9

Achieving Cost Competitiveness with an Agent-Based Integrated Process Planning and Production Scheduling System

Ming Lim[1] and David Zhang[2]
[1]*Aston University,*
[2]*University of Exeter,*
United Kingdom

1. Introduction

As globalisation takes place, the market is getting more and more competitive. Manufacturing enterprises are facing tremendous pressure to succeed in the market with promising market shares. This has led to enterprises seeking for competitive advantage in order to vie with their rivals. In manufacturing context, efforts have been put in, for instance, to reduce production lead times, maximise productivity and optimise resource utilisation. These efforts aim to reduce all types of costs incurred, as a means to achieve cost competitiveness. There is also a challenge for the manufacturers to be competent to efficiently and cost-effectively cope with dynamic changes of customer demands in the market. These demands are related to a wide range of product mixes with short product lifespan and with unpredicted demand pattern (Zhang, 2011).

From a manufacturing perspective, the efficiency of manufacturing operations (such as process planning and production scheduling) are the key element for enhancing manufacturing competence. Process planning and production scheduling functions have been traditionally treated as two separate activities, and have resulted in a range of inefficiencies. These include infeasible process plans, non-available/overloaded resources, high production costs, long production lead times, and so on (Saygin & Kilic, 1999; Khoshnevis & Chen, 1993; Zhang, 1993). Above all, it is unlikely that the dynamic changes can be efficiently dealt with. Despite much research has been conducted to integrate process planning and production scheduling to improve manufacturing efficiency, there is still a gap to achieve the competence required for the current global competitive market.

In this research, the concept of multi-agent system (MAS) is adopted as a means to address the aforementioned gap. A MAS consists of a collection of intelligent autonomous agents able to solve complex problems. These agents possess their individual objectives and interact with each other to fulfil a global goal. This chapter describes a novel use of an autonomous agent system to facilitate the integration of process planning and production scheduling functions to cope with unpredictable demands. This refers to the uncertainties in product mixes and demand patterns. The novelty lies with the currency-based iterative agent bidding mechanism to allow process planning options and production scheduling options to be evaluated simultaneously, so as to search for an optimised and cost-effective solution. This agent based system aims to achieve manufacturing competence by means of enhancing the flexibility and agility of manufacturing enterprises.

This chapter is organised as follows. Section 2 reviews the literature of the existing approaches to integrated process planning and production scheduling. The limitations of these approaches will also be discussed. Section 3 describes the concept of MAS and Section 4 introduces the currency-based iterative agent bidding mechanism proposed in this study. A Tabu search optimisation technique to facilitate the adjustment of current values for agent bidding is presented in Section 5. Section 6 discusses the findings of the simulation results for the iterative bidding mechanism and further analyses the bidding results with three heuristic integrated process planning and scheduling approaches. Section 7 concludes this chapter.

2. Integrated process planning and production scheduling

In the literature, there is a vast number of research works reported contributing to the integration of process planning and production scheduling. These include non-linear process planning, flexible process planning, closed-loop process planning, dynamic process planning, alternative process planning, and just-in-time process planning (Cho & Lazaro, 2010; Kim et al., 2009; Moslehi et al. 2009; Omar & Teo, 2007; Wang et al., 2003; Saygin & Kilic, 1999; Usher & Fernandes, 1996; Khoshnevis & Mei, 1993). According to Larsen & Alting (1990), these works can be classified into three broad categories: non-linear process planning (NLPP), closed-loop process planning (CLPP), and distributed process planning (DTPP).

2.1 Non-Linear Process Planning (NLPP)

NLPP entails a planning system that generates a list of possible alternative plans for each part prior to actual production on the shop floor. This means that NLPP is based on a static shop floor condition. All these possible plans are ranked according to process planning criteria. The first priority plan is always used when the job is required. If the plan is not suitable, e.g. due to resource unavailability, the lower priority plan will be chosen. This procedure is repeated until a suitable plan is found. Examples of such system include FLEXPLAN that uses reactive re-planning strategies to allow fast reaction when unexpected events occur on the shop floor during the execution of a schedule (Tonshoff et al., 1989), and a framework by Hou & Wang (1991) that firstly disaggregating the process planning problems and followed by generating alternative process plans for the parts to be manufactured. Other similar works include Ho & Moodie (1996), Hutchinson & Pflughoeft (1994), and Srihari & Greene (1988).

Recognising the weaknesses of NLPP, some researchers proposed the idea of a two-stage approach to improve NLPP. In the first stage, all possible alternative process plans that do not take into account of operational status of the shop floor resources are generated. The second stage is dynamic process planning whereby the generated process plans are evaluated by taking into account of the availability of the shop floor resources and the objectives or rules are specified by the scheduler. The result of this two-stage approach is a set of ranked near-optimum alternative plans and schedules. The systems applying such an approach are PARIS (Usher & Fernandes, 1996), DYNACAPP (Ssemakula & Wesley, 1994), and THCAPP-G (Wang et al., 1995).

2.2 Closed-Loop Process Planning (CLPP)

NLPP offers flexibility to the scheduling department with a list of alternative process plans. However, process planners do not take into account of the shop floor condition and an arbitrary set is generated based on their experience. In turn, production schedulers only use the alternatives that are available. To make the process planning more efficient, there is a need to have feedback from the shop floor with detailed information of the shop floor condition as well as requirements from scheduling department. With this information, no further effort will be spent on investigating alternatives that are of no use. Furthermore, the risk of overlooking important aspects (e.g. machine reliability and utilisation, bottlenecks) is also reduced. CLPP is an approach that could provide such feedback.

CLPP generates plans for jobs in real time based on the status of the resources at that time. Production schedulers provide process planners with information in relation to resource availability so that every plan is feasible with respect to the current availability of production facilities. Real time status has become a crucial element for CLPP and therefore, CLPP is also referred to as real time process planning or dynamic process planning. The research works based on this approach include a heuristic algorithm proposed by Khoshnevis & Chen (1989) developing a dynamic list of available machines and a list of features for each part. When a match is found between the two lists, the part will be assigned to that machine. However, the algorithm has neglected one issue in relation to the allocation of producing the features to machines. For instance, the algorithm may have allocated a feature to a less desirable machine at a given instant, whereas had it waited for a short while, a more desirable machine might have become available. The authors then introduced the concept of time window into their improved algorithm to deal with this problem (Khoshnevis & Chen, 1990). Although the improved algorithm can yield better results, the computational complexity is increased. In a later work by Chen & Khoshnevis (1992), the integration problem is viewed as a scheduling problem with flexible process plans. The priority is given to the scheduling module. Whenever an assignment of an operation to a machine is made by the scheduling module, the process planning module is invoked to check the validity of the assignment. Other examples of using CLPP are Kiritis & Porchet (1996) and Iwata & Fukuda (1989).

In NLPP, feedback information from the shop floor (i.e. information on the shop floor condition and requirements from scheduling department) is provided to the process planning department and as a result, process planning can be performed more efficiently and infeasible plans (i.e. due to unavailability of resources) can be eliminated. However, the aforementioned manufacturing competence is still not yet achieved in the proposed works. Despite the elimination of infeasible plans, the cost reduction through optimisation of utilisation of resources and minimisation of bottlenecks are not achieved in NLPP.

2.3 Distributed Process Planning (DTPP)

DTPP is a promising approach that performs both process planning and production scheduling simultaneously in a distributed manner, starting from a global level (i.e. pre-planning) and ending at a detailed level (final planning). In DTPP process planning and production scheduling activities are carried out in parallel and in two phases. The first phase is pre-planning whereby process planning function analyses the jobs/operations to be carried out. The features and feature relationships are recognised and the corresponding manufacturing processes are determined. The required machine capabilities are also estimated. The second phase is final planning, which matches the required operations with the operational capabilities of the available manufacturing resources. The integration occurs at the point when resources are available and the operation is required. In this integration, process planning and production scheduling are carried out simultaneously. This approach is sometimes also referred to as just-in-time process planning. The result of this approach is dynamic process and production scheduling constrained by real-time events. Such approach includes the early works by Mamalis et al. (1996), Zhang (1993), Mallur et al. (1992), and more recently by Li et al. (2010), Moon et al. (2009), and Wang et al. (2009).

Despite the effort to integrate process planning and scheduling to find satisfactory solutions, further work is required for the solutions to be optimised in respond to dynamic changes in order to enhance the agility of manufacturing systems. To achieve overall optimality, rescheduling alone may not be effective (e.g. Wang et al., 2011). Process planning options should be taken into consideration to provide flexibility and optional scenarios in using alternative resources to respond constantly to dynamic changes. This means that process planning options and production scheduling options should be integrated and optimised dynamically, so that constraints from both functions can be fulfilled simultaneously and a near-optimum integrated plan and schedule can then be produced. Furthermore, the integration of process planning and production scheduling should also be able to provide scenarios where the production operational structures and possible reconfiguration of manufacturing systems can be assessed. By enhancing this manufacturing competence, the cost competitiveness will be achieved. In this research, a multi-agent system (MAS) is employed aiming to achieve this.

3. Multi-Agent System (MAS)

MAS is a popular research technique applied in various disciplines. A MAS is a distributed intelligent system consisting of a population of agents that pursue individual objectives and interact closely with each other to achieve a global goal. Each agent represents an entity (e.g. a machine or a job) and is endowed with a certain degree of autonomy and intelligence, which includes the ability to perceive its environment and to make decisions based on its knowledge (Ferber, 1999).

In a MAS, a complex system is decomposed into autonomous and loosely-coupled subsystems represented by agents (Wooldridge, 1997). The term autonomous refers to the independency of control between the agents. Each agent determines its course of actions and other agents may influence an agent’s decision by means of coordination (through collaboration or competition/negotiation). The term loosely coupled refers to the dependency on information between the agents. This dependency may exist for some tasks and shall not oblige to overload one agent’s capability. Agents that represent the subsystems are able to solve problems in their domain with their own thread of control and execution. They carry out tasks autonomously without depending on other agents. The agent characteristics of intelligence and autonomous decision-making architecture have attracted many researchers in manufacturing domain solving complex manufacturing problems, including research related to process planning and production scheduling.

In general, the agent-based process planning and production scheduling approaches found in the literature can be grouped into two categorises based on the interaction mechanism used by the agents. They are bidding based methods and non-bidding based methods. Bidding based methods include the works by Robu et al. (2011), Kumar et al. (2008), Liu et al. (2007), Lima et al. (2006), Wong et al. (2006), and non-bidding works by Hajizadeh et al. (2011), Blum & Sampels (2004), Caridi & Sianesi (2000), and Ottaway & Burns (2000).

For any MAS, the design of agents is crucial to ensure the global goal and individual objectives are both fulfilled. This includes the agent functions and network, agent interaction mechanism and its protocols for coordination. In this chapter, the authors proposed a novel use of an autonomous agent system to facilitate the integration of process planning and production scheduling functions in order to maximise the manufacturing competence to cope with unpredictable demands. The novelty lies with the currency-based iterative agent bidding mechanism to allow process planning options and production scheduling options to be evaluated simultaneously, so as to search for an optimised, cost-effective solution. This agent based system aims to provide the flexibility and agility of manufacturing enterprises required to cope with the uncertainties in the market. The following section discusses this iterative agent bidding mechanism in detail.

4. Iterative agent bidding mechanism

In the proposed iterative agent bidding mechanism, a currency-like metric is used whereby each operation to be performed is assigned with a virtual currency value. These operations will then be announced to the agents (e.g. representing resources on the shop floor) and they will bid for the operations based on the currency values. These currency values are used as a parameter to control the bidding process between agents. Agents will only put forward the bids for the operation if they make a virtual profit (i.e. the difference between the given currency for the operation and the cost of performing the operation) that is above a virtual profit threshold set by that agent. This means that these parameters have a direct influence over the decisions of agent bidding for operations and forwarding the bids; therefore the adjustment of parameters will result in different bids constructed. In this mechanism, the virtual currency values will be adjusted iteratively, and so does the bidding process between agents based on the new set of currency values generated. This is to search for better and better bids, leading to near-optimality. The iterative bidding mechanism aims to achieve the lowest possible total production cost while satisfying the delivery due dates. Moreover, with the adjustment of currency values it is able to drive the behaviour of agents in a way that agents become proactive if they know they can perform the job with greater amount of virtual profit earned and vice versa.

The iterative bidding mechanism is illustrated in Figure 1. Assume that machine agents representing the machines on the production shop floor and a job agent representing a job (e.g. to produce a component) to be performed which can be broken down into a number of operations (e.g. to produce the features of the component). The iterative bidding mechanism takes place between the job agent and machine agents. As depicted in Figure 1, the bidding process begins when the job agent announces the job to be performed to all machine agents to bid (*Step 1*). The announcement includes information related to the machining operations to be carried out, such as the number and type of machining operations, recommended type of machining processes for the operations, etc., and the virtual currency value assigned to each operation. Machine agents that are able to perform the first operation will come forward to become 'leaders' whose responsibility is to group other machine agents to perform the remaining operations (*Steps 2-3*). The number of leaders indicates the number of virtual machine groups.

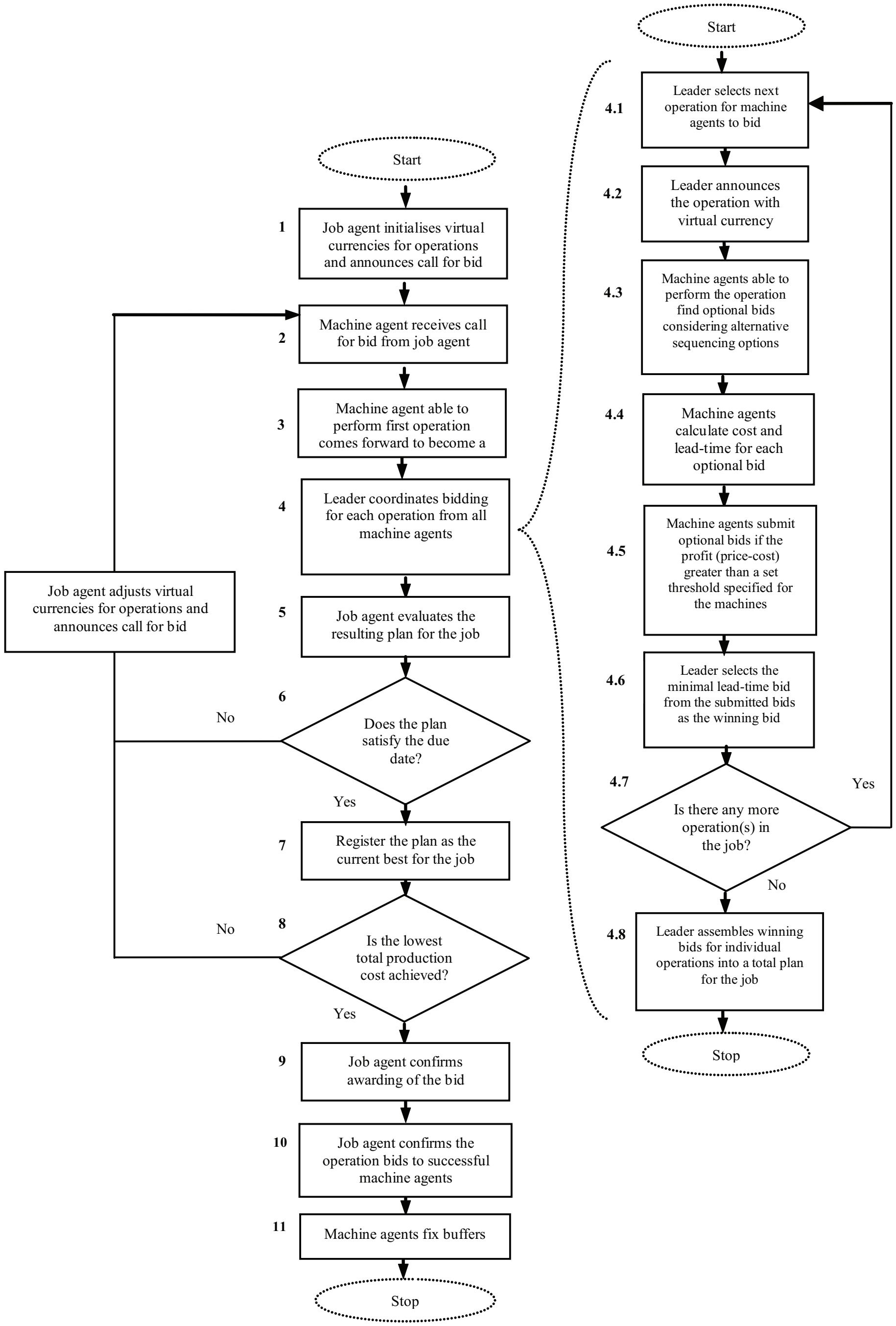

Fig. 1. Iterative agent bidding mechanism

After the leaders are selected, they announce the second operation to all machine agents, including the leaders themselves (*Step 4*). Machine agents that are able to carry out the operation will come forward to bid for the job. They may reschedule and optimise their job buffer by shifting jobs if other operations' due dates are not violated. This aims to produce optional and better bids *BOi,x,k* (*BOi,x,k* denotes the kth bid option from machine *rx* for operation *Oi*). In this way, bottlenecks can be reduced and machine utilisation can also be optimised. By shifting jobs in the job buffer, some bids may eliminate tool change and setup activities and as a result, the time needed to carry out the operation could be reduced. However, extra cost might be involved due to the job shifting in the job buffer, e.g. holding for work-in-process. Machine agents work out their bids in terms of production cost and lead time. The individual machine production cost is obtained as:

$$C_i = C_{ti} + C_{wi} + C_{si} + C_{pi} + C_{ri} \tag{1}$$

where

$$C_{ti} = C_{ti/d}(D) \tag{2}$$

$$C_{pi} = C_{pi/t}\left(\frac{V_{removed}}{MRR}\right) \tag{3}$$

where

C_{ti} = transportation cost from the location of preceding machine (unit of cost),

$C_{ti/d}$ = transportation cost / unit of distance (unit of cost),

D = distance from the location of preceding machine (m),

C_{wi} = holding cost (unit of cost),

C_{si} = setup cost (unit of cost),

$V_{removed}$ = volume to be removed in order to produce the feature (mm3),

MRR = material removal rate (mm3 / unit of time),

C_{pi} = processing cost (unit of cost),

$C_{pi/t}$ = processing cost / unit of time (unit of cost),

C_{ri} = rescheduling cost (unit of cost).

The machine production cost function used in this study does not, however, truly reflect the actual production cost in real production. The cost function is developed for evaluation purposes and the costs such as material cost and labour cost are disregarded.

The individual lead time is worked out as:

$$T_i = T_{ti} + T_{wi} + T_{si} + T_{pi} \tag{4}$$

where

$$T_{ti} = T_{ti/d}(D) \tag{5}$$

$$T_{wi} = \sum_{j=1}^{n} t_{wi[j]} \tag{6}$$

$$T_{pi} = \frac{V_{removed}}{MRR} \tag{7}$$

where

T_{ti} = transportation lead time from the preceding machine (unit of time),

D = distance from the location of preceding machine (m),

$T_{ti/d}$ = transportation lead time / unit of distance (unit of cost),

T_{wi} = waiting time at buffer, i.e. queuing time or bottlenecks (unit of time),

$\sum_{j=1}^{n} t_{wi[j]}$ = total waiting time of n jobs scheduled in the job buffer before the currently bidding job (unit of time),

T_{si} = setup time (unit of time),

T_{pi} = processing lead time (unit of time)

$V_{removed}$ = volume to be removed in order to produce the feature (mm^3), and

MRR = material removal rate (mm^3 / unit of time).

Each machine agent decides whether to forward a bid based on the amount of virtual profit earned:

$$P_{i,x,k} = CU_i - C_{i,x,k} \tag{8}$$

where $P_{i,x,k}$ is the virtual profit that could be made by machine r_x on operation O_i with bid option $BO_{i,x,k}$, CU_i is currency value assigned to operation O_i , and $C_{i,x,k}$ is the production cost for r_x to carry out O_i with bid $BO_{i,x,k}$ as defined by Eq. 1. If $P_{i,x,k}$ is above a set threshold P_{tx} (i.e. $P_{i,x,k} \geq P_{tx}$), the bid will be put forward to the leader. P_{tx} is a mark-up profit that is based on the production cost $C_{i,x,k}$ i.e. $P_{tx} = C_{i,x,k} + C_{i,x,k} \cdot M_{i,x,k}$., where $M_{i,x,k}$. is a random value in the range [0,N], and N is a limiting percentage value. By shifting jobs in the job buffer, a machine agent may put forward more than one bid as long as the virtual profits of the bids are above the set threshold. The threshold varies from one machine to another based on the cost of machine. However, if the profit is below the set threshold ($P_{i,x,k} < P_{tx}$), the machine agent will not forward the bid to the leader. In mathematical terms:

$$B_{i,l} = BO_{i,x,k},\ T_i^{(l)} = T_{i,x,k},\ \ C_i^{(l)} = C_{i,x,k} \quad if\ P_{i,x,k} \geq P_{tx} \tag{9}$$

where $B_{i,l}$ denotes the l^{th} bid submitted for operation O_i, $T_{i,x,k}$ is the lead time for r_x to carry out O_i with bid option $BO_{i,x,k}$, $T_i^{(l)}$ and $C_i^{(l)}$ are the lead time and cost for carrying out O_i with bid $B_{i,l}$. When the bids are received, the leader selects the best bid that provides the shortest lead time from all bids put forward by machine agents:

$$B_i^{win} = B_{i,l}, \ \ T_i^{win} = T_i^{(l)}, \ C_i^{win} = C_i^{(l)} \\ if \ \ T_i^{(l)}=\min\left(T_i^{(1)}, T_i^{(2)}, ..., T_i^{(L)}\right) \tag{10}$$

where B_i^{win} represents the winning bid for O_i, T_i^{win} and C_i^{win} are the lead time and cost corresponding to the winning bid, L is the total number of bids submitted for O_i.

The bid messages can be used to reflect a variety of dynamic status information (e.g. machine status, order condition), and therefore making the bidding mechanism suitable for real-time operational controls. This grouping process continues until all the operations in the job have been scheduled to the most appropriate machines. When the leaders have virtually grouped other machines to perform all operations *(O1, O2, …, On)*, they put together all the individual production costs (i.e. total production cost) and lead times (i.e. total lead time) of the selected machines, and forward the one complete bid as a machine group to the job agent for evaluation (*Step 5*). This bid consists of the total lead time and total production cost denoted as follow:

$$T = \sum_{i=1}^{n} T_i^{win}, \ \ \mathrm{C}=\sum_{i=1}^{n} C_i^{win} \tag{11}$$

The job agent evaluates the bids by means of satisfying the due date D at minimal total production cost

$$Min\left(C = \sum_{i=1}^{n} C_i\right) \\ T = \sum_{i=1}^{n} T_i \le D \tag{12}$$

If the due date is not satisfied (i.e. $T > D$), the virtual currency allocated to operations will be adjusted in the next iteration to look for a better plan (*Steps 6-8*). The lead time and cost of a plan resulting from a bidding iteration are dependent on the virtual currencies. Higher virtual currencies for operations increase the attractiveness of the operations to machine agents and encourage the agents to submit more bids for the operations (even though some bids may bear higher costs) and vice-versa. The iterative loop stops when a near-optimum plan that satisfies the due date with considered near-minimum cost is found. When the near-optimum plan is obtained, the job agent will award the job to the machine group that meets the due date and provides the minimum total production cost. The machine agents in the awarded machine group will then commit to the operations awarded by updating their loading schedules (*Steps 9-11*). If the product orders are large and consistent, there could be a need to group the machines in this virtual machine group physically (i.e. reconfiguring the layout of the existing manufacturing system), which may improve the system, as well as cost efficiency. In this way, the reconfiguration of manufacturing systems can be assessed.

Each agent has individual objectives and a global goal to achieve. For this proposed MAS, the global goal is to find an optimised process plan and schedule that gives the lowest production cost while satisfying all requirements such as due date and product quality. As for individual objectives, the machine agents strive to give the best performance in order to win the jobs and optimise its machine utilisation, and the job agent is responsible for assigning the operations to the outstanding group of machines. Via the iterative bidding mechanism and bid evaluation, agents with different objectives will come to a point where the agents' objectives and the global goal can be satisfied.

A Tabu search optimisation technique is employed in this study to investigate how and to what degree the currency values should be adjusted in each iteration in order to obtain better solutions (leading to near-optimality) for integrated process planning and scheduling problems. This technique will be discussed in the following section.

5. Tabu search optimisation technique

The basic form of TS approach was founded on the ideas proposed by Glover (1989, 1990). This approach is based on the procedures designed to cross boundaries of feasibility or local optimality, which are usually treated as barriers. In other words, TS is a meta-heuristic that guides a local heuristic search procedure to explore the solution space beyond local optimality. The key parameters used in TS are as follows:

- Tabu move – a move that is forbidden because it has previously been taken in the search process.
- Aspiration criterion – a criterion to remove the Tabu move that considered to be sufficiently attractive leading to a better solution.
- Intensification strategy (a.k.a. short-term memory) – a rule that encourages moves surrounding the solution that previously found good.
- Diversification strategy (long-term memory) – a rule that encourages the search process to examine unvisited regions and to generate solutions that are difference from those visited before.

With the simplicity of applying the concept of TS, many researchers have adopted TS in production research for optimisation purposes. Baykasoglu & Ozbakir (2009) proposed a multiple objective TS framework to generate flexible job shop scheduling problems with alternative process plans in order to analyse its performance and efficiency. Demir et al. (2011) proposed a TS approach to optimise production buffer allocation in order to enhance manufacturing efficiency. Baykasoglu & Gocken (2010) used TS to solve fuzzy multi-objective aggregate production planning system. Xu et al. (2010) used a two-layer TS approach to schedule jobs with controllable processing times on a single machine in order to meet costumer due dates.

The TS approach proposed in this study is described in Figure 2. With an illustration of a component that has five features, the approach started off with initialising all the relevant parameters such as initial solution (i.e. a set of currency values), Tabu list size for intensification and diversification strategies, and stopping criteria. This approach consists of two main operators or moves, i.e. intensification (currency values adjustment) and diversification (pairwise exchange). For intensification, every currency value has an equal opportunity to be selected for currency adjustment. If a move j is tabu-active (i.e. Tabu [j] $\neq$ 0), it is not supposed to be chosen again. However, an aspiration criterion can be applied in the case if the tabu-active move j creates a better solution (i.e. lower cost) than the overall best solution found so far. At each move, the solutions generated will be evaluated and compared to the overall best solution and subsequently the overall best solution will be updated if the new solution found outperforms the overall best solution. Eventually, after the pre-determined number of intensifications to be carried out (*M*) or the number of moves when no consecutive improvement was found (*K*) is reached, diversification will take place to explore new regions (i.e. pairwise exchange of currency in the initial solution X_0). This process continues until all the regions are explored. The simulation results of this approach will be discussed in next section.

Step 1: *Initialise the following parameters*

- Initial solution, X_0 (CU_1 CU_2 CU_3 CU_4 CU_5) and objective function, $F(x_0)$.
- Best solution, $X^* = X_0$ and best objective function, $F^* = F(x_0)$.
- Tabu list size *t* for Intensification tabu list.
- Number of intensifications to be carried out, *M*.
- Counter for diversification, *n*.
- Diversification size that depends on the array size (stopping criterion), *N*.
- Counter for no consecutive improvement was made, *k*.
- Size of no consecutive improvement was made, *K*.

Intensification Tabu List (currency values alteration)

(short-term memory)

j	1	2	3	4	5

A currency will be randomly selected to be decreased/increased by α%. Once a currency is selected, it will be marked with *t* (i.e. Tabu-active for *t* iterations). Tabu [*j*] = *t*, where *t* is a scalar, representing Tabu list size. The value of *t* is reduced by 1 at every iteration.

Diversification Tabu List (pairwise exchange)

(Long-term memory)

a \ b	2	3	4	5
1				
2				
3				
4				

Tabu [*a*][*b*] (for *a* < *b*) stores numerical value in the a^{th} row and b^{th} column of the array. When *a* exchanges with *b*, Tabu [*a*][*b*] will be marked. The pairwise exchange is in random order. This move is in the tabu list, which means no such move is permitted in the whole searching process. This is used to diversify the search into new regions. The search process will terminate when all Tabu [*a*][*b*] have been marked.

Step 2: Iteration *i* = 0,1,2,…*M*. X_i denotes the current solution.

Fig. 2. Tabu search approach for iterative agent bidding mechanism

6. Test case and results analysis

The proposed MAS was implemented on a Java platform; a test case was used to simulate the effectiveness of the currency-based iterative bidding mechanism. In the test case, 10 machines (4 lathe machines, 3 milling machines and 3 drilling machines) were operating on the shop floor, and each machine has different capacity and capability. The machines data is depicted in Table 1. These machines are served by automated guided vehicles (AGVs) and each machine has its own buffer of jobs (of components C1, C2, C3 and C4) that have been previously scheduled (Table 2).

Machine	*Process*	*Co-ordinate X (location)*	*Co-ordinate Y (location)*	**Reliability**	**Setup time**	**Setup Cost**	**Misc Cost**	**Machining Cost**	**Holding Cost**	**MRR**
Lathe L1	Turning Drilling	40	0	0.8	25	2.5	160	1.6	0.25	36
Lathe L2	Turning Drilling	80	0	0.75	30	2.6	160	2.0	0.25	40
Lathe L3	Turning Drilling	120	0	0.65	35	2.9	170	2.1	0.20	30
Lathe L4	Turning Drilling	160	0	0.9	28	2.6	160	1.5	0.25	29
Mill M1	Milling	100	40	0.95	20	2.4	180	1.2	0.25	32
Mill M2	Milling	100	80	0.75	22	2.6	180	1.4	0.25	32
Mill M3	Milling	100	120	0.85	24	2.5	200	1.5	1.5	36
Drill D1	Drilling	0	40	0.75	32	3	200	1.8	0.4	37
Drill D2	Drilling	0	80	0.9	28	2.8	190	1.9	0.35	40
Drill D3	Drilling	0	120	0.85	29	2.8	200	1.5	0.25	30

Table 1. Machines data

Lathe	Schedule	Milling	Schedule	Drilling	Schedule
L1	C4(2)*	M1	C3(2)	D1	C1(1)
	620-830^		500-792		0-240
			C3(3)		
L2	C1(2)		792-988	D2	C2(2)
	280-620				460-650
		M2	C2(1)		
L3	C3(1)		0-380	D3	C2(4)
	0-415		C4(3)		1200-1380
	C2(3)		1000-1179		
	750-925				
		M3	C4(1)		
	C1(4)		0-432		
L4	940-1265		C1(3)		
			670-878		

*Cx(y) means job sequence y of component x
^ in unit of time

Table 2. Machines job schedule

To evaluate the bidding mechanism, this test case consists of three components of which orders were placed at interval times. These components are ComA, ComB and ComC. Table 3 listed the process sequence of producing the features of the components, and the information related to the currency values, removal volumes, and tolerance requirements of each feature in the components.

The simulation process begins with the job agent announcing the jobs of producing ComA to all the machine agents. This process repeats for ComB and ComC. To discuss the implementation in details, the simulation process for ComC is predominantly discussed in this section. The assumptions made in the implementation are:

- A machine only performs one process at a time
- A component can be machined by the same machine more than once
- All the machines are accessible by AGVs
- The AGVs are considered to be always available
- Each machine has infinite capacity input and output buffers
- Auxiliary processes for surface treatment such as grinding and reaming are not considered
- Material and labour costs are disregarded
- Chip formation, cutting fluids, temperature rise, and tool wear due to cutting process are neglected.

During the simulation, two test runs are carried out with the value of α (for the adjustment of currency values in intensification process) set at 15% and 30% respectively. In both test runs, the number of moves for intensification process is set to 10 and the Tabu list is 1. In the first test run, the simulation completed at 110th moves and the near-optimum bid obtained has a production cost of 3224 units and a lead time of 1374 units. Figure 3 illustrates all the bids obtained at each TS move.

Com ID	Quantity	Due date (units of time)	Features to produce in sequence	Process required	Currency value	Removal volume (cm³)	Tolerance (+/- mm)
ComA	50	900	Hole (Blind, flat-bottomed)	Drilling	575	55	0.75
			Hollow Cylinder (Through)	Turning	770	140	1.25
			Slot	Milling	450	30	1.00
			Hole (Centre, blind, flat-bottomed)	Drilling	500	40	075
ComB	45	1000	Slot	Milling	780	180	1.25
			Hole (Centre, blind, flat-bottomed)	Drilling	460	30	1.00
			Slot	Milling	440	60	1.25
ComC	70	1400	Hollow Cylinder (Through)	Turning	850	150	1.25
			Hollow Cylinder (Through)	Turning	425	40	1.25
			Hollow Cylinder (Through)	Turning	600	70	1.25
			Slot	Milling	380	30	1.00
			Hole (Blind, flat-bottomed)	Drilling	600	50	0.75

Table 3. New components to produce

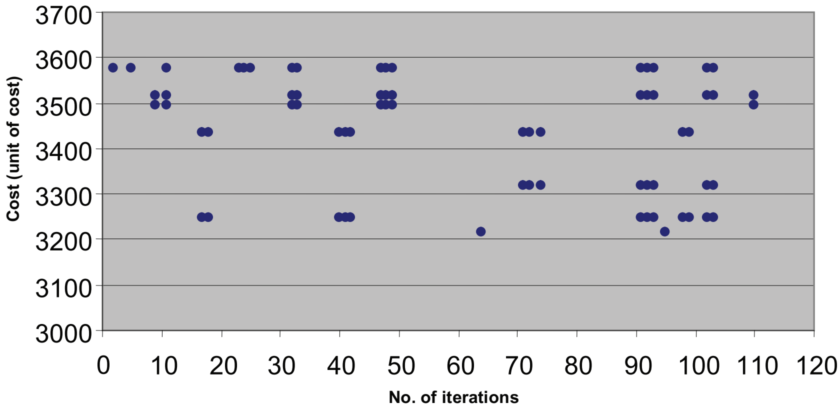

Fig. 3. Bids received at each TS move (α = 15%).

In Figure 4, the plotted line depicts the near-optimum bid recorded at each move during the entire simulation. This shows that lower costs of producing the components are gradually found as the currency values are adjusted iteratively. The first near-optimum bid was obtained at the 62nd move. The new job schedule for all machines is depicted in Table 4.

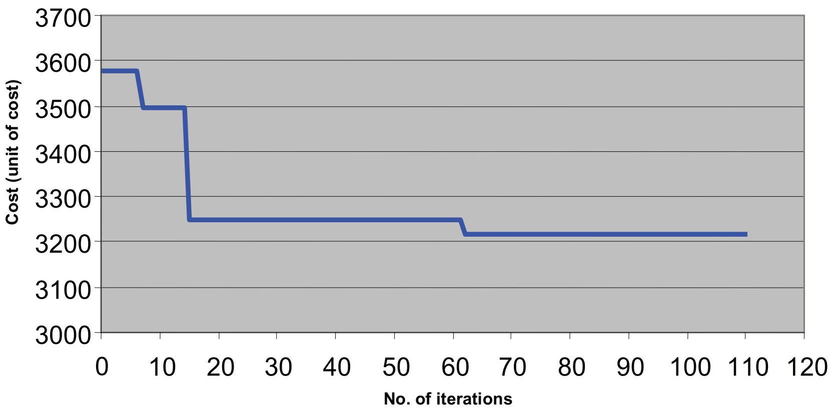

Fig. 4. Optimum bid recorded at each TS move (α = 15%).

Figures 5 and 6 show the results obtained in second test run, the bids received at each TS move and the optimum bid recorded during each move respectively. When the simulation completed, the near-optimum bid obtained was the same as the first test run and the first near-optimum bid was obtained at the 58th move. These results show that in many moves there are no bids received from the machine agents. This happens predominantly when diversification takes place. As each of the currency values is particularly allocated to a unique job to produce a particular feature, exchanging currency values from one job with another is inappropriate. For instance, assume that the currency values for the first feature of a component is relatively large (say 1500) and for the second feature is small (e.g. 500).

Lathe	Schedule	Milling	Schedule	Drilling	Schedule
L1	**ComA(2)*** **212-470^** **ComC(3)** **530-788** *C4(2)* *788-988*	M1	**ComB(1)** **0-242** **ComB(3)** **446-515** **ComA(3)** **520-625** *C3(2)* *625-917* **ComC(4)** **945-1080** *C3(3)* *1080-1276*	D1	**ComA(1)** **0-180** *C1(1)* *180-420* **ComC(5)** **1200-1374**
L2	**ComC(1)** **0-384** **ComC(2)** **384-522** *C1(2)* *522-862*			D2	C2(2) 460-650
				D3	C2(4) 1200-1380
L3	C3(1) 0-415 C2(3) 750-925	M2	C2(1) 0-380 *C4(3)* *1168-1336*		
L4	**ComB(2)** **302-386** **ComA(4)** **728-844** *C1(4)* *1186-1511*	M3	C4(1) 0-432 *C1(3)* *912-1120*		

*C*x*(*y*) means job sequence *y* of component *x*
^ in unit of time
Highlighted in bold = new jobs being scheduled
Highlighted in Italic = existing jobs being rescheduled

Table 4. New machines job schedule

When diversification strategy (pairwise exchange) takes place, the currency value for the first feature is now 500 and as a result, there will not be any bids put forward by the machine agents throughout the entire intensification process until the next diversification takes place. However, the diversification strategy in TS leads to a great opportunity for the search process to explore new region aiming to obtain better solutions. This can be observed in Figure 3 during the moves from 90th to 105th that many bids have been put forward.

To evaluate further the effectiveness of the bidding mechanism, the simulation results obtained were further analysed by comparing with three heuristic integrated process planning and scheduling approaches by Khoshnevis & Chen (1993), Usher & Fernandes (1996) and Saygin & Kilic (1999). Khoshnevis & Chen (1993) proposed an integrated process planning and scheduling system whereby the two stages of process planning and production scheduling are treated as a unified whole. This system uses a six-step heuristic approach based on opportunistic planning to generate feasible process plans through the creation of detailed routing, scheduling and sequencing information. Usher & Fernandes (1996) proposed PARIS (Process planning ARchitecture for Integration with Scheduling) – a two-phased architecture for process planning that supports the integration with scheduling. Saygin & Kilic (1999) proposed a framework that integrates predefined flexible process plans with off-line (predictive) scheduling in flexible manufacturing systems.

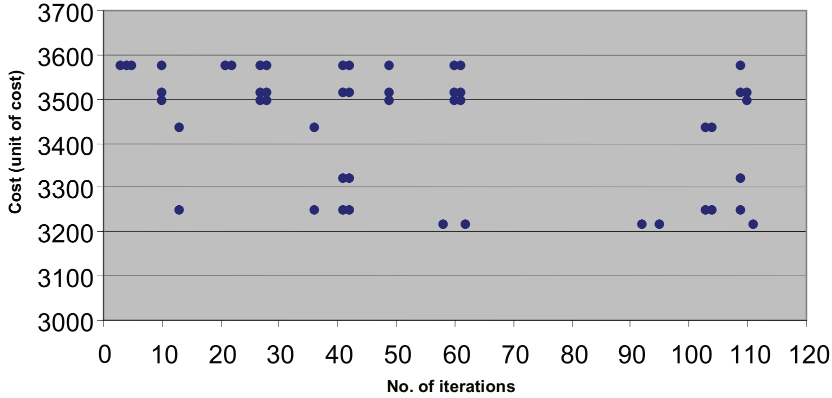

Fig. 5. Bids received at each TS move (α = 30%).

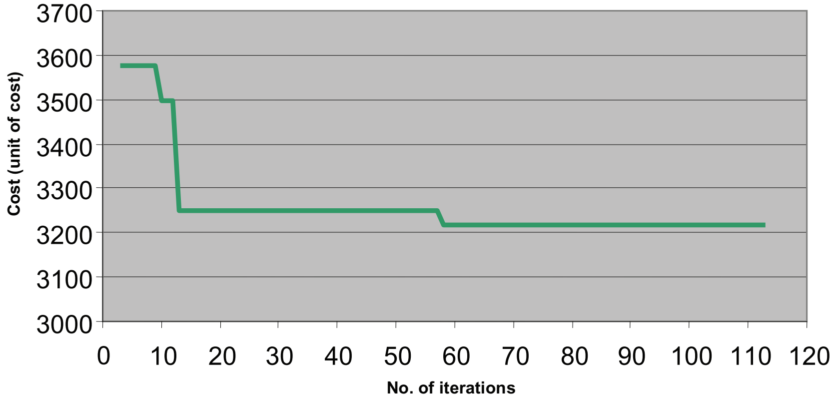

Fig. 6. Optimum bid recorded at each TS move (α = 30%).

In order to make a rational comparison with the iterative bidding MAS developed in this study, the same test case is used to simulate the three heuristics approaches. Based on the simulation results obtained, Table 5 can be drawn for comparison purposes between the four approaches. The highlighted sections indicate the best results between these approaches. Based on the results, the approach by Khoshnevis & Chen (1993) is not able to achieve more promising result (i.e., lower lead time and production cost) than the MAS. The results obtained for ComA and ComC are no better than those achieved by the MAS. However, this approach manages to achieve the same lead time and production cost for ComB as the MAS. For PARIS system, the static phase involves the determination of suitable processes for each feature and followed by machine-group selection, to produce a list of alternative process plans. In the dynamic phase, all of these alternative process plans are scheduled based on the operational status of the machine on the shop floor. In order to make a relevant comparison with the MAS, the criteria used in the process of scheduling are the production cost and lead time (i.e., to meet the delivery due dates). The results, once again, show that the MAS is able to obtain better results than this approach. Furthermore, this approach performs poorer than Khoshnevis & Chen (1993). As for Saygin & Kilic's approach, after rescheduling the results are improved which are the same as the ones obtained in Khoshnevis & Chen (1993).

These results noticeably show that the iterative bidding MAS proposed in this study outperforms these heuristic approaches. Not only it is capable of obtaining better solutions but also the way of the system performs (i.e., autonomous approach) is well suited to integrate process planning and production scheduling with time- and cost-efficiency. Unlike heuristics approaches, the MAS does not generate a list of process plans and allocate machines to these plans, and subsequently determine the best plan based on certain criteria. The MAS allows agents that represent the machines to decide what the best is for them (e.g. maximise their utilisation) by letting them bid for jobs based on their capability. In this way, a near-optimum solution can be achieved and the utilisation of manufacturing resources can also be optimised.

Component	**MAS**		**Khoshnevis & Chen**		**Usher & Fernandes**		**Saygin & Kilic (after rescheduling)**	
	Total Lead Time	Total Production Cost	Total Lead Time	Total Production Cost	Total Lead Time	Total Production Cost	Total Lead Time	Total Production Cost
ComA	**844**	**1894**	1254	2493	1276	2538	1254	2493
ComB	**515**	**1302**	**515**	**1302**	826	1475	**515**	**1302**
ComC	**1374**	**3224**	1882	3620	2430	4032	1882	3620

Table 5. Comparative results between MAS and heuristic approaches

7. Conclusion

In order to achieve manufacturing competence (through cost-competitiveness), this chapter introduced a multi-agent system (MAS) to enable process planning options and production scheduling options to be evaluated and optimised dynamically. The proposed MAS helps to enhance the agility and flexibility of manufacturing systems to cope with dynamic changes in the market by achieving near-optimum solutions to integrated process planning and scheduling problems. To achieve this, a novel currency-based iterative agent bidding mechanism is used as an agent coordination protocol. Agents representing the machines on the shop floor will bid for jobs to produce components; as iterative bidding takes place it aims to lead to better and better solutions to achieve cost-effectiveness.

To facilitate the iterative bidding mechanism, a Tabu search optimisation technique was developed to adjust the current values. A test case was used to simulate the agent bidding mechanism and test runs were executed to evaluate the effectiveness of the bidding mechanism. The simulation results show that as the currency values were adjusted at each TS move, the production cost of producing the components was gradually reduced. The results were then compared to the results obtained based on three heuristic approaches (Khoshnevis & Chen, 1993; Usher & Fernandes, 1996b; Saygin & Kilic, 1999). The comparative results show that the MAS outperforms the heuristic approaches. The MAS evaluates and optimises process plans and production schedules simultaneously. It allows agents that representing the machines to bid for jobs based on their capability and best performance (e.g. to maximise their machine utilisation). Furthermore, the MAS also provides a platform where the possible reconfiguration of manufacturing systems can be assessed and the utilisation of manufacturing resources can be optimised. For future work, the MAS could be enhanced with machine learning capability in order to facilitate the iterative bidding mechanism to achieve optimised solutions more rapidly and efficiently.

8. References

Blum, C. & Sampels, M. (2004). An Ant Colony Optimisation Algorithm for Shop Scheduling Problems. *Journal of Mathematical Modelling and Algorithm,* Vol.3, pp. 285–308.

Baykasoglu, A. & Gocken, T. (2010). Multi-Objective Aggregate Production Planning with Fuzzy Parameters. *Advances in Engineering Software*, Vol.41, No.9, pp. 1124-1131.

Baykasoglu, A. & Ozbakir, L. (2009). A Grammatical Optimization Approach for Integrated Process Planning and Scheduling. *Journal of Intelligent Manufacturing,* Vol.20, No.2, pp. 211-221.

Caridi, M. & Sianesi, A. (2000). Multi-Agent Systems in Production Planning and Control: An Application to the Scheduling of Mixed Model Assembly Lines. *International Journal of Production Economics,* Vol.68, pp. 29–42.

Chen, Q. & Khoshnevis, B. (1992). Scheduling with Flexible Process Plans. *Production planning and Control,* Vol.3, pp. 1-11.

Cho, S.Y. & Lazaro, A. (2010). Control theoretic model using PID controller for just-in-time production scheduling. *International Journal of Acvanced Manufactruing Technology,* Vol.51, Nos.5-8, pp. 699-709.

Demir, L.; Tunali, S. & Lokketangen, A. (2011). A Tabu Search Approach for Buffer Allocation in Production Lines with Unreliable Machines. *Engineering Optimisation*, Vol.43, No.2, pp 213-231.

Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence,* England: Addison-Wesley.

Glover, F. (1989). Tabu Search: Part I. *ORSA Journal of Computing,* Vol.1, pp. 190-206.

Glover, F. (1990). Tabu Search: Part II. *ORSA Journal of Computing,* Vol.2, pp. 4-32.

Hajizadeh, Y.; Christie, M. & Demyanov, V. (2011). Ant Colony Optimization for History Matching and Uncertainty Quantification of Reservoir Models. *Journal of Petroleum Science and Engineering*, Vol.77, No.1, pp. 78-92.

Ho, Y.C. & Moodie, C.L. (1996). Solving Cell Formation Problems in a Manufacturing Environment with Flexible Processing and Routeing Capabilities. *International Journal of Production Research,* Vol.4, No.10, pp. 2901-2923.

Hou, T. & Wang, H. (1991). Integration of a CAPP System and an FMS. *Computers and Industrial Engineering,* Vol.20, No.2, pp. 231-242.

Hutchinson, G.K. & Pflughoeft K.A. (1994). Flexible Process Plans: Their Value in Flexible Automation Systems. *International Journal of Production Research,* Vol.32, No.3, pp. 707-719.

Iwata, K. & Fukuda, Y. (1989). A New Proposal of Dynamic Process Planning. *Proceedings of CIRP International Workshop on CAPP*, Hannover, Germany.

Khoshnevis, B. & Chen, Q. (1989). Integrated Process Planning and Scheduling Functions. *Proceedings of the IIE Integrated Systems Conference and Society for Integrated Manufacturing Conference*, pp. 415-420.

Khoshnevis, B. & Chen, Q. (1990). Integrated Process Planning and Scheduling Functions. *Journal of Intelligent Manufacturing,* Vol.1, pp. 165-176.

Khoshnevis, B. & Chen, Q. (1993). Scheduling with Flexible Process Plan. *Production Planning and Control,* Vol.4, No.4, pp. 333-343.

Kim, H.J.; Chiotellis, S. & Seliger, G. (2009). Dynamic Process Planning Control of Hybrid Disassembly Systems. *International Journal of Advanced Manufacturing Technology,* Vol.40, Nos.9-10, pp. 1016-1023.

Kiritsis, D. & Porchet, M. (1996). A Generic Petri Net Model for Dynamic Process Planning and Sequence Optimization. *Advances in Engineering Software,* Vol.25, pp. 61-71.

Kumar, N.; Tiwari, M.K. & Chan, F.T.S. (2008). Development of a Hybrid Negotiation Scheme for Multi-Agent Manufacturing Systems. *International Journal of Production Research,* Vol.46, No.3, pp. 539–569.

Larsen, N.E. & Alting, L. (1990). Simultaneous Engineering within Process and Production Planning. *Pacific Conference on Manufacturing, Sydney and Melbourne*, pp. 1024-1031.

Li, X.Y.; Shao, X.Y.; Gao, L. & Qian W.R. (2010). An Agent-Based Approach for Integrated Process Planning and Scheduling. *Expert Systems with Applications,* Vol.37, No.2, pp. 1256-1264.

Lima, R.M.; Sousa, R.M. & Martins, P.J. (2006). Distributed Production Planning and Control Agent-Based System. *International Journal of Production Research,* Vol.44, Nos.18/19, pp. 3693–3709.

Liu, Q.; Lv, L. & Feng, T.T. (2007). Research on Reconfiguration of Manufacturing Resource Based on Multi-Agent. *Journal of Wuhan University of Technology,* Vol.29, No.12, pp. 119–122.

Mallur, S.; Mei, J. & Zhang, H. (1992). An Introduction to an Integrated Process Planning. *ASME Manufacturing International*, pp. 335-347.

Mamalis, A.G.; Malagardis, I. & Kambouris, K. (1996). On-Line Integration of a Process Planning Module with Production Scheduling. *International Journal of Advanced Manufacturing Technology,* Vol.12, No.5, pp. 330-338.

Moon, C.; Lee, Y.H.; Jeong, C.S. & Yun, Y.S. (2009). Integrated Process Planning and Scheduling in a Supply Chain. *Computer and Industrial Engineering,* Vol.57, No.4, pp. 1484-1484.

Moslehi, G.; Mirzaee, M.; Vasei, M.; Modarres, M. & Azaron, A. (2009). Two-Machine Flow Shop Scheduling to Minimize the Sum of Maximum Earliness and Tardiness, *Interntional Journal of Production Economics*, Vol.122, No.2, pp. 763-773.

Omar, M.K. & Teo, S.C. (2007). Hierarchical Production Planning and Scheduling in a Multi-Product, Batch Process Environment. *International Journal of Production Research,* Vol.45, No.5, pp. 1029-1047.

Ottaway, T.A. & Burns, J.R. (2000). An Adaptive Production Control System Utilising Agent Technology. *International Journal of Production Research,* Vol.38, No.4, pp. 721–737.

Robu, V.; Noot, H.; La Poutre, H. & van Schijndel, W. (2011). A Multi-Agent Platform for Auction-Based Allocation of Loads in Transportation Logistics. *Expert Systems with Applications*, Vol.38, No.4, pp. 3483-3491.

Saygin, C. & Kilic, S.E. (1999). Integrating Flexible Process Plans with Scheduling in Flexible Manufacturing Systems. *International Journal of Advanced Manufacturing Technology,* Vol. 15, No. 4, pp. 268-280.

Srihari, K. & Greene, T.J. (1988). Alternative Routeings in CAPP Implementation in a FMS. *Computers and Industrial Engineering,* Vol.15, Nos.1-4, pp. 41-50.

Ssemakula, M.E. & Wesley J.C. (1994). Functional Specification of a Dynamic Process Planning System. *Computers and Industrial Engineering,* Vol.27, Nos.1-4, pp. 99-102.

Tonshoff, H.K.; Beckendorff, U. & Anders, N. (1989). FLEXPLAN: A Concept for Intelligent Process Planning and Scheduling. *Proceedings of CIRP International Workshop on CAPP*, pp. 87-106.

Usher, J.M. & Fernandes, K. (1996). A Two-Phased Approach to Dynamic Process Planning. *Computers and Industrial Engineering*, Vol.31, Nos.1/2, pp. 173-176.

Wang, J.; Zhang, Y.F. & Nee, A.Y.C. (2009). Reducing Tardy Jobs by Integrating Process Planning and Scheduling Functions, *International Journal of Production Research,* Vol.47, No.21, pp. 6069-6084.

Wang, L.H.; Feng, H.Y. & Cai, N.X. (2003). Architecture Design for Distributed Process Planning. *Journal of Manufacturing Systems,* Vol.22, No.2, pp. 99-115.

Wang, X.; Li, Z., Liu, C. & Tian, W. (1995). Process Planning Under Job Shop Environment. *SPIE,* 2620, pp. 143-148.

Wang, Y.F.; Zhang, Y. & Fuh J. Y. H. (2011). Job Rescheduling by Exploring the Solution Space of Process Planning for Machine Breakdown/Arrival Problems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* Vol.225, No.2, pp. 282-296.

Wong, T.N.; Leung, et al. (2006). Dynamic Shop Floor Scheduling in Multi-Agent Manufacturing Systems. *Expert Systems with Applications,* Vol.31, pp. 486–494.

Wooldridge, M. (1997). Agent-Based Software Engineering. *IEE Proceedings on Software Engineering*, Vol.144, No.1, pp. 26-37.

Xu, K.; Feng, Z. & Jun, K. (2010). A Tabu-Search Algorithm for Scheduling Jobs with Controllable Processing Times on a Single Machine to Meet Due-Dates. *Computers and Operations Research*, Vol.37, No.11, pp. 1924-1938.

Zhang, H. (1993). IPPM-A Prototype to Integrate Process Planning and Job Shop Scheduling Functions, *CIRP,* Vol.42, No.1, pp. 513-518.

Zhang, Z. (2011). Towards Theory Building in Agile Manufactruing Strategies – Case Studies of an Agility Taxonomy. *International Journal of Production Economics,* Vol.131, No.1, pp. 303-312.