International Journal of Management Science
Vol. 2, No. 1, December 1996

A Distributed Task Assignment Method and its Performance

Kap Hwan Kim*

Abstract

We suggest a distributed framework for task assignment in the computer-controlled shop floor where each of the resource agents and part agents acts like an independent profit maker. The job allocation problem is formulated as a linear programming problem. The LP formulation is analyzed to provide a rationale for the distributed task assignment procedure. We suggest an auction based negotiation procedure including a price-based bid construction and a price revising mechanism. The performance of the suggested procedure is compared with those of an LP formulation and conventional dispatching procedures by simulation experiments.

Keywords: Automated Manufacturing Systems, Shop Floor Control, Distributed Scheduling.

1. Introduction

Optimization theories on scheduling have revealed various limitations in solving practical problems, although they have provided basic intuitions for the development of heuristic rules which have been widely used in practice. For example, the dynamic nature of the shop floor is not considered in the

---

* Department of Industrial Engineering(Research Institute of Mechanical Technology), Pusan National University

analytic models. In addition, unrealistically restrictive assumptions are imposed on the characteristics of the shop floor in order to guarantee the optimality of the solution.

Although dispatching rules can be easily adopted as a dynamic control method because of their ease of use and the limited necessity of information on the shop floor[1], their performance highly depends on the state of the shop. Thus, no single rule will consistently outperform all the other rules during the entire planning horizon as the state of the shop changes continuously[2,3].

When computer integrated manufacturing systems which consist of a large number of automated facilities are considered, it is recognized that these very complex systems are beyond centralized direct control. Thus, a new approach to scheduling, called auction-based shop floor control, is suggested for the computer integrated manufacturing environment[7,8,9,10,11,12,13,14,15,16]. In this cooperative operating scheme, each interacting subsystem has its own objectives and modes of operation. This framework of shop floor control is based on the architecture of CIM which has intelligent and powerful local controllers connected to each other by communication links.

We can characterize this scheme as a distributed information processing, a distributed decision making, and a heterarchical market-like model. Each physical unit of orders and work centers is equiped with a microprocessor together with an intelligent software. Each software element acts like an independent decision maker communicating and negotiating with other software elements in real time through message passing and a bidding protocol to achieve mutual agreements for task assignments.

The advantages of this type of control method are the robustness of the system to various failures and breakdowns of some components and the central controller in the system, the efficiencies supported by a market-like framework, and the possibilities of utilizing the local information in real time.

T. Vamos[4] suggested an evolutionary perspective of distributed cooperative systems. He explained why the distributed cooperative system will be required in the future and how it can be realized under the present technological condition to solve the present complicated problems. He also recommended the automated manufacturing system as a promising field for the application of the concept.

Davis and Smith[5] presented a framework, called the contract net, which specifies communication and control in a distributed problem solver. Under the contract net model, negotiations are performed between a node with a task to be executed and nodes which have capabilities to do the task. Another paper[6] written by Smith describes the contract net protocol in more detail.

In Lin and Solberg's paper[7], a generic framework for this control method is described in detail. Their framework may be characterized by a combination of price and objective mechanism, multiple-way and mutiple-step negotiation, and dynamic resource unification scheme. They introduced control modules such as part agents, resource agents, intelligence agents, monitor agents, communication agents and database management agents. For each module, the organization and responsibilities are explained in detail. In another paper[8] written by them, they evaluated the impact of the flexibility of the part process plan on the performance of the system.

In most studies, parts have been treated with equal importance and thus there is no way for priority parts to advance themselves. In Yang et al.'s paper[9], they introduced an auction based scheme which recognizes the part priority as well as part status.

In Shaw's paper[10], the bidding algorithms which employ rules of the earliest finish time and the shortest processing time are compared with the centralized dispatching procedure using the shortest processing time rule. Although simple rules are employed, he showed that the distributed approach has good performances.

Butler and Ohtsubo[12] report the development of a prototype architecture for distributed scheduling. They show how a task may be divided into multiple subtasks and may be reallocated to sub-agents through the negotiation process. An application of the concept is illustrated for the case of the ship building process.

Parunak et al.[13] report another prototype factory control system, called Yet Another Manufacturing System, which utilizes the concept of distributed computing for industrial control.

Maley[14] developed a distributed manufacturing scheduling / control system which is called the Computer Automation Distributed Environment for Network Coordination and Execution based on contract net and written in

Smalltalk.

Morton and Pentico[17] suggested a scheduling framework, called "bottle-neck dynamics", which utilizes the price of each activity and resource to guide the searching process of the solution. The concept of the price in the paper is basically similar to the idea of this paper. However, the focus of their study is to give a higher priority to bottleneck resources using the concept of price instead of making a decision in a distributed manner.

An auction-based control framework is recommended in order to gain such advantages as distributed control, robustness, modularity, and stability. Some researchers point out, however, that a shortcoming of the framework is the difficulty in forecasting changes of the shop status and the progressing of orders. They also doubt the efficiency of the control method because of it's distributed nature.

In this paper, we will suggest a typical auction-based task assignment procedure in which we specify a detail pricing mechanism, a bid construction method, and a negotiation procedure. However, the main emphasis is placed on the efficiency of the auction-based method.

Afterwards, we will explain the characteristics of the task assignment problem. Suppose that three products are produced and the process of each product consists of four operations, respectively. Then, there are twelve different operations to be performed which we call "jobs". In the practical shop floor, multiple orders for a product arrive dynamically at the shop. Suppose that five orders for each of the three products arrive at the shop during a planning period. Since each order has four operations to be performed, the total number of operations for all the orders will be sixty each of which we call a "task".

In the next section, we will discuss job allocation among work centers under the condition that all the jobs are ready to process from the beginning. An illustration of the job allocation may be stated as "50% of operation 1 of orders for product 1 should be processed at work center 1 and the other 50% should be processed at work center 2". But, the task assignment deals with each operation of a specific order for a product. Later, we will treat the task assignment problem in the dynamic situation which is the main issue in this paper.

The task assignment consists of the task allocation and sequencing tasks at each work center. In task allocation, each task is matched with a work center. When an operation of a production order, we call a task, may be processed by two or more alternative work centers, we have to select one work center among multiple alternatives. Since we have adopted a dynamic decision strategy considering the continuously-changing shop status, we will assume the assignment for the next operation is determined only after the previous operation is completed. Each task has unique characteristics of technical constraints and operational requirements. And, every work center has a different number of waiting tasks and processing capabilities. Thus, decisions on task allocation highly depend on the status of tasks and work centers at the moment of the allocation.

Another important issue in the task assignment is how to sequence tasks at each work center. Although there are many conventional sequencing rules such as shortest processing time rule, earlist due date rule,etc., each rule shows different performances according to the state of the shop floor. In reality, the state of a shop floor changes continuously from time to time and each task has a different requirement to be satisfied. Thus, it is not an adequate solution of such a complex sequencing problem to apply a simple heuristic rule during the entire planning period.

In the previous studies, although general system frameworks of the negotiation-based procedure have been suggested, the basic rationale is not described and the validity of the procedure has not been evaluated based on the rationale.

In this paper, we will suggest a new auction procedure where each agent behaves in a really market-like manner. And we will discuss efficiencies of the distributed decision-making procedure by comparing it with a static version of the problem (job allocation problem) which can be formulated as a linear programming model.

The following assumptions are introduced:

1. **Fixed Operation Sequence** : Each part has a fixed sequence of operations. Thus, we do not consider the problem of selecting the next operation but consider the problem of selecting a most promising work center among many alternatives for the next predetermined operation.

2. **Event-driven Negotiation** : The negotiation process is triggered by the occurance of one of the following predetermined events:

(1) An order arrives at the shop.

(2) An operation of an order is completed.

(3) A work center completes a task.

3. Single Resource Type : Although there may be many different types of resources required for the processing of a part, such as work center, transporter, tool, fixture, etc., we will assume that the work center is the only critical resource.

4. Objective of Cost Minimization : Each part agent has the objective function of minimizing its own total cost which consists of the processing cost, the penalty cost for delayed delivery behind the due date, and the subcontracting cost.

The study in this paper is distinctive in the following aspects:

1. We suggest an event-driven negotiation process where both resource agents and part agents are assumed to behave as independent profit makers by incorporating a price system.

2. LP formulation is provided for the static version of the job allocation problem.

3. We derive a rationale for the distributed task assignment procedure from LP formulation.

4. The efficiency of the distributed decision-making procedure is compared with those of conventional dispatching rules.

We assume that part agent (PA) is in charge of a specific order and has the objective to deliver the order at a cost as low as possible. And resource agent (RA) is assumed to be a profit center who is in charge of a work center and has an interest in making money by offering services to PAs at a service charge as high as possible.

Thus, PA has to determine which work center to select as the processor for the next operation considering the service charge and the expected delivery date. RA has to determine which order to accept for the next service, and how much to receive the service charge considering the utilization of the work center and the revenue from the order simultaneously. Note that the subcontracting manager is also considered as an RA who competes with RAs for internal resources. The scheme in this study is that those decisions are made through an auction-based negotiation process between PAs and RAs instead of isolated determinations.

2. A static version of job allocation problem

In this section, the characteristics of the resource allocation problem are discussed and we explain how the problem may be decomposed into simple subproblems. And, we will show that each of the subproblem coincides with the decision which an RA or a PA tries to make independently in the dynamic situation.

We use the following notations:

u_{ij} = the unit processing time of job i on work center j (u_{ij} are set to ∞ for infeasible processing station),

c_j = the variable cost per unit time of processing on work center j,

x_{ij} = the number of items of job i processed on work center j during the planning period, which is a decision variable,

CAP_j = the capacity of work center j during the planning period,

D_i = the processing requirement of job i during the planning period,

m = the number of in-house work centers,

n = the number of jobs to be processed during the planning period.

The objective of the job allocation is to minimize the total processing cost including subcontracting cost. Each work center has limitations on its processing capacity while subcontractors are assumed to have unlimited capacities. In addition, the processing requirement of each product must be satisfied by the production system including subcontractors during the planning period.

Note that subcontractors are represented as work center $m+1$.

Then, the allocation problem may be formulated as follows:

(P1)

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{m+1} c_j u_{ij} x_{ij} \quad \cdots (1)$$

subject to

$$\sum_{i=1}^{n} u_{ij} x_{ij} \leqq CAP_j \text{ for } j=1, 2, \cdots, m \quad \cdots (2)$$

$$\sum_{j=1}^{m+1} x_{ij} = D_i \text{ for } i=1, 2, \cdots, n \quad \cdots (3)$$

$$x_{ij} \geq 0 \text{ for all i and j}$$

Note that problem (P1) is LP formulation and so can be easily solved.

The type of decision-making as in problem (P1) can be considered to be "centralized" because the job allocation is performed considering all the information of the jobs and work stations. But, in reality, events such as a work center completing a task or an order arriving at the shop occur dynamically. Thus, the decision for all the future work centers and orders cannot be made simultaneously because all the information about them are not available. But, as soon as an order arrives at the shop, a decision has to be made promptly on where to route it. And, at the moment a work center completes a task, the next task should be loaded without delay. That is, for routing problem, the work center for the next operation has to be decided based on only local information such as the processing time and the expected waiting time only at candidate work centers.

In the following, we discuss two contrasting cases. In the first case, the distributed decision by each PA doesn't satisfy the capacity constraints of work centers. But, in the second case, we show how the independent distributed decision by each individual PA is induced to satisfy the capacity constraints by modifying the cost matrix.

Suppose that Table 1 represents an example of the processing cost matrix, where c_j and u_{ij} are enclosed in the parenthesis, and that each PA allocates his / her job among work centers in a way of minimizing the processing cost. But we assume that each PA has only the data in each corresponding row of Table 1 and the capacity limit of each work center. That is, each PA doesn't take into account the state and cost information for the other jobs in his / her decision making. Then, the resulting decision becomes like the values outside the parenthesis in Table 1. Note that constraint (2) of (P1) is not satisfied with this solution.

Suppose that we revise the cost matrix of Table 1 to the one of Table 2 by adding 1 to every c_2 of column 2 and every c_3 of column 3, which is a valid revision in a sense that it does not change the final optimal solution of (P1) (This will be proven in the following discussion). Then, when each PA chooses the least-cost work center based on only the cost data only in each corresponding row, the result will be the values of x_{ij} in Table 2.

Table 1. An example of the cost matrix $(c_j, u_{ij})\ x_{ij}$

work center / job	1	2	3	4 subcontractor	D_i
1	(1,7) 0	(1,2) 4	(1,4) 1	(3,3) 0	5
2	(1,10) 0	(1,5) 1	(1,3) 3	(2,4) 0	4
3	(1,5) 0	(1,4) 0	(1,3) 3	(2,4) 0	3
$\sum_{i=1}^{3} u_{ij}x_{ij}$	0	13	22		
CAP_j	22	8	9		

Table 2. The revised cost matrix $(c_j, u_{ij})\ x_{ij}$

work center / job	1	2	3	4 subcontractor	D_i
1	(1,7) 1	(2,2) 4	(2,4) 0	(3,3) 0	5
2	(1,10) 0	(2,5) 0	(2,3) 3	(2,4) 1	4
3	(1,5) 3	(2,4) 0	(2,3) 0	(2,4) 0	3
$\sum_{i=1}^{3} u_{ij}x_{ij}$	22	8	9		
CAP_j	22	8	9		

For example, since the processing costs for job 1 at work center 1,2,3, and 4 are 7,4,8, and 9 respectively, PM 1 will decide to allocate 4 units to work center 2, which exhaust all the capacity of work center 2, and 1 unit to work center 1.

Note that the solution in Table 2 satisfies constraint (2) of (P1).

The validity of the cost matrix revision in the above and the optimality of distributed decision-making are formally proven in the following:

The problem (P1) can be rewritten as

(P2)

$$\text{Minimize } \sum_{i=1}^{n}\sum_{j=1}^{m+1} c_j\, u_{ij}\, x_{ij} \quad \cdots (4)$$

subject to

$$\sum_{i=1}^{n} u_{ij}\, x_{ij} + z_j = CAP_j \text{ for } j=1, 2, \cdots, m \quad \cdots (5)$$

$$\sum_{j=1}^{m+1} x_{ij} = D_i \text{ for } i=1, 2, \cdots, n \quad \cdots (6)$$

$$x_{ij},\ z_j \geqq 0 \text{ for all i and j} \quad \cdots (7)$$

By adding each constraint of (5) multiplied by a constant α_j to the objective function, we get

(P3)

$$\text{Minimize } [\sum_{i=1}^{n}\{\sum_{j=1}^{m}(c_j+\alpha_j)u_{ij}\, x_{ij} + c_{(m+1)}u_{i(m+1)}x_{i(m+1)}\}+\sum_{j=1}^{m}(\alpha_j z_j - \alpha_j CAP_j)]$$

subject to constraints (5), (6), and (7).

Note that (P3) has the same optimal solution as (P1)[18].

Then, consider the following problem:

(P4)

$$\text{Minimize } [\sum_{i=1}^{n}\{(\sum_{j=1}^{m}(c_j+\alpha_j)u_{ij}\, x_{ij} + c_{(m+1)}u_{i(m+1)}x_{i(m+1)}\}+\sum_{j=1}^{m}(\alpha_j z_j - \alpha_j CAP_j)] \cdots (8)$$

subject to

$$u_{ij}\, x_{ij} \leqq CAP_j \text{ for } i=1, 2, \cdots, n \text{ and } j=1, 2, \cdots, m \quad \cdots (9)$$

$$\sum_{j=1}^{m+1} x_{ij} = D_i \text{ for } i=1, 2, \cdots, n$$

$$x_{ij},\ z_j \geq 0 \text{ for all i and j}$$

In problem (P4), constraint (5) of problem (P3) is replaced by a more relaxed constraint (9). Thus, the value of the optimal objective function of (P4) becomes a lower bound of (P3).

If we drop the last term of (8) which is independent of the other decision variables, problem (P4) can be broken down into n independent problems as the following:

(P5)

$$\text{Minimize } [\{\sum_{j=1}^{m}(c_j+\alpha_j)u_{ij}\, x_{ij} + c_{(m+1)}u_{i(m+1)}x_{i(m+1)}\}] \quad \cdots (10)$$

subject to

$$u_{ij}\, x_{ij} \leqq CAP_j \text{ for all } j \cdots\cdots (11)$$

$$\sum_{j=1}^{m+1} x_{ij} = D_i \cdots\cdots (12)$$

$$x_{ij} \geq 0 \text{ for all } j$$

This is the problem which each PA solved in Table 2. It can be solved easily only by sequencing work centers in an increasing order of the coefficient of x_{ij} in the objective function and increasing x_{ij} up to the capacity of the corresponding work center (constraint (11)) until the total production requirement is satisfied (constraint (12)). Thus, the decision making is distributed in the sense that each PA solves the problem independently based on only such local informations as processing costs of his / her own order at candidate work centers.

Since problem (P5) is a relaxed version of the original problem (P1), if the optimal solutions of n independent problems (P5) satisfy constraint (2) of problem (P1), it is also the optimal solution of (P1).

Implications of distributed decision-making

In the dynamic shop floor, we will try to realize the solution of problem (P1) by satisfying the following conditions:

Condition 1. x_{ij} is the optimal solution of (P5).

Condition 2. The optimal solutions of n independent problem (P5) satisfy constraint (2).

In the next section, we will show how the above conditions are satisfied by the negotiation process.

The last unanswered problem is how to set α_j for each work center. In the example of Table 2, the role of α_j was to make the solution of the distributed decision-making problem satisfy the global capacity constraints of (2). Determination of α_j is what we try to do by setting charge prices of work centers through the negotiation process between PAs and RAs. $(C_j+\alpha_j)$ corresponds to the charge price for an unit processing time of work center α_j in the next section.

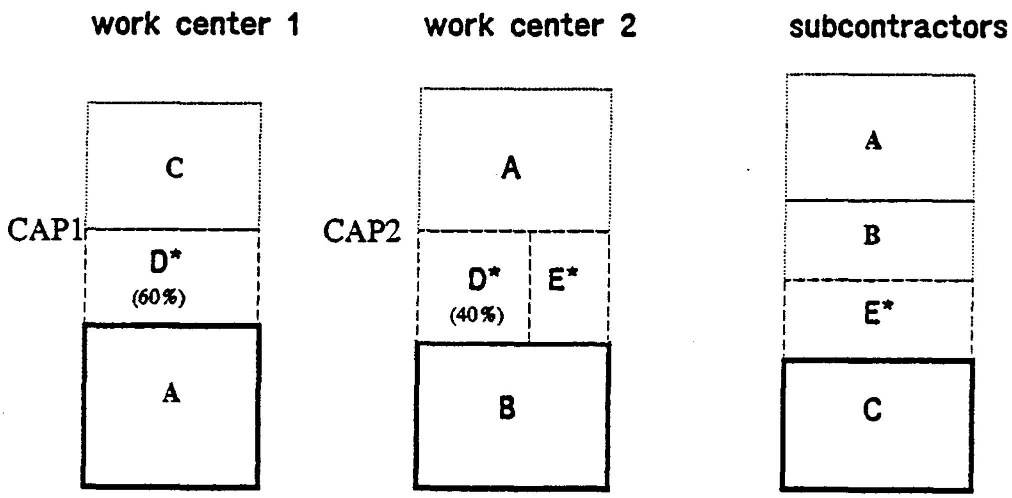

Figure 1. An illustration of job allocation

Now, we will illustrate a solution of the static version of the job allocation problem (P1). The solid line in Figure 1 indicates that the corresponding job is assigned to the work center and the dashed line denotes that the job is shared among multiple work centers. Although the job in the dotted box may be assigned to the corresponding work center according to the process plan, the work center is less attractive than another alternative work center because of the relatively high processing charge. The solution of Figure 1 may be interpreted as follows from the viewpoint of dynamic and distributed decision-making process:

Although work center *1* can process jobs A, D, and C, work center *1* is less attractive to PA of job C than a corresponding subcontractor considering charge prices. For job D, work center *2* is equally competitive as work center *1*. Thus, we enclose job D within the dashed box in the figure for work center *1* and work center *2*, which means that the product may be processed by work centers *1* or *2* at the same cost. In this case, the cost includes the delay cost as well as the processing cost. Note that the amount of work for job D should be split into 60% to 40% between work center *1* and *2* due to limitations of processing capacities. We will call jobs C, D, and A in workcenter 1 "unacceptable", "acceptable", and "preferable" jobs, respectively.

We will represent the set of "preferable", "acceptable", and "unacceptable" jobs for the work center j as B_{j1}, B_{j2}, and B_{j3}, respectively, for a given set of α_j's. And let $B_j = B_{j1} \cup B_{j2} \cup B_{j3}$. In the example, $B_{11}=\{A\}$, $B_{12}=\{D\}$, $B_{13}=\{C\}$, $B_{21}=\{B\}$, $B_{22}=\{D,E\}$, and $B_{23}=\{A\}$.

3. A negotiation-based task assignment procedure

We mean by "task" an operation of a specific order. "Job" in the last section means the works related to a specific operation of all the orders for a product during the planning period. In this section, we discuss the assignment of tasks but not jobs. In dynamic situations, tasks are not always available. In addition, RAs cannot anticipate when orders arrive. The problem is how to realize the optimality of results of the job allocation problem in the previous section even in the dynamic situation.

We suggest a framework in which RA and PA coordinate to assign tasks to work centers efficiently. In the following discussion, RA can be considered as a profit maximizer who sells his / her processing time at a price as high as possible and PA as a cost minimizer who tries to process his / her tasks at a cost as low as possible.

The shop floor control method in this section is basically event-driven. Auction procedure is invoked either when a new order arrives at the shop or a work center completes a task. When an order arrives at the shop or an operation of an order is completed, the corresponding PA activates a "PA-initiated auction procedure". Then, RAs who have available input buffer spaces participate in the negotiation. When a work center finishes processing a task, the corresponding RA starts the "RA-initiated auction procedure". PAs whose next operation can be processed on the corresponding work center become involved with the negotiation process. The overall procedures are summarized in Figure 2 and 3.

Below are various terminologies on prices which are introduced:

$P_a(i,j)$ = the bidding price of PA i to RA j in RA-initiated auction procedure,

$P_r(i,j)$ = the bidding price of RA i to PA j in PA-initiated auction procedure,

P_{jc} = the charge price for a unit processing time at work center(RA) j, which is realized most recently,

P_{je} = the expected charge price for a unit processing time at work center j for the next commitment.

In RA-initiated auction procedure, RA announces the expected price for the next commitment to help PA construct a bid. Based on the expected price, PA prepares the next bidding price. In addition, RA can control his / her own utilization by adjusting the expected price. As a forecasting method, we assume RA uses the exponential smoothing model to get the expected price. That is,

$$P_{je} = (1-\beta)\, P_{je} + \beta\, P_{jc} \qquad (13)$$

where $0 \le \beta \le 1$.

P_{ja} = the minimum acceptable price for RA j. For any bidding price to be committed by PA j, it should exceed at least the minimum acceptable price. In the experiment, we set this price at the level of the processing cost. But, generally, this can be used for RA to screen out bids with too low bidding prices.

Note that since the subcontracting cost is not an internal decision variable, we assume that $P_{je} = P_{jc} = c_j$ for subcontractors.

RA-initiated procedure

1. **Resource Availability Announcement** : When a work center completes a task, RA j in charge of work center j announces its availability, the expected start time, and the expected charge price for his / her resource to all the PAs. The expected start time is calculated by adding the present time, the sum of processing time of the committed tasks, and the expected waiting time for the commitment. RA evaluates the expected waiting time by calculating the exponentially weighted average of the realized waiting time.

2. **Bid Construction and Submission** : PAs, who consider work center j as one of candidate resources to process their waiting orders, submit bids to RA j. The bid information of each PM is composed of the bidding price that he / she is willing to pay and the required resource amount (service time). The bidding price will be decided by PA in a way of minimizing the

expected cost including the processing cost, the subcontracting cost, and the expected penalty cost for the late delivery. In the experiment of the next section, the penalty cost also plays the role of preventing excessive long delays which occur frequently when the conventional dispatching rules are used.

3. **Bid Collection, Evaluation, and Acceptance** : RA j collects and reviews all the bids and selects a bid with the highest bidding price and announces the accepted bid.

4. **Bid Rejection and Bid Revision** : RA announces rejected bids. The RA also announces the expected start time for every rejected task and a new expected charge price. RA calculates the expected start time by adding the present time, the remaining processing time of the task on the work center, the sum of the processing time of the committed tasks, and the expected waiting time of the rejected task before commitment. RA estimates the expected waiting time before commitment of a specific task by sequencing tasks in a descending order of the bidding price and adding cumulatively processing times of tasks whose bidding prices were higher than the one of the corresponding task.

PAs who received a rejection notice from the RA revise bids for the next auction process.

PA-initiated procedure

1. **Task Availability Annoucement** : When an operation of order i is just completed, PA i announces the availability of the task.

2. **Bid Construction and Submission** : RAs, who have available resources which can process the next operation of order i, send to the PA bids with bidding price data and the expected start time. In the experiment, a work center is assumed to be available when there is at least one empty space at the input buffer. The expected charge price is used as the bidding price.

3. **Bid Collection and Evaluation** : Based on bid data collected, PA selects the most attractive bid. Bid evaluation is based on the total cost which consists of the processing cost and the expected penalty cost. PA then informs each corresponding RA the result of the evaluation.

4. **Price Revision** : The rejected RAs lower the next bidding price so that

their work centers may be utilized more intensively. Since RA uses the expected charge price as the bidding price in the experiment, RA actually lowers the expected charge price.

Note that when multiple tasks are committed to a workcenter, FIFO rule is used to process the committed tasks.

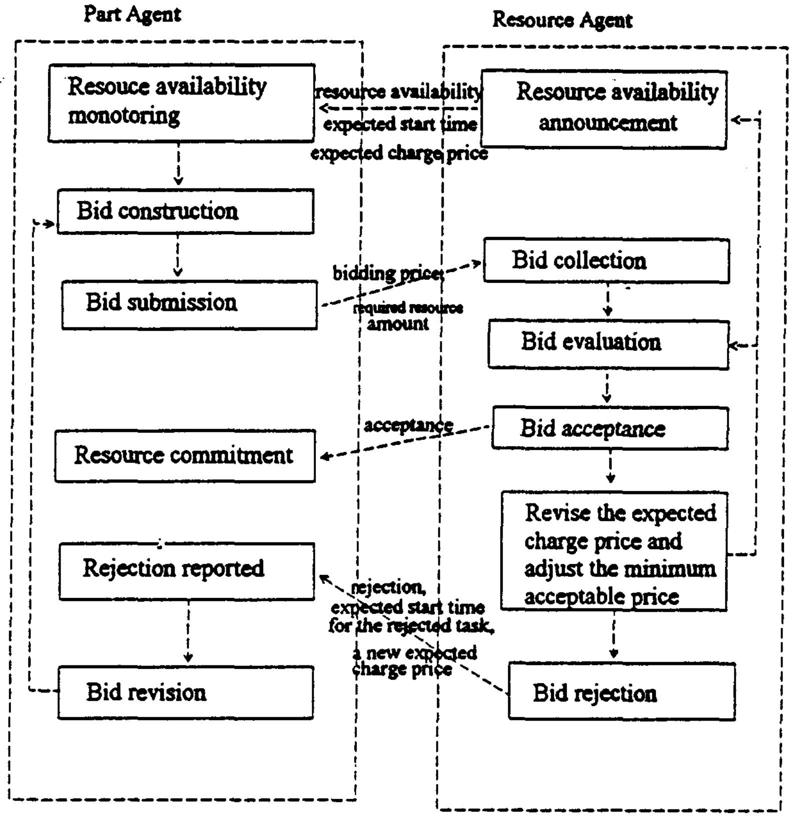

Figure 2. Negotiation process in resource-initiated procedure

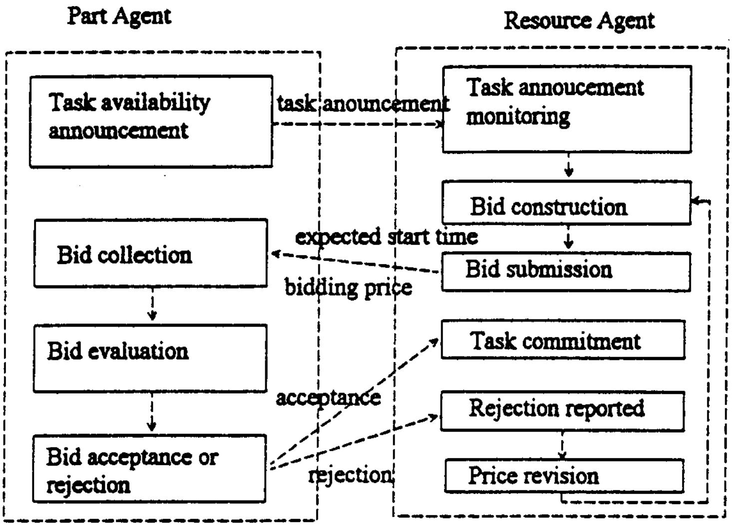

Figure 3. Negotiation process in part-initiated procedure

The pricing and evaluation mechanism

The following notations are used:

π_i = the penalty cost for late delivery of order *i* per unit time,

d_i = the due date of the next operation of order *i*. The due date of each operation is calculated by the backward scheduling as follows: Let order *i* have three operations. Then,

(due date of the third operation) = (delivery due date of order *i*),…(14)

and (due date of the second operation) = (due date of the third operation)

−(expected processing time of the third operation)

−(expected waiting time for the third operation).……………………(15)

The expected setup time, the unit processing time, and the waiting time in (15) are estimated by the exponential smoothing model like equation (13). That is, PA updates these estimated data using the realized data whenever an order completes its process in the shop. This updated data will be used for the next order of the same product.

t_{ij} = the processing time of the next operation when order *i* is processed at work center *j*,

P_{ij} = the bidding price of PA i to RA j,

c_{ij} = the expected completion time of the next operation of order i when it is processed at work center j.

Generally, this is evaluated as follows:

(expected completion time) = (present time) + (remaining processing time of the task on work center j) + (sum of processing time of tasks already committed) + (expected waiting time PA i before the commitment by RA j) +(processing time of the task of PA i under consideration). ....(16)

When RA rejects the bid from PA i, RA informs the PA of the expected starting time which is the sum of the first four terms of the above equation. RA estimates the expected waiting time of each rejected task by sequencing tasks in an ascending order of the submitted bidding price and adding cumulatively processing times of tasks whose bidding prices are higher than the one of the corresponding task. For PAs who are new comers into the negotiation process, RA announces the expected start time for the PA when it initiates the auction process. At that time, RA uses the exponentially averaged waiting time of all the tasks processed so far at the corresponding work center as the expected waiting time.

In PA-initiated procedure, since RAs who have an available buffer space respond to the task availability announcement, the expected waiting time for commitment equals zero.

In the following, we suggest several conditions for the price system in order to be used as an effective method of the shop floor control:

(1) When an order is delayed behind its schedule, the bidding price for the order should become higher than when it is not delayed in order to expedite the acceptance of the bid in the next negotiation process.

(2) PA should increase the possibility of acceptance of a bid when it is submitted to the most attractive RA. That is, the bid which is submitted to the most attractive RA should have a higher bidding price than the expected charge price for the next commitment.

(3) When an RA is the most attractive alternative to many PAs, the charge price should be increased to earn more money.

(4) RA of a less attractive work center should lower the charge price to increase the utilization of the corresponding work center.

In the experiment, bidding prices by a PA to alternative RAs are calculated as follows:

The expected cost incurred when the next operation of order i is processed by work center j is expressed as

$$COST_{ij} = P_{je}\, u_{ij} + \pi_i \max\{c_{ij} - d_i,\ 0\} \quad (17)$$

Let work center j^* be the least costly one and work center j^{**} be the second least costly. Then, since $COST_{ij}$ includes the expected delay cost, the delayed order is automatically expedited by offering a higher price to RA. But, to increase the possibility of acceptance by the most attractive RA, $(COST_{ij^{**}} - COST_{ij^*}) / 2$ is added to $COST_{ij^*}$ in calculating the bidding price for RA j^*. And for the less attractive RA, PA suggests the bidding price at which the cost becomes equal to the one at the most attractive work center. Finally, we get equations for bidding prices as follows:

$$P_{ij^*} = (COST_{ij^*} + COST_{ij^{**}}) / (2u_{ij^*}) \quad (18)$$

$$P_{ij} = COST_{ij^*} / u_{ij} \qquad \text{for } j \neq j^*. \quad (19)$$

In the following, we will show how above conditions (1)-(4) can be satisfied under the price system which is represented by equations (13) and (17)-(19):

(1) Suppose that order i is delayed behind the schedule, then the expected cost, $COST_{ij}$, increases for all j. Thus, bidding prices of (18)-(19) also increase.

(2) From (18),

$$\begin{aligned} P_{ij^*} &= (COST_{ij^*} + COST_{ij^{**}}) / (2u_{ij^*}) \\ &\geq 2COST_{ij^*} / (2u_{ij^*}) \text{ (by the fact } COST_{ij^*} = \min_j COST_{ij}) \\ &\geq P_{je} \text{ (The equality holds only when there is no possibility of delay).} \end{aligned} \quad (20)$$

The PA will submit a bid with the bidding price higher than the previous expected price to the most attractive RA.

(3) If an RA is the most attractive alternative to at least one PA, the highest bidding price, which will be the next committed price, becomes higher than the previous expected charge price by the same reasoning as the one in (2). Then, the expected charge price for the next negotiation increases.

(4) Let RA j ($j \neq j^*$) be not the most attractive alternative. When order i is not delayed,

$$P_{ij} = COST_{ij^*} / u_{ij} \text{ for } j \neq j^*$$
$$\leq COST_{ij} / u_{ij} (\text{by the fact } COST_{ij^*} = \underset{j}{Min}\ COST_{ij})$$
$$= P_{je}. \quad \cdots\cdots (21)$$

That is, the bidding price becomes lower than the expected charge price. If RA j accepts the bid, then the charge price decreases and the expected charge price also decreases. Then, owing to the more competitive price, the utilization of work center j becomes higher than before. But when an order is delayed behind the schedule, the corresponding PA may suggest a higher price than the current expected charge price to expedite the processing of the order.

Let's return to the final two conditions of the previous section for the optimality of the solution.

In the following, we discuss how two conditions in the previous section are satisfied under the auction-based assignment (ABA) procedure. But, since we assume the dynamic situation in the shop floor, we cannot expect that the optimal solution of the static problem can be perfectly realized.

CONDITION 1: In the price system of ABA, the expected charge price corresponds to $C_j + \alpha_j$ or $C_{(m+1)}$ in problem (P5). Since PA tries to seize the most attractive RA (whose cost coefficient, $(C_j + \alpha_j)u_{ij}$ or $C_{(m+1)}u_{i(m+1)}$, is the smallest), the task assignment process of ABA is equivalent to minimizing the objective function of equation (10). In a real shop floor, we can assume that waiting queues don't build up infinitely. Then, since we assign a task only when a work center becomes available and all the orders are processed in the shop eventually, constraints (11) and (12) of problem (P5) are satisfied naturally. The validity of the finite-queue-length assumption will be provided in the explanation of condition 2. In the following, we explain how the independent decision of the individual PA is realized in the negotiation procedure:

Suppose that PA i chooses work center j^* as the least costly one. Then, from equation (20), the bidding price becomes higher than the expected charge price for the least costly work center. And bidding prices to the other work centers become lower than their expected price from (21). Thus, even in the dynamic shop environment, work center j^* has the highest possibility to process order *i*.

Generally speaking, the expected charge price (P_{je}) increases when an order of a preferable product (a product in B_{j1}) in the static model is committed by RA *j* while it decreases when an order of a product in B_{j3} is committed. In a dynamic situation, there is no job in B_{j2}, since the expected charge price of work center *j* changes continuously. Jobs, which should be shared among multiple work centers in the static model, change their membership from time to time responding to the fluctuation of expected charge prices.

Refering to Figure 1, as long as orders of product "A" are waiting for work center 1, orders of product "D" have to wait in the queue. But, when no order of "A" is in the queue of work center 1, an order of "D" will be assigned to work center 1. Note that there is higher possibility for an order of "D" to be assigned to work center 1(about 60%) than to work center 2 (about 40%).

CONDITION 2: In a dynamic situation, an order which arrived at the shop will be processed within a finite length of time unless the length of waiting line grows infinitely. When the number of waiting jobs increases, the expected charge price increases too and so tasks will be transfered to other work centers or subcontractors. Thus, the waiting lines don't build up infinitely in ABA procedure. Note that subcontractors have constant expected charge prices and bidding prices.

4. Simulation experiments

The dynamic task assignment procedure in this paper is evaluated by a simulation experiment. In the example, we assume that there are three types of products, four operations per product, and two alternative machines for each operation. We can illustrate a situation that fits with this specification as follows: In a machine shop, there are three different types of parts(products) to be machined. Five different types of operations may be performed in this shop such as turning, milling, drilling, boring, and grinding. For each type of operation, there are two alternative machines to perform it which are different from each other in the performance and the degree of automation. All parts require four operations and each type of part has a different set and sequence of operations.

SIMAN for IBM PC is used as a modeling software.

The situation assumed for the simulation may be summarized as follows:

1) Three products are manufactured in the shop each of which has four operations. Three product-mix scenarios are used such as (1) 50:30:20, (2) 30:40:30 and (3) 20:30:50 which is expressed in terms of percentages of the number of orders of each product.

2) Orders each of which consists of multiple units of a single product arrive at the shop at every exponentially distributed interval with the mean interarrival time of (1) 80, (2) 100, or (3) 120. And the number of units for a specific order is randomly selected from the uniform distribution, U(30,150).

3) There are ten in-house work centers each of which consists of one machine.

4) Every operation of three products can be processed at one of two alternative work centers or by subcontractor (Refer to Table 4).

5) Processing cost of a subcontractor is higher than those of in-house work centers (Refer to Table 6).

6) Each order is assigned a due date. When a delivery is delayed beyond the assigned due date, a penalty cost is imposed. The penalty cost per unit time delay per order is assumed to be (1) 1, (2) 2 or (3) 4.

7) The total cost consists of in-house processing cost, subcontracting cost, and penalty cost.

8) One unit of the buffer space is allowed for the committed tasks at all the work centers.

9) The performance of the auction-based allocation (ABA) procedure in this paper is compared with those of the following eight conventional dispatching procedures:

- Dispatching by the Earliest Completion Time Rule, subcontracting by the Least Cost Decision Rule, and sequencing by FIFO (ECF) : A task is dispatched to the in-house work center which can complete it as soon as possible. If all the total expected costs for the task at in-house work centers are higher than the one of the subcontractor, it is subcontracted. Of course, the total expected cost includes the expected penalty cost. And, at each work center, FIFO rule is applied to decide the order of the processing.
- Dispatching by the Earliest Completion Time Rule, subcontracting for No Delays, and sequencing by FIFO (EDF) : This rule is the same as ECF

except that if a delay is expected, the task is immediately subcontracted.

• Dispatching and subcontracting by the Least Cost Rule, and sequencing by FIFO rule (LCF) : A task is dispatched to the work center which can process it at the least cost including the expected penalty cost. In this case, the subcontractor is also considered as a work center. And, orders are sequenced by FIFO rule at each work center.

• Dispatching by the Least Cost Rule, subcontracting for No Delay, and sequencing by FIFO rule (LDF) : This rule is the same as LCF except that if a delay is expected, the task is immediately subcontracted.

• Dispatching by the Earliest Competion Time Rule, subcontracting by the Least Cost Rule, and sequencing by the Earliest Due Date Rule (ECE) : This procedure is the same as ECF except that the Earliest Due Date Rule is applied to sequence orders at each work center.

• Dispatching by the Earliest Completion Time Rule, subcontracting for No Delays, and sequencing by EDD (EDE) : This procedure is the same as EDF except that EDD rule is used as the sequencing rule.

• Dispatching and subcontracting by the Least Cost Rule, and sequencing by EDD rule (LCE) : This procedure is the same as LCF except that EDD rule is used as the sequencing rule.

• Dispatching by the Least Cost Rule, subcontracting for No Delay, and sequencing by EDD rule (LDE) : This procedure is the same as LDF except that EDD rule is used as the sequencing rule.

Table 3. Comparisons of conventional heuristic procedures

notation	dispatching	subcontracting	sequencing
ECF	earliest completion time	least cost	FIFO
EDF	earliest completion time	no delay	FIFO
LCF	least cost	least cost	FIFO
LDF	least cost	no delay	FIFO
ECE	earliest completion time	least cost	EDD
EDE	earliest completion time	no delay	EDD
LCE	least cost	least cost	EDD
LDE	least cost	no delay	EDD

In Table 4, alternative work centers and processing time per unit for each operation of three products are provided. And in the last column, we provide the processing time of subcontractors. In the last row of Table 4, the processing cost per unit time of each work center is shown and subcontracting cost per unit processing time is provided in Table 5. We assume that the processing cost is higher and the processing time is longer at the subcontractor's facility than at in-hour work centers. In Table 6, the data in Table 4 and 5 are converted into the processing cost per unit.

Each simulation run was conducted from 0 to 60,000 time units while the first 10,000 time units are considered as a warm-up period. Thus, in the static model, we set the processing capacities as 50,000 time units for all the in-house work centers in the static model which we solved for the comparison purpose.

Detailed output data for an example problem are shown in Table 7. Parameters used in the example problem are (1) mean interval time = 80, (2) product-mix is 50:30:20, and (3) penalty cost = 2. Table 7 shows the solution of LP formulation for job-allocation problem together with the results of simulation runs by the auction-based procedure and eight conventional dispatching rules which are introduced above.

The numeric value in each entry of Table 7 indicates the percentage of the amount of the job allocated to each work center. Note that the distribution of the job amount allocated by ABA resembles that of LP formulation more closely than those by any other heuristic procedure. As a measure of the deviation of solutions of a heuristic procedure from those of LP formulation, we suggest the following formula (This formula is an illustration for the case of ABA.):

$$\sum_{i=1}^{3}\sum_{j=1}^{4}\sum_{k=1}^{3} |P_{OPT}(i,j,k) - P_{ABA}(i,j,k)| \,/\, 36, \quad \cdots\cdots (22)$$

where $P_{ABA}(i,j,k)$ denotes the percentage of job (i,j) (operation j of product i) allocated to the k^{th} work center when ABA procedure is used.

In Table 8, the average deviation of solutions and the average total cost per order are compared. If we agree that the result of LP formulation is the most efficient allocation of resources under the assumption of static shop environment, ABA is the most successful in the assignment of tasks from the viewpoint of efficiency among simulated heuristic procedures. Furthermore, ABA obtains the least cost solution among nine heuristic procedures.

Table 9 shows the processing cost per unit time, the shadow price of each work center in LP model, the average charge price, and utilization of each work center in ABA. Notice that work center 10 is underutilized and so the average charge price for work center 10 is the same as its processing cost per unit time, which coincides with the fact that the shadow price for work center 10 is zero in the final solution of LP formulation.

Table 4. Alternative work centers and processing time per unit of each product

product	operation	WC1	WC2	WC3	WC4	WC5	WC6	WC7	WC8	WC9	WC10	subcontractor
1	1	2.0	3.0									3.1
1	2			2.0	2.3							2.4
1	3					2.3	2.5					2.6
1	4							2.8	3.2			3.8
2	1			4.0	5.5							5.0
2	2					2.9	3.5					3.6
2	3							2.9	2.5			3.6
2	4									3.1	3.2	3.5
3	1	4.0	4.0									4.6
3	2									2.7	2.5	3.5
3	3							2.6	3.1			3.2
3	4					4.1	3.8					3.9
cost per unit time		15	10	10	6	7	15	14	8	9	14	

Table 5. Subcontracting cost per unit processing time

operation / product	1	2	3	4
1	20	20	20	20
2	30	30	30	30
3	40	40	40	40

Table 6. Alternative work centers and processing cost per unit

product	operation	alternative work center		processing cost per unit		
		A	B	A	B	C
1	1	1	2	30.0	30.0	62.0
1	2	4	3	13.8	20.0	48.0
1	3	5	6	16.1	37.5	52.0
1	4	8	7	25.6	39.2	76.0
2	1	4	3	33.0	40.0	150.0
2	2	5	6	20.3	52.5	108.0
2	3	8	7	20.0	40.6	108.0
2	4	9	10	27.9	44.8	105.0
3	1	2	1	40.0	60.0	184.0
3	2	9	10	24.3	35.0	140.0
3	3	8	7	24.8	36.4	128.0
3	4	5	6	28.7	57.0	156.0

A : the least-cost work center
B : the other alternative work center
C : the subcontractor

Table 7. Percentage of the production quantity allocated to each work center for each case of the solution procedure for the example problem

product		1	1	1	1	2	2	2	2	3	3	3	3
operation		1	2	3	4	1	2	3	4	1	2	3	4
LP	A	89	60	12	15	14	100	100	100	100	16	0	0
	B	5	0	10	26	86	0	0	0	0	84	100	100
	C	6	40	78	59	0	0	0	0	0	0	0	0
ABA	A	71	42	10	16	30	79	87	64	77	57	13	21
	B	18	16	9	23	67	19	13	36	23	43	87	79
	C	11	42	81	61	4	2	0	0	0	0	0	0
ECF	A	51	34	23	29	36	58	46	53	49	46	32	27
	B	38	24	6	28	63	35	33	47	51	54	57	73
	C	11	42	71	43	1	7	21	0	0	0	11	0
LCF	A	88	29	5	18	37	58	72	82	93	44	29	48
	B	11	26	24	33	62	27	17	18	7	56	61	51
	C	1	45	71	49	1	15	11	0	0	0	10	1
EDF	A	55	49	25	31	24	49	27	51	46	39	37	28
	B	36	37	19	36	50	35	28	43	43	52	37	45
	C	9	14	56	33	26	16	45	6	11	9	26	27
LDF	A	75	23	69	9	41	4	51	50	80	82	65	10
	B	12	41	26	34	44	26	37	43	14	15	26	42
	C	13	36	5	57	15	70	12	7	6	3	9	48
ECE	A	56	37	25	22	34	52	57	48	50	52	36	32
	B	37	21	8	24	66	33	40	52	49	48	61	68
	C	7	42	67	54	0	15	3	0	1	0	3	0
LCE	A	88	30	12	18	37	57	76	78	93	51	23	40
	B	11	26	14	28	61	29	17	12	7	49	73	60
	C	1	44	74	64	2	14	7	0	0	0	4	0
EDE	A	52	43	51	22	27	19	44	51	54	41	40	21
	B	31	28	42	29	58	13	31	45	45	55	52	29
	C	17	29	7	49	15	68	25	4	1	4	8	50
LDE	A	74	10	79	11	49	1	31	50	84	84	82	0
	B	11	43	21	33	43	42	45	42	12	16	16	28
	C	15	47	0	56	8	57	24	8	4	0	2	72

Table 8. The average percent deviation of the solution and the average total cost per order of each procedure for the example

procedure	average percent deviation	average total cost per order
LP	0	13807
ABA	13	14430
ECF	23	17878
LCF	16	17921
EDF	30	16808
LDF	33	16839
ECE	22	17739
LCE	15	18269
EDE	31	16860
LDE	36	17045

Table 9. Processing cost and price data of each work center in the example problem

work center		1	2	3	4	5	6	7	8	9	10
processing cost per unit time		15	10	10	6	7	15	14	8	9	14
shadow price of LP model		16	11	19	15	16	6	13	16	4	0
ABA	charge price	27.1	20.8	24.7	18.2	27.2	24.4	27.5	26.4	9.5	14.0
	utilization	0.98	0.99	0.98	1.00	1.00	1.00	0.99	1.00	0.93	0.59

We construct 27 problems, each of which has a different combination of parameters of mean interarrival time of (80,100,120), the product-mix of (50:30:20, 30:40:30, 20:30:50), and the penalty cost per unit time of (1,2,4).

In Table 10, we compare average values of cost components and the total cost of 27 problems. We can see that ABA procedure outperforms all the other heuristic procedures in the total cost. The total cost of ABA procedure is found to be the lowest among those of all the heuristic procedures for all 26 problems. Note that the relative distributions of cost components of ABA resemble those of LP formulation more closely than any other heuristic procedure. ECF, LCF, ECE, and LCE, all of which are procedures with least-cost subcontracting rule, result in higher penalty cost than those of ABA. Moreover, subcontracting costs are higher for EDF, LDF, EDE, and LDE compared with the results of ABA.

The average percent deviation of solutions of ABA from those of LP formulation is compared with other heuristic procedures in Table 11. It is calculated by averaging the average deviation of each simulation run, which may be expressed as in equation (22). It shows that solutions of ABA resemble those of LP more closely than other heuristic procedures.

The number of orders in the system and the number of orders completed during the simulation period are compared with each other. ABA is ranked in third place among all the procedures on both of the evaluation measures.

Table 10. Comparisons of the average values of cost components and the total cost per order for 27 problems

procedure	subcontracting cost	in-house process. cost	penalty cost	cost of product 1	cost of product 2	cost of product 3	total cost
LP	3187	10257	0	12590	11454	16256	13386
ABA	3413	10415	25	13126	12838	15503	13814
ECF	2162	11182	1557	11802	15153	17364	14937
LCF	2465	10213	2909	13025	16132	17083	15622
EDF	4798	10781	1	10932	15756	19296	15526
LDF	5949	9847	29	11462	16997	17998	15690
ECE	1896	11346	1715	12465	15064	17123	14927
LCE	2377	10409	3078	14083	16066	17052	15797
EDE	4700	10785	0	10691	15714	19213	15417
LDE	6116	9903	12	10720	17251	19028	15860

Table 11. Comparisons of average percent deviation, number of orders in the system, and total number of orders completed for 27 problems

heuristics	average percent deviation	average no. of orders in the system	total no. of orders completed
ABA	5.0	41	467
ECF	7.8	58	438
LCF	5.9	70	455
EDF	9.9	40	464
LDF	10.3	44	477
ECE	7.7	62	433
LCE	5.6	73	446
EDE	9.9	40	466
LDE	11.2	44	476

5. Conclusions

We have proposed an auction-based task assignment method which can be used in a large automated manufacturing shop floor which has intelligent and powerful local controllers connected with each other by communication links.

The proposed control scheme is characterized as a distributed information processing, a distributed decision making and a heterarchical market-like model.

In this scheme, we have suggested two types of actors, resource agent and part agent. We assume that part agent is in charge of orders of a specific product and has the objective to deliver orders at a cost as low as possible while resource agent is a profit maker who is in charge of a work center. We have suggested a detail procedure where task assignments are made through an auction-based negotiation process between part agents and resource agents instead of the isolated determinations.

To provide a rationale for the auction-based procedure, we analyzed a linear programming model for the static version of job allocation problem. It is shown that the independent selection of the least-cost work center for each job results in the global optimal solution in the job allocation problem only if the processing cost of each work center is properly adjusted.

The performance of the auction-based allocation (ABA) procedure in this paper is compared with those of eight conventional dispatching procedures by a simulation experiment. The results of the experiments show that the solution of ABA resembles the optimal solution of LP formulation most closely and ABA procedure obtains the least-cost solutions on the average among the heuristic procedures simulated.

Although the price system in this paper is very simple, the results of performance evaluation is promising. A more elaborate price system will improve the performance of the task assignment further. Studies are needed on how to make resource agent and part agent more intelligent. We can make them have more forecasting capabilities, reservation functions, and adaptive control functions, etc. In this paper, although every agent may have different weights on objectives because of a different state of each agent at the moment, the basic logic for negotiation procedure is the same for all the agents. But, in a more general case, agents may have different negotiation strategy from each other. Research is needed on the effects of heterogeneous bidding algorithms on the performance of the global system.

References

1. Blackstone, Jr., J. H., Phillips, D. T. and Hogg, G. L. (1982). A State-of-the-art surveys of dispatching rule for manufacturing job shop operations. *Int. J. Prod. Res.*, Vol. 20, No. 1, 27-45.
2. Ishii, N. and Talavage, J. J. (1991). A transient-based real time scheduling algorithm in FMS. *Int. J. Prod. Res.*, Vol. 29, No. 12, 2501-2520.
3. Nakasuka, S. and Yoshida, T. (1992). Dynamic scheduling system utilizing machine learning as a knowledge aquisition tool. *Int. J. Prod. Res.*, Vol. 30, No. 2, 411-431.

4. Vamos, T. (1985, August). Cooperative systems based on non-cooperative people. *IEEE Control Systems Magazine*, 9-14.
5. Davis, R. and Smith, R. G. (1983). Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, Vol. 20, 63-109.
6. Smith, R. G. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, Vol. c-29, No. 12, 1104-1113.
7. Lin, G. Y. and Solberg, J. J. (1992). Integrated shop floor control using autonomous agents. *IIE Trans.*, Vol. 24, No. 3, 57-71.
8. Lin, G. Y. and Solberg, J. J. (1991). Effectiveness of flexible routing control., *The Int. J. of Flexible Manufacturing Systems*, Vol. 3, 189-211.
9. Yang, E. H., Barash, M. M. and Upton, D. M. (1993). Accommodation of priority parts in a distributed computer-controlled manufacturing system with aggregate bidding schemes. *The 2nd Industrial Engineering Research Conference Proceedings*, 827-831.
10. Shaw, M. J. (1989). Dynamic scheduling in cellular manufacturing systems: A framework for networked decision making. *J. of Manufacturing Systems*, Vol. 7, No. 2, 83-94.
11. Shaw, M. J. (1987) A distributed scheduling method for computer integrated manufacturing: The use of local area networks in cellular systems. *Int. J. Prod. Res.*, Vol. 25, No. 9, 1285-1303.
12. Butler, J. and Ohtsubo, H. (1992). ADDYMS: Architecture for distributed dynamic manufacturing scheduling. *Artificial Intelligence Applications in Manufacturing*, Edited by A. Famili, D. S. Nau, and S. H. Kim, AAAI Press, 199-213.
13. Parunak, H. V. D., Irish, B. W., Kindrick, J. and Lozo, P. W. (1985, December). Fractal actors for distributed manufacturing control. *The second conference on artificial intelligence applications*, 653-660.
14. Maley, J. G. (1987). *Part flow orchestration in distributed manufacturing processing*, Ph. D. Dissertation, School of Industrial Engineering, Purdue University, West Layfayette, IN.
15. Ting, J. J. (1990). A cooperative shop-floor control model for computer-integrated manufacturing. *Proceedings of CIM CON '90*, Albert Johnes, Editor, U. S. Government Printing Office, Washington, D. C., 446-465.

16. Upton, D. M. (1991). An organic structure for a computer-controlled manufacturing system. Working paper, Graduate School of Business Administration, Harvard University.
17. Morton, T. E., Pentico, D. W. (1993). *Heuristic Scheduling Systems*, John Wiley & Sons, Inc., 30-34.
18. Murty, K. G. (1976). *Linear and Combinational Programming*, John Wiley & Sons. Inc., 37.