Song YJ, Shen WM (2007) A novel function block based integration approach to process planning and scheduling with execution control. Int J Manuf Technol Manag 11(2):228–250
7. Kumar M, Rajotia S (2006) Integration of process planning and scheduling in a job shop environment. Int J Adv Manuf Technol 28:109–116
8. Yang YN, Parsaei HR, Leep HR (2001) A prototype of a feature-based multiple-alternative process planning system with scheduling verification. Comput Ind Eng 39:109–124
9. Grabowik C, Kalinowski K, Monica Z (2005) Integration of the CAD/CAPP/PPC systems. J Mater Process Technol 164–165:1358–1368
10. Shao XY, Li XY, Gao L, Zhang CY (2009) Integration of process planning and scheduling—a modified genetic algorithm-based approach. Comput Oper Res 36:2082–2096
11. Lee H, Kim SS (2001) Integration of process planning and scheduling using simulation based genetic algorithms. Int J Adv Manuf Technol 18:586–590
12. Lim MK, Zhang DZ (2003) A multi-agent based manufacturing control strategy for responsive manufacturing. J Mater Process Technol 139:379–384
13. Lim MK, Zhang DZ (2004) An integrated agent-based approach for responsive control of manufacturing resources. Comput Ind Eng 46:221–232
14. Wong TN, Leung CW, Mak KL, Fung RYK (2006) An agent-based negotiation approach to integrate process planning and scheduling. Int J Prod Res 44(7):1331–1351
15. Wong TN, Leung CW, Mak KL, Fung RYK (2006) Integrated process planning and scheduling/rescheduling—an agent-based approach. Int J Prod Res 44(18–19):3627–3655
16. Wong TN, Leung CW, Fung RYK (2006) Dynamic shop floor scheduling in multi-agent manufacturing system. Exp System App 31:486–494
17. Leung CW, Wong TN, Mak KL, Fung RYK (2010) Integrated process planning and scheduling by an agent-based ant colony optimization. Comput Ind Eng 59:166–180
18. Shukla SK, Tiwari MK, Son YJ (2008) Bidding-based multi-agent system for integrated process planning and scheduling a data mining and hybrid tabu-SA algorithm-oriented approach. Int J Adv Manuf Technol 38:163–175
19. Li XY, Zhang CY, Gao L, Li WD, Shao XY (2010) An agent-based approach for integrated process planning and scheduling. Exp System App 37:1256–1264
20. Weintraub A, Cormier D, Hodgson T, King R, Wilson J, Zozom A (1999) Scheduling with alternatives: a link between process planning and scheduling. IIE Trans 31:1093–1102
21. Tan W, Khoshnevis B (2004) A linearized polynomial mixed integer programming model for the integration of process planning and scheduling. J Intell Manuf 15:593–605
22. Li WD, McMahon CA (2007) A simulated annealing-based optimization approach for integrated process planning and scheduling. Intl J Comput Integr Manuf 20(1):80–95
23. Li XY, Shao XY, Gao L, Qian WR (2010) An effective hybrid algorithm for integrated process planning and scheduling. Int J Prod Eco 126:289–298
24. Li XY, Shao XY, Zhang CY (2010) Mathematical modeling and evolutionary algorithm-based approach for integrated process planning and scheduling. Comput Ope Res 37:656–667
25. Moon C, Kim J, Hur S (2002) Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain. Comput Ind Eng 43:331–349
26. Moon C, Seo Y (2005) Evolutionary algorithm for advanced process planning and scheduling in a multi-plant. Comput Integ Eng 48:311–325
27. Moon C, Lee YH, Jeong CS, Yun YS (2008) Integrated process planning and scheduling in a supply chain. Comput Ind Eng 54:1048–1061
28. Kim YK, Park K, Ko J (2003) A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. Comput Ope Res 30:1151–1171
29. Qiao LH, Kao ST, Zhang YZ (2011) Manufacturing process modeling using process specification language. Int J Adv Manuf Technol. doi:10.1007/s00170-010-3115-3
30. Lee DY, Dicesare F (1994) FMS scheduling using Petri nets and heuristic search. IEEE Transa Robot Autom 10(2):123–132
31. Yang Z, Qiao L, Jiang L (1998) Improving the performances of part dispatching based on multiple process plans using graph theory. Int J Prod Res 36(7):1987–2003
32. Ho YC, Moodie CL (1996) Solving cell formation problems in a manufacturing environment with flexible processing and routing capabilities. Int J Prod Res 34:2901–2923
33. Bierwirth C, Mattfeld DC (1999) Production scheduling and rescheduling with genetic algorithms. Evol Comput 7:1–17
34. Leung CW, Wong TN, Mak KL, Fung RYK (2006) Integrating process planning and scheduling by an agent-based ant colony system. In Proceedings of the 36th International Conference on Computers and Industrial Engineering 587–596
35. Chryssolouris G, Pierce JE, Dicke K (1992) A decision-making approach to the operation of flexible manufacturing systems. Int J Flex Manuf Sys 3(4):309–330
36. Jain AK, Elmaraghy HA (1997) Production scheduling/rescheduling in flexible manufacturing. Int J Prod Res 35:281–309
37. Sundaram RM, Fu SS (1988) Process planning and scheduling. Comput Ind Eng 15:296–307
38. Palmer GJ (1996) A simulated annealing approach to integrated production scheduling. J Intell Manuf 7:163–176
39. Kim YK (2003) A set of data for the integration of process planning and job shop scheduling. Available at: http://syslab.chonnam.ac.kr/links/data-pp&s.doc. Accessed 4 Oct 2010