Computers & Operations Research 36 (2009) 2082 – 2096

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

Integration of process planning and scheduling—A modified genetic algorithm-based approach

Xinyu Shao, Xinyu Li, Liang Gao*, Chaoyong Zhang

The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China

ARTICLE INFO

Available online 17 July 2008

Keywords:
Process planning
Scheduling
Integration of process planning and scheduling
Genetic algorithm

ABSTRACT

Traditionally, process planning and scheduling for parts were carried out in a sequential way, where scheduling was done after process plans had been generated. Considering the fact that the two functions are usually complementary, it is necessary to integrate them more tightly so that performance of a manufacturing system can be improved greatly. In this paper, a new integration model and a modified genetic algorithm-based approach have been developed to facilitate the integration and optimization of the two functions. In the model, process planning and scheduling functions are carried out simultaneously. In order to improve the optimized performance of the modified genetic algorithm-based approach, more efficient genetic representations and operator schemes have been developed. Experimental studies have been conducted and the comparisons have been made between this approach and others to indicate the superiority and adaptability of this method. The experimental results show that the proposed approach is a promising and very effective method for the integration of process planning and scheduling.

1. Introduction

Process planning and scheduling are two of the most important sub-systems in manufacturing systems. A process plan specifies what raw materials or components are needed to produce a product, and what processes and operations are necessary to transform those raw materials into the final product. The outcome of process planning is the information required for manufacturing processes, including the identification of the machines, tools, and fixtures. Typically, most jobs may have a large number of alternative process plans. Process planning is the bridge of the product design and manufacturing. Scheduling plans receive process plans as their input and their task is to schedule the operations on the machines while satisfying the precedence relations given in the process plans. It is the link of the two production steps which are the preparing processes and putting them into action [1]. Although there is a strong relationship between process planning and scheduling, the integration of them is still a challenge in both research and applications.

In traditional approach, process planning and scheduling were carried out sequentially, where scheduling was done separately after the process plan had been generated. This approach has become an obstacle to enhance the productivity and responsiveness of manufacturing systems, and it may cause the following problems [2,3]:

(1) In a manufacturing organization, process planning function works in static. It considers the resources on the shop floor in an ideal way. Process planners assume unlimited resources on the shop floor and plan for the most recommended alternative process [4]. This may lead to the process planners favoring to select the desirable machines repeatedly. Therefore, the generated process plans are somewhat unrealistic and cannot be readily executed on the shop floor [5]. Accordingly, the resulting optimum process plans often become infeasible when they are carried out in practice at the later stage.

(2) Even if, in the planning phase, process planners consider the current resources on the shop floor, the constraints considered in the process planning phase may have already changed greatly because of the time delay between planning phase and execution phase. This may lead to the optimized process plan infeasibility. Investigations have shown that 20–30% of the total production plans in a given period have to be modified to adapt to the dynamic changing of a production environment [3].

(3) Scheduling plans are often determined after process plans. In the scheduling phase, scheduling planners have to consider the determined process plans. Fixed process plans may drive scheduling plans to end up with severely unbalanced resource load and create superfluous bottlenecks.

(4) In most cases, both for process planning and scheduling, a single criterion optimization technique is used for determining the

* Corresponding author. Tel.: +86 27 87557742; fax: +86 27 87543074.
E-mail address: gaoliang@mail.hust.edu.cn (L. Gao).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.07.006

best solution. However, the real production environment is best represented by considering more than one criterion simultaneously [3]. Furthermore, the process planning and scheduling may have conflicting objectives. Process planning emphasizes the technological requirements of a task, while scheduling involves the timing aspects of it. If there is no appropriate coordination, it may create conflicting problems.

To overcome these problems, there is thus a major need for an integrated process planning and scheduling system. The integration of the two functions may introduce 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 [5]. Without the integration of process planning and scheduling (IPPS), a true computer integrated manufacturing system (CIMS), which strives to integrate the various phases of manufacturing in a single comprehensive system, may not be effectively realized.

The remainder of this paper is organized as follows. Section 2 introduces a literature survey of the problem. IPPS is discussed in Section 3. A modified genetic algorithm (GA) for IPPS is given in Section 4. Experimental studies and discussion are reported in Section 5. Section 6 is the conclusion.

2. Literature survey

In the early studies of CIMS, it has been found that the IPPS is very important to the development of CIMS [6,7]. Chryssolouris and Chan [8,9] were the first to propose the preliminary idea of the IPPS. Beckendorff [10] used alternative process plans to improve the flexibility of manufacturing systems. Khoshnevis and Chen [11] introduced the concept of dynamic feedback into the IPPS. The integration model proposed by Zhang and Larsen [12,13] extended the concepts of alternative process plans and dynamic feedback. It also gave an expression to the methodology of hierarchical approach. In recent years, in the area of IPPS, integration models have been reported with different implementation approaches.

2.1. Integration models of IPPS

Many models have been proposed on the IPPS, and they can be classified into three basic models based on IPPS: nonlinear process planning (NLPP) [10], closed loop process planning (CLPP) [11] and distributed process planning (DPP) [12,13].

2.1.1. Nonlinear process planning

The methodology of NLPP is to make all alternative plans for each part with a rank according to process planning optimization criteria. The plan with highest priority is always ready for submission when the job is required. If the first-priority plan is not suitable for the current shop floor status, the second-priority plan will be provided to the scheduling.

NLPP is a basic model of the IPPS. Jain [14] proposed a scheme for IPPS that can be implemented in a company with existing process planning and scheduling departments when multiple process plans for each part type are available. Because the integration methodology of this model is very simple, most of the current research works on the integration model focuses on the improvement and implementation of the model. However, through a number of experimental computations, Usher [15] concluded that the advantage gained by increasing the number of alternative process plans for a scheduling system diminishes rapidly when the number of the plans reaches a certain level. The computational efficiency needs to be improved when applying to a complex system with a large number of alternative solutions.

2.1.2. Closed loop process planning

The methodology of CLPP is using a dynamic process planning system with a feedback mechanism. CLPP can be used to generate real-time process plans by means of a dynamic feedback from production scheduling system. The process planning mechanism generates process plans based on available resources. Production scheduling tells process planning which machines are available on the shop floor for an incoming job, so that every plan is feasible and respects the current availability of production facilities. This dynamic simulation system can enhance the real-time, intuition and manipulability of process planning system and also enhance the utilization of alternative process plans.

CLPP brings the IPPS to a real integration system very well. Usher and Fernandes [4] divided the dynamic process planning into the static phase and the dynamic phase. Seethaler and Yellowley [16] presented a dynamic process planning system which gave the process plans based on the feedback of scheduling system.

2.1.3. Distributed process planning

The methodology of DPP is to perform both the process planning and production scheduling simultaneously with a hierarchical approach. It divides the process planning and production scheduling tasks into two phases. The first phase is an initial planning phase. In this phase, the characteristics of parts and the relationship between the parts are analyzed, and the primary process plans are determined. The process resources are also evaluated simultaneously. The second phase is a detailed planning phase. In this phase, the process plans are adjusted to the current status of shop floor. The detailed process plans and scheduling plans are obtained simultaneously.

Wu and Fuh [17] gave the integration model of process planning and scheduling in the distributed virtual manufacturing environment. Zhang and Gao [18] presented a framework of concurrent process planning based on Holon. Wang and Song [19] presented a framework of collaborative process planning system supported by a real-time monitoring system.

2.1.4. Comparison of integration models

Every model has its advantages and disadvantages. A comparison among integration models is given in Table 1.

2.2. Implementation approaches of IPPS

2.2.1. Agent-based approaches of IPPS

The concept of an agent came from the research of artificial intelligence (AI) [20]. A typical definition of an agent is given by Nwana and Ndumu [21] as: "An agent is defined as referring to a component of software and/or hardware which is capable of acting exactly in order to accomplish tasks on behalf of its user". Based on the definition, one conclusion is that an agent is a software system that communicates and cooperates with other software systems to solve a complex problem that is beyond the capability of each individual software system [22]. Therefore, the agent-based approach has been considered as one important method for studying distributed intelligent manufacturing systems. A number of researchers have attempted to use this method to solve the IPPS problem. Zhang and Xie [20] reviewed the agent technology for collaborative process planning. The focus of the research was on how the agent technology can be further developed in support of collaborative process planning as well as its future research issues and directions in process planning. Wang and Shen [22] provided a literature review on the IPPS, particularly on the agent-based approaches for the problem. It was also provided with the advantages of the agent-based approach for scheduling. Shen and Wang [23] reviewed the research on manufacturing process planning, scheduling as well as their integration. Lim and Zhang [24] introduced a multi-agent based framework in the

Table 1
Comparison of integration models.

	Advantages	Disadvantages
NLPP	Providing all the alternative process plans of, and enhancing the *flexibility* and the availability of process plans.	Because of the need to give all alternative process plans of the parts, this will cause a *combinational-explosive* problem
CLPP	Based on the *current shop floor status*, the process plans are all very useful.	CLPP needs the real-time data of the current status, if it has to re-generate process plans in every scheduling phase, the *real-time data* is hard to be assured and updated
DPP	This model works in an *interactive, collaborative, and cooperative* way.	Because the basic integration principle of DPP is a hierarchical approach, it can not optimize the process plans and scheduling plans *as a whole*

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 configurations of manufacturing systems can be assessed. Wang and Shen [25] proposed a new methodology of DPP. It focused on the architecture of the new approach, using multi-agent negotiation and cooperation, and on the other supporting technologies such as machining feature-based planning and function block-based control. Wong and Leung [26] presented an online hybrid agent-based negotiation multi-agent system (MAS) for integrating process planning with scheduling/rescheduling. With the introduction of the supervisory control into the decentralized negotiations, this approach is able to provide solutions with a better global performance. Based on the comments listed above, one conclusion can be made that the agent-based approach is a useful method to solve the IPPS problem. Because single-agent environments cannot solve the problem effectively, MAS is more suitable to solve it [20]. Although the architecture and the negotiation among agents may be very complex, MAS will have a promising future in solving this problem [20,23,26]. However, when the number of the agents is large, agents will spend more time processing message than doing actual work, and it is often difficult to apply the generic agent architectures directly to IPPS systems [20]. Further research is needed to make the MAS approach simpler, and more effective and workable for IPPS applications [23].

2.2.2. Algorithm-based approaches of IPPS

The algorithm-based approach for IPPS synthesizes the methodology of NLPP and simulation. The basic steps of this approach are as follows. First, process planning system is used to generate the alternative process plans for all jobs and select user-defined number optimal plans based on the simulation results. Then the algorithm in the scheduling system is used to simulate scheduling plans based on the alternative process plans for all jobs. Finally, based on the simulation results, the process plan of each job and the scheduling plan are determined. This approach is workable, but the biggest shortcoming is that the simulation time may be long and the approach cannot be used in the real manufacturing system. Therefore, one important research direction is to find an effective algorithm for IPPS. In this direction, some researches have been done. Lee and Kim [5] presented the NLPP model which was based on the GA. Tan and Khoshnevis [27] presented a linearized polynomial mixed-integer programming model for IPPS. Chan and Kumar [28] proposed an artificial immune system (AIS) based AIS-FLC algorithm embedded with the fuzzy logic controller (FLC) to solve the complex IPPS problem. Li and McMahon [29] used a simulated annealing-based approach for flexible process plans. Kim and Park [30] used a symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. Moon and Seo [31] proposed an advanced process planning and scheduling model for multi-plant, and used an evolutionary algorithm to solve the model. Morad and Zalzala [32] described a GA-based algorithm that only considered the time aspect of the alternative machines, and then they extended this scope to include the processing capabilities of alternative machines, with different tolerance limits and process costs. Although this paper considered more aspects of alternative machines, it only focused on the alternative machines.

In the research presented in this paper, a new approach for IPPS has been developed in order to overcome the disadvantages of these models by synthesizing the integration methodology of NLPP and DPP. Based on this model, a simulation approach based modified GA has been developed. And, in the proposed method, the alternative process plans, alternative sequences and alternative machines are considered. Through experimental studies, the merits of this approach can be identified clearly.

3. Integration of process planning and scheduling

3.1. Proposed integration model

In this section, the IPPS model is introduced. This model is illustrated in Fig. 1.

The basic integration methodology of this model is to utilize the advantages of NLPP (alternative process plans) and DPP (hierarchical approach). This integration model is based on the concurrent engineering principle where the computer aided process planning (CAPP) and scheduling systems are working simultaneously. In the whole integration decision-making phase, this model gives expression to interactive, collaborative, and cooperative working. And it also exploits the flexibility of alternative process plans.

The detailed working steps of this model are given as follows:

Step 1: CAPP system is working based on the ideal shop floor resources, and generates all the initial alternative process plans for each job. Shop floor resource module provides the current shop floor status to the CAPP system.

Step 2: Based on the shop floor resource information by the given optimization criteria, the CAPP system optimizes all the alternative process plans and selects s (s is determined by the user of the system, and it is in the range of 3–5) near optimal process plans for each job [33]. Because there may be many alternative process plans for each job, this step is to filter out the poor process plans without affecting the flexibility of the model very much, but to avoid the *combinational-explosive* problem effectively.

Step 3: The integration of process planning (use the selected process plans) and scheduling is optimized based on the current shop floor status, generating the optimal scheduling plan and selecting one optimal process plan for each job from the selected process plans in *Step* 2.

Step 4: CAPP system is used to generate the detailed process plan for each job, and scheduling system is used to generate the detailed scheduling plan.

Through the working steps, the advantages of NLPP and DPP are utilized effectively in the integration model. CAPP and scheduling

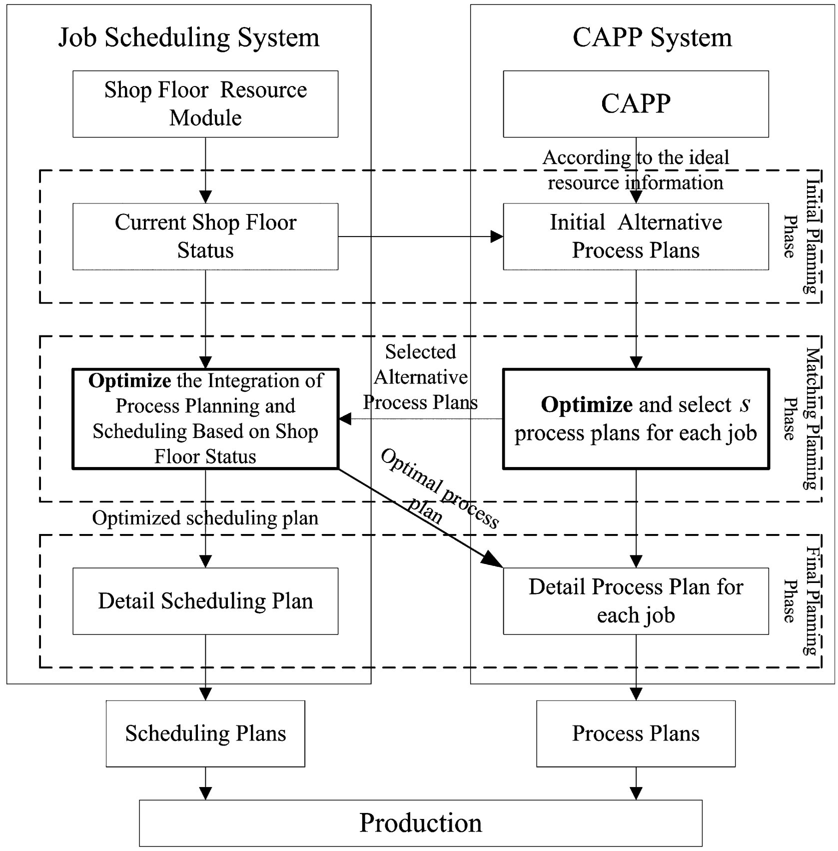

Fig. 1. Integration model.

systems are working simultaneously, and the flexibility of alternative process plans remain.

3.2. Representations for process plans and schedules

There are three types of flexibility considered in process plans [2,29]: operation flexibility, sequencing flexibility and processing flexibility [29]. Operation flexibility [30], also called routing flexibility, relates to the possibility of performing one operation on alternative machines, with possibly distinct processing time and cost. Sequencing flexibility is decided by the possibility of interchanging the sequence of the required operations. Processing flexibility is determined by the possibility of processing the same manufacturing feature with alternative operations or sequences of operations. Better performance in some criteria (e.g. production time) can be obtained through the consideration of these flexibilities [30].

There are many methods used to describe the types of flexibility explained above [34], such as Petri net, and/or graphs and networks. In this research, a network representation proposed by Kim [30] and Sormaz [35] is adopted here. There are three node types in the network: starting node, intermediate node and ending node [30]. The starting node and the ending node which are dummy ones indicate the start and the end of the manufacturing process of a job. An intermediate node represents an operation, which contains the alternative machines that are used to perform the operation and the processing time required for the operation according to the machines. The arrows connecting the nodes represent the precedence between them. OR relationships are used to describe the processing flexibility which the same manufacturing feature can be processed by different process procedures. If the links following a node are connected by an OR connector, they only need to traverse one of the OR-links (the links connected by the OR-connector are called OR-links). OR-link path is an operation path that begins at an OR-link and ends as it merges with the other paths, and its end is denoted by a JOIN-connector. For the links that are not connected by OR-connectors, all of them must be visited [30].

Fig. 2 shows three jobs alternative process plan networks (job 1, 3, 5, and job 2, 4, 6 will be given in Section 5) which are used in Section 5. In the network of Fig. 2 (1), paths {2, 3, 4}, {5, 6, 7} and {5, 8} are three OR-link paths. An OR-link path can definitely contain the other OR-link paths, e.g. paths {6, 7} and {8}.

In this paper, scheduling is often assumed as job shop scheduling [36]. In solving this scheduling problem, the following assumptions are made [29,30]:

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

A Gantt chart has been popularly used to represent a schedule of a group of parts. In the Gantt chart, the order in which the parts and their operations are carried out, is laid out, and the dependencies of

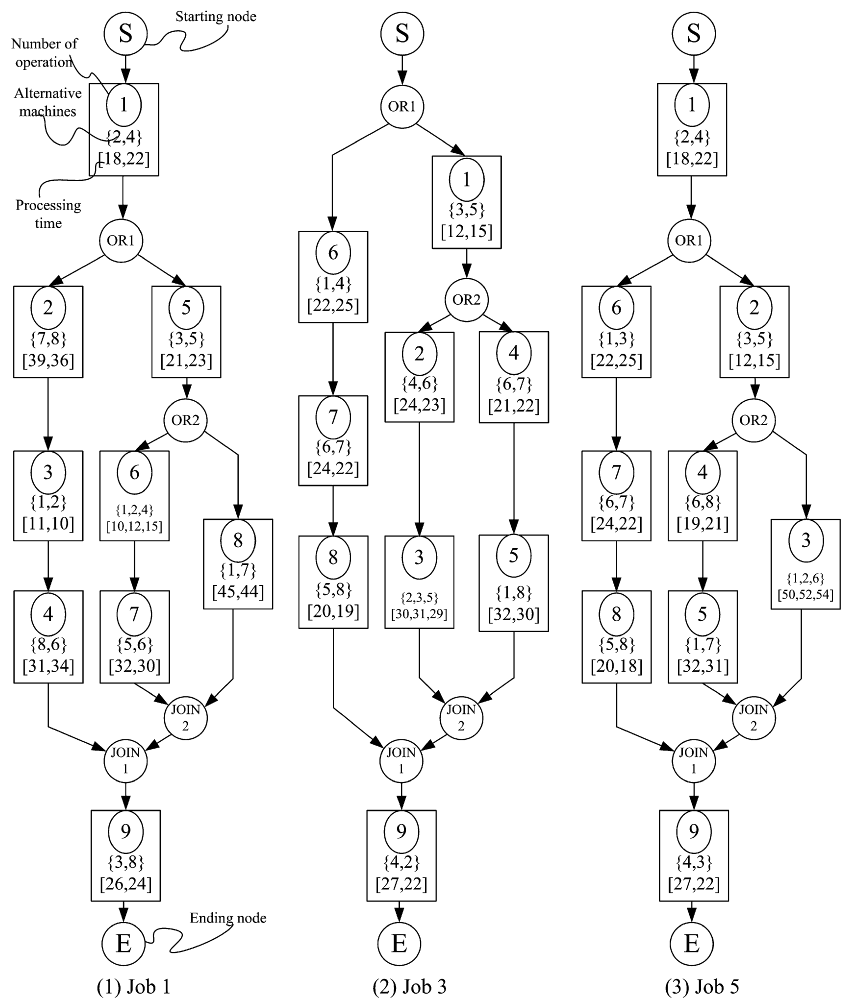

Fig. 2. Alternative process plans networks.

the tasks are managed. *X*-axis of the Gantt chart represents time. Each row in *Y*-axis represents a machine and the specific arrangement for the operations of the parts on the machine. In this paper, Gantt chart is used to represent schedule.

4. Modified GA-based optimization approach

4.1. Flow chart of proposed approach

Fig. 3 shows flow chart of the proposed method. First, the CAPP system gives alternative process plans. They are optimized by GA and the near optimal process plans are found. The next step is to select s near optimal process plans. And then, the integration of process plan and scheduling is optimized by GA. Finally, the optimized process plan for each job and the scheduling plan can be determined.

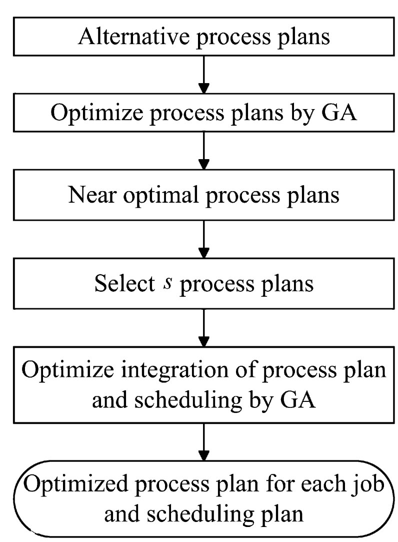

Fig. 3. Flow chart of proposed approach.

4.2. Genetic components for process planning

4.2.1. Encoding and decoding

Each chromosome in process planning population consists of two parts with different lengths as shown in Fig. 4. The first part of chromosome is the process plan string, and is made up of Gene. Gene is a structure, made up of two numbers. The first number is the operation. It can be all the operations for a job, even those that may not be performed because of alternative operation procedures. The second number is the alternative machine, which is the i th element of which represents the machine on which the operation corresponding to the i th element of part 1 is processed. The second

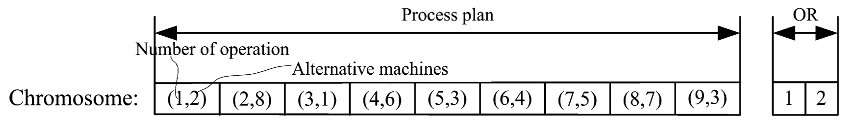

Fig. 4. Individual of process plan.

part of chromosome is the OR string, and is made up of discrimination values. The discrimination value encodes OR-connectors (see Fig. 2) as an integer in decimal system. It is relevant to the process plan string to decide which OR-link is chosen.

Fig. 4 shows an example of individual job 1 (see Fig. 2). Taking Gene (2, 8) for example, 2 is the operation of job 1; 8 is the alternative machine, which is corresponding to the operation 2. Part 1 (process plan string) of the chromosome shown in Fig. 4 is made up of nine Genes; part 2 (OR string) is made up of two discrimination values.

The encoding is directly decoded. The selection of the OR-link paths which contain operations and the corresponding machines are decided by the interpretation of part 2 of the chromosome. And then the orders appearing in the resulting part 1 are interpreted as an operation sequence and the corresponding machining sequence for a job. In the above encoding example, the operation sequence together with the corresponding machining sequence is (1, 2)–(2, 8)–(3, 1)–(4, 6)–(9, 3).

4.2.2. Initial population and fitness evaluation

Initial population: In order to operate evolutionary algorithm, an initial population is needed. The generation of the initial population in GA is usually done randomly. However, when generating the individuals for an initial population, feasible operation sequence in a process plan has to be taken into account. Feasible operation sequence means that the order of elements in the used encoding does not break constraints on precedence relations of operations [30]. As mentioned above, a method has been proposed to generate a random and feasible chromosome.

The procedure of the method is as follows [33]:

Step 1: The process plan string of the initial chromosome contains all the alternative operations, and the sequence of operations is fixed.

Step 2: The second number of process plan string is created by randomly assigning a machine in the set of machines that can perform the operation placed at the corresponding position in the process plan string.

Step 3: The OR string of the initial chromosome, which represents OR-link paths, is initiated by randomly generating an integer for each component of this part. The selection range of each discrimination value is decided by the number of OR-link paths which are controlled by this value. For example, if it has three OR-link paths, the selection range of the discrimination value is the random integer in [1,3].

Fitness evaluation: The objective of the alternative process planning is to minimize the production time (containing working time and transportation time) in the given condition Adjusted fitness has been used as the objective. In order to calculate fitness [33], the following notations are used to explain the model.

N	the total number of jobs;
G_i	the total number of flexible process plans of the i th job;
S	the size of population;
M	the maximal number of generations;
t	$1, 2, 3, \ldots, M$ generations;
o_{ijl}	the j th operation in the l th flexible process plan of the i th job;
P_{il}	the number of operations in the l th flexible process plan of the i th job;
k	the alternative machine corresponding to o_{ijl};
$TW(i,j,l,k)$	the working time of operation o_{ijl} on the k th alternative machine;
$TS(i,j,l,k)$	the starting time of operation o_{ijl} on the k th alternative machine;
$TT(i,l,(j,k_1),(j+1,k_2))$	the transportation time between the k_1 th alternative machine of the o_{ijl} and the k_2 th alternative machine of the $o_{i(j+1)l}$;
$TP(i,t)$	the production time of i th job in the t th generation;

Then the production time is calculated as

$$TP(i,t)=\sum_{j=1}^{P_{il}} TW(i,j,l,k)+\sum_{j=1}^{P_{il}-1} TT(i,l,(j,k_1),(j+1,k_2)),$$
$$i\in[1,N],\ \ j\in[1,P_{il}],\ \ l\in[1,G_i] \tag{1}$$

Because each machine can handle only one job at a time, the constraint is,

$$TS(i,j_2,l,k)-TS(i,j_1,l,k)>TW(i,j_1,l,k),$$
$$i\in[1,N],\ \ j_1,j_2\in[1,P_{il}],\ \ l\in[1,G_i] \tag{2}$$

Because different operations of one job cannot be processed simultaneously, it becomes the constraint of different processes for one job.

$$TS(i,(j+1),l,k_2)-TS(i,j,l,k_1)>TW(i,j,l,k_1),\quad i\in[1,N],$$
$$j\in[1,P_{il}],\ \ l\in[1,G_i] \tag{3}$$

The objective function is

$$\max \quad f(i,t)=\frac{1}{TP(i,t)} \tag{4}$$

The fitness function is calculated for each individual in the population as described in Eq. (4), and the two constraints are Eqs. (2) and (3).

4.2.3. Genetic operations for process planning

In order to result in excellent individuals residing in the population, it is important to employ good operators for dealing with the problem effectively and efficiently. The GA operators can be generally divided into three classes: reproduction, crossover and mutation. And in each class, a large number of operators have been developed.

Reproduction: The tournament selection scheme has been used for reproduction operation. In tournament selection, a number of individuals are selected randomly (dependent on the tournament size, typically between 2 and 7) from the population and the individual with the best fitness is chosen for reproduction. The tournament selection approach allows a tradeoff to be made between exploration and exploitation of the gene pool [37]. This scheme can modify the selection pressure by changing the tournament size.

Crossover: Point crossover has been used as the crossover operator in this study. Each part of the selected chromosomes executes the crossover operation distinguishingly. This crossover generates feasible offspring chromosomes that are within the precedence restrictions and avoids the duplication or omission of operations as follows. The cut point is chosen randomly and the substring before the cut point in one parent (P1) is passed on to the same position as in the offspring (O1). The other part of the offspring (O1) is made up of the substring after the cut point in the other parent (P2). The other offspring (O2) is made up of the substring before the cut point in one parent (P2) and the substring after the cut point in the other parent (P1). An example of the crossover is presented in Fig. 5, where in order to express clearer, two P1s are figured out and the cut point is marked out. Each part of the chromosome executes the crossover operation distinguishingly. The crossover operator produces feasible individuals since both parents are feasible and offspring are created without violating the feasibility of the parents.

Mutation: Point mutation has been used as the mutation operator in this study. Each of the selected chromosomes is mutated as follows. First, a point mutation scheme is applied in order to change the alternative machine represented in a Gene (see Fig. 4). The Gene is randomly chosen from the selected chromosome. Then, the second element of Gene is mutated by altering the machine number to another one of the alternative machines in random. Second, the other mutation is carried out to alter the OR-link path. This is associated with part 2 of chromosome. A discrimination value is randomly chosen from the selected chromosome. Then, it is mutated by changing its value in the selection range randomly. In the example depicted in Fig. 6, mutation point is marked out. Gene (5, 3) has changed into (5, 5), and the selected discrimination value has changed from 1 to 2.

4.3. Genetic components for scheduling

4.3.1. Encoding and decoding

Each chromosome in scheduling population consists of two parts with different length as shown in Fig. 7. The first part of chromosome is the scheduling plan string. In this study, the scheduling encoding is an operation-based representation with job numbers. It is represented by a permutation of the operations. It is natural in representing the operation as a sequence and then the crucial information containing parents can be easily passed on to the offspring. The job numbers are used to represent the operations of the jobs. The different appearances of this number in the chromosome represent the different operations of the job, and the sequence of the appearances of this number is the same as the sequence of the operations of the job. It is assumed that there are n jobs, and q is the number of operations of the process plan which has the most operations among all the alternative process plans of n jobs. The length of the scheduling plan string is equal to $n \times q$.

The second part of chromosome is the process plan string. The positions from 1 to n in this string represent the job from 1 to n. The number in the i th position represents the alternative process plan of the i th job chosen. The number of appearances of i in the scheduling plan string is equal to the number of operations of the alternative process plan which has been chosen. Based on this principle, the composition elements of scheduling plan string are determined. If the number of the elements cannot attain $n \times q$, all the other elements are filled with 0. So, the scheduling plan string is made up of job numbers and 0. One scheduling plan string is generated by arraying all the elements randomly. And the process plan string is generated by choosing the alternative process plan randomly for every job.

Table 2 shows an example of 6 jobs and each job has 3 alternative process plans. Fig. 7 shows an individual scheduling plan of this example. In this example, n is equal to 6, and q is equal to 5. Therefore, the scheduling plan string is made up of 30 elements and the process plan string is made up of six elements. For job 1, the first alternative process plan is chosen, with four operations in the process plan. The four elements of scheduling plan string are 1. For job 2, the second alternative process plan is chosen, with four operations in the process plan. The four elements of scheduling plan string are 2. For job 3, the third alternative process plan is chosen, with four operations in the process plan. The four elements of scheduling plan string are 3. For job 4, the second alternative process plan is chosen, with three operations in the process plan. The three elements of scheduling plan string are 4. For job 5, the second alternative process plan is chosen, with five operations in the process plan. The five elements of scheduling plan string are 5. For job 6, the first alternative process plan is chosen, with four operations in the process plan. The four elements of scheduling plan string are 6. Therefore, the scheduling plan string is made up of four 1s, four 2s, four 3s, three 4s, five 5s and four 6s. The other elements of this string are 0, and the number of 0 is equal to $6=(30-4-4-4-3-5-4)$. And all these elements are arrayed randomly to generate a scheduling plan string.

Table 2
An example of 6 jobs and 3 alternative process plans of each job.

Job	3 alternative process plans	Job	3 alternative process plans
1	(1,2)–(5,3)–(8,7)–(9,8) (1,2)–(5,3)–(8,7)–(9,3) (1,4)–(5,3)–(8,7)–(9,8)	4	(6,1)–(8,5)–(9,2) (6,1)–(7,6)–(9,2) (1,3)–(4,6)–(5,1)–(9,2)
2	(1,2)–(5,3)–(8,7)–(9,8) (1,2)–(4,6)–(8,7)–(9,8) (1,2)–(5,3)–(8,7)–(9‘,3)	5	(1,2)–(2,3)–(3,2)–(9,3) (1,2)–(2,3)–(4,6)–(5,7)–(9,3) (1,2)–(2,3)–(3,1)–(9,3)
3	(1,3)–(2,6)–(3,2)–(9,2) (1,3)–(2,4)–(3,3)–(9,2) (1,3)–(4,6)–(5,1)–(9,2)	6	(1,2)–(2,3)–(3,2)–(9,3) (1,2)–(2,3)–(4,8)–(9,3) (1,2)–(6,1)–(7,7)–(8,8)–(9,3)

The permutations can be decoded into semi-active, active, nondelay, and hybrid schedules. The active schedule is adopted in this research. Recall that at this decoding stage, a particular individual of a scheduling population has been determined, that is, a fixed alternative process plan for each job is given. The notations used to explain the procedure are described below:

m	the total number of machines;
o_{ij}	the j th operation of the i th job;
as_{ij}	the allowable starting time of operation o_{ij};
s_{ij}	the earliest starting time of operation o_{ij};
p_{ij}	the processing time of operation o_{ij};
t_{xyi}	the transportation time between the machine x of pre-operation and machine y of current operation of the i th job
c_{ij}	the earliest completion time of operation o_{ij}, i.e. $c_{ij}=s_{ij}+p_{ij}$;

The procedure of decoding is as follows:

Step 1: Generate the chromosome of machines based on the job chromosome.

Step 2: Determine the set of operations for every machine: $M_a = \{o_{ij}\}, 1 \leqslant a \leqslant m$.

Step 3: Determine the set of machines for every job: $JM_d = \{machine\}, 1 \leqslant d \leqslant N$.

Step 4: The allowable starting time for every operation: $as_{ij} = c_{i(j-1)} + t_{xyi}$ $(o_{ij} \in M_a, x, y \in JM_d)$, $c_{i(j-1)}$ is the completion time of the pre-operation of o_{ij} for the same job.

Step 5: Check the idle time of the machine of o_{ij}, and get the idle ranges $[t\_s, t\_e]$, check these ranges in turn (*if*: $\max(as_{ij}, t\_s)+p_{ij} \leqslant t\_e$, where the earliest starting time is $s_{ij}=t\_s$, *else*: check the next range), if there is no range satisfying this condition: $s_{ij} = \max(as_{ij}, c(o_{ij}-1))$, $c(o_{ij}-1)$ is the completion time of the pre-operation of o_{ij} for same machine.

Step 6: The completion time of every operation: $c_{ij} = s_{ij} + p_{ij}$.

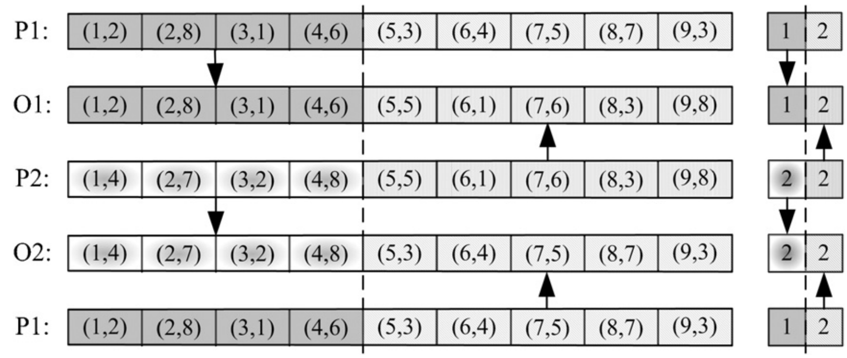

Fig. 5. Crossover for process planning.

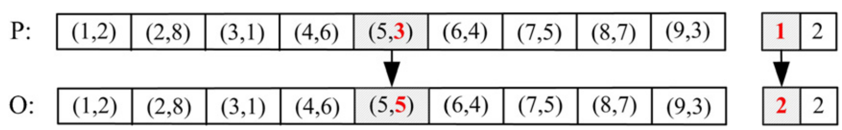

Fig. 6. Mutation for process planning.

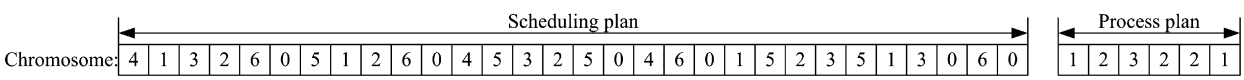

Fig. 7. Individual of scheduling plan.

Step 7: Generate the sets of starting time and completion time for every operation of each job: $T_d(s_{ij}, c_{ij}), 1 \leqslant d \leqslant N$.

In the procedure, the sets of starting time and completion time for every operation of each job can be obtained. The sets are one scheduling plan for the job shop.

4.3.2. Initial population and fitness evaluation

The encoding principle in this study is an operation-based representation. It cannot break the constraints on precedence relations of operations. The initial population is generated based on the encoding principle.

In this paper, two objective functions of the scheduling problem are calculated from:

$$Object_1 = \max(c_{ij})\ (c_{ij} \in T_d(s_{ij}, c_{ij}))\quad \text{the first objective is the makespan.} \tag{5}$$

$$Object_2 = Object_1 + \sum_{a=1}^{m} \left| \sum p_{ij} - avgmt \right| (o_{ij} \in M_a)\quad \text{(see Section 4.4.1)} \tag{6}$$

$\sum p_{ij}$ is the total working time for a machine. *avgmt* is the average working time of all the machines: $avgmt = (\sum_{a=1}^{m} \sum p_{ij})/m$. $\sum_{a=1}^{m} | \sum p_{ij} - avgmt | (o_{ij} \in M_a)$ is the summation of the absolute values of the difference of the total working time and *avgmt* for every machine. The goal of this objective function is lined on the synthetic consideration of the makespan and balanced level of machine utilization.

4.3.3. Genetic components for scheduling

The reproduction operator for scheduling is the same as in process planning. In this section, the crossover and mutation operators are introduced in detail.

(1) *Crossover*: The procedure of crossover for scheduling is described as follows:

Step 1: Select a pair of chromosomes P1 and P2 by the selection scheme and initialize two empty offspring: O1 and O2.

Step 2: First, crossover the process plan strings of P1 and P2 and get the process plan strings of O1 and O2.

Step 2.1: Compare the process plan string of P1 with the process plan string of P2, if the element of P1 is the same as P2, record the value and position of this element. This process is repeated until the end of comparing all the elements of process plan string.

Step 2.2: The recorded elements in process plan string of P1 in *Step* 2.1 are appended to the same positions in O1, while the recorded elements in process plan string of P2 in *Step* 2.1 are appended to the same positions in O2. The other elements (they are the different elements between P1 and P2) in process plan string of P2 are appended to the same positions in O1, while the other elements in process plan string of P1 are appended to the same positions in O2.

Step 3: Secondly, in order to match the process plan strings of O1 and O2 and avoid getting unreasonable O1 and O2, the scheduling plan strings of P1 and P2 are crossovered as follows:

Step 3.1: If the values of elements in scheduling plan string of P1 are the same as the recorded positions in process plan string, these elements (including 0) are append to the same positions in O1 and they are deleted in P1. If the values of elements in scheduling plan string of P2 are the same as the recorded positions in process plan string, these elements (including 0) are append to the same positions in O2 and they are deleted in P2.

Step 3.2: Get the numbers of the remaining elements in scheduling plan of P1 and P2, they are n_1 and n_2. If $n_1 \geqslant n_2$, for O1, it implies that the number of empty positions in O1 is larger than the number of remaining elements in P2. Therefore, $n_1 - n_2$ empty positions in O1 are selected randomly and be filled with 0. Then, the remaining elements in scheduling plan of P2 are appended to the remaining empty positions in O1 seriatim. For O2, $n_1 \geqslant n_2$ implies that the number of empty positions in O2 is smaller than the number of remaining elements in P1. So, $n_1 - n_2$ 0s are selected randomly in O2

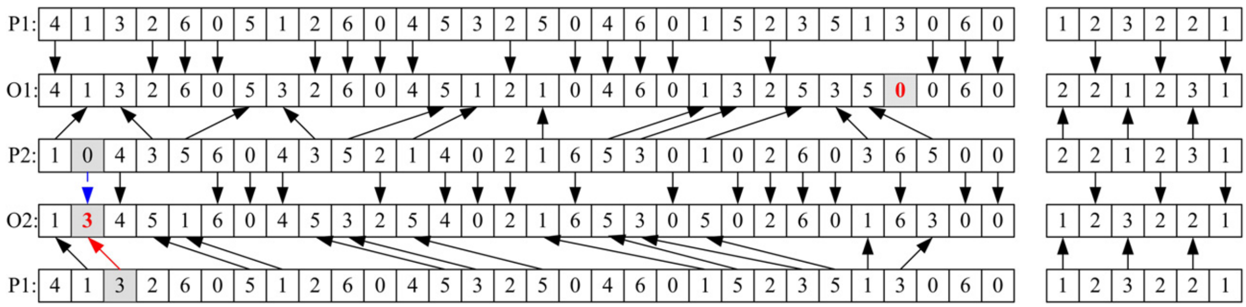

Fig. 8. Crossover for scheduling.

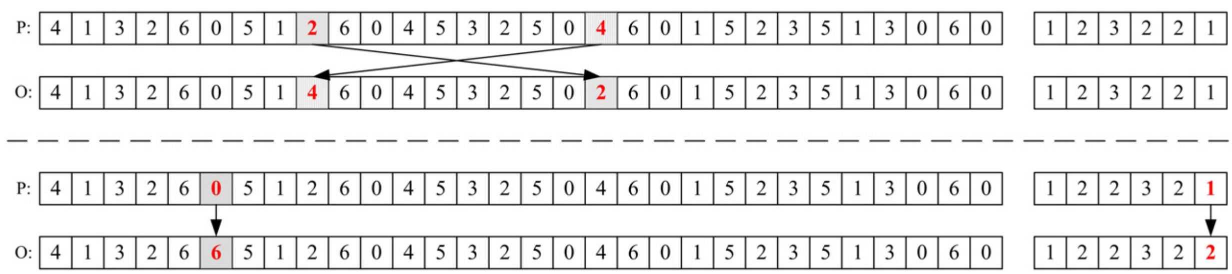

Fig. 9. Mutation for scheduling.

and are set to empty. And then, the remaining elements in scheduling plan of P1 are appended to the empty positions in O2 seriatim. If $n_1 < n_2$, the procedure is reversed.

Step 4: Then, two valid offspring O1 and O2 are obtained.

Take six jobs in Table 2 as an example. An example of the crossover is presented in Fig. 8.

Step 1: Select a pair of chromosomes P1 and P2 and initialize two empty offspring: O1 and O2 (see Fig. 8).

Step 2: First, crossover the process plan strings of P1 and P2 and get the process plan strings of O1 and O2.

Step 2.1: Compare the process plan string of P1 with the process plan string of P2, record the second, fourth and sixth elements in P1 and P2.

Step 2.2: The second, fourth and sixth elements in process plan string of P1 are appended to the same positions in O1, while the second, fourth and sixth elements in process plan string of P2 are appended to the same positions in O2. The other elements which are the first, third and fifth elements in process plan string of P2 are appended to the same positions in O1, while the other elements in process plan string of P1 are appended to the same positions in O2.

Step 3: Secondly, in order to match the process plan strings of O1 and O2 and avoid getting unlawful O1 and O2, the scheduling plan strings of P1 and P2 are crossovered as follows:

Step 3.1: The elements which equate 2, 4 or 6 (including 0) in scheduling plan string of P1 are append to the same positions in O1 and they are deleted in P1; the elements which equate 2, 4 or 6 (including 0) in scheduling plan string of P2 are append to the same positions in O2 and they are deleted in P2.

Step 3.2: In this example, $n_1 = 13$, $n_2 = 12$, $n_1 > n_2$ and $n_1 - n_2 = 1$. For O1, one empty position in O1 is selected randomly and is filled with 0, which has marked out in O1 in Fig. 8. Then, the remaining elements in scheduling plan of P2 are appended to the remaining empty positions in O1 seriatim. For O2, one 0 is selected randomly in O2 and is set to empty, which has marked out in O2 in Fig. 8. And then, the remaining elements in scheduling plan of P1 are appended to the empty positions in O2 seriatim.

Step 4: Then, two valid offspring O1 and O2 are obtained (see Fig. 8).

(2) *Mutation*: In this paper, there are two mutation operators used. One is two-point swapping mutation, and the other one is changing one job's alternative process plan. In the evolution procedure, one operator has been chosen randomly in every generation.

The procedure of two-point swapping mutation for scheduling is described as follows:

Step 1: Select one chromosome P by the selection scheme.

Step 2: Select two points in the scheduling plan string of P randomly.

Step 3: Generate a new chromosome O by interchanging these two elements;

The procedure of the other mutation (changing one job's alternative process plan) for scheduling is described as follows:

Step 1: Select one chromosome P by the selection scheme.

Step 2: Select one point in the process plan string of P randomly.

Step 3: Change the value of this selected element to another one in the selection range (the number of alternative process plans).

Step 4: Judge the number of the operations of the selected job's alternative process plan which has been changed. If it increases, a new chromosome O is generated by changing the margin 0s which are selected randomly to the job number in the scheduling plan string of P seriatim. If it decreases, a new chromosome O is generated by changing the margin job numbers which are selected randomly in the scheduling plan string of P to 0 seriatim.

An example of the mutation is presented in Fig. 9. Above the dash line, it is an example of two-point swapping mutation, the selected two points (2 and 4) are marked out, and O is generated by interchanging 2 and 4. And under the dash line, is an example of the mutation of changing one job's alternative process plan. The selected element 1 (for job 6) which is marked out in the process plan string is changed to 2. Because the number of the second alternative process plans for job 6 is greater than the first one, the first 0 which is selected randomly in scheduling plan string of P is changed to 6 in O.

5. Experimental studies and discussion

Some experiments have been conducted to measure the adaptability and superiority of the proposed GA-based integration approach. The approach is compared with a hierarchical approach and other methods. The performance of the approach is satisfactory from the experimental results and comparisons.

5.1. Test problems and experimental results

The proposed modified GA approach procedure was coded in C++ and implemented on a computer with a 2.40 GHz PentiumIV CPU. To illustrate the effectiveness and performance of the method in this paper, we consider five problem instances. The GA parameters for process planning and scheduling are given in Table 3. The algorithm terminates when the number of generations reaches to the maximum value.

5.1.1. Experiment 1

(1) *Test problem*: For doing the experiments of the proposed approach, six jobs with flexible process plans have been generated. Jobs 1, 3 and 5 have been given in Fig. 2. And jobs 2, 4 and 6 will be given in Fig. 10. There are eight machines in the shop floor. The machines' codes in these six jobs are same. The transportation time (the time units is same as Processing time in Fig. 2) between the machines are given in Table 4. The objective for process planning is the maximum

Table 3
GA parameters.

Parameters	Process planning	Scheduling
The size of the population, S	40	500
Total number of generations, M	30	100
Tournament size, b	2	2
Probability of reproduction operation, p_r	0.10	0.10
Probability of crossover operation, p_c	0.60	0.80
Probability of mutation operation, p_m	0.10	0.10

Table 4
Transportation time between the machines.

Machine code	1	2	3	4	5	6	7	8
1	0	3	7	10	3	5	8	12
2	3	0	4	7	5	3	5	8
3	7	4	0	3	8	5	3	5
4	10	7	3	0	10	8	5	3
5	3	5	8	10	0	3	7	10
6	5	3	5	8	3	0	4	7
7	8	5	3	5	7	4	0	3
8	12	8	5	3	10	7	3	0

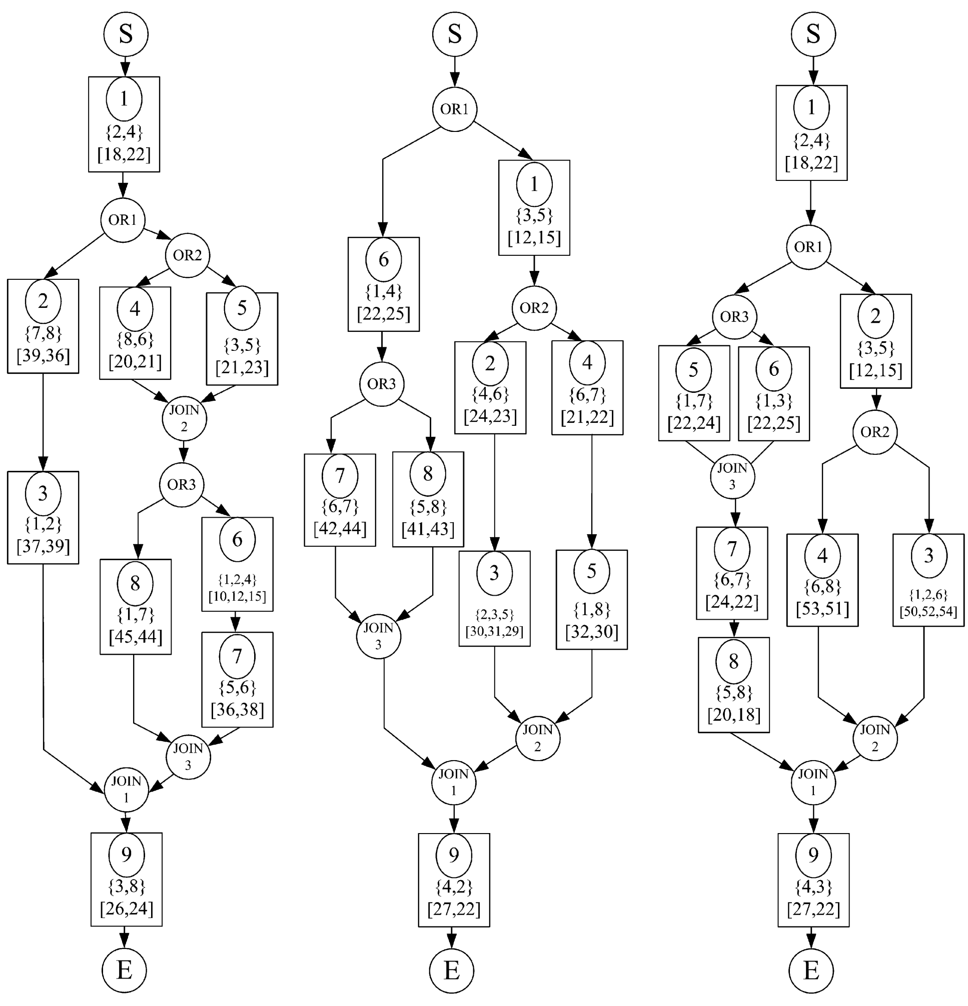

Fig. 10. Alternative process plans networks.

objective function $f(i,t)$ (Eq. (4)). And the objectives for the IPPS are minimum makespan $Object_1$ (Eq. (5)) and synthetic consideration of the makespan and balanced level of machine utilization $Object_2$ (Eq. (6)).

(2) *Experimental results*: The experiments of process planning were carried out for one objective: minimizing the production time. Table 5 shows the experimental results where the adjusted fitness Eq. (4) is applied for all the jobs. The results include three near optimal alternative process plans with their fitness and production time for each job (see Section 4.3.2). Fig. 11 illustrates convergence curve of the job 1 for process planning. The curve shows the search capability and evolution speed of the algorithm.

The computation time for job 1 is 1015 ms. The experimental results in Table 5 and Fig. 11 show that GA-based process planning reaches a good solution in short time. This means that the first optimization step (the optimization of process planning) in the proposed model needs very little computation time. And even in the more complex system, based on the development of computer, the computer time is also very little. Therefore, the proposed model with two optimization steps is a promising approach for the IPPS.

The experiments on IPPS were carried out for two objectives: minimizing makespan $Object_1$ (Eq. (5)) and the synthetic consideration of the makespan and balanced level of machine utilization $Object_2$ (Eq. (6)). Table 6 shows the experimental results which include the selected process plan of each job. Fig. 12 illustrates Gantt charts for scheduling. The schedule in Fig. 12(1) yields to a makespan of 162 time units, and $Object_2$ is equals to 712. In the Fig. 12(2), makespan is equal to 165, $Object_2$ is equal to 640. This means that if the makespan and balanced level of machine utilization are synthetically considered, some optimization of makespan may be lost. If only makespan is considers, some balanced level of machine utilization may be lost. Therefore, which objective is chosen should be based on the goal of operator.

Table 5
Experimental results of process planning.

Job	3 Alternative process plans	Fitness	Production time
1	(1,2)–(5,3)–(8,7)–(9,8)	0.00862069	116
	(1,2)–(5,3)–(8,7)–(9,3)	0.00847458	118
	(1,4)–(5,3)–(8,7)–(9,8)	0.00833333	120
2	(1,2)–(5,3)–(8,7)–(9,8)	0.00862069	116
	(1,2)–(4,6)–(8,7)–(9,8)	0.00854701	117
	(1,2)–(5,3)–(8,7)–(9,3)	0.00847458	118
3	(1,3)–(2,6)–(3,2)–(9,2)	0.01052632	95
	(1,3)–(2,4)–(3,3)–(9,2)	0.01010101	99
	(1,3)–(4,6)–(5,1)–(9,2)	0.01000000	100
4	(6,1)–(8,5)–(9,2)	0.01075269	93
	(6,1)–(7,6)–(9,2)	0.01063830	94
	(1,3)–(4,6)–(5,1)–(9,2)	0.01000000	100
5	(1,2)–(2,3)–(3,2)–(9,3)	0.00862069	116
	(1,2)–(2,3)–(4,6)–(5,7)–(9,3)	0.00847458	118
	(1,2)–(2,3)–(3,1)–(9,3)	0.00833333	120
6	(1,2)–(2,3)–(3,2)–(9,3)	0.00862069	116
	(1,2)–(2,3)–(4,8)–(9,3)	0.00854701	117
	(1,2)–(6,1)–(7,7)–(8,8)–(9,3)	0.00826446	121

5.1.2. Experiment 2

Experiment 2 is adopted from Moon et al. [31]. In this experiment, the problem was constructed with five jobs with 21 operations and six machines in two plants. The makespan was used as the objective. In this paper, the transportation times between machines and the lot sizes were not considered. Table 7 shows the comparison of the results of operation sequences with machine selection between evolutionary algorithm [31] and modified GA, and Fig. 13 illustrates the Gantt chart of this problem.

5.1.3. Experiment 3

Experiment 3 is adopted from Morad and Zalzala [32]. In this experiment, the problem was constructed with five jobs and five

Table 6
Experimental results of the selected process plan for every job.

Job	Optimization criterion	
	$Object_1$	$Object_2$
1	(1,4)–(5,3)–(8,7)–(9,8)	(1,2)–(5,3)–(8,7)–(9,8)
2	(1,2)–(4,6)–(8,7)–(9,8)	(1,2)–(4,6)–(8,7)–(9,8)
3	(1,3)–(2,4)–(3,3)–(9,2)	(1,3)–(2,4)–(3,3)–(9,2)
4	(6,1)–(8,5)–(9,2)	(6,1)–(8,5)–(9,2)
5	(1,2)–(2,3)–(3,1)–(9,3)	(1,2)–(2,3)–(4,6)–(5,7)–(9,3)
6	(1,2)–(2,3)–(3,2)–(9,3)	(1,2)–(2,3)–(4,8)–(9,3)

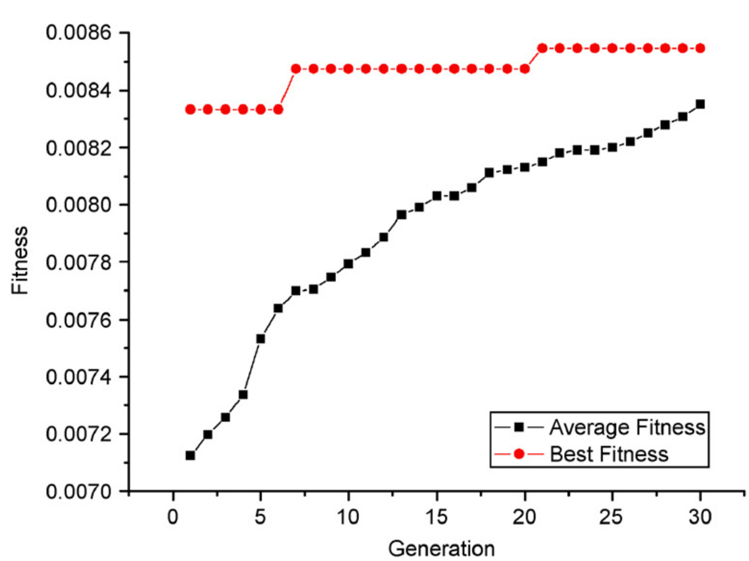

Fig. 11. Convergence curve of the job 1 for process planning.

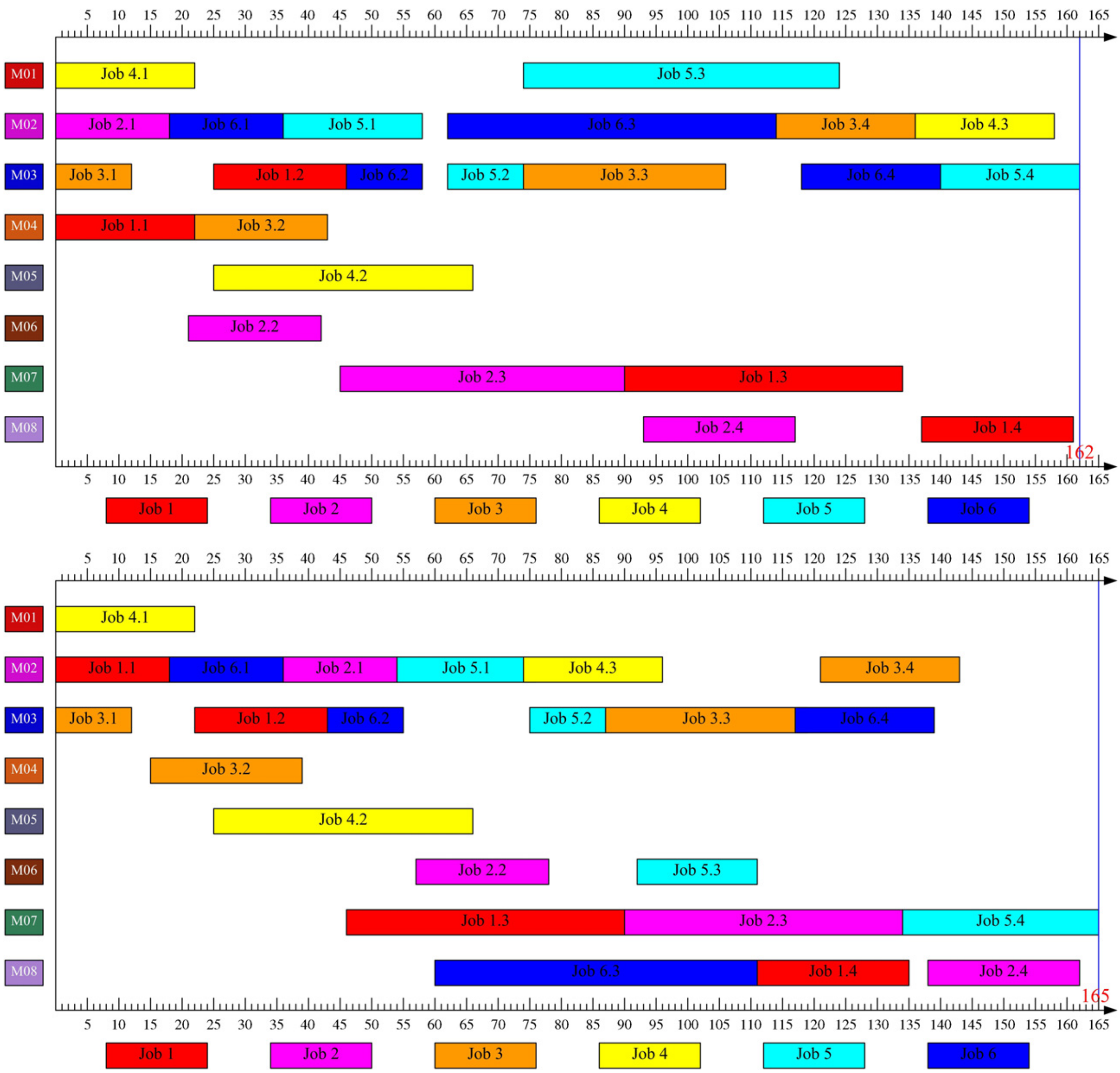

Fig. 12. (1) Gantt chart of experiment 1 based on $Object_1$ (Makespan = 162, $Object_2$ = 712). (2) Gantt chart of experiment 1 based on $Object_2$ (Makespan = 165, $Object_2$ = 640).

Table 7
Comparison of the results of operations sequences with machine selection (The data which is marked by * is adopted from Moon et al. [31].).

Job	Evolutionary algorithm*	Modified GA
1	1(M4)–3(M3)–2(M1)–4(M6)	1(M1)–2(M4)–4(M6)–3(M3)
2	6(M1)–5(M2)–7(M1)	5(M2)–6(M1)–7(M1)
3	9(M6)–11(M6)–8(M3)–10(M2)–12(M5)	9(M6)–11(M6)–8(M3)–10(M2)–12(M5)
4	13(M3)–14(M5)–15(M3)–16(M6)–17(M5)	13(M3)–15(M3)–14(M2)–16(M4)–17(M1)
5	19(M2)–18(M4)–20(M5)–21(M4)	18(M4)–19(M5)–20(M5)–21(M4)

machines. The makespan was used as the objective. Each part undergoes four different operations in a specified order. Alternative machines for processing the parts are given in Table 8, along with the respective processing times.

Table 9 gives the experimental results of experiment 3. Using the SA, GA and modified GA-based approach, the best value obtained is 33 compared with the value of 38 obtained using the heuristic. The advantage of the GA and modified GA approaches compared with the SA approach is the availability of multiple solutions that introduces some flexibility into the IPPS system. In the SA approach, only one solution is produced for every run. Fig. 14 illustrates Gantt chart of this problem (modified GA).

5.1.4. Experiment 4

Experiment 4 is also adopted from Morad and Zalzala [32]. In this experiment, the problem was constructed with four jobs and

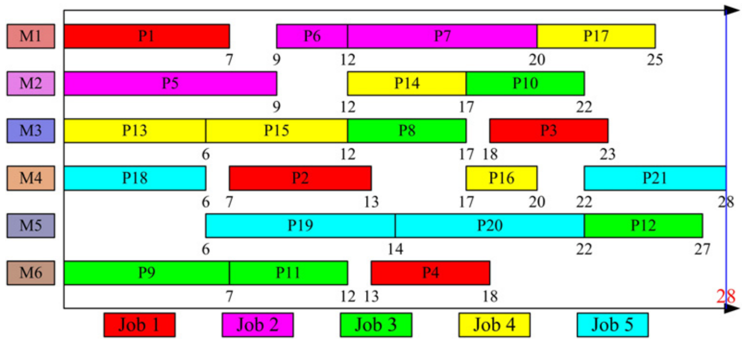

Fig. 13. Gantt chart of experiment 2 (P1 means Operation 1, Makespan = 28).

Table 8
Sundaram and Fu data in Ref. [32] (The numbers in parenthesis are the machine numbers).

Job	Operation 1	Operation 2	Operation 3	Operation 4
1	5(M1) 3(M2)	7(M2)	6(M3)	3(M4) 4(M5)
2	7(M1)	4(M2) 6(M3)	7(M3) 7(M4)	10(M5)
3	4(M1) 5(M2) 8(M3)	5(M4)	6(M4) 5(M5)	4(M5)
4	2(M2) 6(M3)	8(M3)	3(M3) 8(M4)	7(M4) 4(M5)
5	3(M1) 5(M3)	7(M3)	9(M4) 6(M5)	3(M5)

Table 9
Experimental results for experiment 3 (The data which is marked by * is adopted from Morad and Zalzala [32].).

Solution methodology	Heuristic*	Simulated annealing*	GA-based approach*	Modified GA
Makespan	38	33	33	33

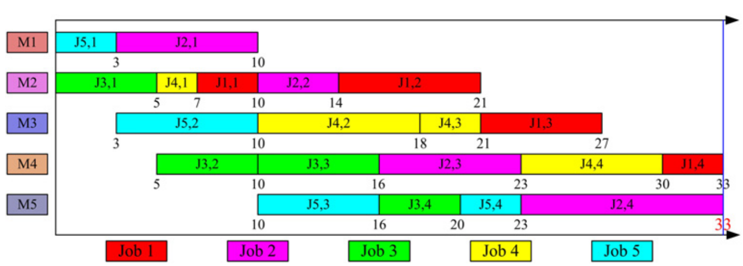

Fig. 14. Gantt chart of experiment 3 (Makespan = 33).

Table 10
Comparison among the approaches (The data which is marked by* is adopted from Morad and Zalzala [32].).

	Traditional method*	Approach in [32]*	Modified GA
Makespan	2030	1265	1100

three machines. The makespan was used as the objective. Table 10 gives a comparison among them Fig. 15 illustrates Gantt charts of experiment 4.

As indicated in Table 10 and Fig. 15, the best solution found by modified GA is not detected using the method in Morad and Zalzala [32]. It may not be the global optimization. However, the result implies that the proposed approach is more effective to escape from the local optimization point and easier to get a better solution than the method in Morad and Zalzala [32].

5.1.5. Experiment 5

Experiment 5 is adopted from Moon et al. [38]. In this experiment, the problem was constructed with five jobs and five machines. The makespan is used as the objective. Table 11 shows the experimental results, and Fig. 16 illustrates the Gantt chart of this problem.

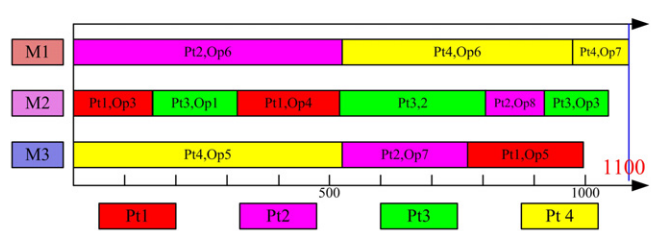

Fig. 15. Gantt chart of experiment 4 (Pt: Part; Op: Operation, Makespan = 1100).

Based on the experimental results of experiment 5, the best solution found by the modified GA is not detected using the method in Moon et al. [38]. The modified GA can get better scheduling plans.

5.2. Comparison with hierarchical approach

The proposed integration approach has been compared with a hierarchical based approach in terms of optimum results. The hierarchical approach is coded in C++, and implemented on the same computer with GA.

Hierarchical approaches have been widely used to solve an aggregated problem combining several sub-problems that are interlinked. In this approach, the process planning problem is first solved, and then the scheduling problem is considered under the constraint of the solution. For the process planning problem, minimizing the production time is used as an objective, which is maximum $f(i,t)$ (Eq. (4)). The scheduling objective of hierarchal approach is to minimize the makespan $Object_1$ (Eq. (5)). To solve the two problems of the process planning and the scheduling problem hierarchically, in this paper GA is adopted. The parameters of hierarchical approach are the same with those for the proposed method (see Table 4). The solution to the process planning problem obtained becomes an input to the scheduling problem. The selected process plan of each job is the optimal process plan. For the first experiment, they are the first ones of 6 jobs in Table 5. Fig. 17 illustrates the Gantt chart of hierarchical approach yields to a makespan of 250 time units, and $Object_2$ is equal to 808.

The experimental results of Table 6 show that the selected process plans in the proposed approach are not all of the optimal plans in Table 5, such as, the selected process plan of job 1 is the third optimal plan. Comparing the results between the proposed method and the hierarchical approach, the selected process plans of each job among them are different, and the optimized result of scheduling of the hierarchical approach is not as good as the results from integration model. The reasons for this are that the hierarchical approach lacks the integration and flexibility. There is no connection between process planning and scheduling. Therefore, the IPPS is necessary and it can enhance the productivity of manufacturing system largely.

Table 11
Experimental results of experiment 5 (The data which is marked by * is adopted from Moon et al. [38]. $v1$ means operation 1. M1 means machine 1).

	Moon et al.*	Modified GA
Job 1	v1M2–v2M2	v1M1–v2M2
Job 2	v3M4–v4M5	v3M4–v4M5
Job 3	v5M1–v6M3–v7M2	v7M2–v5M2–v6M3
Job 4	v8M3–v9M4	v8M3–v9M4
Job 5	v10M1–v11M3–v12M1–v13M5	v12M3–v10M1–v11M1–v13M5
Makespan	16	14

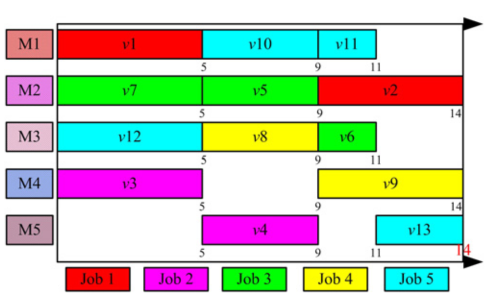

Fig. 16. Gantt chart of experiment 5 (Makespan = 14).

5.3. Discussion

Overall, the experimental results indicate that the proposed approach is a more acceptable approach for the IPPS. The reasons are as follows. First, in the proposed approach, the selected process plans are not all the optimal ones, and it considers all the conditions synthetically. Second, in some experiments, the modified GA-based approach can be used to get better results than that of other methods. This means that the proposed approach has more possibilities to get the best results.

6. Conclusion

In the traditional approach, process planning and scheduling were regarded as two separate tasks and performed sequentially. However, the functions of the two systems are usually complementary. Therefore, the research on the integration of process planning and scheduling is necessary. The research presented in this paper developed a new integration model with a modified GA-based approach have been developed to facilitate the integration and optimization of these two systems. With integration model, the process planning and scheduling systems are working simultaneously. To improve the optimization performance of the proposed method, more efficient genetic representations and operator schemes have been developed. Experimental studies have been used to test the method and the comparisons have been made among this method and other methods to indicate the superiority and adaptability of the proposed approach. The experimental results show that the proposed method is a promising and very effective method in the research of integration of process planning and scheduling.

With the new method developed in this work, it would be possible to increase the efficiency of manufacturing systems. One future work is to use the proposed method to practical manufacturing systems. The increased use of this approach will most likely enhance the performances of future manufacturing systems.

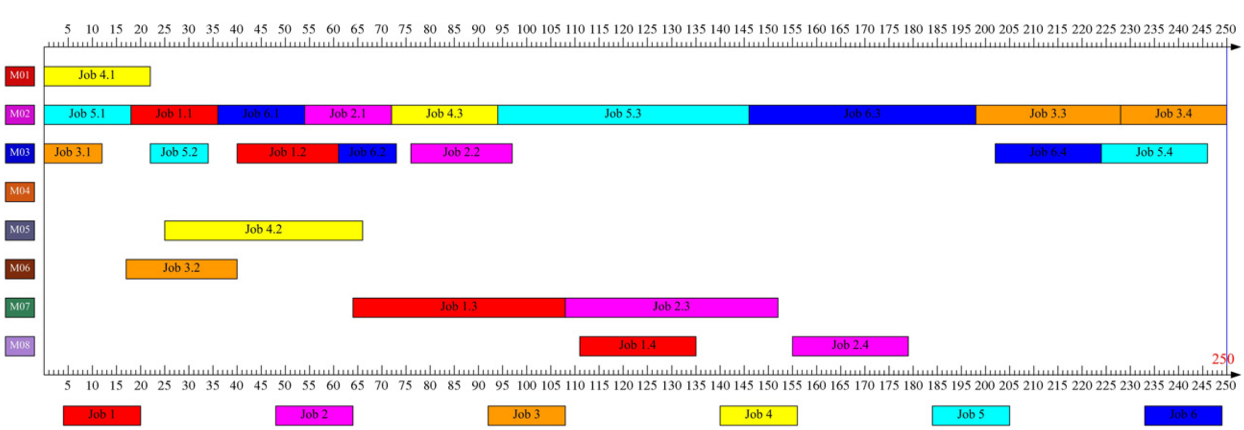

Fig. 17. Gantt chart of hierarchical approach based on $Object_1$ (Makespan = 250, $Object_2$ = 808).

Acknowledgements

The authors would like to thank Professor Yiming Rong from Worcester Polytechnic Institute for his helpful comments and suggestions. The authors would like to thank anonymous referees whose comments helped a lot to improve this paper. This research work is supported by the National Basic Research Program of China (973 Program) under Grant no.2004CB719405, the National High-Tech Research and Development Program of China (863 Program) under Grant nos.2007AA04Z107 and 2006AA04Z131.

References

[1] Sugimura N, Hino R, Moriwaki T. Integrated process planning and scheduling in holonic manufacturing systems. In: Proceedings of IEEE international symposium on assembly and task planning soft research park, vol. 4, Japan, Fukuoka; 2001. p. 250–4.
[2] Saygin C, Kilic SE. Integrating flexible process plans with scheduling in flexible manufacturing systems. International Journal of Advanced Manufacturing Technology 1999;15:268–80.
[3] Kumar M, Rajotia S. Integration of scheduling with computer aided process planning. Journal of Materials Processing Technology 2003;138:297–300.
[4] Usher JM, Fernandes KJ. Dynamic process planning-the static phase. Journal of Materials Processing Technology 1996;61:53–8.
[5] Lee H, Kim SS. Integration of process planning and scheduling using simulation based genetic algorithms. International Journal of Advanced Manufacturing Technology 2001;18:586–90.
[6] Tan W, Khoshnevis B. Integration of process planning and scheduling—a review. Journal of Intelligent Manufacturing 2000;11:51–63.
[7] Kumar M, Rajotia S. Integration of process planning and scheduling in a job shop environment. International Journal of Advanced Manufacturing Technology 2005;28(1–2):109–16.
[8] Chryssolouris G, Chan S, Cobb W. Decision making on the factory floor: an integrated approach to process planning and scheduling. Robotics and Computer-Integrated Manufacturing 1984;1(3–4):315–9.
[9] Chryssolouris G, Chan S. An integrated approach to process planning and scheduling. Annals of the CIRP 1985;34(1):413–7.
[10] Beckendorff U, Kreutzfeldt J, Ullmann W. Reactive workshop scheduling based on alternative routings. In: Proceedings of a conference on factory automation and information management. Boca Roton, FL: CRC Press; 1991. p. 875–85.
[11] Khoshnevis B, Chen QM. Integration of process planning and scheduling function. In: Proceedings of IIE integrated systems conference and society for integrated manufacturing conference. Atlanta, GA: Industrial Engineering and Management Press; 1989. p. 415–20.
[12] Zhang HC. IPPM-a prototype to integrated process planning and job shop scheduling functions. Annals of the CIRP 1993;42(1):513–7.
[13] Larsen NE. Methods for integration of process planning and production planning. International Journal of Computer Integrated Manufacturing 1993;6(1–2): 152–62.
[14] Jain A, Jain PK, Singh IP. An integrated scheme for process planning and scheduling in FMS. International Journal of Advanced Manufacturing Technology 2006;30:1111–8.
[15] Usher JM. Evaluating the impact of alternative plans on manufacturing performance. Computers and Industrial Engineering 2003;45:585–96.
[16] Seethaler RJ, Yellowley I. Process control and dynamic process planning. International Journal of Machine Tools and Manufacture 2000;40:239–57.
[17] Wu SH, Fuh JYH, Nee AYC. Concurrent process planning and scheduling in distributed virtual manufacturing. IIE Transactions 2002;34:77–89.
[18] Zhang J, Gao L, Chan FTS. A holonic architecture of the concurrent integrated process planning system. Journal of Materials Processing Technology 2003;139:267–72.
[19] Wang LH, Song YJ, Shen WM. Development of a function block designer for collaborative process planning. In: Proceedings of CSCWD2005. Coventry, UK, 2005. p. 24–6.
[20] Zhang WJ, Xie SQ. Agent technology for collaborative process planning: a review. International Journal of Advanced Manufacturing Technology 2007;32:315–25.
[21] Nwana H, Ndumu D. An introduction to agent technology. Software agents and soft computing: toward enhancing machine intelligence, concepts and applications 1997. p. 3–36.
[22] Wang LH, Shen WM, Hao Q. An overview of distributed process planning and its integration with scheduling. International Journal of Computer Applications in Technology 2006;26(1–2):3–14.
[23] Shen WM, Wang LH, Hao Q. Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews 2006;36(4):563–77.
[24] Lim MK, Zhang Z. A multi-agent based manufacturing control strategy for responsive manufacturing. Journal of Materials Processing Technology 2003;139:379–84.
[25] Wang LH, Shen WM. DPP: an agent-based approach for distributed process planning. Journal of Intelligent Manufacturing 2003;14:429–39.
[26] Wong TN, Leung CW, Mak KL, Fung RYK. Integrated process planning and scheduling/rescheduling—an agent-based approach. International Journal of Production Research 2006;44(18–19):3627–55.
[27] Tan W, Khoshnevis B. A linearized polynomial mixed integer programming model for the integration of process planning and scheduling. Journal of Intelligent Manufacturing 2004;15:593–605.
[28] Chan FTS, Kumar V, Tiwari MK. Optimizing the performance of an integrated process planning and scheduling problem: an AIS-FLC based approach. In: Proceedings of IEEE conference on cybernetics and intelligent systems (CIS); 2006. p. 1–8.
[29] Li WD, McMahon CA. A simulated annealing-based optimization approach for integrated process planning and scheduling. International Journal of Computer Integrated Manufacturing 2007;20(1):80–95.
[30] Kim YK, Park K, Ko J. A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. Computers and Operations Research 2003;30:1151–71.
[31] Moon C, Seo Y. Evolutionary algorithm for advanced process planning and scheduling in a multi-plant. Computers and Industrial Engineering 2005;48: 311–25.
[32] Morad N, Zalzala A. Genetic algorithms in integrated process planning and scheduling. Journal of Intelligent Manufacturing 1999;10:169–79.
[33] Li XY, Shao XY, Gao L. Optimization of flexible process planning by genetic programming. International Journal of Advanced Manufacturing Technology 2008;38:143–53.
[34] Catron AB, Ray SR. ALPS—a language for process specification. International Journal of Computer Integrated Manufacturing 1991;4:105–13.
[35] Sormaz D, Khoshnevis B. Generation of alternative process plans in integrated manufacturing systems. Journal of Intelligent Manufacturing 2003;14:509–26.
[36] Fattahi P, Mehrabad MS, Jolai F. Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. Journal of Intelligent Manufacturing 2007;18(3):331–42.
[37] Langdon WB, Qureshi A. Genetic programming—computers using Natural Selection to generate programs. Technical report RN/95/76, Gower Street, London WCIE 6BT, UK, 1995.
[38] Moon C, Lee YH, Jeong CS, Yun YS. Integrated process planning and scheduling in a supply chain. Computers and Industrial Engineering 2008;54:1048–61.