EXTENDED ABSTRACT

Negotiation in Multi–Agent Manufacturing Systems*

Naga K.C. Krothapalli and Abhijit V. Deshmukh†
Department of Mechanical & Industrial Engineering
University of Massachusetts
Amherst, MA 01003

1 Manufacturing System Control

Hierarchical or central control is the prevalent control architecture in manufacturing systems. In hierarchical control, a central controller plans the sequence of operations for a group of machines and parts to be processed, either *a priori* or on–line. Although hierarchical control architecture has been widely used, it has limitations (Deshmukh et al., 1993). The primary drawback of hierarchically controlled manufacturing systems is the difficulty of modifying these systems. Modifying the configuration of hierarchically controlled manufacturing systems is expensive and time consuming as it involves expensive software rewriting. Another disadvantage relates to the deterioration of the quality of information as it flows up and down the hierarchy. The potential single point of failure at the central controller poses another significant problem. The inclusion of fault tolerance methods results in increased system complexity (Duffie, 1990). Furthermore, hierarchical systems require huge databases, which can result in data retrieval delays and data consistency maintenance problems.

Decentralized control architecture has been used as an alternative to the hierarchical control and its limitations. Decentralized systems, also referred to as multi–agent manufacturing systems, are composed of autonomous components (parts, machines, etc) which communicate, and potentially negotiate, with other components to perform tasks based on available local information without directions from a central controller. Thus, the central

*Based on manuscript accepted for publication in the *International Journal of Production Research*. The authors acknowledge the support provided by the National Science Foundation under Grant No. DMI-9696017.

†Corresponding author. Phone: (413) 545–1615 Fax: (413) 545–1027 email: `deshmukh@ecs.umass.edu`

controller is replaced by several small controllers, one for each individual component of the manufacturing system. Each controller or *agent* is assigned a set of protocols which govern the operations of the manufacturing entity. An agent, in this context, is a software entity that has a knowledge base, an inference mechanism, and an explicit model of the problem to solve (Grossner et al., 1995). Overall performance of the manufacturing system depends on the interactions among the agents and the quality of data used to make these decisions. The design of negotiation protocols between agents is thus critical for proper functioning of the decentralized control.

Past research in multi–agent manufacturing systems has drawn heavily from the fields of computer science and distributed artificial intelligence . The contract net protocol for decentralized task allocation is one of the earliest multi–agent paradigms in literature (Smith, 1980; Smith and Davis, 1981). Recently, some researchers have investigated the effects of different communication rules for agents on the overall system performance (Sandholm, 1993; Cohen et al., 1994; Sandholm and Lesser, 1995). Several researchers have also applied the multi–agent ideas to the field of mobile robots (Parker, 1993; Lueth et al., 1995; Parker, 1994). Researchers in computer science have approached the problem of distributed computations by applying a collection of processors to tackle a single problem (Davis and Smith, 1983). In manufacturing systems literature, (Lewis, 1981) investigated the use of the data flow architectures for controlling part/job flow in a computer–controlled manufacturing system. A distributed manufacturing system was proposed in (Shaw, 1987), which uses a network–wide bidding scheme. (Maley et al., 1988; Maley, 1988) developed a distributed manufacturing control system, called CADENCE. (Upton, 1988) presents a hybrid system model in which parts select the machine which would take the least amount of processing time. A market–driven contract net was proposed by (Baker, 1996) which provides direct negotiation between customers and shop floor agents associated with resources. (Lin and Solberg, 1992) develops a market–like system which uses currency in decision making in order to achieve optimal performance. (Yang, 1994) proposes a system which allows the preemption of the parts based on priority rankings. (Prabhu and Duffie, 1996; Duffie and Prabhu, 1996) use a closed–loop feedback structure for controlling the arrival of parts at processing machines in a distributed manufacturing system. (McDonnell and Joshi, 1997) and (Ben-Arieh and Chopra, 1997) propose game–theoretic approaches for shop floor control in distributed systems. The prior work in multi–agent manufacturing systems, and specifically the research using currency for negotiations, does not consider the nonlinear relationship between currency and time, especially as the part nears its due date. Also, the interaction between currency allocation and preemption is not explicitly considered in a common framework.

The research described in this paper proposes decentralized control protocols for discrete part manufacturing systems which incorporate both the currency and the preemption schemes within a common framework. Currency functions are used to help the agents to meet their individual objectives, as well as achieve the overall objectives of the system. The preemption scheme is used to expedite the processing of parts based on their due dates. The control schemes were implemented using Swarm, an object–oriented simulation platform. Different control protocols for agents were tested in large flexible job shop environments. The system performance was observed under different operating conditions.

2 Multi–Agent Manufacturing Systems

In this research, parts and machines are modeled as agents with communication capabilities. The material handling devices are assumed to be always available and transportation times are included in machining times. A part agent or a machine agent is programmed to meet its objective based on the local available data. The local data is gathered by communicating with other agents and also from local sensors. The primary objective of a part is to finish all the processing before the due date, while that of a machine is to maximize the utilization rate. Both parts and machines are governed by a set of rules, which help them to realize these objectives. The main advantage of this architecture is that agents do not have to rely on a particular component (central controller) to execute their instructions. Failure of one component does not halt the operation of the entire manufacturing system. The following paragraphs outline the construction of machine and part agents.

2.1 Machine Agent

A machine agent is responsible for all the decisions made by the machine related to the selection of parts for processing. All the attributes of the machine, such as its identification, location, type, buffer limit, speed factor, success rate and the parts waiting in buffer, are made available to the machine agent. Machines are grouped together based on their processing capabilities. A part with a particular process requirement can be processed by any one of the machines in the group which can perform the given operation. The speed factor is defined as the ratio of the processing speed of a machine with respect to a base machine, which is the slowest machine in the group. The buffer limit is the maximum number of parts that be placed in queue of a machine. If the buffer limit is reached, the machine can no longer accept parts. The success rate of a machine is defined as the ratio of the number of parts that select the machine out of all the parts that request bids. The success rate is used in estimating the completion time for parts. If β_W is the number of bids won in last hour and β_T is the total bids submitted by the machine in that time, then the success rate $\psi = \frac{\beta_W}{\beta_T}$. The queueing time is estimated using the average processing requirements for all the parts waiting in the buffer.

2.2 Part Agent

Part agents communicate with machine agents to complete processing done before due dates. Part agents are designed with attributes such as identification number, part type, location, routing table, nominal processing times and due dates. Each part has a unique identification number which is used in the broadcasts to machine agents. Parts are grouped into different classes based on the similarity of processing requirements. Parts in a given group have the same routing requirements. The part routing chart contains the sequence of operations to be performed and the processing requirements. Each part enters the manufacturing system with a due date. The objective of every part is to complete processing before its due date. Nominal time is defined as an estimate of processing time taken by the base machine.

2.3 Other Agents

Several other agents, such as material handling agents and information agents, also play an important role in the operation of multi–agent manufacturing systems. Material handling agents wait for part agents to request for transportation and provide transportation to the requested location. Information agents serve as data brokers and provide the necessary information when requested by the parts or machines. This work does not model the support agents explicitly. However, the proposed protocols can be extended with minimal modifications to include auxiliary agents.

2.4 Negotiation Protocols

Negotiation protocols are rules used by machine and part agents to make decisions on the shop floor. This paper compares different interaction schemes for agent–based manufacturing systems. The negotiation protocols studied can be categorized into three control schemes: 1) shortest processing time (SPT) scheme 2) currency scheme, and 3) preemption scheme. The performance of these agent–based protocols is compared with a hierarchical system model. The protocols are discussed in detail in the following sections.

3 Shortest Processing Time (SPT) Scheme

In the SPT scheme, a part selects the machine with the shortest expected processing time in the system. In this scheme, the parts request for bids and the machine tools submit the bids. A part makes a decision based on the estimated completion time submitted by the machines and selects the machine with the least estimated processing time. The control protocols for part and machine agents are explained in the following paragraphs.

3.1 Part Control Protocol

When parts enter the manufacturing system, they request bids from the machines. The parts wait for a fixed time after which they evaluate all the bids and select the machine with the least expected waiting time. In this control scheme, the parts do not consider their criticality while making decisions. Once the selection is made, parts send their acceptance to the selected machine and request for transportation as soon as the machine confirms the bid. The parts queue in the buffer of the selected machine till they can be machined. During this period, each part maintains contact with the selected machine. If it looses contact with the machine, the part assumes that the destination machine has failed and request for bids again. The part repeats this procedure until all the processing is completed. This procedure makes the control scheme robust to machine failures.

3.2 Machine Control Protocol

The machine receives requests for bids from the parts entering the system. When a request for a bid has a reference to a NC program which the machine can process, then the machine submits a bid with the expected completion time, provided the input buffer is not full. The expected completion time ω is given by

$$\omega = Q + \frac{Q}{b}\,\psi\,\beta_K + \frac{m_i}{S} \tag{1}$$

where Q is the expected queue time (this takes into account all the parts in the buffer), b is the number of parts in the buffer, β_K is number of bids submitted in the last K time units, K is the time a part waits, before evaluating bids, ψ is the success rate of the machine, m_i is the nominal processing time for the ith operation of the parts, and S is the speed factor. Thus, ω is the summation of the expected machining time of the committed parts (Q), the machining time of expected commitments from currently negotiated parts ($\frac{Q}{b}\psi\beta_K$) and the expected machining time of the requesting part ($\frac{m_i}{S}$).

When a part sends a message requesting a confirmation of a machine bid, the machine once again checks the buffer limit. If the buffer capacity has not been exceeded then the machine sends a confirmation to the part. The machine maintains contact with all the parts that are headed for the input buffer by sending periodic messages. If a machine recognizes any problems or fails, it stops sending messages to the parts which have selected the machine. The machine processes a part when the part arrives in front of the queue.

4 Currency Scheme

In this control scheme, a currency like metric is used to direct agents in a manufacturing system. This control scheme is based on the market paradigm. If a customer wants to acquire a service offered by an organization, a currency transaction is involved. Consider the machine as the service provider and the part as the customer. The part enters the system with some currency, which depends on the priority of the part. The machine computes the charge based on the expected completion time and also the recent bid success rate. The part then converts the expected waiting time into equivalent currency, which is based on the due date and adds the machine charge to it. The part only considers the bids which have a completion time less than the due date and a charge under its currency limit. The part selects bids which have the lowest overall charge. This approach differs from the currency schemes proposed in (Lin and Solberg, 1992) by providing currency functions which are based on the criticality or slack available to the parts and the utilization of the machines. This provides a self–correcting method for generating stable bids under different operating conditions. The market–driven contract net presented in (Baker, 1991) provides the customers a set of product delivery times and the associated costs, of which the customers select the ones which satisfies their constraints. The currency approach differs from the market–driven model by including fault tolerance and by avoiding the requirement of a schedule. Further, the approach presented in this paper dynamically allocates faster resources to critical parts, thereby increasing the efficiency of the system. The following paragraphs discuss the part and machine agent control protocols in detail.

4.1 Part Control Protocol

A part enters the manufacturing system with some currency. The amount of currency depends on the due date and the nominal processing time. The part with a lower ratio of due date over nominal processing time carries more currency per unit nominal processing time than the part with a higher ratio of due date over nominal processing time. The part requests bids from machines which can process the NC program in its routing table. After a fixed time, the part evaluates the bids. If there are no bids received, the part broadcasts the request again. If there are n processes in the routing of a part, then due dates for each process are computed by,

$$d_i = D\left(\frac{m_i}{M}\right) \qquad \forall i = 1, \ldots, n \tag{2}$$

where d_i is the due date for the ith process, D is the due date for the part, m_i is the nominal processing time for the ith process, M is the total nominal time of all processes.

Similarly, currency spending limits are set for each processing stage. If C is the total currency allocated to a part, the currency limit c_i for the ith process is given by

$$c_i = C\left(\frac{m_i}{M}\right) \qquad \forall i = 1, \ldots, n \tag{3}$$

The part converts expected waiting time into equivalent currency based on the waiting time and the shape factor.

$$C_p = 2.0\ tan^{-1}(e^{(A\omega)}) \qquad where\ \ 0.01 \leq A \leq 0.1 \tag{4}$$

where C_p is the equivalent currency per unit time, A is a shape parameter, ω is the expected waiting time submitted by the machine. C_p is multiplied by ω to get the equivalent currency value of the waiting time. The currency function, C_p, for different shape factors is shown in Figure 1.

The shape factor A determines the curvature of the currency function. A is a function of the ratio of d_i/m_i, as shown below.

$$A = 0.11 - \frac{1 + tanh\left(\frac{d_i}{m_i} - 1\right)}{20} \tag{5}$$

The lower limit for A is set at 0.01 where the upper limit is set at 0.1. The shape factor function is shown in Figure 2. The shape factor and the equivalent currency function together determine the different levels of criticality as the part nears the due date. The currency function and the shape function were chosen to reflect the asymmetric value of time when comparing earliness and lateness. It is much more expensive to have a tardy job, than a early job. Such value functions are used to represent human decision making under risk, as proposed in (Kahneman and Tversky, 1979). Although the specific functions and parameters were chosen arbitrarily, the use of such functions represents a novel approach. Moreover, the general behavior of these functions reflects the decision process used by expediters and schedulers on the shop floor.

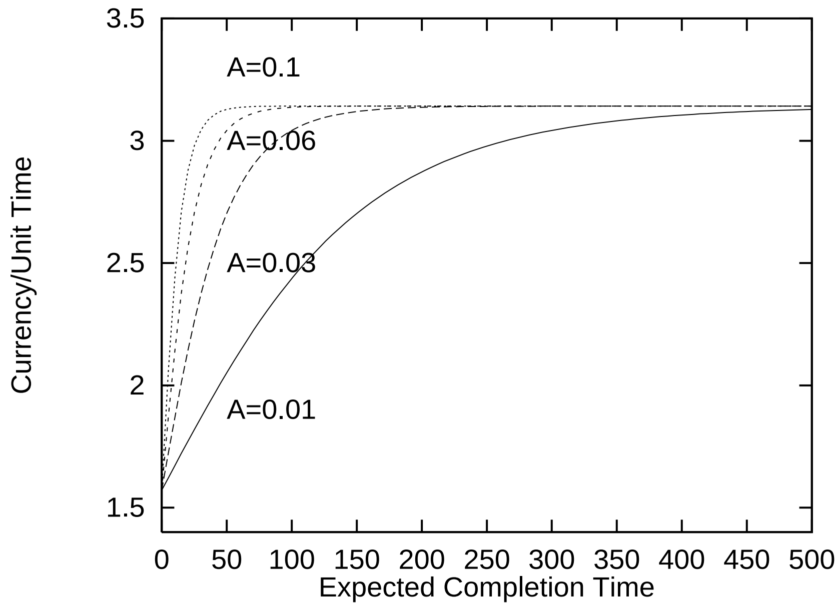

Figure 1: The currency function for parts

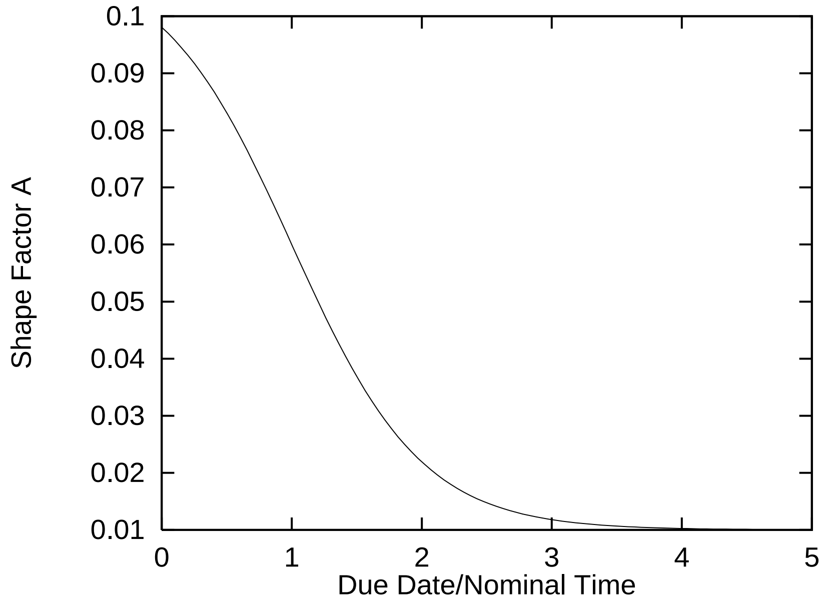

Figure 2: The shape factor for parts

4.2 Machine Control Protocol

The machine waits for requests for bids from the parts. If the machine can process the requested NC program and the buffer count is below the limit, then the machine computes the estimated completion time and the charge for processing. The completion time is computed by Equation 1, similar to the SPT control scheme. The cost of machining is computed using a currency function, which depends on prior success rate and is given by

$$C_m = 4.7 - (2.0tan^{-1}(e^{(0.11-B)\omega})) \qquad where\ \ 0.01 \leq B \leq 0.1 \tag{6}$$

where C_m is the charge per unit time for processing, B is the shape factor which depends on success rate, and ω is the processing time. The machine's currency function of machine is shown in Figure 3.

The shape factor B is given by

$$B = 0.1 - \frac{1 + tanh(1 - 0.1\psi)}{20} \tag{7}$$

where ψ is the success rate of the machine. The shape factor function is shown in Figure 4.

If a part selects the machine, then the machine confirms the selection if the buffer count is below the limit. The machine periodically sends contact messages to all the parts placed in the queue. If the machine fails, it does not communicate with the parts or detect requests for bids. The machine processes the part at the head of the queue.

5 Preemption Scheme

This control scheme facilitates the preemption of parts that are of lower priority based on the due date criticality. Both machine and part agents use currency as a means of valuing preemption. This control scheme achieves performance improvements by incorporating priorities in the processing of parts. Parts with closer due dates preempt parts that are not required to be processed immediately. This decreases the average tardiness of the parts in the system.

5.1 Part Control Protocol

A part enters the manufacturing system with currency which is used to pay for machining and preemption. The part requests the machines that can process it to submit bids. The

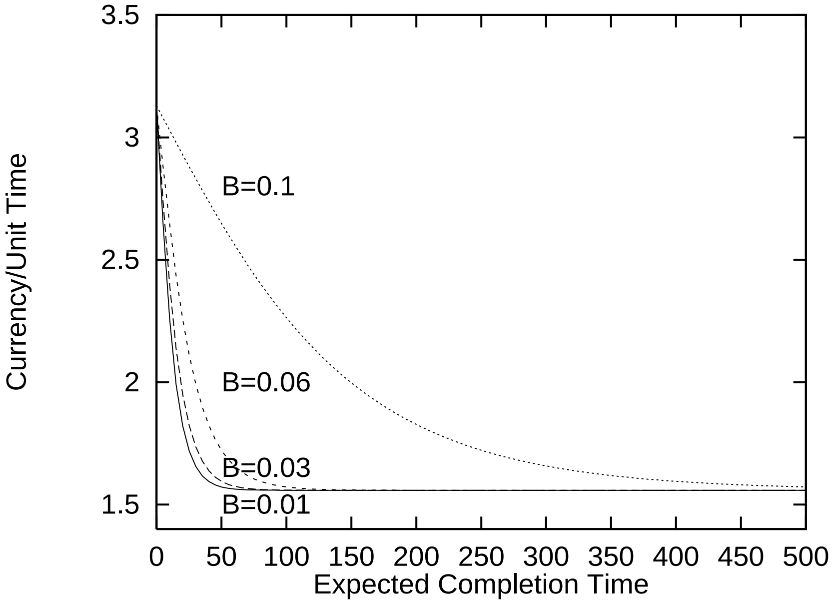

Figure 3: The currency function for machines

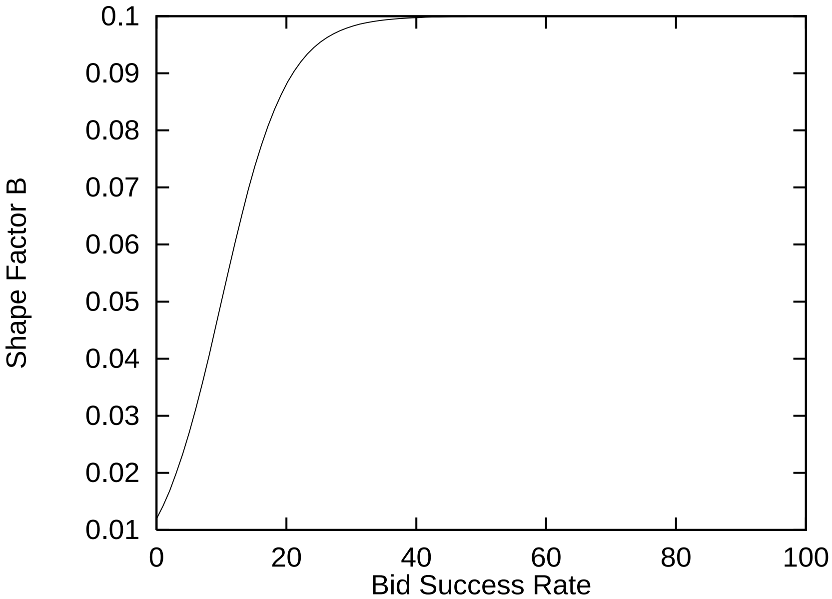

Figure 4: The currency shape factor for machines

part evaluates the bids after waiting for a fixed time, and then selects the machine with the least compound currency, as in the currency scheme. Once the machine is selected, the part confirms the bid and waits in the buffer of the selected machine. Whenever the selected machine notifies the part of a change in the expected waiting time, the part checks to see if the new completion time is less than its due date. If the completion time exceeds the due date, the part requests to preempt the part ahead in the queue. The part in front of it estimates the change in its completion time, and if it is less than its due date, it computes the currency that it charges based on its currency function. If the part that requested for preemption holds that currency, preemption is allowed. This is repeated till the completion time is less than the due date. If the other parts do not allow its preemption, the part requests the other machines for bids. If any machine can meet its requirements, the part leaves the current queue to join the new one.

5.2 Machine Control Protocol

The machine reads the request from the part and computes the expected completion time and the charge, provided that its buffer count is less than the limit. The machine then confirms the selection of the part, if the buffer count is still less than its limit. The machine maintains contact with all the parts that selected it as long as the machine is functional. It processes the part in front of the buffer and updates the change in completion time to all the parts in the queue.

6 Experimental Verification

A multi–agent manufacturing system was modeled to test the performance of the different control schemes. The manufacturing system consisted of 40 machines. These machines were grouped into 5 types, each group consisting of 8 machines with different speed factors. The system was designed with five part types, each with different routings as shown in Table 1. The due dates for these parts were set based on the priority of the parts. The arrival rates of the parts were exponentially distributed. All the machines had failure rates and average recovery times, which were used to examine the robustness of the control schemes to component failures. We present the simulation results for the following control protocols:

Model 1: Hierarchical Control Scheme

Model 2: Multi–Agent Shortest Processing Time Control Scheme

Model 3: Multi–Agent Currency Control Scheme

Model 4: Multi–Agent Preemption Model

Table 1: Routing table for parts

		Part Routing				
Part 1	Process Type	1	2	3	4	5
	Nominal Time	30	50	40	55	35
Part 2	Process Type	2	3	4	5	1
	Nominal Time	30	50	40	55	35
Part 3	Process Type	3	4	5	1	2
	Nominal Time	30	50	40	55	35
Part 4	Process Type	4	5	1	2	3
	Nominal Time	30	50	40	55	35
Part 5	Process Type	2	3	4	5	1
	Nominal Time	30	50	40	55	35

The hierarchical control scheme was used as a baseline model to compare the performance of multi–agent manufacturing systems models. In the hierarchical control model a group of machines was controlled by a central controller. In the simulation experiment each central controller coordinated four machines. When a controller failed, all the machines controlled by it ceased functioning as well. Controller failures were simulated at rates much larger than machine failure rates. Parts were assigned to individual machines by the central controller.

The following performance measurements were used to compare the different control schemes: 1) the average waiting time of parts in system and 2) the average tardiness of parts.

The manufacturing system models were simulated using Swarm (Burkhart et al., 1996), an object oriented simulation tool. Parts and machines were designed as instances of classes. In this system, an agent was represented by an object and communication between agents was modeled by messages between objects. Each object had its own state variables, while the generic definition of its behavior was provided by the class. The framework of Swarm facilitates the easier implementation of multi–agent systems. All the objects were created in a zone, where the objects and their associated data were stored. The time management of activities was accomplished by schedules. Probe objects were employed to acquire properties of agents in simulation. In addition, two more agents were created to assist in simulation – the scheduler and the generator. The primary function of the scheduler was to maintain a schedule of events for creating parts, failure and recovery of machines. It invoked these events based on its calendar and also collected the performance measurements in output data files. The generator provided random numbers following different distributions to the requesting objects.

6.1 Experimental Design

The system performance measures were observed by varying the three parameters–system load, failure rates and recovery times. At high load, the mean arrival rate was 1/22 parts per time unit, whereas at low load, the mean arrival rate was 1/35 parts per time unit. Failure rates were set at a mean values of 1000 time units and 2000 time units. Repair times were set at means of 200 and 100 time units. The 2^k full factorial design of experiments for the three factors is shown in Table 2. Each configuration is replicated 10 times. The simulation run length for each run was 10000 time units.

Table 2: 2^k Full factorial experimental design

	Experiments							
	1	2	3	4	5	6	7	8
Load	−	−	−	−	+	+	+	+
Failure Rate	−	−	+	+	−	−	+	+
Recovery Time	+	−	+	−	+	−	+	−

6.2 Performance of Negotiation Schemes

This section presents the comparison of different negotiation schemes based in the results from the simulation experiments. The negotiation schemes are compared with respect to each performance measure. Note that the experiment numbers in the graphs correspond to the numbers in Table 2.

6.2.1 Average Waiting Time of Parts

The average waiting time of parts in the system is shown in Table 3. Figure 5 shows the histograms of the average waiting time when the load is low and Figure 6 shows the the plots of the average waiting time at high load. The hierarchical control scheme had the highest waiting time at both loads. The SPT and preemption models had the least average waiting time when the load was low. The SPT had a low waiting time because it processed the parts in the quickest machines without prioritizing them. At high load, the preemption model had the least average waiting time. Note that the variance of the average waiting time for agent based models is significantly smaller than the hierarchical control models.

Table 3: Average waiting time of parts

		Experiments							
		1	2	3	4	5	6	7	8
Model 1	Mean	56.0	55.8	70.1	64.4	285.8	346.8	418.4	460.9
	SD	31.2	36.9	33.0	28.9	115.4	197.5	157.4	287.6
Model 2	Mean	28.7	26.3	30.8	27.8	218.8	164.9	252.7	178.3
	SD	1.5	2.0	1.5	2.0	30.3	27.9	44.1	25.8
Model 3	Mean	39.3	35.5	40.1	38.3	217.4	197.2	252.0	177.9
	SD	3.9	4.5	3.5	2.9	20.2	17.3	44.4	15.6
Model 4	Mean	30.9	28.5	31.1	28.7	140.9	107.4	200.8	125.4
	SD	1.5	2.2	1.3	1.9	18.4	22.3	34.1	25.6

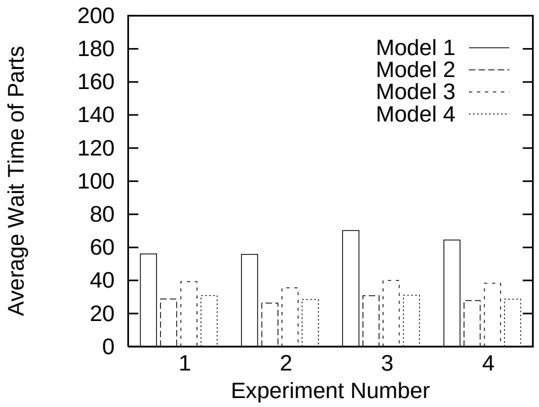

Figure 5: Average waiting time of parts at low load

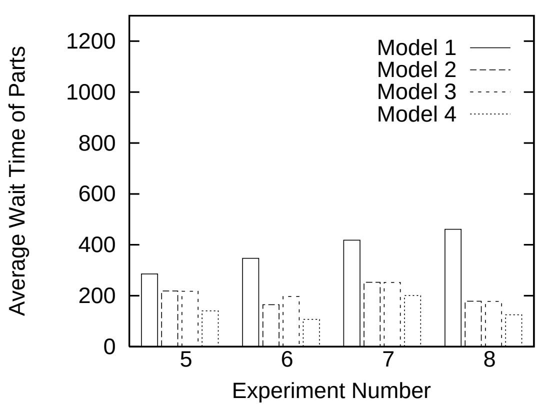

Figure 6: Average waiting time of parts at high load

6.2.2 Average Tardiness of Parts

Table 4 shows the average tardiness of the parts in the system. The average tardiness values at low load and high load are shown in Figures 7 and 8. The hierarchical system has significantly higher average tardiness at both low and high loads. At low load, all the multi–agent manufacturing systems have average tardiness levels close to zero. The preemption model has the lowest tardiness at the high load, which can be explained by the ability of parts to expedite their processing as they near the due dates.

Table 4: Average tardiness of parts

		Experiments							
		1	2	3	4	5	6	7	8
Model 1	Mean	10.1	13.2	18.6	13.9	138.3	201.1	252.6	310.8
	SD	10.9	22.6	20.0	11.7	91.3	179.5	147.2	263.4
Model 2	Mean	0.00	0.00	0.02	0.01	61.1	28.1	89.2	34.7
	SD	0.01	0.00	0.02	0.03	20.1	15.6	33.2	15.5
Model 3	Mean	0.07	0.00	0.02	0.01	55.6	40.6	84.7	29.0
	SD	0.13	0.02	0.02	0.03	14.5	13.9	38.6	9.5
Model 4	Mean	0.005	0.006	0.01	0.002	16.7	5.4	57.3	13.4
	SD	0.01	0.01	0.017	0.006	8.2	5.4	23.7	10.6

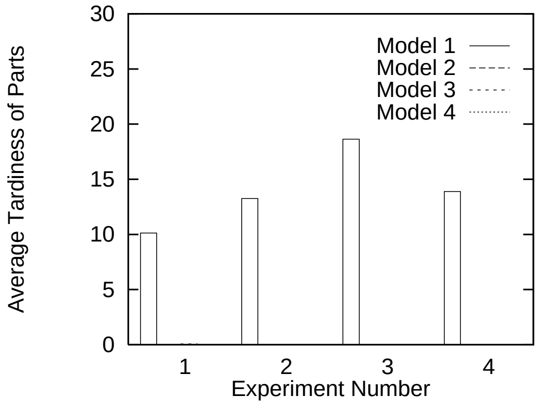

Figure 7: The Average tardiness of parts at low load

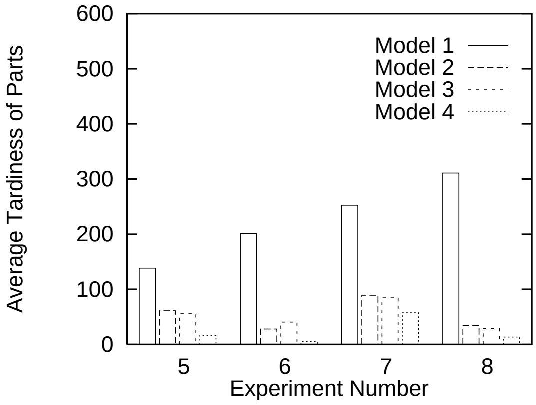

Figure 8: The Average tardiness of parts at high load

7 Summary

Multi–agent systems overcome several disadvantages of centrally controlled systems by distributing the decision making capability over several autonomous agents. Hence, the rules used by individual agents for making decisions are of significant interest in understanding the overall performance of agent based systems. This research presented three negotiation schemes, the SPT scheme, the currency scheme and the preemption scheme. In the currency and the preemption schemes, the agents use a value function to quote prices and evaluate overall costs of bids. The value or currency functions were developed to reflect the changing utility of time as the parts near their due dates.

A large manufacturing system, comprising of 40 machines and 5 part families, was simulated. The different control protocols, for both the part and the machine agents, were implemented using the Swarm multi–agent simulation platform. The performance of the agent based protocols was compared to that of a hierarchical control model. The results obtained from the simulations show that these control schemes have better performance than the traditional hierarchical control schemes under a variety of operating conditions. The results also indicate that the multi–agent manufacturing systems are more fault tolerant than the hierarchical manufacturing systems. However, multi–agent systems have higher overheads due to the communications between parts and machines. Moreover, analytical methods do not currently exist for predicting performance in multi–agent systems using different negotiation schemes.

Further work needs to be conducted in developing analytical models of multi–agent systems in order to design negotiation protocols which meet certain performance expectations. These modeling tools will facilitate the analysis of families of currency functions, thereby reducing the design and implementation time for multi–agent manufacturing systems.

References

Baker, A. (1991). *Manufacturing Control with a Market-Driven Contract Net*. PhD thesis, ECSE Department, Rensselaer Polytechnic Institute, Troy, NY.

Baker, A. (1996). *Market–Based Control: A Paradigm for Distributed Resource Allocation*, chapter Metaphor or Reality: A Case Study Where Agents Bid With Actual Costs to Schedule a Factory, pages 184–223. World Scientific Publishing Co. Pte. Ltd.

Ben-Arieh, D. and Chopra, M. (1997). A game theoretic approach to real-time distributed shop floor control. In *Proceedings of Sixth Industrial Engineering Research Conference*, pages 304–309.

Burkhart, R., Askenazi, M., and Minar, N. (1996). Swarm release documentation. Available at `http://www.santafe.edu/projects/swarm`.

Cohen, P., Cheyer, A., Wang, M., and Baeg, S. (1994). OAAA: An open agent architecture. In *Proceedings of AAAI Spring Symposium*.

Davis, R. and Smith, R. (1983). Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63–109.

Deshmukh, A., Benjaafar, S., Talavage, J., and Barash, M. (1993). Comparison of centralized and distributed control policies for manufacturing systems. In *Proceedings of 2nd Industrial Engineering Research Conference Proceedings*, pages 744–748.

Duffie, N. (1990). Synthesis of heterarchical manufacturing systems. *Computers in Industry*, 14:167–174.

Duffie, N. and Prabhu, V. (1996). Heterarchical control of highly distributed manufacturing systems. *International Journal of Computer Integrated Manufacturing*, 9(4):270–281.

Grossner, C., Preece, A., Radhakrishnan, T., and Newborn, M. (1995). Sharing data in multi-agent systems. Available at `http://www.cs.concordia.ca/~staffcs/cliff/dai/dai-list.html`.

Kahneman, D. and Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291.

Lewis, W. (1981). *Data Flow Architectures for Distributed Control of Computer Operated Manufacturing Systems: Structure and Simulated Applications*. PhD thesis, Purdue University, West Lafayette, IN.

Lin, G. and Solberg, J. (1992). Integrated shop floor control using autonomous agents. *IIE Transactions*, 24(3):57–71.

Lueth, T., Grasman, R., Laengle, T., and Wang, J. (1995). Cooperation among distributed controlled robots by local interaction protocols. In *Proceedings of 6th International Symposium on Robotics and Manufacturing*, pages 405–410.

Maley, J. (1988). Managing the flow of intelligent parts. *Robotics & Computer-Integrated Manufacturing*, 4(3/4):525–530.

Maley, J., Ruiz-Mier, S., and Solberg, J. (1988). Dynamic control in automated manufacturing: a knowledge integration approach. *International Journal of Production Research*, 26(11):1739–1748.

McDonnell, P. and Joshi, S. (1997). A game theoretic approach to reconfiguration decisions by heterarchical machine controllers. In *Proceedings of Sixth Industrial Engineering Research Conference*, pages 316–321.

Parker, L. (1993). Designing control laws for cooperative agent teams. In *Proceedings of IEEE Conference on Robotics and Automation*, pages 582–587.

Parker, L. (1994). Alliance: An architecture for fault tolerant, cooperative control of heterogeneous mobile robots. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 776–783.

Prabhu, V. and Duffie, N. (1996). Modeling and analysis of heterarchical manufacturing systems using discontinuous differential equations. *Annals of CIRP*, 45(1):445–448.

Sandholm, T. (1993). An implementation of the contract net protocol based on marginal cost calculations. *AAAI*, pages 256–262.

Sandholm, T. and Lesser, V. (1995). Coalition formation among bounded rational agents. In *Proceedings of 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 662–669, Montreal, Canada.

Shaw, M. (1987). A distributed scheduling method for computer integrated manufacturing: the use of local area networks in cellular systems. *International Journal of Production Research*, 25(9):1285–1303.

Smith, R. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 23(12):1104–1113.

Smith, R. and Davis, R. (1981). Framework for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1):61–70.

Upton, D. (1988). *The Operation of Large Computer Controlled Manufacturing Systems*. PhD thesis, School of Industrial Engineering, Purdue University, West Lafayette, IN.

Yang, E.-H. (1994). *Operation Strategies for Large Computerized manufacturing Systems*. PhD thesis, Purdue University, West Lafayette, IN.