Iterative Multi-Agent Bidding and Co-ordination Based On Genetic Algorithm

M K LIM[1] and Z ZHANG[2]
[1, 2] School of Engineering & Computer Science,
University of Exeter, Harrison Building, North Park Road,
Exeter EX4 4QF, United Kingdom.
{M.K.Lim, D.Z.Zhang}@exeter.ac.uk

Abstract. This paper introduces a multi-agent system which integrates process planning and production scheduling, in order to increase the flexibility of manufacturing systems in coping with rapid changes in the market. This system consists of various autonomous agents which have the capability of communicating with each other and making decisions based on its knowledge. An iterative bidding mechanism is proposed to facilitate the process of job assignment to machines and handle the negotiation between agents. This mechanism enables optimum process plans and production schedules to be produced concurrently. A currency scheme is proposed to encourage or prohibit machine agents to put forward their bids. To deal with the optimisation problem, i.e. to what degree and how the currency values are adjusted in each iteration, a genetic algorithm (GA) approach is developed. This system is implemented in a Java platform. A test case is used and the results will be discussed in this paper.

1 Introduction

In recent years, manufacturing enterprises are facing greater pressures to competently cope with dynamic changes of demand patterns and product mix in the market due to global competition and shorter product life cycles. As a result, customer satisfaction has become tougher to achieve and manufacturing enterprises have to be agile and responsive towards market changes in order to stand firm in the competitive market. Manufacturing enterprises begin to pay their attention to the responsiveness issue in the manufacturing systems. The development of the concept of agile manufacturing was driven by the need to rapidly respond to dynamic changing customer requirements [1]. This concept requires manufacturing systems to be able to produce a large variety of products efficiently and be responsive and reconfigurable to accommodate dynamic changes in product mix, product designs and demand patterns in the market [2,3].

In manufacturing, conventional approach to production planning and control where process planning and production scheduling are carried out in two stages, has become an obstacle to achieving agile manufacturing (i.e. to obtain high responsiveness of manufacturing systems). This approach has brought up a number of problems within manufacturing systems which are elaborated as follows [4, 5]:

- Process planners assume unlimited resources on the shop floor. The most desirable processes and machines are always selected repeatedly with respect to the preference of the planners. When actual production takes place, constraints such as unavailability of machines may be encountered, making the process plans unfeasible.
- The throughput of orders in a workshop often suffers from disruptions caused by bottlenecks or non-availability of equipment and tools. This is due to the lack of communication between process planning and production scheduling.
- Scheduling takes place after the process planning stage and therefore it is restricted to fixed process plans, which are not to be altered.
- Single criteria optimisation is usually considered for both process planning and scheduling. In a real production environment (i.e. dynamic shop floor condition), more than one criterion needs to be considered simultaneously.

- The time delay between the planing phase and the execution phase causes problems. Due to the dynamic nature of a production environment, it is likely that when a design is ready to be manufactured the constraints that were used in producing the plan may have already changed.

For an effective and efficient agile manufacturing system, process planning and production scheduling are two important manufacturing functions that need to be integrated. In this study, an agent-based system is proposed to integrate these two manufacturing functions in order to minimise the problems above and most importantly, to maximise the responsiveness of manufacturing systems.

In recent years, multi-agent systems (MAS) have become an increasingly popular subject in manufacturing research. A MAS is a distributed artificial intelligence (DAI) system composed of a number of autonomous agents which are able to communicate and collaborate with each other to achieve common goals. The skill of negotiating with other agents to bid for jobs is one of the characteristics of autonomous agents. Each autonomous agent should have the following properties [6,7,8]:

- *Autonomy*: agent encapsulates some state of its environment and makes decisions about what to do based on this state;
- *Reactivity*: agent perceives its environment and respond to changes that occur in the environment;
- *Pro-activeness*: agent takes the initiatives to exhibit goal-directed behaviour; and
- *Social ability*: agent interacts with one another via an agent communication language and engages in social activities in order to achieve goals;

In the literature, the application of MAS can be found in various areas of manufacturing research. Krothapalli & Deshmukh [9] proposed inter-agent and intra-agent negotiation protocols by using several schemes such as shortest processing time, currency, and pre-emption, to improve the performance of decentralised manufacturing systems. Sluga, *et al*. [10] and Dornfeld, *et al*. [11] elaborated their ideas of MAS in process planning, while Teresedai and Ramesh [12] and Sousa and Ramos [13] applied the concept of MAS in dynamic scheduling in manufacturing systems. Lim and Zhang [14] developed an agent-based framework which utilises contract net protocols to enhance the negotiation mechanism of agents, to integrate process planning and production scheduling. The aim of the framework is to increase the responsiveness of manufacturing systems to cope with dynamic changes in market. For a highly uncertain and changing manufacturing environment, Lin and Solberg [15] proposed a framework for controlling the work flow in computer controlled manufacturing systems. The framework allows multiple steps negotiation between parts and resources.

A major issue in multi-agent systems research is the co-ordination and interaction of agents within a multi-agent system. Several interaction protocols can be found in the research literature. The most common interaction mechanism appearing in the literature is the contract net protocol. Smith [16] introduced the contract net protocol as a means of achieving co-ordination within a distributed problem-solving system. The main element of this protocol is the negotiation which is proceeded by the allocation of tasks to a set of problem solvers. Contract net based systems represent a concept that can be used to establish mechanisms of cooperation and interaction between agents. In the contract net, agents act as managers and contractors. A manager broadcasts the existence of a task and the remaining agents assess their ability to contribute to the solution of the task. Agents bid for the task based on their perceived suitability to the task. The manager then evaluates the bids and awards the task to the most appropriate agent. This agent then becomes the contractor. Contract net protocol has been used in such diverse areas as air traffic management [17], integrated process planning and scheduling system [18], electrical power grid management

[19] and computer task allocation [20,21]. The following is a summary of the steps involved in the contract net protocol:

a) *Task announcement*: A manager agent broadcasts an announcement to other agents or sub-agents to carry out a task;
b) *Bidding*: Those agents that receive the announcement and have the appropriate knowledge to help send bids to the manager agent;
c) *Awarding*: The manager agent awards the task to the most appropriate agent; and
d) *Expediting*: The agent performs the task that has been awarded.

This paper introduces a multi-agent system which integrates process planning and production scheduling, in order to increase the flexibility of manufacturing systems in coping with rapid changes in the market and dealing with internal uncertainties such as machine breakdown or resources shortage. This system consists of various agents which are capable of communicating and co-ordinating with each other to perform tasks related to process planning, optimisation and scheduling, making decisions based on their knowledge and if necessary on information provided by other agents. Machine agents which represent the machines play an important role in the system in that they negotiate with each other to bid for jobs. An iterative bidding mechanism is proposed to facilitate the process of job assignment to machines and handle the negotiation between agents. This mechanism enables optimum process plans and production schedules to be produced concurrently, so that dynamic changes in the market can be coped with at a minimum cost, and the utilisation of manufacturing resources can also be optimised. In addition, a currency scheme with currency-like metrics is proposed to encourage or prohibit machine agents to put forward their bids for the jobs announced. The values of the metrics are adjusted iteratively so as to obtain an integrated plan and schedule which results in the minimum total production cost while satisfying products due dates. To deal with the optimisation problem, i.e. to what degree and how the currencies should be adjusted in each iteration, a genetic algorithm (GA) approach is developed. Java programming language is used in this study to implement the system. In order to verify the implementation, a test case is used and the results will be discussed in this paper.

2 System Architecture and Descriptions

The multi-agent system developed in this study consists of an order handling agent, a component agent, various machine agents, a tool agent, and a material handling agent. Fig. 1 depicts the architecture of this system. The architecture can be described in three sections: demand related activities, component related activities, and shop floor activities. As shown in Fig. 1, the top left section is demand related activities. When there is an order input to the system, order handling agent interprets the order and inputs the order into a Master Production Schedule (MPS). MPS along with other two inputs: the inventory records and the BOM which are derived from a CAD system, are used to generate a material requirements plan (MRP). A component MPS is derived from the MRP and it consists of all the components of products to be manufactured over a period of time as well as the quantities and due dates. To the top right section of Fig. 1, the activities are related to the component issue. Component data files are derived from a CAD system. These data files record the technical information for each component, which include component ID, materials to be used, materials hardness, a list of geometrical features as well as the feature relationships (i.e. the sequence of ordered features where the first order feature must be manufactured before the second order feature and so on), and estimated removal volume, and tolerance and surface quality requirements. Manufacturing database records possible processes which can be provided on the shop floor, to produce different types of features. Machine agents will always update this database to ensure the record is kept up to date. Component agent retrieves information from the component MPS regarding what components are to be produced and obtains technical data of the components from the component data files. Based on data from manufacturing database, component agent

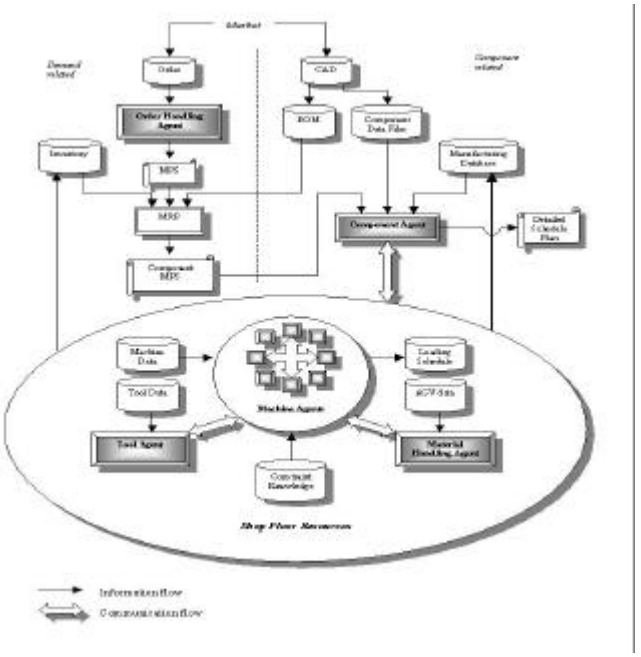


Fig. 1. The architecture of the multi-agent system

recommends suitable processes to produce the features of each component. At this stage, alternative process sequences can be identified. The lower section of Fig. 1 is related to shop floor activities, which consist of various machine agents, a tool agent and a material handling agent, where agents represent machines, tools, and material handling devices respectively. The iterative bidding process will take place between the component agent and the machine agents.

The component agent announces one component at a time to all machine agents. The contents of announcement include the ordered feature sequences and the recommended processes for each feature in the component. In addition, the announcement also includes a currency scheme where virtual currency values are assigned to all features in the component. Agents put forward their bids for the jobs announced if they make a virtual profit based on the currency values and vice versa. Machine agents which represent machines able to produce the first feature, will come forward to become "leaders" whose responsibility is to group other machine agents for the remaining features in the component. Process planning and production scheduling are integrated at a detailed level at this stage. The leaders announce the second feature along with the allocated currency to all machine agents, including the leaders themselves and other leaders. Machine agents which represent machines that are able to provide the processes to produce the second feature will come forward to bid for the job. The bidding process is illustrated in Fig. 2. Machine agents will first check with constraint knowledge to make sure the process sequence constraint is not violated (i.e. process provided by the machine can be carried out on the component after the preceding process). Machine agents will then ask tool agent to recommend a tool to be used to produce the feature. Tool agent checks its tool list and based on the component material and its hardness, tolerance and surface quality required, as well as the tool availability at that period of time, the tool which is available and gives the least cost while all the requirements are met will be selected. Tool agent will provide the machine agents with information on the selected tool ID and tool cost per unit of time. Machine agents will also request material handling agent to provide the information on the travelling cost per unit of time and travelling time per unit of distance from the preceding machine. Machine agents may optimise and reschedule its job queue (of those components which

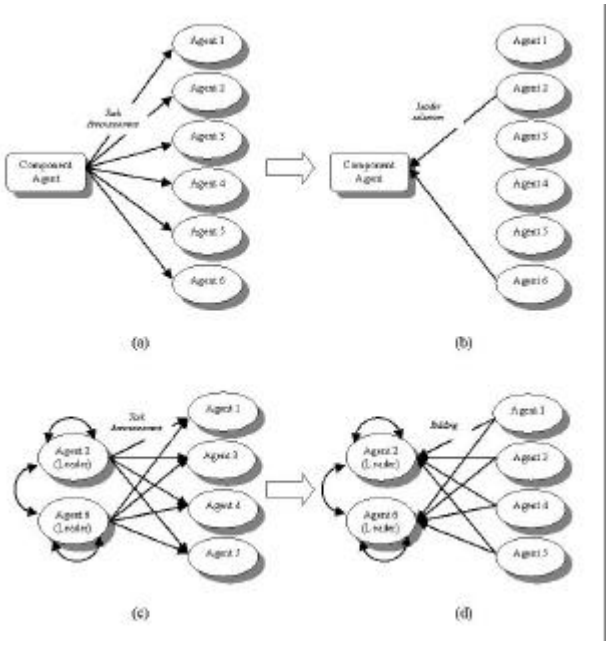


Fig. 2. Iterative agent bidding mechanism: (a) Component agent makes jobs announcement, (b) Agents come forward to become leaders, (c) Leaders announce the following jobs to other agents, (d) Agents put forward bids to the leaders.

have been planned and scheduled) locally by swapping jobs if other components due dates are not violated, to produce optional bids.

By swapping jobs in the job queue, tool changes could be avoided which may reduce setup times. Current machine loading, material handling system, and tool selection are the issues taken into consideration when machine agents work out their bids which consist of manufacturing cost and lead time. The individual machine manufacturing cost is obtained as:

$$C_{mf} = C_{hand} + C_{setup} + C_{tool} + C_{mc} + C_{ms} \quad (1)$$

where C_{hand} = handling cost from the location of preceding machine (in unit of cost),
C_{setup} = setup cost (in unit of cost),
C_{tool} = tool cost (in unit of cost),
C_{mc} = machining cost (in unit of cost), and
C_{ms} = miscellaneous cost due to jobs swapping (in unit of cost).

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

The individual machine lead time is worked out as:

$$T_{lead} = T_{trav} + T_{wt} + T_{setup} + T_{tc} + T_{mc} \quad (2)$$

where T_{trav} = travelling time from the location of preceding machine (in unit of time),
T_{wt} = waiting time at buffer (in unit of time),
T_{setup} = setup time (in unit of time),
T_{tc} = tool change time (in unit of time), and
T_{mc} = machining time (in unit of time).

Each machine agent tries to earn more profit,

$$P = CU_i - C_{mf} \tag{3}$$

where CU_i = currency allocated to feature i (in unit of cost), and
C_{mf} = individual machine manufacturing cost (in unit of cost).

If the profit for a bid option is above a set threshold δ (i.e. $P \geq \delta$), the bid will be put forward to the leader. A machine agent may put forward more than one bid as long as the profits of the bids are above the set threshold. The threshold varies from one machine to another. The leader will then select a bid which provides the shortest lead time from all bids put forward by machine agents. This grouping process continues until all the processes to produce the features of the component have been scheduled to the most appropriate machines.

At this stage, each leader sums up the manufacturing costs and lead times of all selected machines and forwards the bid as a machine group to component agent. Component agent evaluates the bids based on time and cost. If the bids do not satisfy the due date, the currency allocated will be increased to encourage more bids to come forward in the next round of the bidding process. If the bids satisfy the due date, currency in the next iteration will be decreased to discourage higher cost bids from machine agents to come forward and hence the total production cost can be minimised. Bidding process takes place iteratively until the total production cost is minimised. When the optimum or near optimum solution is obtained, component agent will update the detailed schedule plan. The machine agents in the awarded machine group will confirm with component agent to commit to the job awarded by updating their loading schedule.

3 Genetic Algorithm Approach

Many modern heuristic search methods such as genetic algorithm (GA), simulated annealing (SA), tabu search (TS), and neural networks (NN), have been applied to various manufacturing optimisation problems. Zhou, *et al.* [22] and Al-Hakim [23] adopted a GA approach to deal with production scheduling problems. Suresh, *et al* [24] and Rajasekharan, *et al.* [25] developed GA approaches in finding optimal facility layout configuration in flexible manufacturing systems.

In the iterative agent bidding mechanism, the currency allocated to each component is adjusted according to the performance of bids (in terms of production cost and lead time) put forward by the leaders, in order to obtain a bid which carries the minimum total production cost while satisfying due dates. To deal with the optimisation problem, i.e. to what degree and how the currencies should be upward or downward adjusted, a GA approach is developed. The GA approach is summarised as below:

Step1: Gene coding: A population of chromosomes, *POP_SIZE* and a generation number, *GEN* are determined. The genes in each chromosome represent the currencies for all the features in a component, where CU_{xy} represents the currency for feature y in chromosome x.
Step2: Evaluation of fitness functions (Announcement to machine agents): Component agent announces all the chromosomes to machine agents. Based on the currencies, machine agents bid for the jobs based on their performances. The two fitness functions are:

1) Due date, $d \geq \sum_{i=1}^{k} t_i$ (4)

2) Production cost, $min \sum_{i=1}^{k} C_i$ (5)

where k is the number of features in a particular component, t_i is the time taken to produce feature i, and C_i is the production cost of producing feature i. The chromosome with the least production cost and satisfies product due date is recorded as the best solution found at this iteration.

Step3: Selection of chromosomes: In this approach, all the chromosomes are selected and have equivalent opportunity for crossover and mutation.

Step4: Crossover: Position-based crossover operator is used. There are three types of position-crossover operators used in this approach: point-to-point crossover, single cut-off point crossover, and two cut-off points crossover. Every chromosome will be assigned a random probability. If the random probability is equal or less than the probability of crossover (P_C), chromosome of which will be in the mating pool. Every chromosome in the mating pool will be paired up and crossover will be carried out. A chromosome may be mated twice if the total number of chromosomes in the mating pool is an odd number. Those chromosomes whose random probability is greater than P_C, will remain unaltered. If the two offspring obtained are identical to the parent chromosomes, alternate crossover operator will be selected. After trying the three crossover operators and if no new unique offspring is obtained, two new parent chromosomes will be reselected from the population and crossover will again be carried out. This process carries on until two unique offspring chromosomes which are different from the parent chromosomes, are obtained. The new offspring chromosomes selected will then replace their parent chromosomes if the fitness functions obtained are better than the ones of their parent chromosomes.

Point-to-point crossover:

Parent 1 = [CU $_{11}$ CU $_{12}$ CU $_{13}$ CU $_{14}$] Offspring 1 = [CU $_{11}$ CU $_{22}$ CU $_{13}$ CU $_{24}$]
Parent 2 = [CU $_{21}$ CU $_{22}$ CU $_{23}$ CU $_{24}$] ⇨ Offspring 2 = [CU $_{21}$ CU $_{12}$ CU $_{23}$ CU $_{14}$]

Single cut-off point crossover:

Parent 1 = [CU $_{11}$ CU $_{12}$ CU $_{13}$ CU $_{14}$] Offspring 1 = [CU $_{11}$ CU $_{22}$ CU $_{23}$ CU $_{24}$]
Parent 2 = [CU $_{21}$ CU $_{22}$ CU $_{23}$ CU $_{24}$] ⇨ Offspring 2 = [CU $_{21}$ CU $_{12}$ CU $_{13}$ CU $_{14}$]

Two cut-off points crossover:

Parent 1 = [CU $_{11}$ CU $_{12}$ CU $_{13}$ CU $_{14}$] Offspring 1 = [CU $_{11}$ CU $_{22}$ CU $_{23}$ CU $_{14}$]
Parent 2 = [CU $_{21}$ CU $_{22}$ CU $_{23}$ CU $_{24}$] ⇨ Offspring 2 = [CU $_{21}$ CU $_{12}$ CU $_{13}$ CU $_{24}$]

Step5: Mutation: In the mutation process, genes within a chromosome will be mutated. The mutation probability (P_m) determines whether a chromosome will be mutated. A random probability is assigned to each chromosome in the population. The chromosome, of which the random probability is equal or less than P_m, will be mutated. The mutation operator used is to increase/decrease the currency with mutation factor, μ, which is randomly selected in the range of $[Z_1, Z_2]$ (where Z_1 is the lower limit of the mutation factor and Z_2 is the upper limit of the mutation factor). Again, a random number is selected in the range of [1, number of genes] to represent the feature of which the currency is to be mutated.



Parent 1 = [CU $_{11}$ CU $_{12}$ CU $_{13}$ CU $_{14}$] Offspring 1 = [CU $_{11}$ *CU $_{12}$ CU $_{13}$ CU $_{24}$]

Step6: Re-evaluation of fitness functions (Re-announcement to machine agents): The fitness functions of each offspring chromosome in the new population will be re-evaluated by re-announcing the offspring chromosomes to machine agents. The chromosome with the least production cost and satisfies product due date is recorded as the best solution found at this iteration. If either one from the pair of the offspring chromosomes obtains better performance in fitness functions than their parent chromosomes, the pair of offspring chromosomes will replace their parent chromosomes. Step 3 to Step 6 is repeated until the condition is satisfied.

4 Test Case and Results

The system implementation is written with Java programming language. There are 25 Java classes constructed. *SystemInterface* class provides an interface to allow users to execute the system and update or create new data. A text area is included in the interface to display the activities of agents at due time. *BaseAgent* class provides the fundamental agent behaviour and implements the *Runnable* interface which requires other agent classes which extend *BaseAgent*, to provide a *run()* method that is served as the body of a thread. This is the mechanism to give the agents autonomy. *AgentMessage*, *AgentEvent*, and *AgentEventListener* classes are developed to facilitate message syntax and format and message passing protocol between agents. An administrator agent is created to deal with the message passing between agents and it maintains a list of active agents which have been registered with it.

In order to verify the implementation, a simple test case is used. In this test case, there are 10 machines, i.e. 3 milling machines, 3 drilling machines, and 4 lathe machines. Each of the machines has its own job loading (i.e. job queue). A new product with 4 features of a quantity of 250 units is introduced to the system. In the GA approach, the population size is 10 and generation

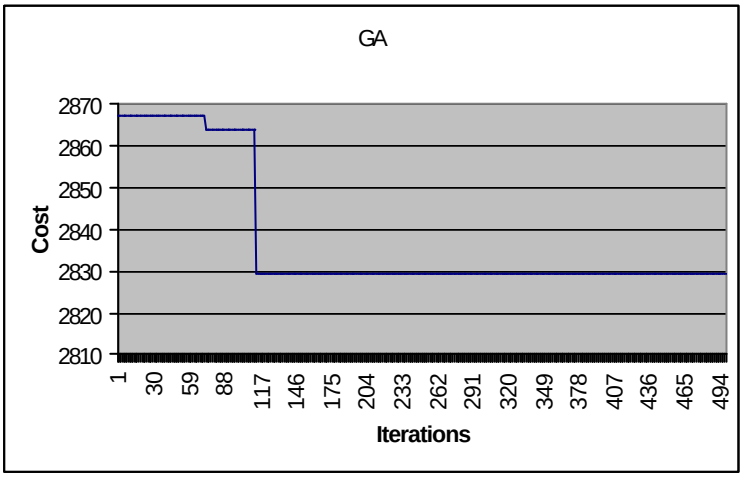


Fig. 3. Performance of the GA approach

is 500. The crossover and mutation probabilities are set at 0.99 and 0.35 respectively. After 500 iterations, the best cost obtained is 2829 units of cost. Fig. 3 illustrates the best result obtained at

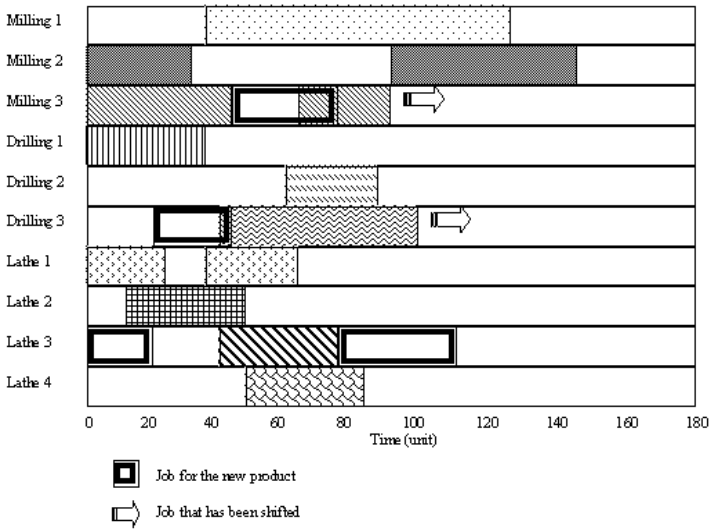


Fig. 4. Job queues of all machines

each iteration and Fig. 4 depicts the job queues of all machines when an optimum solution is obtained. As shown in Fig. 4, the previously scheduled jobs in Milling 3 and Drilling 3 machines have been shifted in the optimum solution.

5 Conclusions

Today, manufacturing enterprises require agility in manufacturing systems in order to cope with increasingly dynamic changes of products requirements in the competitive market. In this study, a multi-agent system is developed to integrate dynamic process planning and scheduling, to increase the responsiveness of manufacturing systems. This system not only integrates dynamic process planning and scheduling but also optimises machine utilisation and provides a platform to assess the reconfiguration of manufacturing systems. The proposed iterative bidding mechanism for MAS enables optimum process plan and production schedule to be produced concurrently, so that dynamic changes in the market can be coped with at a minimum cost. A currency scheme is introduced for agent co-ordination where currency-like metrics are used to encourage or prohibit machine agents to put forward their bids for jobs. A genetic algorithm approach has been proposed to iteratively adjust currency metrics in order to optimise the integrated process plans and schedules. The implementation has been carried out with Java programming language. A test case is used and the results shown that with the currency adjustment at each bidding iteration, the total production cost is gradually minimised. In comparison to others' work in the application of MAS in manufacturing, the approach developed in this study facilitates the interaction between agents with an iterative bidding mechanism. The currency scheme where the currency values are adjusted iteratively based on the performance of bids, drives the behaviour of agents in the bidding process.

References

1. Kidd, P.T.: Agile Manufacturing: A Strategy for the 21st Century. IEE Manufacturing Division: Agile Manufacturing (1996)
2. Kidd, P.T.: Agile Manufacturing – Forging New Frontiers. Addison-Wesley, England (1994)
3. McMullen, T.B.: Theory of Constraints: The Infrastructure of Agile Manufacturing. APICS – the Educational Society of Resource Management (1996) 167-171

4. Kempenaers, J., Pinte, J., Detand, J., Kruth, J.P.: A Collaborative Process Planning and Scheduling System. Advances in Engineering Software **25**(1) 1996 3-8
5. Zhang, H.: IPPM – A Prototype to Integrate Process Planning and Job Shop Scheduling Functions. CIRP **42**(1) (1993) 513-518
6. Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley, England (1999)
7. Ouelhadj, D., Hanachi, C., Bouzouia, B.: Multi-Agent System for Dynamic Scheduling and Control in Manufacturing Cells. IEEE Int. Conf. on Robotics and Automation **3** (1998) 2128-2133
8. Wooldridge, W.: Agent-Based Software Engineering. IEE Proceedings on Software Engineering **44**(1) (1997) 26-37
9. Krothapalli, N.K.C., Deshmukh, A.V.: Design of Negotiation Protocols for Multi-Agent Manufacturing Systems. Int. J. of Production Research **37**(7) (1999) 1601-1624
10. Sluga, A., Butala, P., Bervar, G.: A Multi-Agent Approach to Process Planning and Fabrication in Distributed Manufacturing. Computers and Industrial Engineering **35**(3-4) (1998) 455-458
11. Dornfeld, D., Wright, P.K., Wang, F.C., Sheng, P., Stori, J., Sundararajan, V., Krishnan, N., Chu, C.H.: Multi-Agent Process Planning for a Networked Machine Service. Technical Paper – Society of Manufacturing Engineers MS99-175 (1999)
12. Teredesai, T., Ramesh, V.C.: Multi-Agent Mixed Initiative System for Real-Time Scheduling. Proceedings of the IEEE Int. Conf. on Systems, Man and Cybernetics **1** (1998) 439-444
13. Sousa, P., Ramos, C.: A Dynamic Scheduling Holon for Manufacturing Orders. J. of Intelligent Manufacturing **9** (1998) 107-112
14. Lim, M.K., Zhang, Z.: An Integrated Agent-Based Approach for Responsive Control of Manufacturing Resources. The 27th Int. Conf. on Computers and Industrial Engineering ISBN 7-900043-38-1/TP.38 (2000) No.55
15. Lin, Y.J., Solberg, J.: Integrated Shop Floor Control Using Autonomous Agents. IIE Transactions **24**(3) (1992) 57-71
16. Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers C-2912 (1980) 1104-1113
17. Wangermann, J.P., Stengel, R.F.: Principled Negotiation Between Intelligent Agents: A Model for Air Traffic Management. Artificial Intelligence in Engineering **12** (1998) 177-187
18. Lim, M.K., Zhang, Z.: Integrated Manufacturing Systems Control Using a Multi-Agent System. Proceedings of 33rd Int. MATADOR Conf. ISBN 1-85233-323-5 (2000) 9-14
19. Krishna, V., Ramesh, V.C.: Intelligent Agents for Negotiations in Market Games, Part1: Model. IEEE Transactions on Power Systems **13** (1998) 1103-1108
20. Stankovic, J.A., Sidhu, I.S.: An Adaptive Bidding Algorithm for Process Clusters and Distributed Groups. Int. Conf. on Distributed Computer Systems (1984) 49-59
21. Ferfuson, D., Yemini, Y., Nikolaou, C.: Microeconomic Algorithms for Load Balancing in Distributed Computer Systems. IEEE Int. Conf. on Distributed Computer Systems (1988) 491-499
22. Zhou, H., Feng, Y., Han, L.: The Hybrid Heuristic Genetic Algorithm for Job Shop Scheduling. Computers and Industrial Engineering **40** (2001) 191-200
23. Al-Hakim, L.: An Analogue Genetic Algorithm for Solving Job Shop Scheduling Problems. Int. J. of Production Research **39**(7) (2001) 1537-1548
24. Suresh, G., Vinod, V.V., Sahu, S.: A Genetic Algorithm for Facility Layout. Int. J. of Production Research **33**(12) (1995) 3411-3423
25. Rajasekharan, M., Peters, B.A., Yang, T.: A Genetic Algorithm for Facility Layout Design in Flexible Manufacturing Systems. Int. J. of Production Research **36**(1) (1998) 95-110