Int. J. Production Economics ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Int. J. Production Economics

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

An effective hybrid algorithm for integrated process planning and scheduling

Xinyu Li, Xinyu Shao, Liang Gao *, Weirong Qian

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

ARTICLE INFO

Article history:
Received 12 February 2008
Accepted 28 January 2010

Keywords:
Process planning
Scheduling
Integrated process planning and scheduling
Hybrid algorithm

ABSTRACT

Process planning and scheduling are two of the most important functions in the manufacturing system. Traditionally, process planning and scheduling were regarded as separate tasks performed sequentially, where scheduling was implemented after process plans had been generated. However, their functions are usually complementary. If the two systems can be integrated more tightly, greater performance and higher productivity of manufacturing system can be achieved. In this paper, a new hybrid algorithm (HA) based approach has been developed to facilitate the integration and optimization of these two systems. To improve the optimization performance of the approach, an efficient genetic representation, operator and local search strategy have been developed. Experimental studies have been used to test the performance of the proposed approach and to make comparisons between this approach and some previous works. The results show that the research on integrated process planning and scheduling (IPPS) is necessary and the proposed approach is a promising and very effective method on the research of IPPS.

1. Introduction

Process planning and scheduling are two of the most important sub-systems in a manufacturing system. A process plan specifies raw materials or components needed to produce a product, processes and operations, which are necessary to transform those raw materials into the final product. The outcome of process planning includes the identification of machines, tools and fixtures suitable for a job and the arrangement of operations for a job. And, a job may have one or more alternative process plans. Process planning is the bridge of the product design and manufacturing. With the process plans of jobs as inputs, a scheduling task is to schedule the operations of all the jobs on machines while precedence relationships in the process plans are satisfied. Scheduling is the link of the two production steps, which are the preparing processes and putting them into action. Although there is a close relationship between process planning and scheduling, their integration is still a challenge in both research and applications (Sugimura et al., 2001).

In traditional approaches, process planning and scheduling were carried out in a sequential way, where scheduling was conducted separately after the process plans had been generated. Those approaches have become the obstacles to improve the productivity and responsiveness of the manufacturing systems and to cause the following problems (Kumar and Rajotia, 2002, 2003):

- Traditionally, in a manufacturing organization, the process planning function works in static. Process planner plans jobs separately. For each job, manufacturing resources on the shop floor are usually assigned on it without considering the competition for the resources from other jobs (Usher and Fernandes, 1996). This may lead to the process planners favoring to select the desirable resources for each job repeatedly. Therefore, the resulting optimum process plans often become infeasible when they are carried out in practice at the later stage (Lee and Kim, 2001).
- Even though process planners consider the restriction of the current resources on the shop floor, because of the time delay between planning phase and execution phase, the constraints considered in the planning phase may have already changed greatly, which may lead to the optimum process plans being infeasible (Kuhnle et al., 1994). Investigations have shown that 20–30% of the total production plans in a given period have to be rescheduled to adapt to dynamic changes in a production environment (Kumar and Rajotia, 2003).
- Traditionally, 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.
- In most cases, both for process planning and scheduling, a single criterion optimization technique is used to determine the best solution. However, the real production environment is best represented by considering more than one criterion simultaneously (Kumar and Rajotia, 2003). And, process planning emphasizes the technological requirements of a job,

* Corresponding author. Tel.: +86 27 87543871 85; fax: +86 27 87543074.
E-mail address: gaoliang@mail.hust.edu.cn (L. Gao).

0925-5273/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijpe.2010.04.001

Please cite this article as: Li, X., et al., An effective hybrid algorithm for integrated process planning and scheduling. International Journal of Production Economics (2010), doi:10.1016/j.ijpe.2010.04.001

while scheduling attaches importance to the timing aspects and resource sharing of all jobs. If there is no appropriate coordination, it may create conflicting problems.

To overcome these problems, there is an increasing need for deep research and application of the IPPS system. The IPPS can introduce significant improvements to the efficiency of manufacturing through eliminating or reducing scheduling conflicts, reducing flow-time and work-in-process, improving production resources utilizing and adapting to irregular shop floor disturbances (Lee and Kim, 2001). Without 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 review. Problem formulation is discussed in Section 3. Hybrid algorithm for IPPS is proposed in Section 4. Experimental studies and discussions are reported in Section 5. Section 6 is conclusion.

2. Literature review

In the beginning research of CIMS, some researchers have found that the IPPS is very important to the development of CIMS (Tan and Khoshnevis, 2000). The preliminary idea of IPPS was introduced by Chryssolouris et al. (1984) and Chryssolouris and Chan (1985). Beckendorff et al. (1991) used alternative process plans to improve the flexibility of manufacturing systems. Khoshnevis and Chen (1989) introduced the concept of dynamic feedback into IPPS. The integration model proposed by Zhang (1993) and Larsen (1993) extended the concepts of alternative process plans and dynamic feedback and defined an expression to the methodology of hierarchical approach. Some earlier works of IPPS had been summarized in Tan and Khoshnevis (2000) and Wang et al. (2006). In recent years, in the area of IPPS, several models have been reported, and they can be classified into three basic models based on IPPS: nonlinear process planning (NLPP), closed loop process planning (CLPP) and distributed process planning (DPP) (Li et al., 2010a).

2.1. Nonlinear process planning

The methodology of NLPP is to provide all alternative plans for each job 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 into the scheduling system.

NLPP is the most basic model of IPPS. Because the methodology of this model is very simple, most of the current researches on the integration model focus on the implementation and improvement of this model. Kim et al. (1997) gave a scheduling system that was supported by flexible process plans. Lee and Kim (2001) presented the NLPP model, which was based on the genetic algorithm (GA). Yang et al. (2001) presented a prototype of a feature-based multi-alternative process planning system. Thomalla (2001) investigated an optimization methodology for job shop scheduling with alternative process plans. Kim et al. (2003) used a symbiotic evolutionary algorithm for the IPPS problem. Li and McMahon (2007) used a simulated annealing-based approach for the IPPS problem. Shao et al. (2009) used a modified GA to solve the IPPS problem. Baykasoglu and Ozbakir (2009) analyzed the effect of dispatching rules on the scheduling performance of flexible job shop with different flexibility levels. However, through a number of experimental computations, Usher (2003) concluded that the advantages obtained by increasing the number of alternative process plans for a scheduling system diminishes rapidly when the number of plans reaches a certain level.

2.2. Closed loop process planning

The methodology of CLPP is a feedback mechanism. CLPP is a dynamic process planning system that faces the shop floor. CLPP generates real-process plans by means of a dynamic feedback from scheduling system. The process planning mechanism generates process plans based on available resources. Scheduling provides the information about the current shop floor status to process planning system, 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 it also can enhance the utilization of alternative process plans.

Usher and Fernandes (1996) divided the dynamic process planning to the static phase and the dynamic phase. Seethaler and Yellowley (2000) presented a dynamic process planning system, which can give the process plans based on the feedback of scheduling system. Anosike and Zhang (2009) proposed an agent-based approach for integrating manufacturing operations.

2.3. Distributed process planning

The methodology of DPP is to perform both the process planning and the scheduling simultaneously with a hierarchical approach. It divides the process planning and scheduling tasks into two phases. The first phase is the 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 at this stage as well. The process resources are also evaluated simultaneously. The second phase is the 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.

Kempenaers et al. (1996) demonstrated the three modules of the collaborative process planning system. Wu et al. (2002) gave the integration model of IPPS in the distributed virtual manufacturing environment. Zhang et al. (2003) presented the framework of concurrent process planning based on Holon. Wang et al. (2005) presented the framework of collaborative process planning system supported by a real-time monitoring system.

In this research, a new HA-based approach has been developed to facilitate the integration and optimization of the IPPS problem. Through experimental studies, the merits of the proposed approach can be shown clearly.

3. Problem formulation

3.1. Problem definition

The IPPS problem can be defined as follows (Guo et al., 2009):

> Given a set of N parts which are to be processed on machines with operations including alternative manufacturing resources, select suitable manufacturing resources and sequence the operations so as to determine a schedule in which the precedence constraints among operations can be satisfied and the corresponding objectives can be achieved.

In the manufacturing systems considered in this study, a set of process plans of each part is designed and maintained. The generation of one scheduling plan and the selection of process plan of each job from a set of process plans are determined based on the objectives. If there are N jobs, and each with G_i alternative process plans, the number of possible process plan combinations is $\{G_1 \times G_2 \times \cdots \times G_i \times \cdots \times G_N\}$. This problem is an NP-hard problem. For large problems, it is difficult to find perfect solutions in reasonable time. Therefore, in this research, one HA based approach has been developed to solve it. The mathematical model of IPPS can be found in Li et al. (2010b).

3.2. Representations for process plans and schedules

Three types of flexibility are considered in process planning (Hutchinson and Flughoeft, 1994; Saygin and Kilic, 1999): operation flexibility, sequencing flexibility and processing flexibility (Benjaafar and Ramakrishnan, 1996). Operation flexibility (Kim et al., 2003), which is also called routing flexibility (Lin and Solberg, 1991), relates to the possibility of performing one operation on alternative machines (Wahab and Stoyan, 2008), with possibly distinct processing time and costs. 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 can be obtained by the consideration of these flexibilities (Kim et al., 2003).

There are many methods used to describe the types of flexibility explained above (Catron and Ray, 1991), such as Petri-net (Lee and Junq, 1994), AND/OR graphs and network. Now, a network representation proposed by Ho and Moodie (1996), Kim (2003) and Sormaz and Khoshnevis (2003) is adopted here. There are three types of the nodes in the network: starting node, intermediate node and ending node (Kim et al., 2003). 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 can 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 that the same manufacturing feature can be performed by different process procedures. If the links following a node are connected by an OR connector, it only needs to traverse one of the OR-links (the links connected by the OR-connector are called OR-links), and an OR-link path can of course contain the other OR-link paths. 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 (Kim et al., 2003). One path from the starting node to the ending node is one alternative process plan. Fig. 1 shows one job's alternative process plans network.

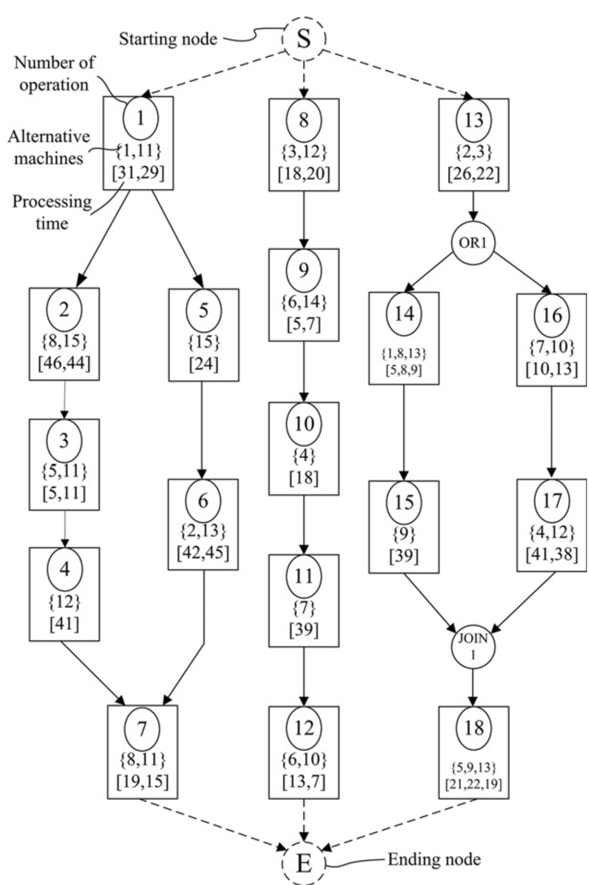

Fig. 1. Alternative process plans network of a job.

A Gantt chart has been popularly used to represent a schedule of a group of parts. The X-axis of the Gantt chart represents time. Each row in the Y-axis represents a machine and the specific arrangement for the operations of the jobs on the machine. In this paper, the Gantt chart is used to represent a schedule.

4. Hybrid algorithm optimization approach for IPPS

4.1. Hybrid algorithm model

4.1.1. Traditionally genetic algorithm

GA is one of the Evolutionary Algorithms. It was developed by Holland and Rechenberg (Franz, 2006). By imitating basic principles of nature evolution, they created an optimization algorithm that has successfully been applied in many areas. GA is able to search very large solution spaces efficiently by providing a concise computational cost, since it uses probabilistic transition rules instead of deterministic ones. It is easy to implement and is increasingly used to solve inherently intractable problems called NP-hard problems.

4.1.2. Local search strategy

Tabu search (TS) (Glover and Laguna, 1997) is a meta-heuristic method, which has been successfully applied in many scheduling problems and other combinatorial optimization problems. TS allows the searching process to explore solutions that do not decrease the objective function value if these solutions are not forbidden. It is usually obtained by keeping track of the last solution in terms of the action used to transform one solution to the next. It consists of several elements that contain the neighborhood structure, the move attributes, the tabu list, aspiration criteria and terminate criteria.

TS has emerged as one of the most efficient local search strategies for scheduling problems. In this study, it has been adopted as the local search strategy for every individual. And the neighborhood structure of the job shop scheduling problem used in Nowicki and Smutnicki (1996) has been adopted here. Because of the differences between job shop scheduling problem and IPPS problem, it also has been relatively modified to avoid the unlawful solution. The basic flow chart of TS is shown in Fig. 2.

In the proposed HA, when an individual is to perform local search, it should be converted to a feasible schedule at first. And then the solution is used as the initial solution of TS. After the local search, the output solution of TS should be encoded to a feasible individual.

4.1.3. Hybrid algorithm model

GA is able to search very large solution space efficiently, but its local search ability is not good. TS as a local search algorithm can search the local space very well. Through analyzing the optimization mechanism of these two algorithms, a hybrid algorithm model that synthesizes the advantages of GA and TS has been proposed to solve the IPPS problem. The basic procedure of the HA model is described as follows:

Step 1: Initialize population randomly, and set the parameters.

Step 2: Evaluate all populations.

Step 3: Set $Gen=Gen+1$.

Step 4: Generate a new generation population through reproduction, crossover and mutation.

Step 5: Local search by TS for every individual.

Step 6: Perform *Steps* 2 and 5 cyclically until terminate criteria satisfied.

According to this model, every individual evolves by the genetic operators firstly, and then it focuses on the local search. Based on the above procedure, to implement the HA model effectively, efficient encoding scheme of individuals, genetic operators and local search strategy are necessary. In the proposed algorithm, TS is adopted as the local search method. In the following sections, the details of the HA have been presented in detail.

4.2. Hybrid algorithm for IPPS

4.2.1. Encoding and decoding

Each chromosome in scheduling population consists of three parts with different lengths as shown in Fig. 3. The first part of chromosome is the alternative process plan string. The positions from 1 to N in this string represent the jobs from 1 to N. The number in the ith position represents the selected alternative process plan of the job i.

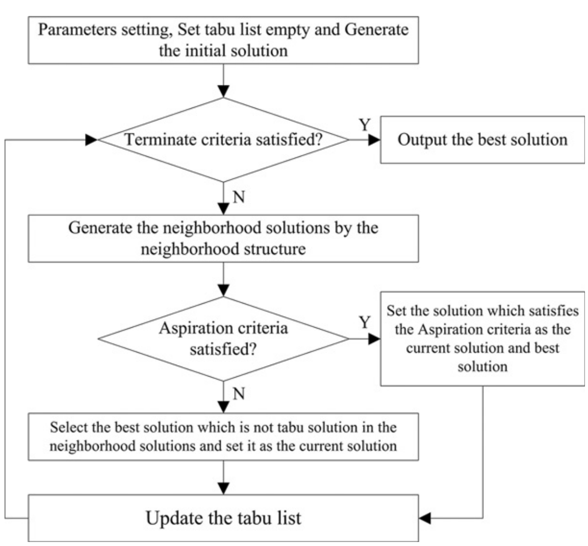

Fig. 2. Basic flow chart of TS.

The second part of chromosome is the scheduling plan string. In this paper, the scheduling encoding made up of the jobs numbers is the operation-based representation. This representation uses an unpartitioned permutation with P_{il}-repetitions of job numbers. In this representation, each job number appears P_{il} times in the chromosome. By scanning the chromosome from left to right, the fth appearance of a job number refers to the fth operation in the selected process plan of this job. The important feature of this representation is that any permutation of the chromosome can be decoded to a feasible solution. It is assumed that there are N jobs, and q_i is the number of operations of the process plan that has the most operation numbers among all the alternative process plans of the job i. Then the length of the scheduling plan string is equal to $\sum q_i$. The number of appearances of i in the scheduling plan string is equal to the number of operations of the selected alternative process plan. Based on this principle, the composition elements of scheduling plan string are determined. If the number of elements is less than $\sum q_i$, all the other elements are filled with 0. Therefore, the scheduling plan string is made up of jobs' numbers and 0. One scheduling plan string is generated by arraying all the elements randomly. And the alternative process plan string is generated by choosing the alternative process plan randomly for each job.

The third part of chromosome is the machine string. It denotes the selected machine set of the corresponding operations of all jobs. The length of this string is equal to the length of the scheduling plan string. It contains N parts, and the length of ith part is q_i. The ith part of this string denotes the selected machine set of the corresponding operations of job i. Assume the hth operation in the selected lth alternative process plan of job i can be processed by a machine set $S_{ilh}=\{m_{ilh1},m_{ilh2},\ldots,m_{ilhc_{ilh}}\}$. The ith part of this string can be denoted as $\{g_{il1}g_{il2}\ldots g_{ilh}\ldots g_{ilq_i}\}$, and g_{ilh} is an integer between 1 and c_{ilh} and it means that the hth operation in the lth selected alternative process plan of job i is assigned to the g_{ilh}th machine $m_{ilhg_{ilh}}$ in S_{ilh}. If the number of operations of the selected alternative process plan of job i is p and p is less than q_i, $\{g_{ilp}\ldots g_{ilq_i}\}$ are equal to 0.

Fig. 3 shows an example of an individual scheduling plan. In this example, N is equal to 6, and $q_1=6$, $q_2=6$, $q_3=6$, $q_4=7$, $q_5=5$, $q_6=5$. $\sum q_i$ is equal to 35. So, the scheduling plan string and machine string are made up of 35 elements, and the process plan string is made up of 6 elements. For job 1, the first alternative process plan has been chosen, and there are 6 operations in the selected process plan, so the 6 elements of scheduling plan string are 1. And in the first part of machine string, the first element of this part is 1 and it means that the first operation in the selected alternative process plan of job 1 is assigned to the first machine m_{1111} in S_{111}. The other elements in the chromosome can be deciphered respectively. The number of 0 in the scheduling plan string and the machine string is equal to 5-35-6-5-4-6-5-4.

The permutations can be decoded into semi-active, active, non-delay, and hybrid schedules. The active schedule is adopted in this paper. Recall that at this decoding stage, a particular individual of a scheduling plan is determined, that is, a fixed alternative process plan for each job is given. The notations used

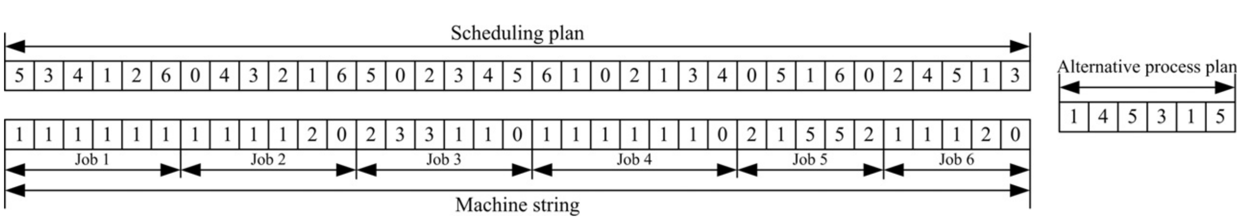

Fig. 3. The chromosome of the integration.

to explain the procedure are described below:

M the total number of machine;
o_{ijl} the jth operation in the lth alternative process plan of the ith job;
as_{ijl} the allowable starting time of operation o_{ijl};
s_{ijl} the earliest starting time of operation o_{ijl};
k the alternative machine corresponding to o_{ijl};
t_{ijlk} the processing time of operation o_{ijl} on machine k, $t_{ijlk}>0$;
c_{ijl} the earliest completion time of operation o_{ijl}, i.e. $c_{ijl}=s_{ijl}+t_{ijlk}$;

The procedure of decoding is as follows:

Step 1: Generate the machine of each operation based on the machine string of the chromosome.

Step 2: Determine the set of operations for every machine, $M_a=\{o_{ijl}\},\quad 1\le a\le M$.

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

Step 4: The allowable starting time for every operation, $as_{ijl}=c_{i(j-1)l}$ ($o_{ijl}\in M_a$), $c_{i(j-1)l}$ is the completion time of the pre-operation of o_{ijl} for the same job.

Step 5: Check the idle time of the machine of o_{ijl}, and get the idle areas [t_s,t_e], check these areas in turn (*if*: $\max(as_{ijl},t\_s)+t_{ijlk}\le t\_e$, the earliest starting time is $s_{ijl}=t\_s$, *else:* check the next area), if there is no area satisfying this condition:$s_{ijl}=\max(as_{ijl},c(o_{ijl}-1))$, $c(o_{ijl}-1)$ is the completion time of the pre-operation of o_{ijl} for the same machine.

Step 6: The completion time of every operation, $c_{ijl}=s_{ijl}+t_{ijlk}$.

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

In the above procedure, it can be obtained that the sets of starting time and completion time are for every operation of every job. It is a schedule for the shop.

4.2.2. Initial population and fitness evaluation

The encoding principle of the scheduling plan string in chromosome in this paper is an operation-based representation. The important feature of this representation is that any permutation of the chromosome can be decoded to a feasible schedule. It cannot break the constraints on precedence relations of operations. The initial population is generated based on the encoding principle. In this paper, the makespan is used as the objective.

4.2.3. Genetic operators for IPPS

It is important to employ good operators that can effectively deal with the problem and efficiently lead to excellent individuals in the population. The genetic operators can generally be divided into three classes: reproduction, crossover and mutation. And in each class, a large number of operators have been developed.

(1) *Reproduction*: The tournament selection scheme with a user-defined reproduction probabilistic has been used for reproduction operation. In tournament selection, a number of individuals are selected at random (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 (Langdon and Qureshi, 1995). This scheme can modify the selection pressure by changing the tournament size.

(2) *Crossover*: There are three parts in a chromosome and two separate crossover operations for each pair of the selected chromosomes.

The procedure of the first 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: Generate a crossover point randomly.
 Step 2.2: The elements in the process plan strings of P1 and P2, which are in the left side of the crossover point, are appended to the same positions in O1 and O2, respectively; the elements in the process plan strings of P1 and P2, which are in the right side of the crossover point, are appended to the same positions in O2 and O1, respectively.
Step 3: Secondly, in order to match the process plan strings of O1 and O2 and avoid obtaining unlawful 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 ones of the positions in the left side of crossover point in process plan string, these elements (include 0) are appended 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 ones of the positions in the left side of crossover point in process plan string, these elements (include 0) are appended to the same positions in O2 and they are deleted in P2.
 Step 3.2: Obtain the numbers of the remaining elements in scheduling plan strings of P1 and P2, n_1 and n_2. If $n_1\ge n_2$, for O1, it implies that the number of empty positions in O1 is bigger than the number of remaining elements in P2. So, n_1-n_2 empty positions in O1 are selected randomly and filled with 0. Then, the remaining elements in scheduling plan string of P2 are appended to the remaining empty positions in O1 seriatim. For O2, $n_1\ge n_2$ 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 and set to empty. And then, the remaining elements in scheduling plan string 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 generated.

An example of the first crossover operation is presented in Fig. 4.

Step 1: Select a pair of chromosomes P1 and P2 and initialize two empty offspring: O1 and O2 (see Fig. 4).
Step 2: First, cross over the process plan strings of P1 and P2 and get the process plan strings of O1 and O2:
 Step 2.1: Generate a crossover point randomly; this example is the third position;
 Step 2.2: The elements in the process plan strings of P1 and P2, which are in the left side of the third position, are appended to the same positions in O1 and O2, respectively; the elements in the process plan strings of P1 and P2, which are in the right side of the third position, are appended to the same positions in O2 and O1, respectively.
Step 3: Secondly, in order to match the process plan strings of O1 and O2 and avoid obtaining unlawful O1 and O2, the scheduling plan strings of P1 and P2 are crossovered as follows:
 Step 3.1: The elements that equate 0, 1, 2 or 3 in scheduling plan string of P1 are appended to the same positions in O1 and they are deleted in P1; the elements that equate 0, 1, 2

or 3 in scheduling plan string of P2 are appended to the same positions in O2 and they are deleted in P2;

Step 3.2: In this example, $n_1 = 14$, $n_2 = 12$, $n_1 > n_2$ and $n_1 - n_2 = 2$. For O1, two empty positions in O1 are selected randomly and are filled with 0, which have been marked out in O1 in Fig. 4. Then, the remaining elements in the scheduling plan of P2 are appended to the remaining empty positions in the O1 seriatim. For O2, two 0s are selected randomly in O2 and are set to empty, which have been marked out in O2 in Fig. 4. And then, the remaining elements in the scheduling plan of P1 are appended to the empty positions in O2 seriatim.

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

Second, a two-point crossover is implemented on the Machine-string. In this operation, two positions are selected by randomly generating two numbers at first, and then two new strings (the Machine strings of O1 and O2) are generated by swapping all characters between the positions of the two parent strings (the Machine strings of P1 and P2). After this procedure, in order to match the process plan strings of O1 and O2 and avoid obtaining unlawful O1 and O2, the Machine strings of O1 and O2 have to be checked as follows:

Step 1: Record the quantity of each job in scheduling plan strings of O1 and O2 (except 0), $n1_i$ and $n2_i$;

Step 2: For O1, compare $n1_i$ and the quantity of the elements (except 0) in the ith part of the machine string of O1 ($r1_i$). If $n1_i \geq r1_i$, $n1_i - r1_i$ 0s from the $r1_i$+1th position to the $n1_i$th position in this part of the machine string are set to g_{ilh} randomly (see Section 4.2.1). And, the elements from the first position to the $r1_i$th position are unchangeable, and the other elements in this part are set to 0. Then, the machine string of O1 matches the other two parts of O1. For O2, do the same procedure with O1. If $n1_i < r1_i$, the procedure is reversed.

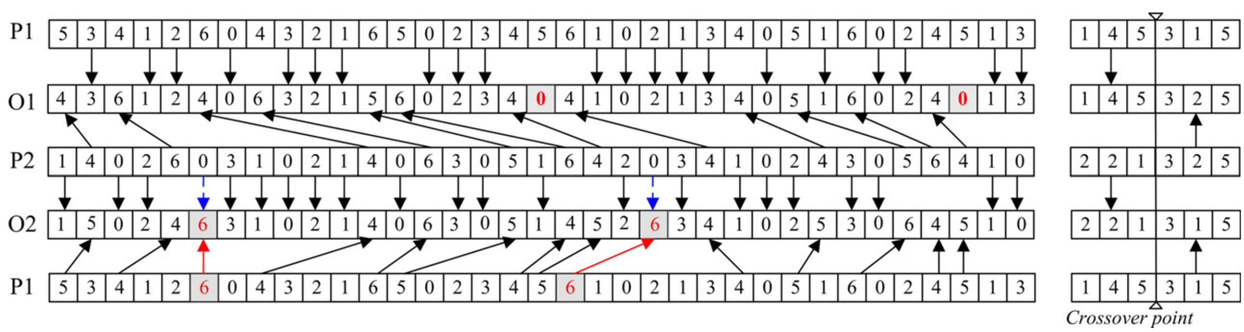

Fig. 4. The crossover operation for scheduling and alternative process plan strings.

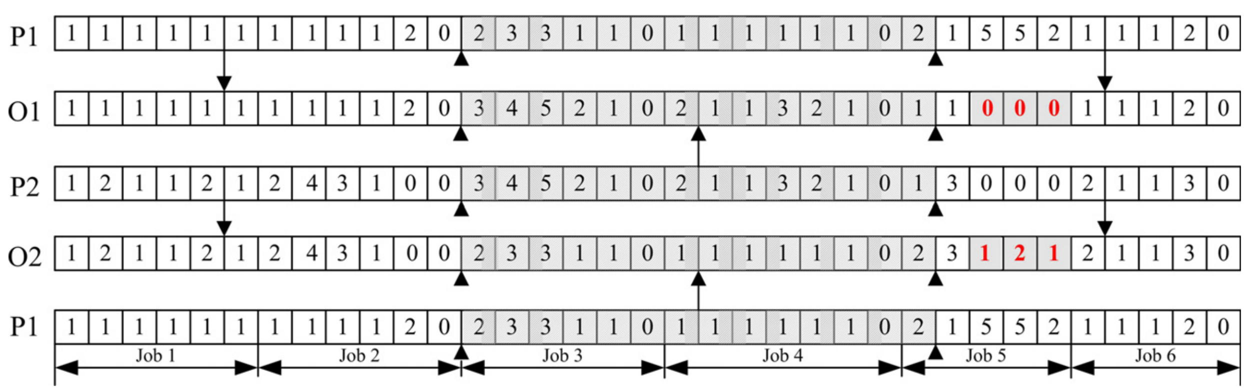

Fig. 5. The crossover operation for Machine-string.

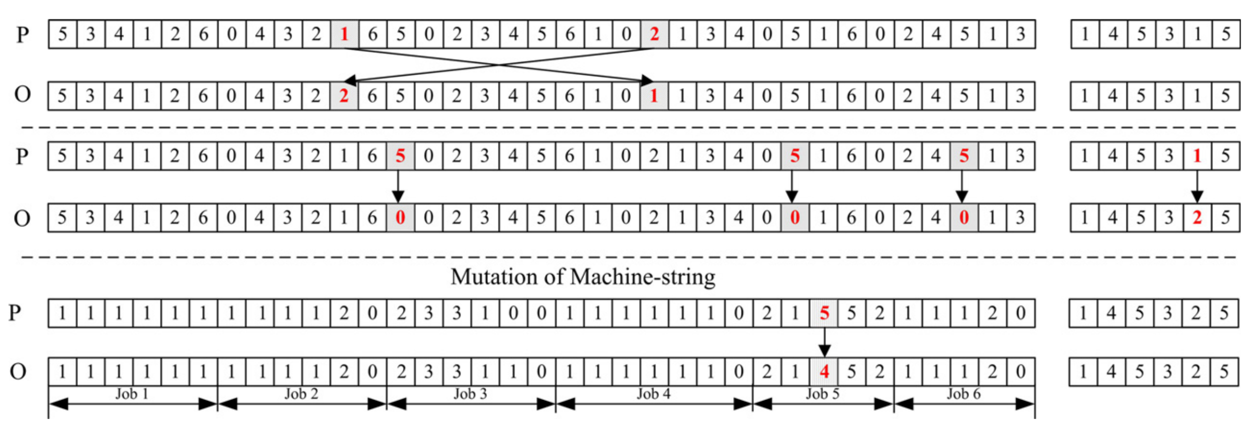

Fig. 6. Mutation operations.

Please cite this article as: Li, X., et al., An effective hybrid algorithm for integrated process planning and scheduling. International Journal of Production Economics (2010), doi:10.1016/j.ijpe.2010.04.001

One example of this crossover procedure is presented in Fig. 5.

(3) *Mutation*: In this paper, three mutation operations have been used. The first one is two-point swapping mutation, the second one is changing one job's alternative process plan, and the third one is the mutation of Machine-string. In the evolution procedure, one operator has been chosen randomly in every generation. Fig. 6 is an example of three mutation operators.

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 second 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 area (the number of alternative process plans).

Table 1
The HA parameters.

The size of the population, *PopSize*	200
Total number of generations, *MaxGen*	100
The permitted maximum step size with no improvement, *MaxStagnantStep*	20
The maximum iteration size of TS, *MaxIterSize*	200 × (*CurIter*/*MaxGen*)
Tournament size, *b*	2
Probability of reproduction operation, p_r	0.05
Probability of crossover operation, p_c	0.8
Probability of mutation operation, p_m	0.1
Length of tabu list, *maxT*	9

Table 2
The experimental results of experiment 1.

Model	No integration	Integration
Makespan	28	24

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

The mutation of Machine-string is applied in order to change the alternative machine represented in the Machine-string of chromosome. One element in the Machine-string is randomly chosen from the selected individual. Then, this element is mutated by altering the machine number to another one of the alternative machines at random.

5. Experimental studies and discussions

5.1. Test problems and experimental results

The proposed HA procedure was coded in C++ and implemented on a computer with a 2.0 GHz Core (TM) 2 Duo CPU. To illustrate the effectiveness and performance of the proposed HA in this paper, three instances from other papers are adopted here. The objective in this paper is to minimize makespan. The HA parameters for these problem instances are given in Table 1.

In HA, GA terminates when the number of generations reaches to the maximum value (*MaxGen*); TS terminates when the number of iterations reaches to the maximum size (*MaxIterSize*, *CurIter* is the current generation of GA) or the permitted maximum step size with no improvement (*MaxStagnantStep*). From the equation (*MaxIterSize* = 200 × (*CurIter*/*MaxGen*)), the *MaxIterSize* becomes bigger along with the *CurIter* in the computation procedure. In the early stage of evolution of HA, because GA cannot supply good initial individual for TS, it is little possible for TS to find the good solutions. The *MaxIterSize* of TS is small. This can save the computation time of HA. In the late stage of evolution of HA, GA can provide good initial individual for TS. In this case, enlarging the *MaxIterSize* of TS can help the TS to find the good solutions. Therefore, the maximum iteration size of TS is adaptive adjustment in the evolution process. This can balance the exploitation and exploration of HA very well and save the computation time.

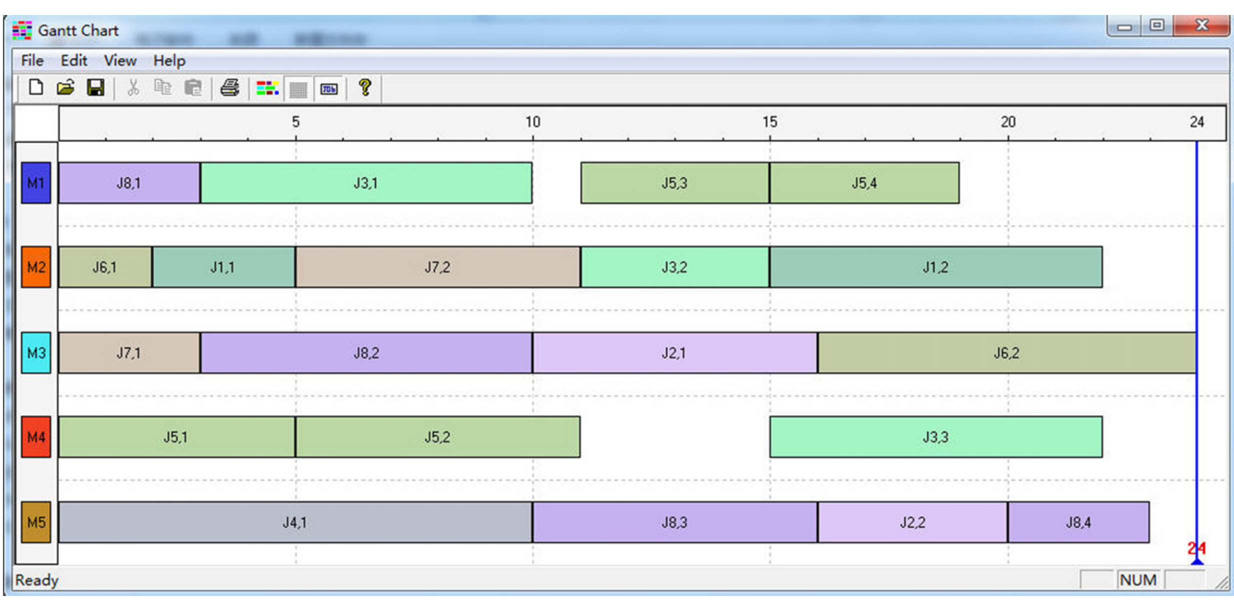

Fig. 7. Gantt chart of experiment 1 (Makespan = 24).

Please cite this article as: Li, X., et al., An effective hybrid algorithm for integrated process planning and scheduling. International Journal of Production Economics (2010), doi:10.1016/j.ijpe.2010.04.001

5.1.1. Experiment 1

Experiment 1 is adopted from Chan et al. (2006). In this experiment, one problem is constructed with eight jobs and five machines. And in this paper, the outsourcing machine was not considered. We only used the data from this paper. In this experiment, the integration model has been compared with the no integration model. No integration model means that there is no integration between process planning and scheduling. The purpose of this experiment is to show that the research on IPPS is necessary. Table 2 shows the experimental results and Fig. 7 illustrates the Gantt chart of this problem.

Table 3
The experimental results and comparisons of experiment 2 (symbiotic evolutionary algorithm—SEA, cooperative coevolutionary genetic algorithm—CCGA, the data that are marked by * is adopted from Kim et al. (2003).)

Problem	Number of jobs	Job number	Hierarchical approach*	CCGA*	SEA*	HA
1	6	1-2-3-10-11-12	483	458	428	427
2	6	4-5-6-13-14-15	383	363	343	343
3	6	7-8-9-16-17-18	386	366	347	345
4	6	1-4-7-10-13-16	328	312	306	306
5	6	2-5-8-11-14-17	348	327	319	322
6	6	3-6-9-12-15-18	506	476	438	429
7	6	1-4-8-12-15-17	386	378	372	372
8	6	2-6-7-10-14-18	376	363	343	343
9	6	3-5-9-11-13-16	507	464	428	427
10	9	1-2-3-5-6-10-11-12-15	504	476	443	430
11	9	4-7-8-9-13-14-16-17-18	413	410	369	369
12	9	1-4-5-7-8-10-13-14-16	361	360	328	327
13	9	2-3-6-9-11-12-15-17-18	505	498	452	436
14	9	1-2-4-7-8-12-15-17-18	423	420	381	380
15	9	3-5-6-9-10-11-13-14-16	496	482	434	427
16	12	1-2-3-4-5-6-10-11-12-13-14-15	521	512	454	446
17	12	4-5-6-7-8-9-13-14-15-16-17-18	474	466	431	423
18	12	1-2-4-5-7-8-10-11-13-14-16-17	417	396	379	377
19	12	2-3-5-6-8-9-11-12-14-15-17-18	550	535	490	476
20	12	1-2-4-6-7-8-10-12-14-15-17-18	473	450	447	432
21	12	2-3-5-6-7-9-10-11-13-14-16-18	525	501	477	446
22	15	2-3-4-5-6-8-9-10-11-12-13-14-16-17-18	560	567	534	518
23	15	1-4-5-6-7-8-9-11-12-13-14-15-16-17-18	533	531	498	470
24	18	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18	607	611	587	544

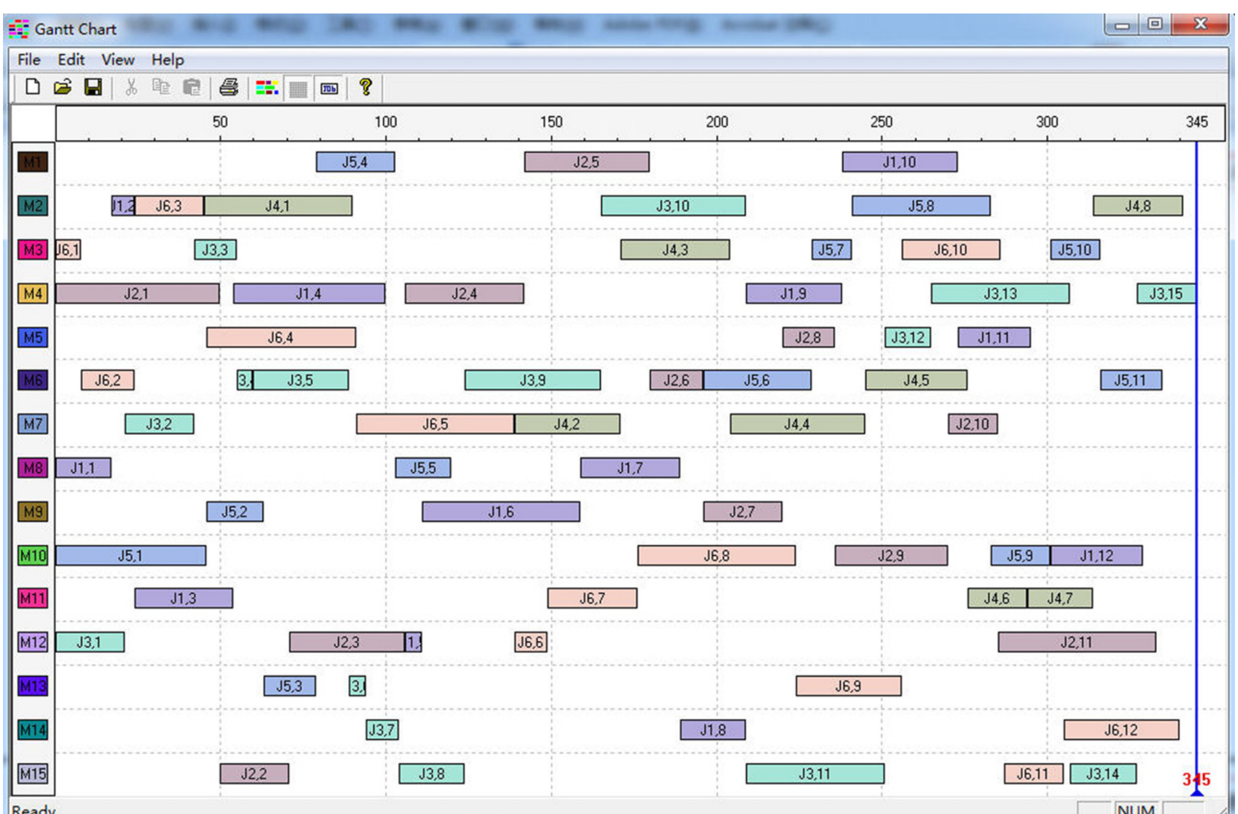

Fig. 8. Gantt chart of problem 3 in experiment 2 (Makespan=345).

Please cite this article as: Li, X., et al., An effective hybrid algorithm for integrated process planning and scheduling. International Journal of Production Economics (2010), doi:10.1016/j.ijpe.2010.04.001

Table 4
The experimental results of experiment 3 (the results marked by * are adopted from Leung et al. (2009)).

Solution methods	Petri-net*	ACO_03*	ACO_Agent06*	ACO_Agent09*	HA
Makespan	439	420	390	380	360

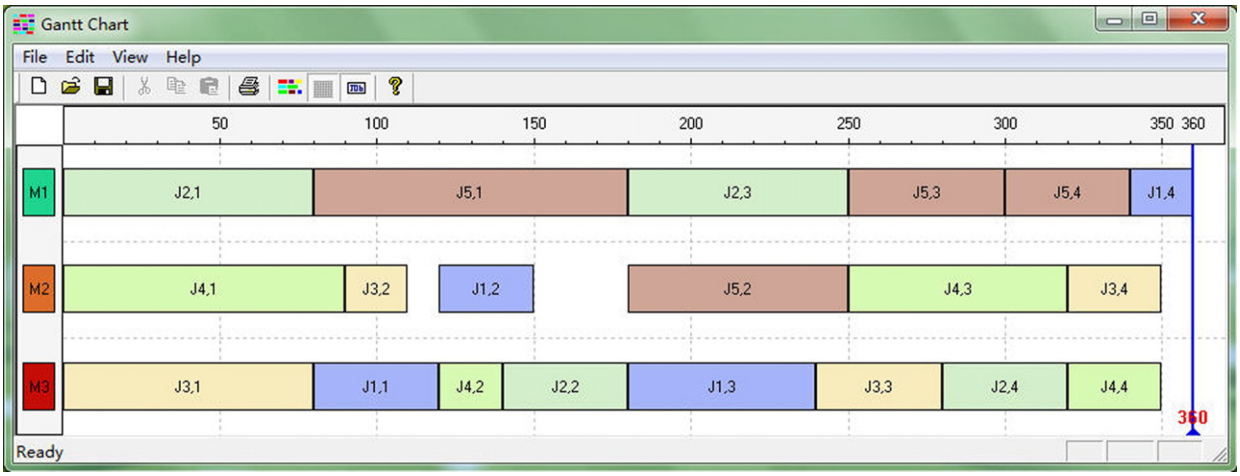

Fig. 9. Gantt chart of experiment 3 (Makespan = 360).

The experimental result of experiment 1 shows that the result of the no integration model is worse than that of the integration model. And the integration model can obtain better scheduling plans. This implies that the research on IPPS is necessary.

5.1.2. *Experiment 2*

Experiment 2 is adopted from Kim et al. (2003) and Kim (2003). In this experiment, 24 test-bed problems are constructed with 18 jobs and 15 machines. While designing the problems, the complexity of these problems as well as the number of jobs has been divided into five levels. Table 3 shows the experimental results, and the comparisons between the proposed HA and the methods in Kim et al. (2003) are also given. Fig. 8 illustrates the Gantt chart of problem 3 in this experiment.

Based on the experimental results of Table 3, only one solution (problem 5) of HA is worse than SEA and a few solutions of HA are equal to SEA, almost solutions (18 problems) of HA are better than the other methods. The merits of HA in large scale problems are obvious. It means that the proposed HA-based approach is more effective to obtain the optimal solutions.

5.1.3. *Experiment 3*

Experiment 3 is adopted from Leung et al. (2009). In this experiment, one problem is constructed with five jobs and three machines. Each job undergoes four different operations and each operation can be processed on one or more machines with respective processing time. Table 4 shows the experimental results, and the comparisons between the proposed HA and previous methods (Leung et al., 2009) are also given. Fig. 9 illustrates the Gantt chart of this problem. The results show that the proposed method can obtain better result.

These experimental results reveal that the proposed method can solve the IPPS problem effectively.

5.2. *Discussion*

Overall, the experimental results indicate that the research on IPPS is necessary and the proposed approach is a very effective and more acceptable approach for IPPS problem. The reasons are as follows. First, in the proposed HA approach, the selected process plans are not all the optimal ones, and it considers all the conditions synthetically. Second, in most experiments, the HA-based approach can obtain better results than other previously developed methods. This means that the proposed approach has more possibilities to obtain the better results of the IPPS problems.

6. Conclusion

Considering the complementarity of process planning and scheduling, the research has been conducted to develop a hybrid algorithm-based approach to facilitate the integration and optimization of these two systems. Process planning and scheduling functions are carried out simultaneously. To improve the optimization performance of the proposed approach, the efficient genetic representations, operator and local search strategy have been developed. To verify the feasibility of the proposed approach, three experimental studies have been carried out to compare this approach with other previous methods. The experimental results show that the research on IPPS is necessary and the proposed approach has achieved significant improvement.

Acknowledgements

The authors would like to thank the editor and anonymous referees whose comments helped a lot in improving this paper. This research work was supported by the National Natural Science Foundation of China (NSFC) under Grant no. 50825503 and the

Program for New Century Excellent Talents in University under Grant no. NCET-08-0232.

References

Anosike, A.I., Zhang, D.Z., 2009. An agent-based approach for integrating manufacturing operations. International Journal of Production Economics 121, 333–352.

Baykasoglu, A., Ozbakir, L., 2009. Analyzing the effect of dispatching rules on the scheduling performance through grammar based flexible scheduling system. International Journal of Production Economics, doi:10.1016/j.ijpe.2009.11.032.

Beckendorff, U., Kreutzfeldt, J., Ullmann, W., 1991. Reactive workshop scheduling based on alternative routings. In: Proceedings of the Conference on Factory Automation and Information Management. CRC Press Inc, Florida, pp. 875–885.

Benjaafar, S., Ramakrishnan, R., 1996. Modeling, measurement and evaluation of sequencing flexibility in manufacturing systems. International Journal of Production Research 34, 1195–1220.

Catron, A.B., Ray, S.R., 1991. ALPS—a language for process specification. The International Journal of Computer Integrated Manufacturing 4, 105–113.

Chan, F.T.S., Kumar, V., Tiwari, M.K., 2006. Optimizing the performance of an integrated process planning and scheduling problem: an AIS-FLC based approach. In: Proceedings of the CIS. IEEE.

Chryssolouris, G., Chan, S., Cobb, W., 1984. Decision making on the factory floor: an integrated approach to process planning and scheduling. Robotics and Computer-Integrated Manufacturing 1 (3–4), 315–319.

Chryssolouris, G., Chan, S., 1985. An integrated approach to process planning and scheduling. Annals of the CIRP 34 (1), 413–417.

Franz, R., 2006. Representations for Genetic and Evolutionary Algorithms. Springer – Verlag, Berling, Heidelberg, Netherlands.

Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic Publishers.

Guo, Y.W., Li, W.D., Mileham, A.R., Owen, G.W., 2009. Applications of particle swarm and optimization in integrated process planning and scheduling. Robotics and Computer-Integrated Manufacturing 25, 280–288.

Ho, Y.C., Moodie, C.L., 1996. Solving cell formation problems in a manufacturing environment with flexible processing and routing capabilities. International Journal of Production Research 34, 2901–2923.

Hutchinson, G.K., Flughoeft, K.A.P., 1994. Flexible process plans: their value in flexible automation systems. International Journal of Production Research 32 (3), 707–719.

Kempenaers, J., Pinte, J., Detand, J., et al., 1996. A collaborative process planning and scheduling system. Advances in Engineering Software 25, 3–8.

Khoshnevis, B., Chen, Q.M., 1989. Integration of process planning and scheduling function. In: IIE Integrated Systems Conference & Society for Integrated Manufacturing Conference Proceedings. Industrial Engineering & Management Press, Atlanta, pp. 415–420.

Kim, K.H., Song, J.Y., Wang, K.H., 1997. A negotiation based scheduling for items with flexible process plans. Computers & Industrial Engineering 33 (3–4), 785–788.

Kim, Y.K., 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〉.

Kim, Y.K., Park, K., Ko, J., 2003. A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. Computers & Operations Research 30, 1151–1171.

Kumar, M., Rajotia, S., 2002. An architecture of computer aided process-planning system integrated with scheduling using decision support system. In: Proceeding of the 10th International Manufacturing Conference in China (IMCC2002), Xiamen, China, October 2002.

Kumar, M., Rajotia, S., 2003. Integration of scheduling with computer aided process planning. Journal of Materials Processing Technology 138, 297–300.

Kuhnle, H., Braun H, J., Buhring, J., 1994. Integration of CAPP and PPC—interfusion manufacturing management. Integrated Manufacturing Systems 5 (2), 21–27.

Langdon, W.B., Qureshi, A., 1995. Genetic programming—computers using "Natural Selection" to generate programs. Technical Report RN/95/76, London, UK.

Larsen, N.E., 1993. Methods for integration of process planning and production planning. International Journal of Computer Integrated Manufacturing 6 (1–2), 152–162.

Lee, K.H., Junq, M.Y., 1994. Petri net application in flexible process planning. Computers & Industrial Engineering 27, 505–508.

Lee, H., Kim, S.S., 2001. Integration of process planning and scheduling using simulation based genetic algorithms. International Journal of Advanced Manufacturing Technology 18, 586–590.

Leung, C.W., Wong, T.N., Mak, K.L., Fung, R.Y.K., 2009. Integrated process planning and scheduling by an agent-based ant colony optimization. Computer & Industrial Engineering, doi:10.1016/j.cie.2009.09.003.

Li, W.D., McMahon, C.A., 2007. A simulated annealing-based optimization approach for integrated process planning and scheduling. International Journal of Computer Integrated Manufacturing 20 (1), 80–95.

Li, X.Y., Gao, L., Zhang, C.Y., Shao, X.Y., 2010a. A review on integrated process planning and scheduling. International Journal of Manufacturing Research 5 (2), 161–180.

Li, X.Y., Gao, L., Shao, X.Y., Zhang, C.Y., Wang, C.Y., 2010b. Mathematical modeling and evolutionary algorithm based approach for integrated process planning and scheduling. Computers & Operations Research 37, 656–667.

Lin, Y.J., Solberg, J.J., 1991. Effectiveness of flexible routing control. International Journal of Flexible Manufacturing Systems 3, 189–211.

Nowicki, E., Smutnicki, C., 1996. A fast taboo search algorithm for the job shop scheduling problem. Management Science 42 (6), 797–813.

Saygin, C., Kilic, S.E., 1999. Integrating flexible process plans with scheduling in flexible manufacturing systems. International Journal of Advanced Manufacturing Technology 15, 268–280.

Seethaler, R.J., Yellowley, I., 2000. Process control and dynamic process planning. International Journal of Machine Tools & Manufacture 40, 239–257.

Shao, X.Y., Li, X.Y., Gao, L., Zhang, C.Y., 2009. Integration of process planning and scheduling—a modified genetic algorithm-based approach. Computers & Operations Research 36, 2082–2096.

Sormaz, D., Khoshnevis, B., 2003. Generation of alternative process plans in integrated manufacturing systems. Journal of Intelligent Manufacturing 14, 509–526.

Sugimura, N., Hino, R., Moriwaki, T., 2001. Integrated process planning and scheduling in holonic manufacturing systems. In: IEEE International Symposium on Assembly and Task Planning Soft Research Park, Fukuoka, Japan, vol. 4, pp. 250–254.

Tan, W., Khoshnevis, B., 2000. Integration of process planning and scheduling—a review. Journal of Intelligent Manufacturing 11, 51–63.

Thomalla, C.S., 2001. Job shop scheduling with alternative process plans. International Journal of Production Economics 74, 125–134.

Usher, J.M., Fernandes, K.J., 1996. Dynamic process planning—the static phase. Journal of Materials Processing Technology 61, 53–58.

Usher, J.M., 2003. Evaluating the impact of alternative plans on manufacturing performance. Computers & Industrial Engineering 45, 585–596.

Wahab, M.I.M., Stoyan, S.J., 2008. A dynamic approach to measure machine and routing flexibilities of manufacturing systems. International Journal of Production Economics 113, 895–913.

Wang, L.H., Song, Y.J., Shen, W.M., 2005. Development of a function block designer for collaborative process planning. In: Proceedings of the CSCWD2005, Coventry, UK, pp. 24–26.

Wang, L.H., Shen, W.M., Hao, Q., 2006. An overview of distributed process planning and its integration with scheduling. International Journal of Computer Applications in Technology 26 (1–2), 3–14.

Wu, S.H., Fuh, J.Y.H., Nee, A.Y.C., 2002. Concurrent process planning and scheduling in distributed virtual manufacturing. IIE Transactions 34, 77–89.

Yang, Y.N., Parsaei, H.R., Leep, H.R., 2001. A prototype of a feature-based multiple-alternative process planning system with scheduling verification. Computers & Industrial Engineering 39, 109–124.

Zhang, H.C., 1993. IPPM—a prototype to integrated process planning and job shop scheduling functions. Annals of the CIRP 42 (1), 513–517.

Zhang, J., Gao, L., Chan, F.T.S., et al., 2003. A holonic architecture of the concurrent integrated process planning system. Journal of Materials Processing Technology 139, 267–272.