Expert Systems with Applications 39 (2012) 288–297

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Application of game theory based hybrid algorithm for multi-objective integrated process planning and scheduling

Xinyu Li [a], Liang Gao [a,*], Weidong Li [b]

[a] State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, 430074 Hubei, China
[b] Faculty of Engineering and Computing, Coventry University, Coventry, UK

ARTICLE INFO

Keywords:
Integrated process planning and scheduling
Game theory
Multi-objective
Hybrid algorithm

ABSTRACT

Process planning and scheduling are two key sub-functions in the manufacturing system. Traditionally, process planning and scheduling were regarded as the separate tasks to perform sequentially. Recently, a significant trend is to integrate process planning and scheduling more tightly to achieve greater performance and higher productivity of the manufacturing system. Because of the complementarity of process planning and scheduling, and the multiple objectives requirement from the real-world production, this research focuses on the multi-objective integrated process planning and scheduling (IPPS) problem. In this research, the Nash equilibrium in game theory based approach has been used to deal with the multiple objectives. And a hybrid algorithm has been developed to optimize the IPPS problem. Experimental studies have been used to test the performance of the proposed approach. The results show that the developed approach is a promising and very effective method on the research of the multi-objective IPPS problem.

1. Introduction

Process planning and scheduling are two key sub-functions in a manufacturing system. 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, fixtures, 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 function is to arrange 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, the integration of them is still a challenge in both researches and applications (Sugimura, Hino, & Moriwaki, 2001).

In traditional approaches, process planning and scheduling were carried out in a sequential way. Those methods have become the obstacles to improve the productivity and responsiveness of the manufacturing systems and to cause the following problems (Kumar & Rajotia, 2003):

- In traditional manufacturing organization, process planners plan 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 & Fernandes, 1996). This may lead to the process planners favoring to choose the desirable resources for each job repeatedly. Therefore, the resulting optimal process plans often become infeasible when they are carried out in practice at the later stage (Lee & Kim, 2001).
- Even though process planners consider the restrictions 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; this may lead to the optimal process plans infeasible (Kuhnle, Braun, & Buhring, 1994).
- 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 loads and create superfluous bottlenecks.
- In most cases, both for process planning and scheduling, a single criterion optimization technique is used for determining the best solution. However, the real production environment is best represented by considering more than one criterion simultaneously (Kumar & Rajotia, 2003). Furthermore, the process planning and scheduling may have conflicting objectives. Process planning emphasizes the technological requirements of an operation, while scheduling involves the timing aspects. If there is no appropriate coordination, it may create conflicting problems.

* Corresponding author. Tel.: +86 27 87549419; fax: +86 27 87543074.
E-mail address: gaoliang@mail.hust.edu.cn (L. Gao).

0957-4174/$ - see front matter
doi:10.1016/j.eswa.2011.07.019

To overcome these problems, there is an increasing need for deep researches and applications of the integrated process planning and scheduling (IPPS) system. It can introduce significant improvements to the efficiency of manufacturing through eliminating scheduling conflicts, reducing flow-time and work-in-process, improving production resources utilizing and adapting to irregular shop floor disturbances (Lee & 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. Therefore, in a complex manufacturing situation, it is ideal to integrate the process planning and scheduling more closely to achieve the global optimum in manufacturing, and increase the flexibility and responsiveness of the systems (Li & McMahon, 2007).

In the beginning research of CIMS, some researchers have found that the IPPS is very important to the development of CIMS (Tan & Khoshnevis, 2000). The preliminary idea of IPPS was introduced by Chryssolouris, Chan, and Cobb (1984) and Chryssolouris and Chan (1985). Beckendorff, Kreutzfeldt, and Ullmann (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 the hierarchical approach. Some earlier works of IPPS had been summarized in Tan and Khoshnevis (2000) and Wang, Shen, and Hao (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 (Li, Gao, Zhang, & Shao, 2010a): non-linear process planning (Kim, Song, & Wang, 1997; Thomalla, 2001), closed loop process planning (Seethaler & Yellowley, 2000; Usher & Fernandes, 1996) and distributed process planning (Wang, Song, & Shen, 2005; Zhang, Gao, & Chan, 2003).

In the past decades, the optimization approaches of the IPPS problems also have achieved several improvements. Especially, several optimization methods have been developed based on the modern artificial intelligence technologies, such as evolutionary algorithms, simulated annealing (SA) algorithm, particle swarm optimization (PSO) algorithm and the multi-agent system (MAS) based approach. Kim, Park, and Ko (2003) used a symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. Shao, Li, Gao, and Zhang (2009) used a modified genetic algorithm (GA) to solve IPPS problem. Li, Gao, Shao, Zhang, and Wang (2010b) proposed the mathematical models of IPPS and an evolutionary algorithm based approach to solve it. Chan, Kumar, and Tiwari (2009) proposed an enhanced swift converging SA algorithm to solve IPPS problem. Guo, Li, Mileham, and Owen (2009a, 2009b) proposed the PSO based algorithms to solve the IPPS problem. Shen, Wang, and Hao (2006) provided a literature review on the IPPS, particularly on the agent-based approaches for the IPPS problem. Wong, Leung, Mak, and Fung (2006) presented an online hybrid agent-based negotiation MAS for integrating process planning with scheduling/rescheduling. Shukla, Tiwari, and Son (2008) presented a bidding-based MAS for solving IPPS. Li, Zhang, Gao, Li, and Shao (2010c) developed an agent-based approach to facilitate the IPPS.

Most of the current researches on IPPS have been concentrated on the single objective. However, because different departments in a company have different expectations in order to maximize their own profits, for example, the manufacturing department expects to reduce costs and improve work efficiency, the managers want to maximize the utilization of the existing resources, and the sale department hopes to better meet the delivery requirements of the customers, in this case, only considering the single objective can not meet the requirements from the real-world production. Therefore, further studies are required for IPPS, especially on the multi-objective IPPS problem. However, only seldom papers focused their researches on the multi-objective IPPS problem. Morad and Zalzala (1999) proposed a GA based on weighted-sum method to solve multi-objective IPPS problem. Li and McMahon (2007) proposed a SA based approach for multi-objective IPPS problem. Baykasoglu and Ozbakir (2009) proposed an approach which made use of grammatical representation of generic process plans with a multiple objective tabu search (TS) framework to solve multi-objective IPPS effectively. Zhang and Gen (2010) proposed a multi-objective GA approach for solving process planning and scheduling problems in a distributed manufacturing system.

In this paper, a novel approach has been developed to facilitate the multi-objective IPPS problem. A game theory based hybrid algorithm has been applied to solve the multi-objective IPPS problem.

The remainder of this paper is organized as follows: problem formulation is discussed in Section 2. In Section 3, the game theory model of the multi-objective IPPS has been presented. A proposed algorithm for solving multi-objective IPPS problem is given in Section 4. Experimental results are reported in Section 5. Section 6 is conclusions.

2. Problem formulation

The IPPS problem can be defined as follows (Guo, Li, Mileham, & Owen, 2009b):

> *"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 this research, scheduling is often assumed as the job shop scheduling, and the mathematical model of IPPS is based on the mixed integer programing model of the job shop scheduling problem (JSP). In this research, the following three criteria are considered to be optimized simultaneously: in order to improve the work efficiency, selecting the maximal completion time of machines, i.e., the *Makespan*, as one objective; in order to improve the utilization of the existing resources, especially for the machines, selecting the maximal machine workload (*MMW*), i.e., the maximum working time spent on any machine, and the total workload of machines (*TWM*), i.e., the total working time of all machines, as the other two objectives.

In order to solve this problem, the following assumptions are made:

(1) Jobs are independent. Job preemption is not allowed and each machine can handle only one job at a time.
(2) The different operations of one job can not 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 transmission time is assumed to be negligible.
(5) Setup time for the operations on the machines is independent of the operation sequence and is included in the processing time.

Based on these assumptions, the mathematical model of the multi-objective IPPS problem is described as follows (Li et al., 2010b):

The notations used to explain the model are described below:

N: the total number of jobs
M: the total number of machines
G_i: the total number of alternative process plans of job i
o_{ijl}: the jth operation in the lth alternative process plan of the job i
P_{il}: the number of operation in the lth alternative process plan of the job i
k: the alternative machine corresponding to o_{ijl}
t_{ijlk}: the processing time of operation o_{ijl} on machine k
c_{ijlk}: the earliest completion time of operation o_{ijl} on machine k
c_i: the completion time of job i
W_k: the workload of machine k
A: a very large positive number

$$X_{il} = \begin{cases} 1 & \text{the lth flexible process plan of job } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ijlpqsk} = \begin{cases} 1 & \text{the operation } o_{ijl} \text{precedes the operation } o_{pqs} \text{ on machine } k \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{ijlk} = \begin{cases} 1 & \text{if machine } k \text{ is selected for } o_{ijl} \\ 0 & \text{otherwise} \end{cases}$$

Objectives:

(1) Minimizing the maximal completion time of machines (*Makespan*):

$$\text{Min} \quad f_1 = Makespan = \text{Max} c_{ijlk} \; i \in [1, N], \quad j \in [1, P_{il}], \quad l \in [1, G_i], \quad k \in [1, M]. \tag{1}$$

(2) Minimizing the maximal machine workload (*MMW*):

$$\text{Min} \quad f_2 = MMW = \text{Max} \quad W_k \; k \in [1, M]. \tag{2}$$

(3) Minimizing the total workload of machines (*TWM*):

$$\text{Min} \quad f_3 = TWM = \sum_{k=1}^{M} W_k \; k \in [1, M]. \tag{3}$$

Subject to:

(1) For the first operation in the lth alternative process plan of job i:

$$c_{i1lk} + A(1 - X_{il}) \geqslant t_{i1lk} \quad i \in [1, N], \quad l \in [1, G_i], \quad k \in [1, M]. \tag{4}$$

(2) For the last operation in the lth alternative process plan of job i:

$$c_{iP_{il}lk} - A(1 - X_{il}) \leqslant Makespan \quad i \in [1, N], \quad l \in [1, G_i], \quad k \in [1, M]. \tag{5}$$

(3) The different operations of one job can not be processed simultaneously:

$$c_{ijlk} - c_{i(j-1)lk_1} + A(1 - X_{il}) \geqslant t_{ijlk} \quad i \in [1, N], \quad j \in [1, P_{il}], \quad l \in [1, G_l], \quad k, k_1 \in [1, M]. \tag{6}$$

(4) Each machine can handle only one job at a time:

$$c_{pqsk} - c_{ijlk} + A(1 - X_{il}) + A(1 - X_{ps}) + A(1 - Y_{ijlpqsk}) \geqslant t_{pqsk}, \tag{7}$$

$$c_{ijlk} - c_{pqsk} + A(1 - X_{il}) + A(1 - X_{ps}) + AY_{ijlpqsk} \geqslant t_{ijlk} \; i, \quad p \in [1, N], \quad j, q \in [1, P_{il,ps}], \quad l, s \in [1, G_{i,p}], \quad k \in [1, M]. \tag{8}$$

(5) Only one alternative process plan can be selected of job i:

$$\sum_{l} X_{il} = 1 \; l \in [1, G_i]. \tag{9}$$

(6) Only one machine for each operation should be selected:

$$\sum_{k=1}^{M} Z_{ijlk} = 1 \; i \in [1, N], \quad j \in [1, P_{il}], \quad l \in [1, G_l]. \tag{10}$$

The objective functions are Eqs. (1)–(3). In this research, these three objectives have been considered for the IPPS problem. The constraints are in-Eqs. (4)–(10). Constraint (6) expresses that the different operations of a job are unable to be processed simultaneously. This is the constraint of different processes for a job. Constraints (7) and (8) show that each machine can handle only one job at a time. This is the constraint of a machine. Constraint (9) ensures that only one alternative process plan can be selected for each job in one schedule. Constraint (10) guarantees that only one machine for each operation should be selected.

Many studies have been devoted to do the research on the multi-objective optimization. These developed methods can be generally classified into three following different types (Hsu, Dupas, Jolly, & Goncalves, 2002):

- The first type is using the weighted-sum method to transform the multi-objective problem to a mono-objective problem.
- The second type is the non-Pareto approach. This method utilizes operators for processing the different objectives in a separated way.
- The third type is the Pareto approach. This method is directly based on the Pareto optimality concept.

In this paper, the Nash equilibrium in game theory based approach has been used to deal with the multiple objectives. After dealing with the multiple objectives, a hybrid algorithm has been used to optimize the multi-objective in the vast search space.

3. Game theory model of multi-objective IPPS

Game theory is a good method to analyze the interaction of several decision makers. It is a very important tool in the modern economy. Recently, it has been used to solve some complex engineering problems, such as power systems, collaborative product design, etc. (Li, Gao, Li, & Guo, 2008). In this paper, Non-cooperative game theory has been applied to deal with the conflict and competition among the multiple objectives in multi-objective IPPS problem. In this approach, the objectives of this problem can be seen as the players in the game, and the Nash equilibrium solutions are taken as the optimal results.

3.1. Game theory model of multi-objective optimization problem

The definition of general multi-objective optimization problem (MOP) is described as follows:

General MOP contains n variables, k objectives and m constraints. The mathematical definition of MOP is described as follows:

$$\begin{aligned} \text{Max/Min} \quad & y = f(x) = \{f_1(x), f_2(x), \ldots, f_k(x)\}, \\ \text{s.t.} \quad & e(x) = \{e_1(x), e_2(x), \ldots, e_m(x)\} \leqslant 0, \\ & x = (x_1, x_2, \ldots, x_n) \in X \; y = (y_1, y_2, \ldots, y_k) \in Y. \end{aligned} \tag{11}$$

x is the variable, y is the objective, X is the variables space, Y is the objectives space, $e(x) \leqslant 0$ is the constraints.

In order to apply game theory to deal with the multiple objectives, the mapping between MOP and game theory should been presented. This means that the game theory model of the MOP should be constructed.

Comparing the MOP with the game theory, the MOP can be described by game theory as follows: k objectives in MOP can be described as the k players in game theory, X in MOP can be described as the decision space S in game theory, $f_i(x)$ in MOP can be described as the utility function u_i in game theory, $e(x)$ in MOP can be described as the constraints in game theory.

Defining mapping φ_i: $X \to S_i$ as the decision strategies space of the ith player, and $\bigcup_{i=1}^{k} S_i = X$; defining mapping $\varphi_i : f_i \to u_i$ as the decision strategies set of the ith player, and then the game theory model of MOP can be defined as follows:

$$G = \{S; U\} = \{S_1, S_2, \ldots, S_k; \quad u_1, u_2, \ldots, u_k\}. \tag{12}$$

3.2. Nash equilibrium and MOP

Nash equilibrium is a very important concept in the non-cooperative game theory. In Nash equilibrium, the strategy of each player is the best strategy when giving the strategies of the other players. If the number of the players is limited, at least, the game has one Nash equilibrium solution.

The Nash equilibrium can be defined as follows:

$s^* = \{s_1^*, s_2^*, \ldots, s_k^*\}$ is a strategy set of the game in Eq. (12). If the s_i^* is the best strategy for the ith player when giving the strategies $(s^*_{...i})$ of the other players, i.e., for the any ith player and the $s_i^j \in S_i$, Eq. (13) is right, s^* can be seen as one Nash equilibrium solution in this game:

$$\begin{aligned} &u_i(s_i^*, s^*_{...i}) \geqslant u_i\left(s_i^j, s^*_{...i}\right), \\ &s^*_{...i} = \left\{s_1^*, s_2^*, \ldots, s_{i-1}^*, \quad s_{i+1}^*, \ldots, s_k^*\right\}. \end{aligned} \tag{13}$$

Therefore, for a MOP (Eq. (11)), $\{f_1(x), f_2(x), \ldots, f_k(x)\}$ can be seen as the k players in a game. The decision strategies space S equals to the variables space X. And the utility function for each player is $f_i(S)$. The Nash equilibrium solution $s^* = \{s_1^*, s_2^*, \ldots, s_k^*\}$ can be seen as one solution of the MOP (Eq. (11)).

Therefore, in Nash equilibrium, each objective has its own effect on the whole decision of the MOP, no one can dominate the decision-making process.

3.3. Non-cooperative game theory for multi-objective IPPS problem

In order to use the Non-cooperative game theory to deal with the multiple objectives in multi-objective IPPS problem, the game theory model of multi-objective IPPS problem should be constructed. In this paper, multi-objective IPPS problem has three objectives. They can be seen as three players in the game. The utility function of the first player is the first objective function ($u_1 = f_1$), the utility function of the second player is the second objective function ($u_2 = f_2$), and the utility function of the third player is the third objective function ($u_3 = f_3$) (f_1, f_2 and f_3 see Section 2). The game theory model of the multi-objective IPPS problem can be described as follows:

$$G = \{S; u_1, u_2, u_3\}. \tag{14}$$

The Nash equilibrium solution of this model is taken as the optimal result of the multi-objective IPPS problem.

4. Applications of the proposed algorithm on multi-objective IPPS

4.1. Work flow of the proposed algorithm

In order to solve the game theory model of the multi-objective IPPS problem effectively, one approach with a hybrid algorithm (the hybrid of GA and TS) has been proposed. The work flow of the method is shown in Fig. 1. The basic procedure of the method is described as follows:

Step 1: Set the parameters of the algorithm, including the parameters of the hybrid algorithm and the Nash equilibrium solution algorithm.
Step 2: Initialize population randomly.
Step 3: Evaluate all population, and calculate all the three objectives of every individual.
Step 4: Use the Nash equilibrium solutions algorithm to find the Nash equilibrium solutions in the current generation, and record them.
Step 5: Is the terminate criteria satisfied?
If yes, go to *Step 8*.
Else, go to *Step 6*.
Step 6: Generate the new population by the hybrid algorithm.
Step 6.1: Generate the new population by the genetic operations, including reproduction, crossover and mutation.
Step 6.2: Local search by TS for every individual.
Step 7: Go to *Step 3*.
Step 8: Use the Nash equilibrium solutions algorithm to compare all the recorded Nash equilibrium solutions in every generation, and select the best solutions.
Step 9: Output the best solutions.

The Nash equilibrium solutions algorithm and the hybrid algorithm are presented in the next sub-sections.

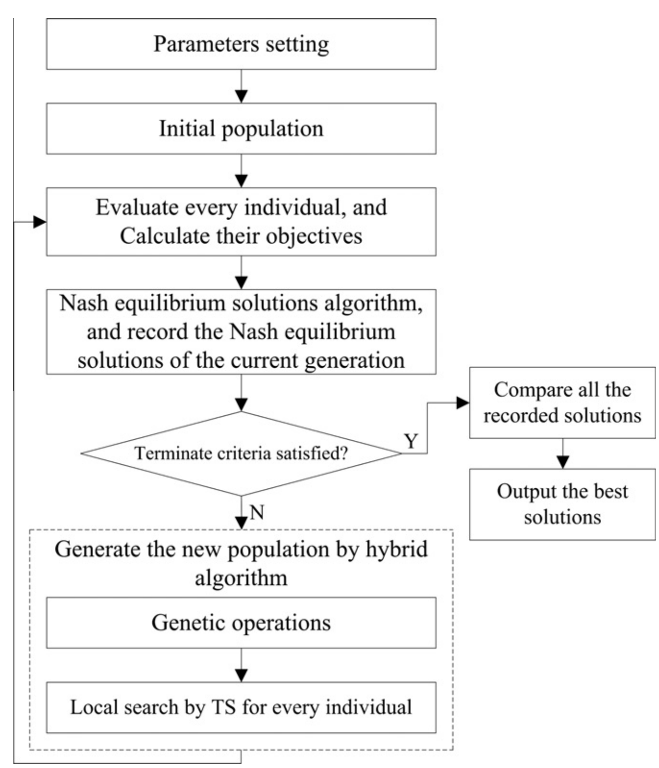

Fig. 1. Work flow of the proposed algorithm.

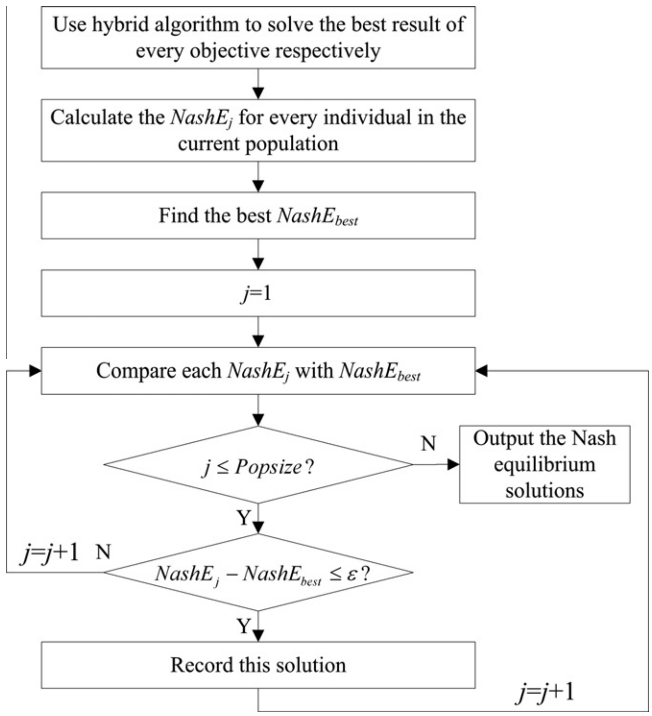

Fig. 2. Work flow of the Nash equilibrium solutions algorithm.

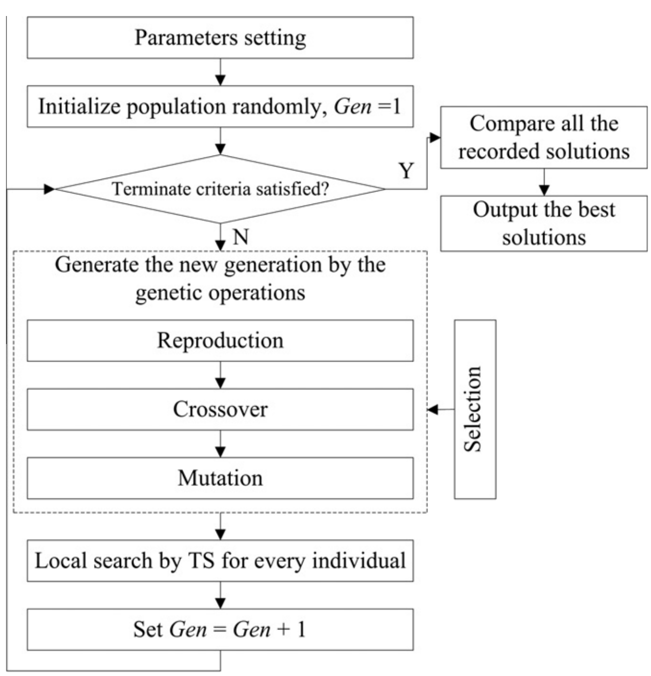

Fig. 3. Work flow of the hybrid algorithm.

4.2. Nash equilibrium solutions algorithm for multi-objective IPPS

In Nash equilibrium, the strategy of each player is the best strategy when giving the strategies of the other players. The main purpose of Nash equilibrium solution is to keep every objective trying its best to approximate to its own best result and can not damage the benefits of the other objectives. In Nash equilibrium, each objective has its own effect on the whole decision of the MOP, no one can dominate the decision-making process, and one criterion which has been proposed to judge the solutions is described as follows:

Table 1
The parameters of the proposed algorithm.

Parameters	
The size of the population, *Popsize*	400
Total number of generations, *maxGen*	200
The permitted maximum step size with no improving, *maxStagnantStep*	20
The maximum iteration size of TS, *maxIterSize*	$200 \times (curIter/maxGen)$
Probability of reproduction operation, p_r	0.05
Probability of crossover operation, p_c	0.6
Probability of mutation operation, p_m	0.1
Length of tabu list, *maxT*	10
Nash equilibrium solution factor, ε	0.1

Table 2
The data of problem 1 (Baykasoglu & Ozbakir, 2009).

Job	Operation	Alternative machines with processing time					Alternative operation sequences
		M_1	M_2	M_3	M_4	M_5	
1	O_1	57	40	88	62	77	O_3–O_4–O_1–O_2
	O_2	7	10	11	10	5	O_4–O_3–O_1–O_2
	O_3	95	74	76	71	93	O_2–O_4–O_1–O_3
	O_4	24	18	22	28	26	O_3–O_1–O_4–O_2
2	O_1	84	76	68	98	84	O_1–O_2–O_3
	O_2	20	10	15	20	19	O_1–O_3–O_2
	O_3	91	88	98	87	90	O_3–O_2–O_1
							O_2–O_1–O_3
3	O_1	65	87	58	80	74	O_2–O_4–O_1–O_3
	O_2	46	21	38	52	18	O_1–O_4–O_2–O_3
	O_3	19	22	19	14	22	O_1–O_4–O_5
	O_4	73	56	64	72	60	O_4–O_2–O_1–O_3
	O_5	96	98	96	95	98	
4	O_1	13	7	13	12	11	O_1–O_2–O_4–O_3
	O_2	52	64	97	47	40	O_2–O_1–O_4–O_3
	O_3	20	30	17	11	14	O_4–O_2–O_1–O_3
	O_4	94	66	80	79	95	O_3–O_2–O_4–O_1
5	O_1	94	97	55	78	85	O_3–O_1–O_2
	O_2	31	23	19	42	17	O_3–O_4
	O_3	88	65	76	64	80	O_3–O_2–O_1
	O_4	88	74	90	92	75	O_1–O_3–O_2

Table 3
Experimental results of problem 1 (the data marked by * was adopted from Baykasoglu and Ozbakir (2009)).

Criteria	Grammatical approach*	Solution 1	Solution 2
Makespan	394	165	170
Maximal machine workload (*MMW*)	328	159	158
Total workload of machines (*TWM*)	770	764	740

Table 4
Selected operation sequence for each job of problem 1.

Job	Solution 1	Solution 2
1	O_3–O_4–O_1–O_2	O_3–O_4–O_1–O_2
2	O_1–O_2–O_3	O_3–O_2–O_1
3	O_2–O_4–O_1–O_3	O_2–O_4–O_1–O_3
4	O_4–O_2–O_1–O_3	O_4–O_2–O_1–O_3
5	O_3–O_4	O_3–O_4

$$NashE_j = \sum_{i=1}^{3} DOBJ_{ji} \tag{15}$$

$$DOBJ_{ji} = \frac{CurrentObjective_{ji} - BestObjective_i}{BestObjective_i}, \tag{16}$$

$NashE_j$ is the Nash equilibrium criterion of the jth individual. $DOBJ_{ii}$ is calculated by the Eq. (16). $CurrentObjective_{ji}$ is the objective of the ith objective of the jth individual. $BestObjective_i$ is the best objective of the ith objective.

The work flow of the Nash equilibrium solutions algorithm is shown in Fig. 2. The basic procedure of this algorithm is described as follows:

Step 1: Firstly, use the hybrid algorithm (Li, Shao, Gao, and Qian, (2010d)) to optimize the mono-objective IPPS problem and getting the best result for every objective.
Step 2: Calculate the $NashE_j$ for every individual in the current population.
Step 3: Find the best $NashE_j$, and set it as the $NashE_{best}$.
Step 4: Compare each $NashE_j$ with the $NashE_{best}$.
Step 5: Is $j \leqslant Popsize$?
If yes, go to *Step 6*.
Else, go to *Step 9*.
Step 6: Is $NashE_j - NashE_{best} \leqslant \varepsilon$? ($\varepsilon$ is the Nash equilibrium solution factor).
If yes, go to *Step 7*.
Else, go to *Step 8*.
Step 7: Record this solution and $j = j + 1$, go to *Step 4*.
Step 8: $j = j + 1$, go to *Step 4*.
Step 9: Output the Nash equilibrium solutions for this generation.

This algorithm is also used to the select the final results from all the recorded Nash equilibrium solutions in every generation.

From the work flow of this algorithm, we can find that every objective is trying its best to approximate to its own best result and no one of them can dominate the whole decision of the multi-objective IPPS problem.

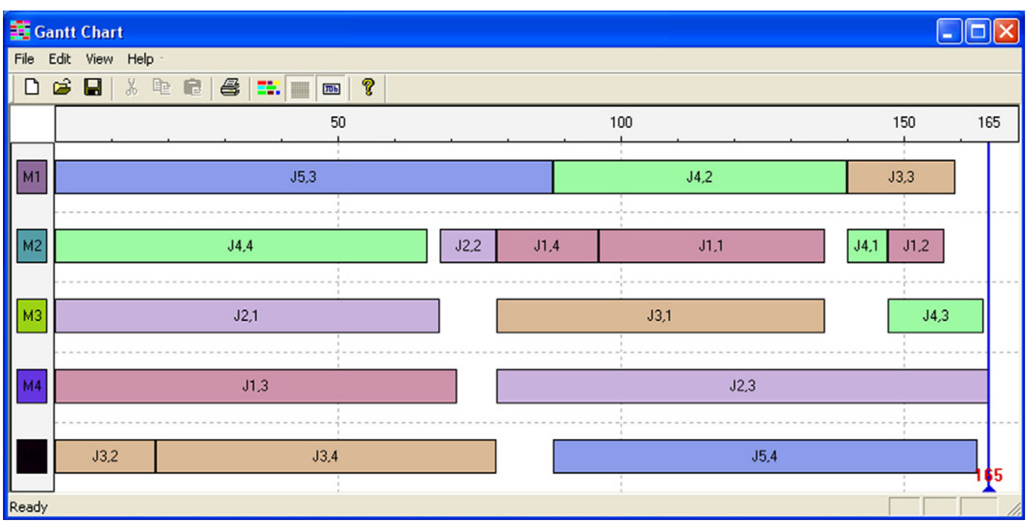

Fig. 4. Gantt chart of the solution 1 of problem 1.

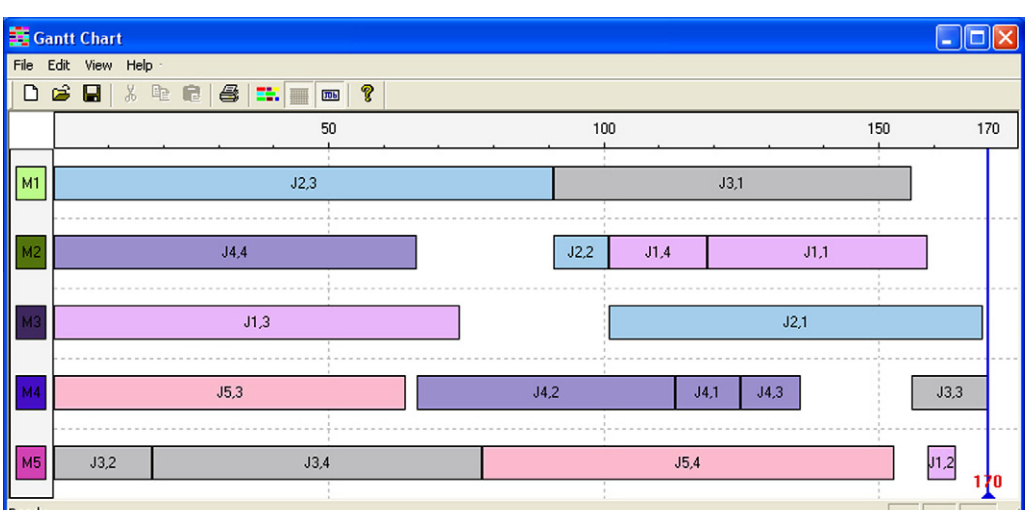

Fig. 5. Gantt chart of the solution 2 of problem 1.

Table 5
The data of problem 2.

Job	Operation	Alternative machines with processing time								Alternative operation sequences
		M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	
1	O_1	7	8	6	7	8	9	10	8	O_1–O_2–O_3–O_4
	O_2	20	19	21	22	18	23	20	21	O_2–O_1–O_3–O_4
	O_3	10	11	12	9	10	13	10	11	O_1–O_2–O_4–O_3
	O_4	30	32	33	29	31	32	33	30	O_2–O_1–O_4–O_3
2	O_1	50	54	48	52	51	53	50	48	O_2–O_1–O_3–O_4
	O_2	10	8	9	10	11	13	12	15	O_2–O_3–O_4–O_1
	O_3	10	10	10	11	12	12	13	10	O_2–O_4–O_3–O_1
	O_4	20	20	19	21	22	24	18	20	
3	O_1	7	7	6	8	10	9	9	8	O_2–O_3–O_1–O_4
	O_2	10	12	14	13	14	15	10	11	O_3–O_2–O_4–O_1
	O_3	30	31	35	29	32	35	33	32	O_3–O_4–O_2–O_1
	O_4	40	42	44	45	38	40	43	40	
4	O_1	57	60	62	61	63	68	58	59	O_1–O_2–O_3–O_4
	O_2	10	12	14	15	13	16	12	11	O_1–O_2–O_4–O_3
	O_3	30	28	32	31	29	30	27	33	
	O_4	20	18	19	21	23	22	20	18	
5	O_1	8	7	6	9	10	12	10	11	O_1–O_2–O_4–O_5–O_3
	O_2	12	11	11	12	10	13	9	10	O_1–O_3–O_4–O_5–O_2
	O_3	20	22	24	25	23	18	19	21	O_1–O_2–O_5–O_4–O_3
	O_4	18	20	21	22	19	25	24	23	O_1–O_2–O_3–O_5–O_4
	O_5	30	33	36	40	28	29	33	34	
6	O_1	9	6	7	8	10	12	11	8	O_5–O_1–O_3–O_2–O_4
	O_2	7	6	7	8	9	12	10	8	O_5–O_3–O_1–O_2–O_4
	O_3	10	12	11	9	10	13	14	8	O_5–O_1–O_2–O_4–O_3
	O_4	20	24	22	23	25	21	18	19	O_1–O_5–O_2–O_3–O_4
	O_5	40	44	42	38	36	37	39	41	
7	O_1	20	22	21	24	26	23	24	25	O_1–O_2–O_3–O_4–O_5
	O_2	10	18	16	12	11	13	15	9	O_2–O_3–O_1–O_4–O_5
	O_3	50	53	55	45	47	48	49	51	O_3–O_2–O_4–O_5–O_1
	O_4	30	32	33	34	29	31	28	35	O_4–O_5–O_3–O_2–O_1
	O_5	21	23	24	26	28	27	20	23	
8	O_1	70	73	79	80	65	67	69	68	O_1–O_2–O_3–O_4–O_5
	O_2	7	8	9	10	8	10	11	13	O_1–O_3–O_2–O_4–O_5
	O_3	16	18	15	14	12	13	20	14	O_3–O_1–O_2–O_4–O_5
	O_4	25	26	25	24	20	23	21	26	O_3–O_2–O_1–O_4–O_5
	O_5	11	12	13	14	16	10	12	15	O_2–O_1–O_3–O_4–O_5

4.3. Applications of the hybrid algorithm on multi-objective IPPS

The JSP had been proved to be a NP-hard problem. The IPPS problem is more complicated than the JSP problem. It is also a NP-hard problem. For the large scale problems, the conventional algorithms (including the exact algorithms) are often incapable of optimizing non-linear multi-modal functions in the reasonable time. To address this problem effectively, one modern optimization algorithm has been used to quickly find a near optimal solution in a large search space through some evolutional or heuristic strategies. In this research, the hybrid algorithm (HA) which is the hybrid of the GA and the TS has been applied to facilitate the search process. This algorithm has been successfully applied to solve the mono-objective IPPS problem (Li et al., 2010d). Here, this algorithm has been developed further to solve the multi-objective IPPS problem. The working steps of this algorithm are explained here for illustration.

The HA is used in this research to generate the new generations. Therefore, there is no fitness function to evaluate the population. The work flow of the HA is shown in Fig. 3. The basic procedure of the proposed algorithm is described as follows:

Step 1: Set the parameters of HA, including size of the population (*Popsize*), maximum generations (*maxGen*), reproduction probabilistic (p_r), crossover probabilistic (p_c), mutation probabilistic (p_m), parameters of the tabu search.
Step 2: Initialize population randomly, and set *Gen* = 1.
Step 3: Is the terminate criteria satisfied?
If yes, go to *Step 7*.
Else, go to *Step 4*.
Step 4: Generate a new generation by genetic operations.
Step 4.1: Selection: the random selection scheme has been used for selection operation.

Table 6
Experimental results of problem 2.

Criteria	Solution 1	Solution 2	Solution 3
Makespan	122	122	123
Maximal machine workload (*MMW*)	106	102	107
Total workload of machines (*TWM*)	751	784	750

Table 7
Selected operation sequence for each job of problem 2.

Job	Solution 1	Solution 2	Solution 3
1	O_1–O_2–O_3–O_4	O_1–O_2–O_3–O_4	O_1–O_2–O_3–O_4
2	O_2–O_3–O_4–O_1	O_2–O_3–O_4–O_1	O_2–O_1–O_3–O_4
3	O_2–O_3–O_1–O_4	O_2–O_3–O_1–O_4	O_2–O_3–O_1–O_4
4	O_1–O_2–O_3–O_4	O_1–O_2–O_3–O_4	O_1–O_2–O_4–O_3
5	O_1–O_3–O_4–O_5–O_2	O_1–O_3–O_4–O_5–O_2	O_1–O_2–O_4–O_5–O_3
6	O_5–O_1–O_2–O_4–O_3	O_5–O_1–O_2–O_4–O_3	O_5–O_1–O_3–O_2–O_4
7	O_4–O_5–O_3–O_2–O_1	O_4–O_5–O_3–O_2–O_1	O_1–O_2–O_3–O_4–O_5
8	O_2–O_1–O_3–O_4–O_5	O_2–O_1–O_3–O_4–O_5	O_3–O_2–O_1–O_4–O_5

Step 4.2: Reproduction: reproduce the *Popsize* $\times$ p_r individuals from the parent generation to the Offspring generation.
Step 4.3: Crossover: the crossover operation with a user-defined crossover probabilistic (p_c) is used for IPPS crossover operation.
Step 4.4: Mutation: the mutation operation with a user-defined mutation probabilistic (p_m) is used for IPPS mutation operation.
Step 5: Local search by TS for every individual.
Step 6: Set *Gen* = *Gen* + 1, and go to *Step 3*.
Step 7: Use the Nash equilibrium solutions algorithm to compare all the recorded Nash equilibrium solutions in every generation, and select the best solutions.
Step 8: Output the best solutions.

According to this algorithm, every individual evolves by the genetic operations firstly, and then it focuses on the local search. More details of the HA can refer to Li et al., (2010d).

5. Experimental results

In this paper, the proposed algorithm 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 algorithm in this paper, two instances have been selected. The first instance is adopted from the other paper. Because of the lack of benchmark instances on the multi-objective IPPS problem, we presents the second instance. The parameters of the proposed

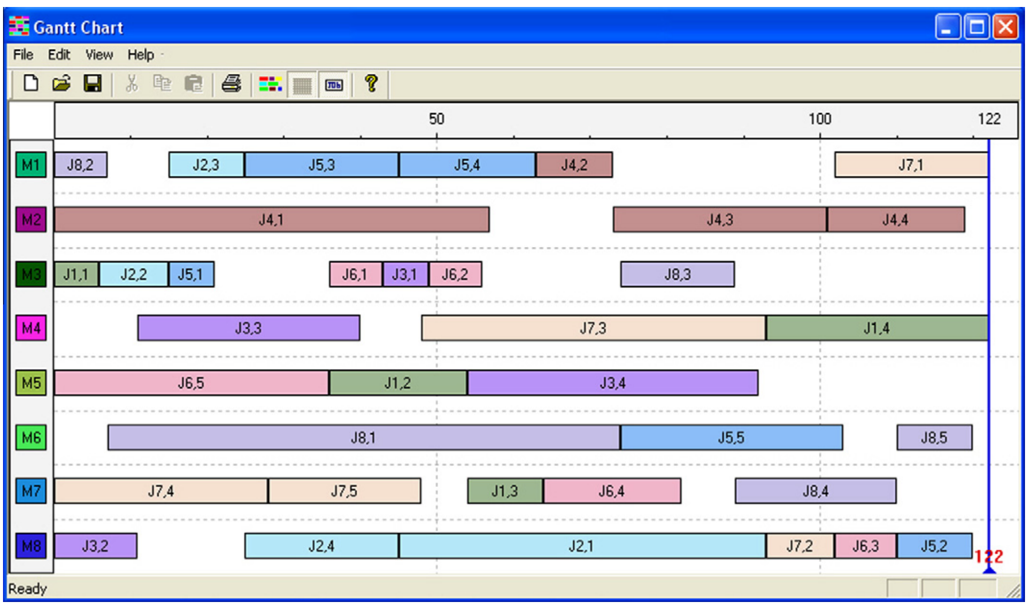

Fig. 6. Gantt chart of the solution 1 of problem 2.

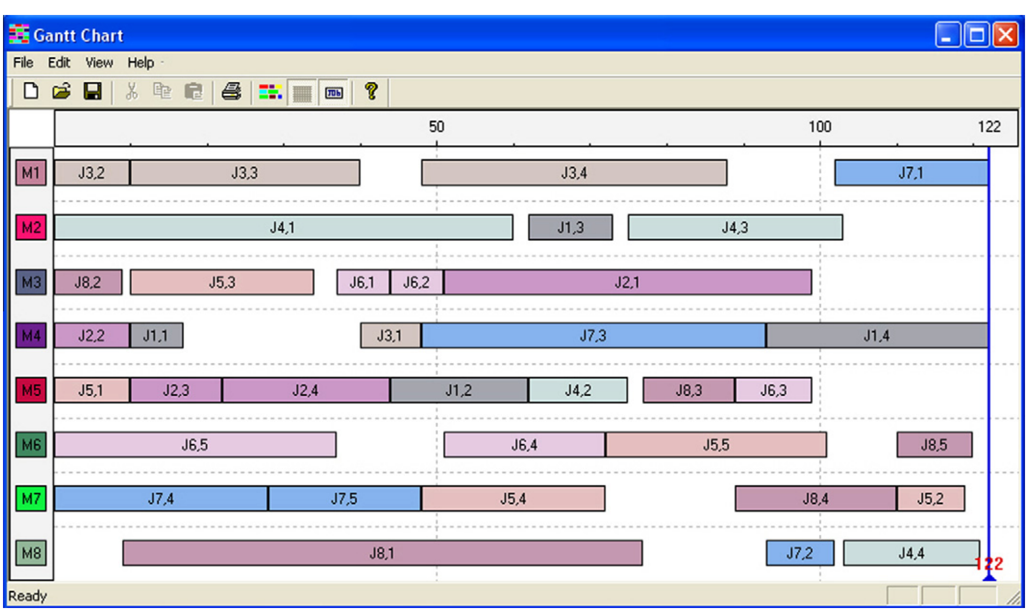

Fig. 7. Gantt chart of the solution 2 of problem 2.

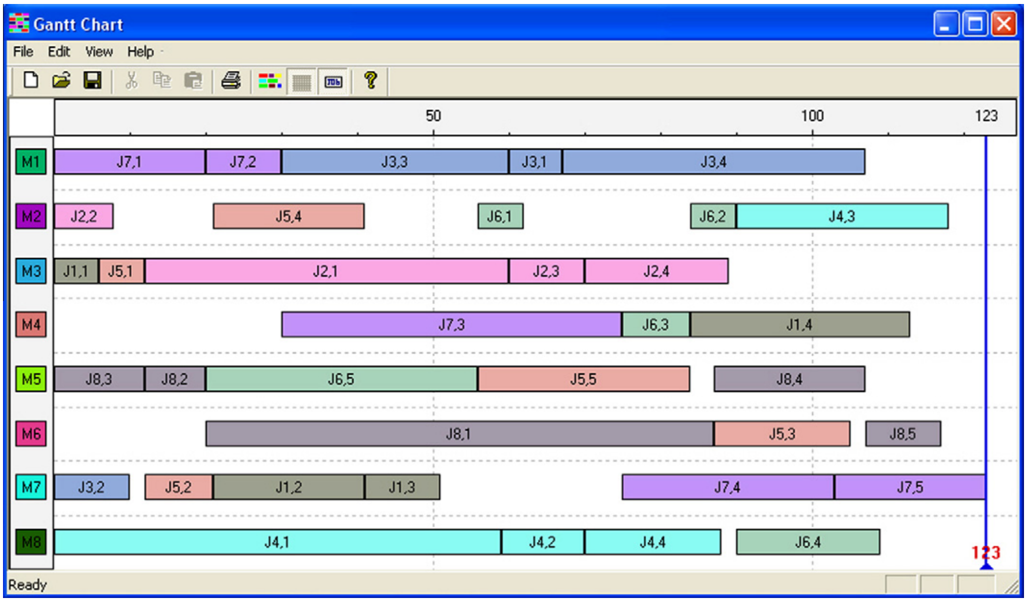

Fig. 8. Gantt chart of the solution 3 of problem 2.

algorithm for these problem instances are given in Table 1. The proposed algorithm 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* was the current generation of GA) or the permitted maximum step size with no improving (*maxStagnantStep*).

5.1. Problem 1

Problem 1 is adopted from Baykasoglu and Ozbakir (2009). It is constructed with 5 jobs and 5 machines. The data is shown in Table 2. Table 3 shows the experimental results and the comparisons with the other algorithm. Table 4 shows the selected operation sequence for each job. Fig. 4 illustrates the Gantt chart of the solution 1 of the proposed algorithm. Fig. 5 illustrates the Gantt chart of the solution 2.

From the experimental results of problem 1 (Table 3), the results of the proposed algorithm can dominate the results of the grammatical approach. They are better than the results of the other algorithm. This means that the proposed approach is more effective to obtain the good solutions of the multi-objective IPPS problem.

5.2. Problem 2

Owing to the lack of the benchmark instances on the multi-objective IPPS problem, we presents the problem 2. The data of problem 2 is shown in Table 5. It is constructed with 8 jobs and 8 machines. Table 6 shows the experimental results. Table 7 shows the selected operation sequence for each job. Fig. 6 illustrates the Gantt chart of the solution 1 of the proposed algorithm. Fig. 7 illustrates the Gantt chart of the solution 2, and Fig. 8 illustrates the Gantt chart of the solution 3.

From the experimental results of problem 2 (Table 6), the proposed approach can obtain the good solutions of the multi-objective IPPS problem effectively.

6. Conclusions

Considering the complementarity of process planning and scheduling, and the multiple objectives requirement from the real-world production, the research has been conducted to develop a game theory based hybrid algorithm to facilitate the multi-objective IPPS problem. In this proposed approach, the Nash equilibrium in game theory has been used to deal with the multiple objectives. And a HA has been used to optimize the IPPS problem. Experimental studies have been used to test the performance of the proposed approach. The results show that the developed approach has achieved satisfactory improvement. The contributions of this research include:

- The game theory has been used to deal with the multiple objectives of the IPPS problem. This is a new idea on the multi-objective manufacturing problems, and it also can be used to deal with the multiple objectives of other problems in the manufacturing field, such as process planning problem, assembly sequencing problem, scheduling problems and so on.
- To find optimal or near optimal solutions from the vase search space efficiently, a HA has been applied to the multi-objective IPPS problem. Experiments have been conducted and the results have shown the effectiveness of applying this approach.

Acknowledgements

The authors would like to thank the editor and anonymous referees for their valuable comments. This research work is supported by the Natural Science Foundation of China (NSFC) under Grant No. 51005088 and 50825503, the National Basic Research Program of China (973 Program) under Grant No. 2011CB706804 and the Program for New Century Excellent Talents in University under Grant No. NCET-08-0232.

References

Baykasoglu, A., & Ozbakir, L. (2009). A grammatical optimization approach for integrated process planning and scheduling. *Journal of Intelligent Manufacturing, 20*, 211–221.

Beckendorff, U., Kreutzfeldt, J., & Ullmann, W. (1991). Reactive workshop scheduling based on alternative routings. In *Proceedings of a conference on factory automation and information management* (pp. 875–885).

Chan, F. T. S., Kumar, V., & Tiwari, M. K. (2009). The relevance of outsourcing and leagile strategies in performance optimization of an integrated process

planning and scheduling model. *International Journal of Production Research, 47*(1), 119–142.
Chryssolouris, G., & Chan, S. (1985). An integrated approach to process planning and scheduling. *Annals of the CIRP, 34*(1), 413–417.
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.
Guo, Y. W., Li, W. D., Mileham, A. R., & Owen, G. W. (2009a). Optimisation of integrated process planning and scheduling using a particle swarm optimization approach. *International Journal of Production Research, 47*(14), 3775–3796.
Guo, Y. W., Li, W. D., Mileham, A. R., & Owen, G. W. (2009b). Applications of particle swarm optimisation in integrated process planning and scheduling. *Robotics and Computer-Integrated Manufacturing, 25*(2), 280–288.
Hsu, T., Dupas, R., Jolly, D., & Goncalves, G. (2002). Evaluation of mutation heuristics for the solving of multiobjective flexible job shop by an evolutionary algorithm. In *Proceedings of the 2002 IEEE international conference on systems, man and cybernetics* (Vol. 5, pp. 655–660).
Khoshnevis, B., & Chen, Q. M. (1989). Integration of process planning and scheduling function. In *Proceedings of IIE integrated systems conference & society for integrated manufacturing conference* (pp. 415–420).
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.
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.
Kuhnle, H., Braun, H. J., & Buhring, J. (1994). Integration of CAPP and PPC – interfusion manufacturing management. *Integrated Manufacturing Systems, 5*(2), 21–27.
Kumar, M., & Rajotia, S. (2003). Integration of scheduling with computer aided process planning. *Journal of Materials Processing Technology, 138*, 297–300.
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, 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.
Li, W. D., Gao, L., Li, X. Y., & Guo, Y. (2008). Game theory-based cooperation of process planning and scheduling. In *Proceeding of the 12th international conference on computer supported cooperative work in design, China* (pp. 841–845).
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.
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, 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., Shao, X. Y., Gao, L., & Qian, W. R. (2010d). An effective hybrid algorithm for integrated process planning and scheduling. *International Journal of Production Economics, 126*, 289–298.
Li, X. Y., Zhang, C. Y., Gao, L., Li, W. D., & Shao, X. Y. (2010c). An agent-based approach for integrated process planning and scheduling. *Expert Systems with Applications, 37*, 1256–1264.
Morad, N., & Zalzala, A. M. S. (1999). Genetic algorithms in integrated process planning and scheduling. *Journal of Intelligent Manufacturing, 10*, 169–179.
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.
Shen, W. M., Wang, L. H., & Hao, Q. (2006). 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, 36*(4), 563–577.
Shukla, S. K., Tiwari, M. K., & Son, Y. J. (2008). Bidding-based multi-agent system for integrated process planning and scheduling: A data-mining and hybrid Tabu-SA algorithm-oriented approach. *International Journal of Advanced Manufacturing Technology, 38*, 163–175.
Sugimura, N., Hino, R., & Moriwaki, T. (2001). 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, 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.
Wang, L. H., Song, Y. J., & Shen, W. M. (2005). Development of a function block designer for collaborative process planning. In *Proceeding of 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.
Wong, T. N., Leung, C. W., Mak, K. L., & Fung, R. Y. K. (2006). Integrated process planning and scheduling/rescheduling – An agent-based approach. *International Journal of Production Research, 44*(18–19), 3627–3655.
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. (2003). A Holonic architecture of the concurrent integrated process planning system. *Journal of Materials Processing Technology, 139*, 267–272.
Zhang, W. Q., & Gen, M. (2010). Process planning and scheduling in distributed manufacturing system using multiobjective genetic algorithm. *IEEJ Transactions on Electrical and Electronic Engineering, 5*, 62–72.