An Iterative Agent Bidding Mechanism for Responsive Manufacturing

MK Lim[1], Z Zhang[2], WT Goh[3]

[1](Corresponding author) School of Engineering and Applied Science, Aston University Aston Triangle, Birmingham B4 7ET UK, Tel: +44 (0)121 2043508, Fax: +44 (0)121 2043683, Email: M.K.Lim@aston.ac.uk

[2]School of Engineering, Computer Science and Mathematics, University of Exeter, Harrison Building, North Park Road, Exeter EX4 4QF, U.K. Tel: +44 (0)1392 263641, fax: +44 (0)1392 217965, Email: D.Z.Zhang@exeter.ac.uk

[3]Department of Mechanical Engineering, University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK. Tel: + 44 114 222 7885, Email: W.T.Goh@sheffield.ac.uk

Abstract

In today’s market, the global competition has put manufacturing businesses in great pressures to respond rapidly to dynamic variations in demand patterns across products and changing product mixes. To achieve substantial responsiveness, the manufacturing activities associated with production planning and control must be integrated dynamically, efficiently and cost-effectively. This paper presents an iterative agent bidding mechanism, which performs dynamic integration of process planning and production scheduling to generate optimised process plans and schedules in response to dynamic changes in the market and production environment. The iterative bidding procedure is carried out based on currency-like metrics in which all operations (e.g. machining processes) to be performed are assigned with virtual currency values, and resource agents bid for the operations if the costs incurred for performing them are lower than the currency values. The currency values are adjusted iteratively and resource agents re-bid for the operations based on the new set of currency values until the total production cost is minimised. A simulated annealing optimisation technique is employed to optimise the currency values iteratively. The feasibility of the proposed methodology has been validated using a test case and results obtained have proven the method outperforming non-agent based methods.

Keywords: Planning and production control, multi-agent system, system modelling and simulation, simulated annealing.

1. Introduction

In this present day, the market is highly competitive, dynamic, and customer driven. This has led to increasing rates of new product introduction (i.e., decreasing product life cycle) and dynamic variations in demand patterns across product mixes. As a result, customers have become harder to satisfy and manufacturing enterprises are facing greater pressures to be responsive and flexible in response to market changes in order to compete with business rivals in the same market focus. The competitive advantage is now largely dependent upon rapid responsiveness to the dynamic changes in product mixes and demand patterns, as well as to new opportunities in the market (i.e., market shifts). The urgent need for high responsiveness and flexibility in coping with the dynamic market changes has been demonstrated by the study carried out by Zhang and Sharifi (2001) involving a case with 12 companies and a questionnaire survey with 1000 companies. The analysis of the study also indicates that, in order to achieve high responsiveness, one of the operational issues to be focused on is production planning and control, particularly process planning and production scheduling, which must be dynamically and cost-effectively integrated. However, the conventional control strategies for manufacturing systems were not designed to achieve such responsiveness.

This paper introduces a multi-agent system (MAS) to model a manufacturing system, aimed at enhancing its agility and responsiveness to cope with the highly dynamic global market. The concept of MAS is adopted in this research because it is proven in the literature of its autonomous, intelligence, distributed decision-making approach of achieving satisfactory solutions to complex problems, such as the integration of process planning and production scheduling. The proposed MAS utilises an iterative agent bidding mechanism, which performs dynamic integration of process planning and production scheduling to generate optimised process plans and schedules in response to changes in the market and production shop floor. The iterative bidding procedure is carried out based on currency-like metrics in which all operations to be performed (e.g. machining processes) are assigned with virtual currency values, and resource agents (e.g. representing machines on the shop floor) bid for the operations if the costs incurred for performing the operations are lower than the currency values. The currency values are adjusted iteratively and resource agents re-bid for the operations based on the new set of currency values until the total production cost is minimised. A simulated annealing optimisation technique is employed to optimise the currency values iteratively, so as to achieving better and better bids, leading to (or near) optimality.

This paper is organised as follows: Section 2 discusses the concept of MAS and its applications in process planning and production scheduling as found in the literature. Section 3 describes in detail the iterative agent bidding mechanism used in the MAS developed in this research. Section 4 illustrates how the simulated annealing method is used in the iterative bidding mechanism to achieve optimisation in integrated process planning and production scheduling. Section 5 discusses the implementation and simulation results of the proposed MAS as well as a comparison of the performance of the proposed MAS and three other process planning and scheduling approaches found in the literature. Finally, Section 6 concludes this paper.

2. Agent based Integrated Process Planning and Production Scheduling

This section discusses the characteristics of a multi-agent system (MAS) and a review of its application in process planning and scheduling as found in the literature, which have formed the reason of its adoption in this research. A MAS is an intelligent system consists of a collection of agents, each of which represents an entity (e.g. a machine or a job) and is capable of interacting with one another to achieve its goals. Each agent 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). Within a MAS problem solving domain, a complex system is decomposed into several autonomous and loosely-coupled subsystems represented by agents. These agents will then interact collectively to solve a given problem, which could be part of a complex problem which has been broken down. Each agent determines its course of actions, although other agents may influence an agent's decision by forwarding appropriate messages. In a MAS, 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 simultaneously without depending on other agents. The characteristics of autonomous, intelligence, distributed decision-making architecture of agents have attracted many researchers in manufacturing domain solving complex manufacturing problems, nevertheless in the research of 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 the bidding based methods and the non-bidding based methods. The following review discusses the research work in both methods.

2.1 Bidding Based Methods

For the bidding based methods, agents perform bidding to produce process plans and production schedules. The bidding process begins with an agent, namely "manager" decomposes a task into manageable sub-tasks and announces these sub-tasks to other agents termed "constructors". Those contractors with the capability of processing the sub-tasks will bid for the tasks. Eventually, the manager will allocate individual sub-tasks to corresponding agents based on some criteria. This simple bidding based problem solving concept has gained great interests of the authors of this paper to examine its suitability for the research.

Gu, et. al. (1996) is one of the earliest bidding based methods employed for process planning and scheduling. Despite a successful development of an agent bidding method, a number of conceptual models are still proposed to verify the feasibility of using MAS in a distributed production planning and control environment (Yu and Krishnan, 2004; Lima, et. al. 2006). The conceptual models have proved their feasibility of employing these models in a simulated production environment. However, no discussion is provided on how to ensure the global performance is achieved in a dynamic scheduling environment. This achievement is an important measure for the research of this paper, in which the ultimate objective is to obtain an optimised process plan and production schedule, given the dynamic variations in demand patterns across products and changing product mixes.

The inspiration of the research of this paper has been provoked by currency based bidding mechanisms found in literature, which can be adopted for improving the coordination of agent bidding and negotiation to achieve system and cost optimisation. Lin and Solberg (1992) use a currency bidding mechanism to ensure the overall shop floor performance is achieved. This happens when the price values employed reach their equilibrium. Other researchers have adopted an optimisation approach such as genetic algorithm (Maione and Naso, 2001 and 2003; Shen, 2002; Goh, 2003; Deshpande and Cagan, 2004; Liu, et. al., 2007) to ensure the attainment of global objectives in a distributed agent bidding environment. As the nature of a MAS involved distributed decision-making where agents bid and negotiation until the objective functions are achieved, high communication overhead with long processing time is resulted. Responding to this issue, some researchers have proposed to integrate hierarchical and heterarchical control mechanisms to form a hybrid coordination and control mechanism for MAS (Wong, et. al., 2006; Kumar, et. al., 2008). A mediator is used to observe the agent negotiation process to avoid exhaustive negotiation which will lead to high communication overhead. Wong, et. al. (2006) has further analysed the hybrid coordination and control mechanism by comparing the performance generated using the hybrid mechanism and the one obtained using the traditional heterarchical mechanism. The results produced using the hybrid mechanism is found to outperform those obtained using pure heterarchical mechanism in terms of producing a shorter makespan. However, the above reviewed works are designed to address the process planning and scheduling without taking into consideration shop floor disturbances such as machine breakdown, change of production volume, change of process plan, etc. This consideration is another aspect of achievement aimed by the research of this paper.

2.2 Non-bidding Based Methods

For the non-bidding based methods, agents do not perform bidding but interact with one another via information exchange, either directly or indirectly to generate process plans and production schedules. Organisational self-design (Ottaway and Burns, 2000) and ant society (Valckenaers, et. al., 2000; Blum and Sampels, 2004; as well as van der Zwaan and Marques, 1999) are two well known non-bidding based methods. Within the ant society, ants (representing agents) do not negotiation or bid but exchange information by updating pheromones deposited at various machines and/or crossings. The information received is used to determine the best scheduling solution for production. The constraint-based Architecture for Multiagent Planning/Scheduling (CAMPS) proposed by Miyahita (1998) is another non-bidding based method proposed to address manufacturing shop floor planning and scheduling. The main drawback of this method is that the Planner Agent follows a fixed process plan (i.e., uses pre-fixed resources for each task). Caridi and Sianesi (2000) also employ a non-bidding based method for planning and scheduling in a mixed product assembly line. Results obtained from a case study reviewed that the proposed method does not perform any better than the traditional heuristic approach proposed by Bautista, et. al. (1996). Hence, this proves the significant of carrying out performance evaluation between agent based methods and non-agent based methods to validate the need for developing yet another MAS based methodology. As overall, not many non bidding based methods were developed because it is difficult for a distributed system to achieve its global performance without the aid of agent bidding or negotiation.

3. Iterative Agent Bidding Mechanism

A currency based iterative agent bidding mechanism is proposed to solve the optimisation issues that have not been attained to by the approaches discussed in the previous sections. The proposed iterative bidding mechanism performs dynamic integration of process planning and production planning functions to enable process planning options and production scheduling options to be evaluated simultaneously. It explores the optimisation method where choosing a machine that carries a higher production cost for preceding operation may well result in better machines selected (with lower production costs) for the subsequent operations, leading to overall optimality for completing an entire job which consists of *n* sequence of operations. This mechanism aims at obtaining an optimised process plan and schedule to achieve cost-effectiveness and manufacturing system efficiency in response to dynamic variations in demand patterns and product mixes in the market and dynamic production environment.

This bidding mechanism is based on currency-like metrics, in which each operation within a job is assigned with a virtual currency value, as a parameter to control the bidding process. The values are set randomly within a range (between 0 and a higher limit defined by the maximal possible cost of the operations based on historical data). The operations will then be announced to the agents (that represent a shop of *m* resources/machines). Machine agents that have the technical ability to perform the operations will come forward to bid based on the currency values. These currency values are iteratively adjusted, and machine agents re-bid for the operations based on the new set of currency in order to find better and better bids, leading to optimum. The bidding process is initiated by a job agent which may be an agent representing a product or a component. The job agent is responsible for finding and allocating the operations in the job to appropriate machine agents for processing.

This iterative bidding process is shown in Figure 1. The bidding process begins when the job agent announces the job to be carried out to all machine agents to bid (Figure 1: step1). The announcement includes information on the machining operations to be carried out (e.g. number and type of machining operations, recommended type of machining processes for the operations, etc.), as well as 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 (step 2-3).

Figure 1: Iterative bidding process

The leaders announce the second operation along with the allocated currency value to all machine agents, including the leaders themselves (step 4). Machine agents that are able to carry out the second operation will come forward to bid for the job. Machine agents may reschedule and optimise its job buffer (consisting of those operations that have been previously planned and scheduled) locally by shifting jobs if other operations due dates are not violated, in order to produce optional and better bids. 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.

Machine agents work out their bids in terms of production cost and lead time. The individual machine production cost is obtained as:

$$C_{prod} = C_{hand} + C_{setup} + C_{tool} + C_{mc} + C_{ms} \tag{1}$$

where C_{hand} = handling cost from the location of preceding machine,

C_{setup} = setup cost,

C_{tool} = tool cost,

C_{mc} = machining cost,

C_{ms} = miscellaneous cost (e.g. the cost involved due to job shifting)

The machine production cost function used does not, however, truly reflect the actual cost in real production. The function is developed for evaluation purposes and costs such as material cost, labour cost and maintenance cost are neglected. The individual machine lead time is worked out as below:

$$T_{lead} = T_{trav} + T_{wt} + T_{setup} + T_{tc} + T_{mc} \tag{2}$$

where T_{trav} = travelling time from the location of preceding machine,

T_{wt} = waiting time at buffer,

T_{setup} = setup time,

T_{tc} = tool change time,

T_{mc} = machining time.

The machine agent compares the cost with the corresponding currency value to work out a virtual profit. The virtual profit PF is:

$$PF = CU_i - C_{prod} \tag{3}$$

where CU_i = currency allocated to i^{th} operation,

C_{prod} = individual machine production cost.

If the profit is higher than a set threshold δ (i.e., $PF \geq \delta$), the machine agent submits the bid to the leader. A machine agent may also produce more than one bid by placing the job in different buffer positions as long as the due dates of existing jobs in the buffer are not violated in which case, additional cost incurred for other jobs will be added to the bid cost. The threshold varies from one machine to another based on the cost of machine.

When the leaders have virtually grouped other machines to perform all operations, they put together all the individual production costs (i.e. total production cost) and lead times (i.e. total lead time) of selected machines, and forward the complete bid as a machine group to the job agent to be evaluated (step 5-6), which consist of the total lead time and total production cost denoted as follow:

$$T = \sum_{i=1}^{n} T_i^{win}, \qquad C = \sum_{i=1}^{n} C_i^{win} \tag{4}$$

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

$$\begin{aligned} & Min\left(C = \sum_{i=1}^{n} C_i \right) \\ & T = \sum_{i=1}^{n} T_i \le D \end{aligned} \tag{5}$$

If the due date is not satisfied ($T > D$), or the cost is not considered minimal, the virtual currency allocated to operations will be adjusted in the next iteration to look for a better plan (step 7-8). The lead time and cost of a plan resulting from a bidding iteration are dependent on values set for virtual currencies. Higher virtual currencies for operations increase the attractiveness of the operations to resources and encourage resources to submit more bids for the operations (even though some bids may bear high costs), making it more likely to find a plan to meet the due date. Lower virtual currencies, on the other hand, reduces the attractiveness of operations to resources and discourage resources from submitting high cost bids for the operations, making it more likely to find a plan that gives a low cost. The iterative loop stops when an optimum (or near-optimum) plan that satisfies the due date with considered near minimum cost is found (step 9-10).

As each agent is autonomous, the job agent is not able to control the order in which the machine agents forwarding the bids, therefore the job agent keeps a list of the bids received. The job agent will then evaluate these bids one at a time on first-come first-serve basis. When all the bids are evaluated and 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 confirm with the job agent to commit to the operations awarded by updating their loading schedules (step 11-12).

If the order and quantity of a product (i.e. a job) to be produced are consistent and large, 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.

Each agent has individual objectives and a global goal to achieve. For instance, the global goal of the proposed MAS 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, while the machine agent's objective is 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 agents' objectives and global goal can be satisfied.

With the iterative adjustment of the currency values, the system 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. In other words, these currency values can be used by the job agent as a means to control (encourage or prohibit) the behaviour of agents to put forward their bids.

In the following section, simulated annealing (SA) is introduced as an optimization technique to investigate how and to what degree the currency values should be adjusted/optimized in each iteration in order to obtain better solutions (leading to optimality) to integrated process planning and production scheduling problem.

4. Simulated Annealing (SA) for Currency Value Adjustment

In this study, the currency values assigned to each operation in a job are adjusted according to the performance of total bids (in terms of production cost and lead time) put forward by the machine agents in the previous round, aiming to obtain a total bid that carries the minimum total production cost while satisfying delivery due dates. A SA approach is proposed to adjust these currency values, as summarised in Figure 2.

Figure 2: SA tuning process

The job agent initialises the SA value tuning process by setting the control SA parameters (step 1), such as initial, current temperature and terminating temperature, and acceptance probability. When the job agent receives the bids from the leaders, it will evaluate the bids by means of satisfying the due date (step 2-4). If all the bids do not satisfy the due date, the job agent will adjust the currency values and re-announce the new set of currency to machine agents (step 5). Amongst the bids received, the job agent selects a bid which satisfies the due date and gives the least production cost and records it as the best bid found so far (step 6). If the SA's terminating temperature (which was set in step 1) is not reached, the SA tuning process and re-announcement of new set of currency value will resume (step 5, 7). Otherwise, the best bid found so far will be the near-optimum bid and the job agent will confirm the awarding the operations to the successful machine agents (step 8).

The SA approach is carried out in two phases. In the first phase, an initial solution (i.e., a set of virtual currency values) for all operations in a job is generated and announced to machine agents by the job agent. All machine agents will come forward to bid for the jobs of producing the features and the leaders will then forward the total bids to the job agent. Subsequently, the job agent selects the best total bid that satisfies the due date and carries the least production cost as the optimal bid. If there is no total bid sent forward by the leaders, the currency values in the initial solution will be increased by an increment factor (ω), and re-announced to the machine agents to let them bid and if there is still no total bid sent forward, the currency values will again be increased until at least one total bid is forwarded.

In the second phase, the perturbation scheme takes place (i.e., the currency values are adjusted). This scheme is not entirely a random search scheme as the classical SA algorithm does, but it is biased in a way that the optimisation searching process will be speeded up. The bias is dependent on the temperature at n^{th} round and $(n-1)^{th}$ round, which is also related to the cooling rate (r_1 or r_2). The cooling rate is not a constant, where the temperature of next round of iteration will be decreased with a greater degree if a new optimal solution is obtained in the current round and vice versa. Similarly, if a new optimal solution is obtained in the previous round, the perturbation on the new set of currency will also be greater as temperature of this round has been decreased with a greater degree.

The new solution generated from the perturbation scheme will then be announced to the machine agents. Amongst the total bids forwarded by the leaders, the best total bid of this round will be selected and compared to the optimal bid. If the current best total bid performs better than the optimal bid, the optimal bid will be replaced and the solution of which will be used for perturbation in the next iteration. This process will be carried out in the length of Markov chain (i.e., the number of iterations). In this study, the length of Markov chain is corresponding to the number of features in the particular component and the degree of temperature reduction. It can be shortened when the pre-determined number of loops of no new optimal solutions found (χ) is reached. In addition, this SA approach includes a counter to record the number of small improvements made in the current round based on a small improvement index (λ) that should be relatively small. The SA process will also terminate once the prescribed number of consecutive small improvements made (γ) is reached. This is to speed up the SA process in finding the optimal bid. All these parameters will be initialised by the job agent (step 1). The currency based SA approach is implemented and the results will be discussed in the following section.

5. Implementation and Results

A MAS is designed as a platform for the iterative agent bidding mechanism to take place and it is implemented on a Java platform. To evaluate the effectiveness of this bidding mechanism, the mechanism is tested on a simplified manufacturing system, which consists of four lathe machines (LATHE 1, LATHE 2, LATHE 3, LATHE 4), three milling machines (MILL 1, MILL 2, MILL 3) and three drilling machines (DRILL 1, DRILL 2, DRILL 3). Each of the milling and drilling machines is capable of providing one type of machining processes, which is milling and drilling processes respectively. As for the lathe machines, two types of processes are available, drilling and turning. The plant layout of these machines is illustrated in Figure 3.

Figure 3: Manufacturing layout

In terms of material transportation, these machines are served by automated guided vehicles (AGVs) and each machine has input and output buffers. In addition, each machine has its own tool magazine as illustrated in Figure 3. Each machine has its own machine data that is different from one another in terms of capacity and capability. Each of the machines has its own job buffer previously scheduled, in this case, of four components (Com 202, Com302, Com 331 and Com 441). The graphical representation of machines job buffer (i.e., schedule plan) is depicted in Figure 4. The figure in the subscript parentheses indicates the features' sequence number of the corresponding component. There are three components to be produced in sequence, namely ComA, ComB and ComC (Table 1). Each of these components uses material AISI 1118 with a hardness of 100 BHN. The process plans in terms of the features sequence to produce these components along with the required machining processes are listed in Table 2. These plans include information such as currency values, removal volumes, and tolerance requirements of each feature in the components. These parameters are developed arbitrarily for system evaluation purposes.

Figure 4: Machines job buffer

Table 1: The order details of the three components

Table 2: Process Plans for ComA, ComB and ComC

The simulation process for ComC is predominantly discussed in this paper. To examine the effectiveness of the proposed SA approach for currency value adjustment, three test runs were carried out: (1) non-random SA (which has been discussed in the previous section in which the perturbation scheme is not entirely a random search scheme but it is biased dependent on the temperature at n^{th} round and $\{n\text{-}1\}^{th}$ round), (2) random SA (which is a completely random scheme and will be discussed later in this section), and (3) acceptance probability *U* was reduced. Each of these test runs will be discussed as follows. The assumptions made in the implementation include:

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

In the first test run, the SA parameters are determined as follows: increment factor ω is 1.2, initial temperature T_0 is 475 units, terminating temperature T_f is 20 units, prescribed cooling factor r_1 (if new optimum solution was found) is 0.96, and prescribed cooling factor r_2 (if no new optimum solution was found) is 0.98, acceptance probability U is 0.8, counter for consecutive loops with no better solution found χ is 50, counter for loops with consecutive small improvements made γ is 10, and small improvement index λ is 0.01. The simulation results of the near-optimum solution/total bid received for ComA are shown in Table 3. This table also highlighted some jobs involved in producing Com202 and Com331 are being shifted without violating their delivery due dates. The costs involved in shifting these jobs such as holding costs are taken into account when the resource agents bid for ComA. Figure 5 is the schematic representation of the new machines job buffer when ComA is scheduled. Arrows are used in this figure to indicate those jobs that have been shifted.

Table 3: Production cost and lead time in the near-optimum bid for ComA

Figure 5: Updated machines job buffer

Table 4 shows the near-optimum total bid received for ComB and the machines involved in the total bid are scheduled as illustrated in Figure 6. Based on the near-optimum total bid results, there are no shifting jobs taking place in the scheduling of ComB because performing jobs at the machines (i.e., MILL 1, LATHE 4 and MILL 1 for features 1, 2 and 3 respectively) that are idle during that particular period of time is preferable to shifting jobs and carrying the accessory holding cost and time.

Table 4: Production cost and lead time in the near-optimum bid for ComB.

Figure 6: Updated machines job buffer

For ComC, the SA simulation terminated at the 95th iteration as no better solution was obtained after the prescribed number of consecutive iterations. The near-optimum total bid obtained to produce ComC is offered by LATHE 2, LATHE 1, MILL 1 and DRILL 1 in sequence with a total lead time of 1412 units of time and a production cost of 3215 units of cost. Table 5 shows the individual production time and cost of the machines involved in this total bid. Figure 7 depicts the schematic view of the new time assignment of all machines when ComC is scheduled. Figure 8 depicts all the total bids (in total production cost) put forward by the leaders at each iteration for ComC. Each dot in the figure represents a total bid received. In Figure 9, the plotted line indicates the optimum total bid recorded ($cost(S_{optimal})$) at each iteration throughout the entire simulation process. This indication shows that the cost to produce the component gradually decreases as the currency values are optimised iteratively.

Table 5: Production cost and lead time in the near-optimum bid for ComC

Figure 7: Updated machines job buffer

Figure 8: Bids received at each SA iteration (non-random approach)

Figure 9: Optimum bid recorded at each SA iteration (non-random approach)

A second test run is simulated with a random SA approach where the perturbation is carried out randomly within specified limits regardless of the temperature. The lower and upper limits are set at 0.8 and 1.2 respectively. Other parameters are set the same as the first SA test run (i.e., the non-random approach): increment factor ω is 1.2, initial temperature T_0 is 475 units, terminating temperature T_f is 20 units, prescribed cooling factor r (there is only one cooling factor for all situations) is 0.98, acceptance probability U is 0.8, counter for consecutive loops with no better solution found χ is 50, counter for loops with consecutive small improvements made γ is 10, and small improvement index λ is 0.01. The random SA simulation ends at the 112th iteration where no better solution had been found after the prescribed number of consecutive iterations. The near-optimum total bid obtained to produce ComC is the same as the one in the non-random approach, 1412 units of time at 3215 units of cost with the same machines involved. The total bid results received at each iteration are depicted in Figure 10. Figure 11 depicts the plotted line indicating the optimum total bid recorded at each iteration. In comparison to Figure 8, random SA approach reaches the near-optimum total bid at the 62nd iteration, while in the non-random SA the total bid is obtained at the 45th iteration. This comparison shows that the non-random SA is able to find better total bids in shorter time and hence the non-random SA performs better in searching for better total bids. In addition, both the random and non-random approaches can be further analysed based on the dots in Figures 8 and 10. It is obvious that the total bid results obtained in the random SA are more scattered and spread out than the results in the non-random

SA. This suggests that the non-random SA is able to optimise the currency values in such a way that when the currency values are adjusted iteratively, better total bids are obtained and higher cost total bids are gradually eliminated (i.e., to discourage resource agents with higher cost bids to come forward to bid for the jobs). However, in certain cases where a greater number of constraints/parameters are involved, random SA may be useful to explore wider non-elite solution spaces with the hope to obtain better results. In conclusion, particularly in this currency optimisation case, the results shown that the non-random SA approach outperforms random SA.

Figure 10: Bids received at each SA iteration (random approach)

Figure 11: Optimum bids recorded at each SA iteration (random approach)

A third test run is carried out with lower acceptance probability U of 0.6. (Note: a higher cost total bid will only be accepted if $exp(-\Delta/T) \geq U$, in other words, the lower the U, the higher possibility the higher cost total bids will be accepted) Other parameters are set the same as in the non-random SA approach. Figure 12 illustrates the total bid results obtained at each iteration. Figure 13 shows the plotted line representing the optimum total bid recorded at each iteration throughout the entire SA simulation. The first near-optimum total bid is obtained at the 48th iteration and the simulation ends at the 98th iteration. The near-optimum total bid is the one obtained in both the random and non-random approaches. Based on Figures 12 and 8, the changes in the value of U have little effect upon the time taken to search for near-optimum solution. However, by referring to the dots in Figure 12, the SA approach with lower U may still obtain high cost total bids as the temperature decreases. Eventually, this reflects the fact that the lower the U, the more chances the high cost total bids will be accepted. In these SA approaches, the idea of shortening the simulation process by using a counter for small improvement has little effect on the simulation due to the nature of the optimisation problem and the number of participating agents. Three additional test runs have also been simulated based upon the former three test runs and by increasing the counter for consecutive loops with no better solution found from 50 to 150. The results show no difference to the near-optimum total bid obtained in the former three test runs.

Figure 12: Bids received at each SA iteration (U =0.6)

Figure 13: Optimum bid recorded at each SA iteration (U =0.6)

In order to validate the effectiveness of the proposed MAS method, the results obtained were compared with those of three non-agent based integrated process planning and production approaches found in the literature. The first approach is proposed by Mei & Khoshnevis (1993), the second approach is PARIS (Process planning ARchitecture for Integration with Scheduling), proposed by Usher and Fernandes (1996) and the third approach is proposed by Saygin and Kilic (1999). In order to make rational comparisons with the MAS developed in this study, the same test case was applied to these three non-agent approaches. Table 6 depicts the comparative results between MAS and the three non-agent approaches for the three components in the test case. The highlighted sections indicate the best results obtained amongst the approaches for each component.

Table 6: Comparative results

Based on the results, the approach by Mei & Khoshnevis is not able to achieve more promising result (i.e., lower lead time and production cost) than 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 relevant comparisons 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 Mei & Khoshnevis. As for Saygin and Kilic's approach, before rescheduling takes place, the results for the three components are no any better than all other approaches. However, after rescheduling, the results are improved which are the same as the ones obtained in Mei & Khoshnevis' approach.

Based on the results, it noticeably shows that the MAS outperforms all other three approaches. Not only it is capable of obtaining better solutions but also the way of MAS performing (i.e., distribution approach) is well suited to integrate process planning and production scheduling with time and cost efficiency.

6. Conclusion and remarks

This paper presents an iterative agent bidding mechanism, which performs the integration of process planning and production scheduling functions dynamically, efficiently and cost-effectively to generate optimised process plans and schedules. This aim at achieving maximum manufacturing system efficiency and flexibility, and utilisation of manufacturing resources to respond rapidly to dynamic variations in demand patterns across products and changing product mixes, as well as in production environment. The iterative bidding mechanism is carried out based on currency-like metrics in which all jobs to be performed are assigned with virtual currency values, and resource agents bid for the jobs if the costs incurred for performing the jobs are lower than the currency values. The currency values are adjusted iteratively and resource agents re-bid for the jobs based on the new set of currency values until the total production cost is minimised while satisfying the delivery due date. A simulated annealing (SA) algorithm is proposed to optimise the currency values. In the implementation of this bidding mechanism, a test case is used to validate the feasibility of the proposed methodology. Three test runs are performed (i.e. the non-random SA, random SA approaches, and lower acceptance probability of SA control parameter) and results obtained show that the non-random SA is able to determine better total bids in a shorter time. Furthermore, these results are also compared with three non-agent based integrated process planning and scheduling approaches proposed by Mei & Khoshnevis (1993), Usher and Fernandes (1996), and Saygin and Kilic (1999); the comparative results noticeably indicate that the MAS developed in this study outperforms the other three approaches with more promising results. Therefore, it can be concluded that, due to the autonomous and intelligent behaviour and distributed decision-making architecture of agent based system, it could be a better approach to achieve near-optimum solutions to solve integrated process planning and production scheduling problems.

REFERENCES

1. Bautista, J., Companys, R., Corominas, A., 1996. Hueristic and exact algorithms for solving the mondem problem. European J. of Operational Research, 88, 101-113.
2. Blum, C., Sampels, M., 2004. An ant colony optimisation algorithm for shop scheduling problems. J. of Mathematical Modelling and Algorithm. 3, 285-308.
3. Caridi, M., Sianesi, A. 2000. Multi-agent systems in production planning and control: an application to the scheduling of mixed model assemble lines. Int. J. of Production Economics, 68, 29-42.
4. Deshpande, S., Cagan, J., 2004. An agent based optimization approach to manufacturing process planning. J. of Mechanical Design, Transactions of the ASME, 126 (1), 46-55.

5. Ferber, J., 1999. Multi-agent systems: an introduction to distributed artificial intelligence. London, Britain: Addison-Wesley.
6. Goh, W. T. 2003. Multi-agent based integrated process planning and scheduling in an agile manufacturing environment. PhD Thesis. The University of Exeter.
7. Gu, P., Balasubramanian, S., Norrie, D., 1995. “Bidding-based autonomous process planning and scheduling”, Proc. of SPIE, Vol. 2620, 184-193.
8. Krothapalli, N.K.C., Deshmukh, A.V., 1999. Design of negotiation protocols for multi-agent manufacturing systems. International Journal of Production Research, 37 (7), 1601-1624.
9. Lin, G.Y.J., Solberg, J. J. 1992. Integrated shop floor control using autonomous agents. IIE Transaction. 24 (3), 57-71.
10. Liu, Q., Lv, L., Feng, T. T., 2007. Research on reconfiguration of manufacturing resource based on multi-agent. J. of Wuhan University of Technology. 29 (12), 119-122.
11. Maione, G., Naso, D., 2001. Evolutionary adaptation of dispatching agents in heterarchical manufacturing systems. Int. J. of Production Research, 39 (7), 1481-1503.
12. Maione, G., Naso, D., 2003. A genetic approach for adaptive multi agent control in heterarchical manufacturing system. IEEE Trans. Sys. Man Cybernetics. A: Sys. Human. 33, 573-588.
13. Mei, Q., Khoshnevis, B, 1993. Scheduling with flexible process plan. Production Planning & Control, 4 (4), 333-343.
14. Morley, R.E., Schelberg, C., 1993. An analysis of a plant-specific dynamic scheduler. In proceedings of the NSF Workshop on Dynamic Scheduling.
15. Miyashita, K., 1998. CAMPS: a constraint-based architecture for multi-agent planning and scheduling. Journal of Intelligent Manufacturing, 9 (2), 147-154.
16. Ottaway, T.A., Burns, J.R., 2000. An adaptive production control system utilising agent technology. International Journal of Production Research, 38 (4), 721-737.

17. Sadeh, N.M., Hildum, D.W., Kjenstad, D., Tseng, A., 2001. MASCOT: An agent-based architecture for dynamic supply chain creation and coordination in the Internet economy. Production Planning and Control, 12 (3), 212-223.

18. Saygin, C., Kilic, S.E., 1999. Integrating flexible process plans with scheduling in flexible manufacturing systems. Int. J. of Advanced Manufacturing Technology, 15 (4), 268-280.

19. Shen, W. 2002. Genetic algorithms in agent-based manufacturing scheduling systems. Integrated Computer-Aided Engineering 9 (3), 207-217.

20. Usher, J.M., Fernandes, K., 1996. A two-phased approach to dynamic process planning. Computers and Industrial Engineering, 31 (1/2), 173-176.

21. Valckenaers, P., Brussel, H.V., Peeters, P., Kollingbaum, 2000. MASCADA: multi-agent manufacturing control for flexible flow shops. Proceedings of Annual Conference of ICIMS-NOE, ASI 2000 & IIMB 2000, 214-221.

22. van der Zwaan., S., Marques, C., 1999. Ant colony optimisation for job shop scheduling. Proc. of the 3rd Workshop in Genetic Algorithms and Artificial Life.

23. Wong, T. N., Leung, et. al., 2006. Dynamic shop floor scheduling in multi-agent manufacturing systems. Expert Systems with Applications, 31, 486-494.

24. Yu, J., Krishnan, K.K., 2004. A conceptual framework for agent-based agile manufacturing. Information Systems Journal, 14 (2), 93-109.

25. Zhang, Z., Sharifi, H., 2001. A methodology for achieving agility in manufacturing organisations. Int. J. of Operations & Production Management, 20 (4), 496-512.

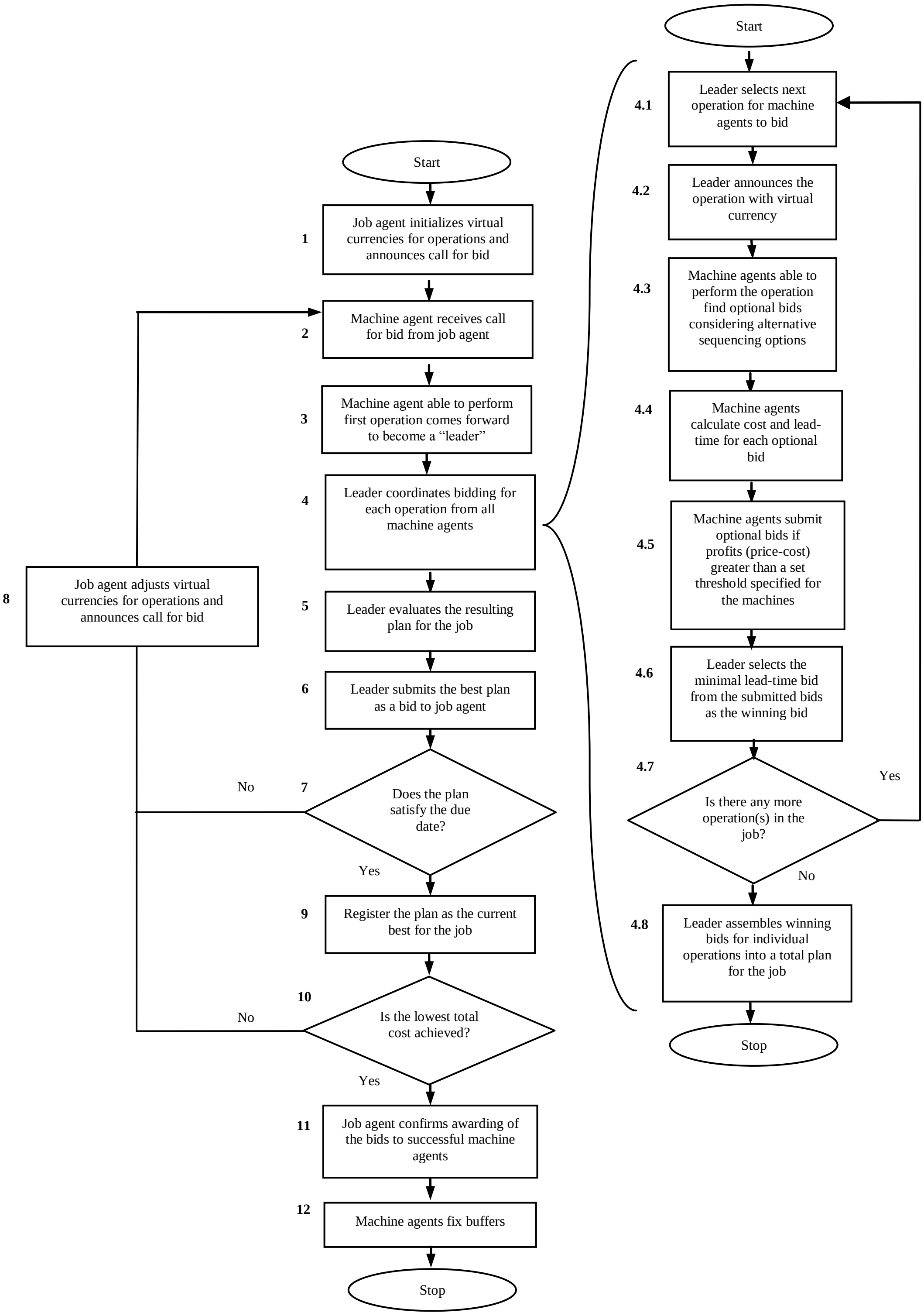

Start
1
Job agent initializes virtual currencies for operations and announces call for bid
2
Machine agent receives call for bid from job agent
3
Machine agent able to perform first operation comes forward to become a "leader"
4
Leader coordinates bidding for each operation from all machine agents
5
Leader evaluates the resulting plan for the job
6
Leader submits the best plan as a bid to job agent
7
Does the plan satisfy the due date?
No
Yes
8
Job agent adjusts virtual currencies for operations and announces call for bid
9
Register the plan as the current best for the job
10
Is the lowest total cost achieved?
No
Yes
11
Job agent confirms awarding of the bids to successful machine agents
12
Machine agents fix buffers
Stop
Start
4.1
Leader selects next operation for machine agents to bid
4.2
Leader announces the operation with virtual currency
4.3
Machine agents able to perform the operation find optional bids considering alternative sequencing options
4.4
Machine agents calculate cost and lead-time for each optional bid
4.5
Machine agents submit optional bids if profits (price-cost) greater than a set threshold specified for the machines
4.6
Leader selects the minimal lead-time bid from the submitted bids as the winning bid
4.7
Is there any more operation(s) in the job?
Yes
No
4.8
Leader assembles winning bids for individual operations into a total plan for the job
Stop

Figure 1: Iterative bidding process

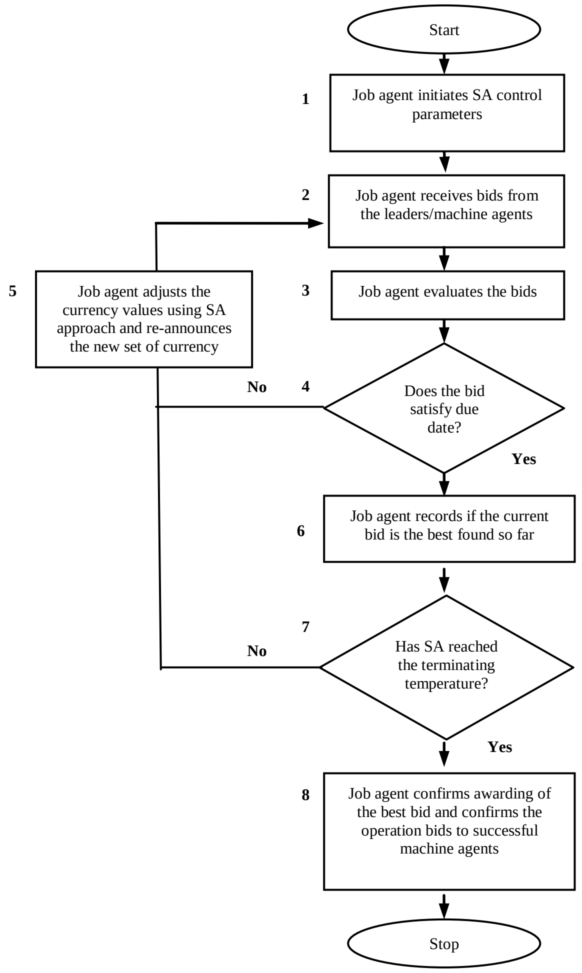

Start
1
Job agent initiates SA control parameters
2
Job agent receives bids from the leaders/machine agents
5
Job agent adjusts the currency values using SA approach and re-announces the new set of currency
3
Job agent evaluates the bids
No
4
Does the bid satisfy due date?
Yes
6
Job agent records if the current bid is the best found so far
7
No
Has SA reached the terminating temperature?
Yes
8
Job agent confirms awarding of the best bid and confirms the operation bids to successful machine agents
Stop

Figure 2: SA tuning process

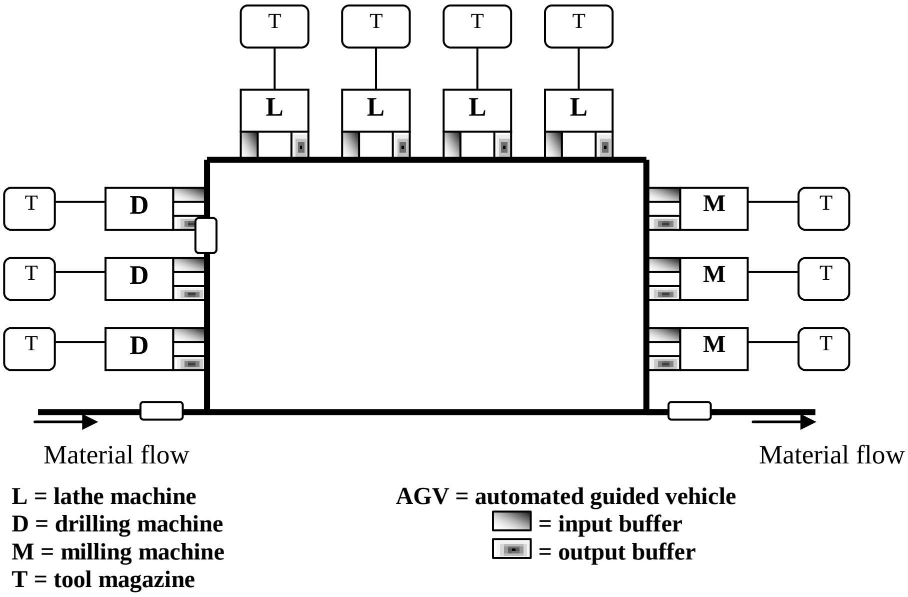
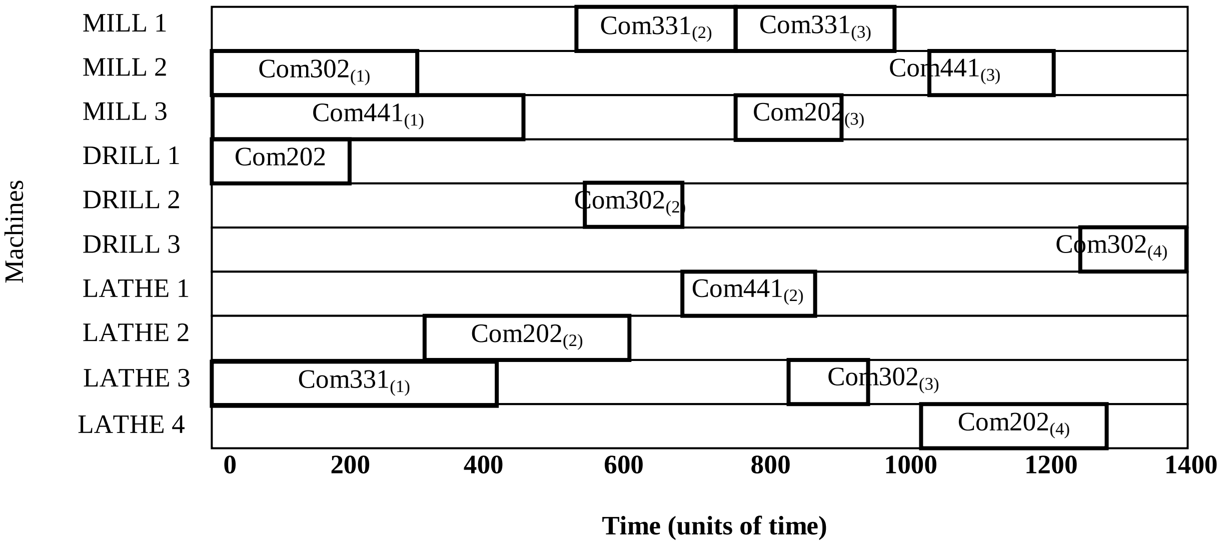


Figure 3: Manufacturing layout



Note: *Com_X$_{(i)}$* means the job to process *i*th feature of Component *X*

Figure 4: Machines job buffer

Table 1: The order details of the three components

Component ID	Quantity	Due date (unit of time)
ComA	50	1200
ComB	40	750
ComC	85	1600

Table 2: Process Plans for ComA, ComB and ComC

Feature	Type of removing features	Process required	Currency value	Removal volume (cm^3)	Tolerance (+/- mm)
ComA					
f1	Hole (Blind, flat-bottomed)	Drilling	500	55	0.75
f2	Hollow Cylinder (Through)	Turning	800	140	1.25
f3	Slot	Milling	440	30	1.00
f4	HoleC (Centre*, blind, flat-bottomed)	Drilling	500	40	075
	*Lathe machines can provide drilling process if the hole is at the centre.				
ComB					
f1	Slot	Milling	700	180	1.25
f2	HoleC (Centre, blind, flat-bottomed)	Drilling	400	30	1.00
f3	Slot	Milling	400	60	1.25
ComC					
f1	Hollow Cylinder (Through)	Turning	1050	150	1.25
f2	Hollow Cylinder (Through)	Turning	500	40	1.25
f3	Hollow Cylinder (Through)	Turning	700	70	1.25
f4	Slot	Milling	440	30	1.00
f5	Hole (Blind, flat-bottomed)	Drilling	660	50	0.75

Table 3: Production cost and lead time in the near-optimum bid for ComA

	Machines involved in sequence order				**TOTAL**
	DRILL 1	**LATHE 1**	**MILL 1**	**LATHE 4**	
Production cost (unit of cost)	445	663	375	405	1888
Production lead time (unit of time)	0–120 *(Shifted Com202 by 120)*	180–429	579–658 *(Shifted Com331 by 153)*	718–839	839

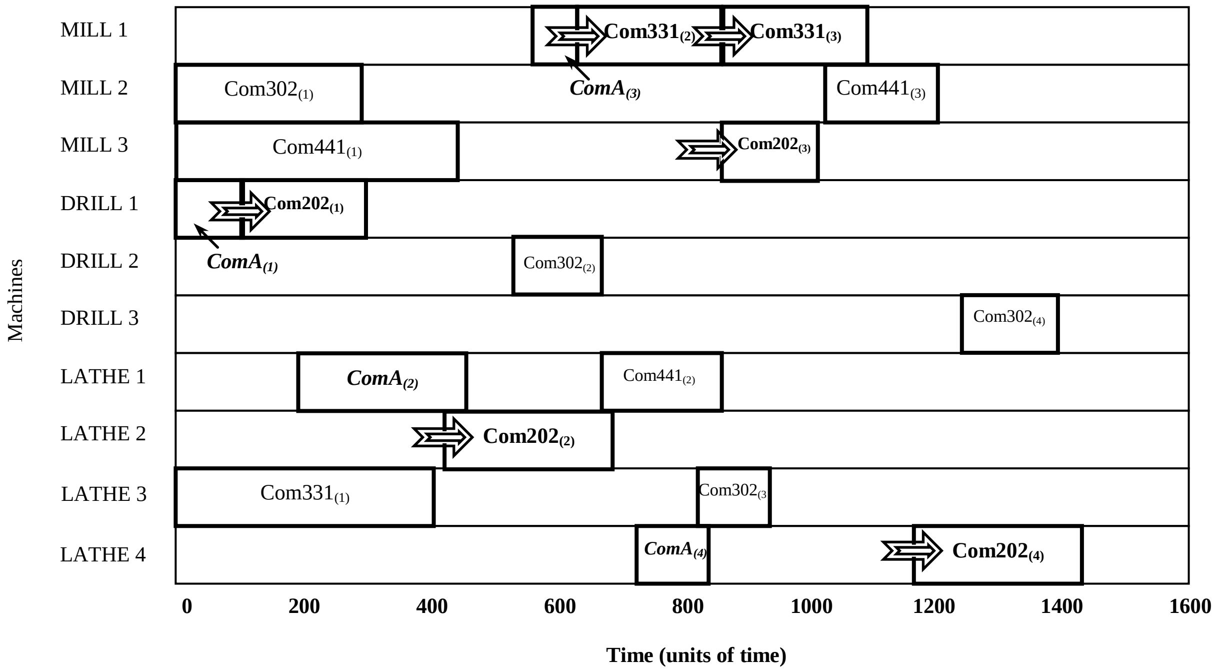


Figure 5: Updated machines job buffer

Table 4: Production cost and lead time in the near-optimum bid for ComB.

	Machines involved in sequence order			**TOTAL**
	MILL 1	**LATHE 4**	**MILL 1**	
Production cost (unit of cost)	605	351	381	1337
Production lead time (unit of time)	0–229	289–382	442–545	545

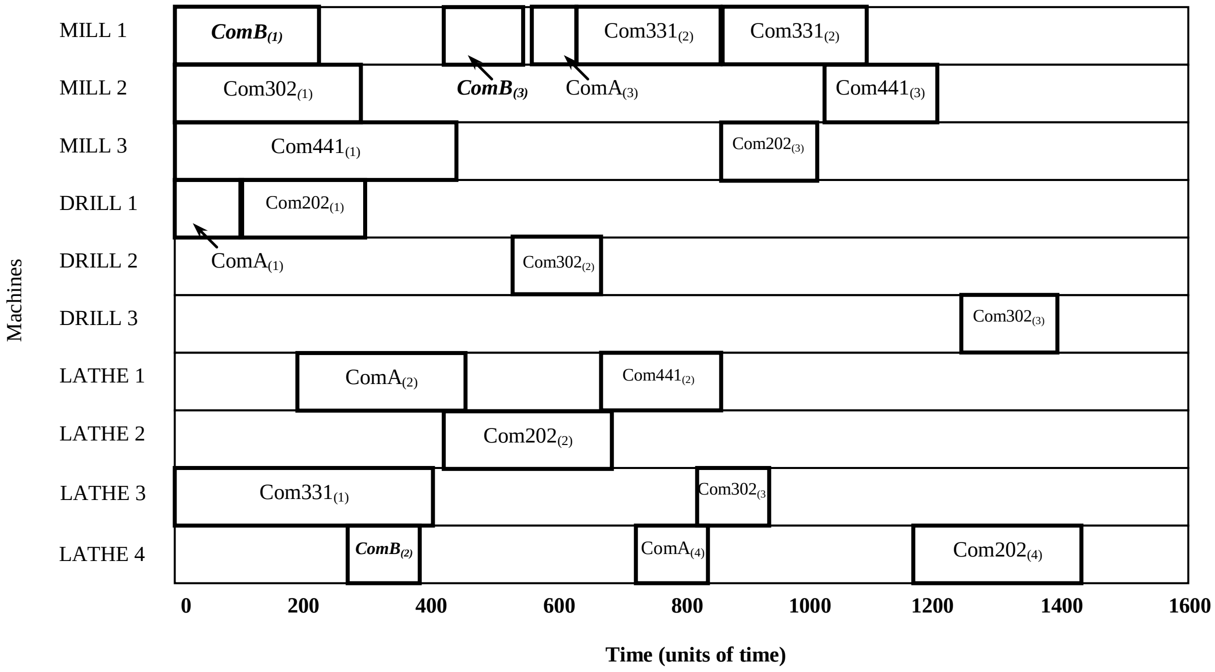


Figure 6: Updated machines job buffer

Table 5: Production cost and lead time in the near-optimum bid for ComC

	Machines involved in sequence order					**TOTAL**
	LATHE 2	**LATHE 2**	**LATHE 1**	**MILL 1**	**DRILL 1**	
Production cost (unit of cost)	1050	465	647	429	624	3215
Production lead time (unit of time)	0–372	372–510 *(Shifted Com202 by 100)*	540–784 *(Shifted Com441 by 114)*	934–1041 *(Shifted Com331 by 163)*	1251–1418	1418

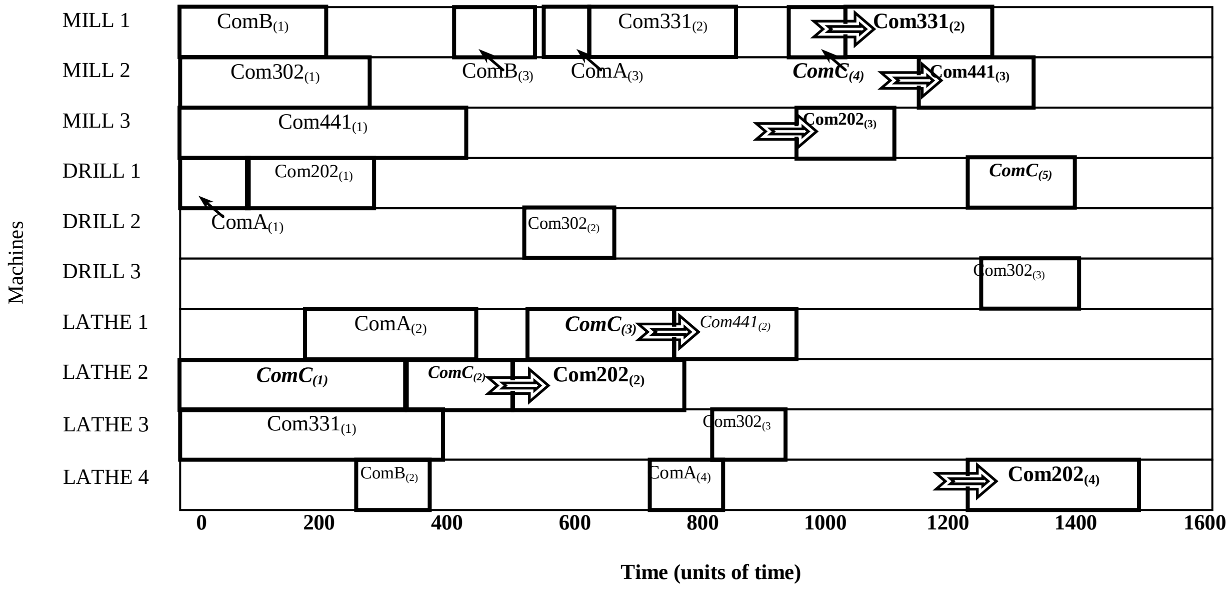


Figure 7: Updated machines job buffer

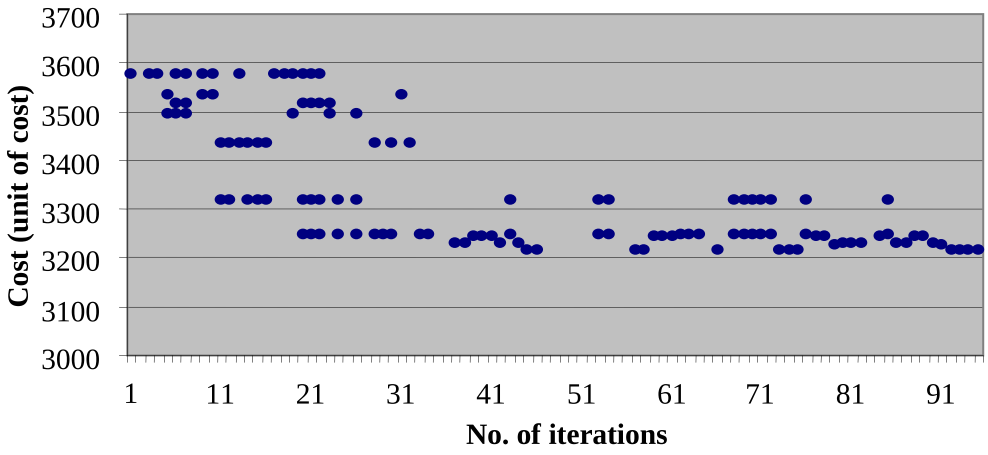


Figure 8: Bids received at each SA iteration (non-random approach)

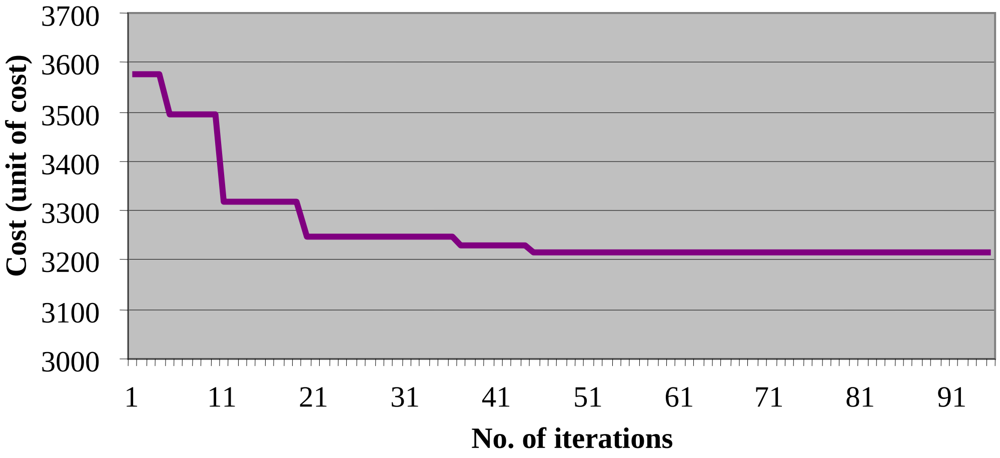


Figure 9: Optimum bid recorded at each SA iteration (non-random approach)

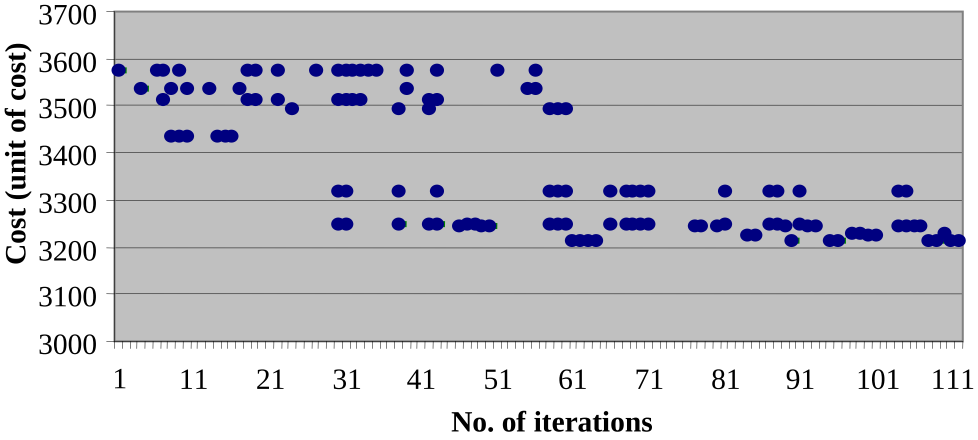


Figure 10: Bids received at each SA iteration (random approach)

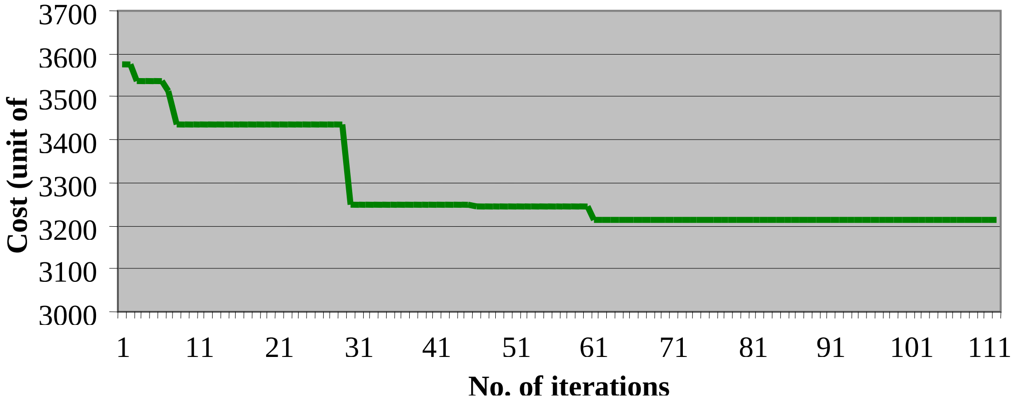


Figure 11: Optimum bids recorded at each SA iteration (random approach)

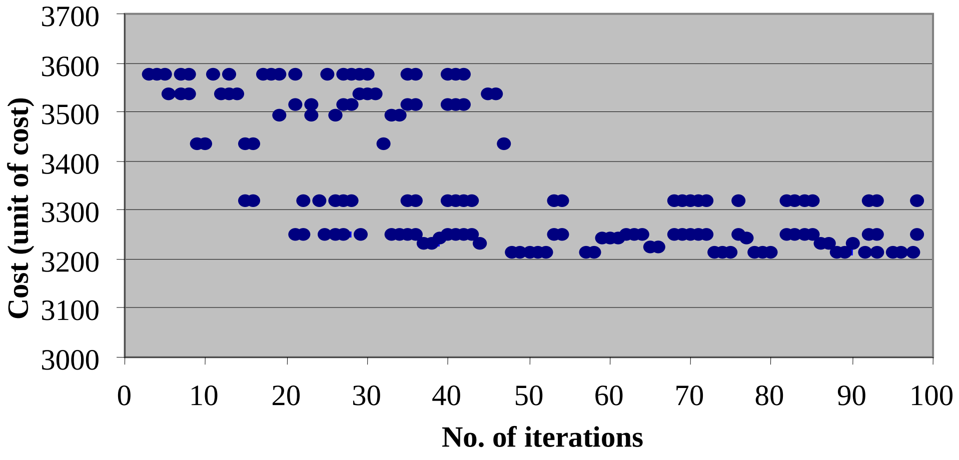


Figure 12: Bids received at each SA iteration (U =0.6)

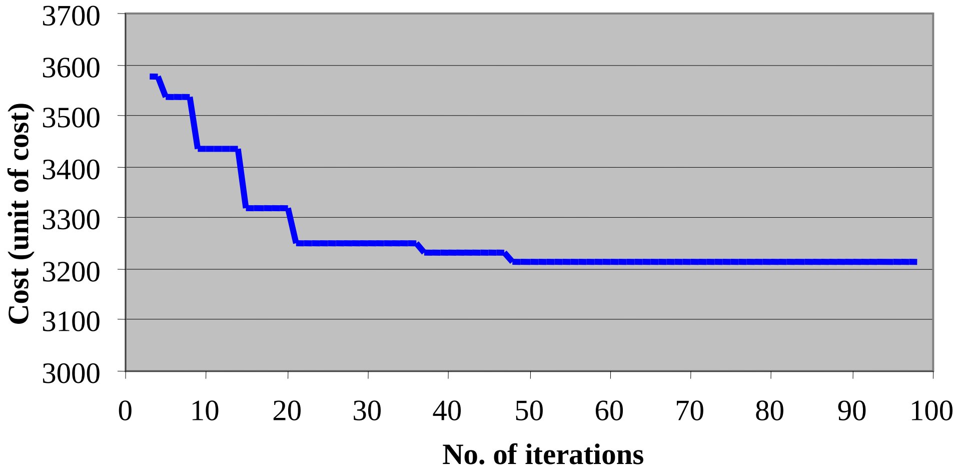


Figure 13: Optimum bid recorded at each SA iteration (U =0.6)

Table 6: Comparative results

Component	**MAS**		**Khoshnevis and Chen**		**Usher and Fernandes**		**Saygin and Kilic (before rescheduling)**		**Saygin and 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	Total Lead Time	Total Production Cost
ComA	***839***	***1888***	1227	2458	1268	2448	1430	2593	1227	2458
ComB	***545***	***1337***	***545***	***1337***	***545***	***1337***	1108	1505	***545***	***1337***
ComC	***1418***	***3215***	1980	3379	2416	3238	2216	3445	1980	3379