BEBR

FACULTY WORKING
PAPER NO. 1244

A Distributed Scheduling Method for Computer Integrated Manufacturing: The Use of Local Area Networks in Cellular Systems

Michael J. Shaw

College of Commerce and Business Administration
Bureau of Economic and Business Research
University of Illinois, Urbana-Champaign

BEBR

FACULTY WORKING PAPER NO. 1244

College of Commerce and Business Administration
University of Illinois at Urbana-Champaign

April 1986

A Distributed Scheduling Method for Computer Integrated Manufacturing: The Use of Local Area Networks in Cellular Systems

Michael J. Shaw, Assistant Professor
Department of Business Administration

Presented at Symposium of Real-Time Optimization in Automated Manufacturing Facilities, National Bureau of Standards, USA, January 1986.

Abstract

This paper describes a distributed scheduling approach that takes into account characteristics of the communication network in the computer-integrated manufacturing environment. The approach is based on a network-wide bidding scheme wherein the scheduling decision is made by collecting the price of each manufacturing cell for taking on the job. I also describe the formalism and model for the distributed scheme that can be incorporated in a communication protocol. A simulation study has been conducted to compare the performance of different strategies or heuristics employed in the scheduling method.

1. Introduction

Flexible automation--automation that can handle a large and constantly changing variety of produced items--has become essential in the efforts to improve manufacturing productivity. The use of computers on the factory floor adds programmability and thus versatility into manufacturing systems. More important, computers also provide on-line execution of planning, decision-making, and control of the processes, coordinating the activities occurring in various parts of the system.

An emerging architecture for such computer-integrated manufacturing (CIM) systems is the cellular system, as shown in Figure 1, consisting of flexible cells (Bourne [1982], Cutkosky [1984], and Simpson et al. [1982]); each cell can communicate with other cells through a local area network (LAN). Such cellular manufacturing systems have played an increasingly important role in the design of the fully automated systems for many reasons; among them are the reduced machine set-up time, reduced tooling, the simplification of planning and control, reduced in-process inventory, the near-constant load-time, and system modularity (McLean [1983], Greene [1984], and Sikha [1984]).

Insert Figure 1 Here

This paper is concerned with the scheduling aspect of the cellular system, where jobs arrive at the system dynamically over time and the system behaves like a network of queues. It is a loosely coupled system of cooperating flexible cells in which each cell can be set up

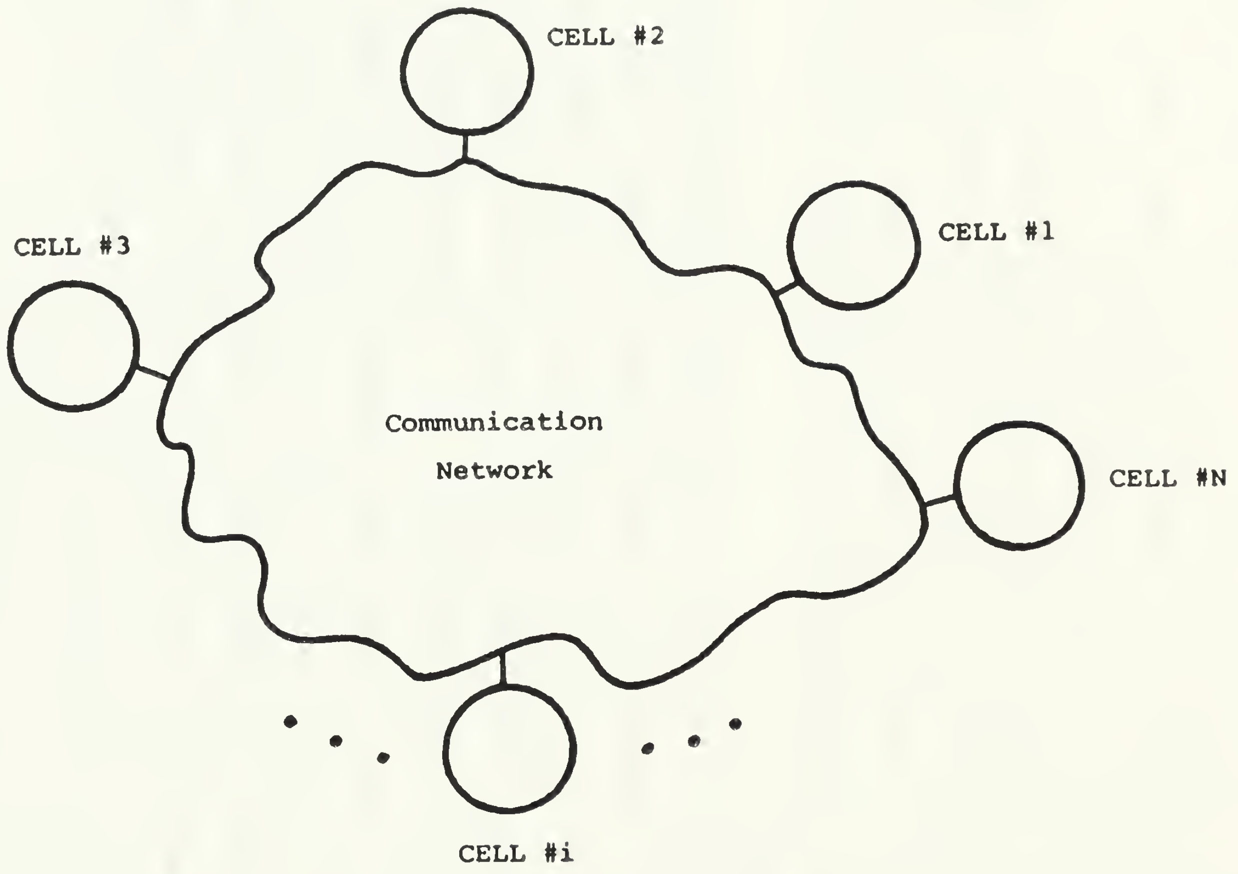


Figure 1. A Conceptual Model of the Cellular CIM

to produce items belonging to a range of several part families, but in which a particular cell holds a competitive advantage over other cells on a specialized subsets of the jobs. A job consisting of operations of different families may be collectively manufactured by several cells; for a overloaded cell, some jobs are tranferred to other temporarily underloaded cells with similar functionalities. These operational decisions can be viewed as the task-assignment problem aiming at matching given jobs with the most capable cells.

The task-assignment problem has been studied in previous scheduling research; assorted techniques have been used in solving the problem, such as the graph theoretic method, queueing network analysis, mathematic programming, or the use of heuristics rules (Baker [1976] and French [1982]). The scheduling problem in flexible manufacturing--characterized by the shorter lead-time, machine flexibility, and dynamic job arrivals--has been studied by simulation techniques (Shanthikumar and Sargent [1980] and Chang et al. [1984]), queueing network analysis (Solberg [1977] and Kimenia and Gershwin [1985]), and artificial intelligence (Shaw [1984] and Shaw and Whinston [1985a] [1985b]). The scheduling methods and characteristics for cellular manufacturing are described in McLean et al. [1982] and Sinha and Hollier [1984]. Mosier et al. [1984] developed and evaluated dispatching rules for scheduling jobs among manufacturing cells formed by group technology. The importance of appropriately incorporating LAN technology in CIM systems has been pointed out by McLean et al. [1983], Cutkosky et al. [1984], and Keil and Dillon [1985]. But there has not been any work evaluating the impact of the LAN technology on the way the scheduling is performed; nor is there any research that considers the networking environment in designing the scheduling method for CIM.

The scheduling method described in this paper takes into account the characteristics of a local area network for communication. The system is treated as a loosely-coupled network of cooperating cells and the scheduling is carried out by a network-wide bidding scheme for determining the assignment of cells to given jobs dynamically. It is a distributed scheduling method in that no node in the network has greater importance, as far as scheduling is concerned, than any other node. Moreover, this scheduling method can incorporate different dispatching rules and can be used for both task allocation and resource allocation. As such, this is the only research in the manufacturing area to date that takes into account the use of local area networks for executing job scheduling and I shall show that there are ample advantages in doing so.

The remainder of this paper will be organized as follows: Section 2 discusses the networking characteristics of the cellular flexible manufacturing systems; Section 3 presents the details of the distributed bidding algorithm; the modeling and implementing of the scheduling scheme is discussed in Section 4; in Section 5, a simulation study is described to evaluate the performance of the bidding scheme; finally, Section 6 concludes the paper by summarizing the characteristics of the methodology developed.

2. The Networking Environment in CIM

In the effort to achieve computer-integrated manufacturing, it has become increasingly important to integrate islands of factory automation and to establish efficient communication means among computer-controlled machines. The economic and technical characteristics of the LAN technology make it very suitable to achieve such integration and communication in CIM systems. A LAN is a data-communication network that services geographical areas spanning distances of no more than a few kilometers. It allows independent devices to communicate with each other, usually implemented with inexpensive transmission medium and interface devices. There are a number of criteria to consider in designing a LAN:

- the transmission medium;
- the transmission technique;
- the network topology; and
- the access control scheme.

The transmission medium that has been employed includes twisted wire pairs, coaxial cables, and optical fibres. There are two primary transmission techniques: broad band and base band. While it is the trend that the coaxial cable will continue to be the widespread choice for general purpose LANs, the choice between the two transmission techniques is less obvious and really depends on the specific communication needs. (For the comparison between broad band and base band techniques, see Krutsch [1981].) The basic topologies currently used for LANs are the ring, the bus, and the star.

As opposed to the star topology with its central control unit, both the bus and the ring topologies provide distributed control. Among the three, the bus topology is the one best suited for the CIM environment for a number of reasons. First, the tree-like organization makes intallation relatively easy on the factory floor. Second, since each attached device has independent access to the bus, adding or removing a device can be done without disturbing the rest of the network. Lastly, the bus topology is easy to service and more reliable than either the star or the ring (Maira [1986]).

For distributed control, an access-control scheme is executed by the interface unit when it has received a message packet from its attached device for transmission onto the network. The access-control scheme is fully distributed; that is, each interface unit can determine when it is appropriate to transmit a packet based on what is observed locally and what the steps of (locally stored) access-control scheme dictate. There are no explicit signals from cental controllers that give out permission to transmit. Two most frequently used access-control schemes are (1) the CSMA/CD (carrier sense, multiple access with collision detection) scheme, as typified by the Ethernet developed by Xerox (Metcalfe and Boggs [1976]); and (2) the token-passing scheme (Box [1982], which is the access control scheme incorporated in the Manufacturing Automation Protocol.

Quickly emerging as the industrial networking standards, the Manufacturing Automation Protocol (MAP) is based on the token-bus network environment, which can be characterized as follows: (1) The network is topologically a logical ring on a physical bus, wired together with a broad-band communication bus. (2) It transmits information by data packets. The sending station designates in the packet the address of the receiving station; the bus topology permits every station to hear all transmissions. (3) The stations monitor all bits passing by in the bus through an interface. A receiving station examines the address field of the message packet; if it recognizes its own address, it takes the appropriate action; if not, it ignores the message. (4) The access control scheme is based on token-passing. It uses a special bit pattern, called the token, circulates around the network. When a station wants to transmit a packet, it is required to seize the token and remove it from the bus before transmitting; after a station has finished transmitting the last bit of its packet, it must regenerate the token so that some other station can grab the token and start transmitting data. Only a station with the token can transmit packets, and then only for a predetermined period of time. Such a token-passing mechanism ensures that all stations have an opportunity to send message packets without any conflicting transmission.

The architecture of MAP is based on a seven-layer network architecture, referred to as the Open System Interconnection (OSI) model (Tanenbaum [1981]). The networking environment based on the OSI model is hierarchical and the communications tasks are divided into seven subtasks, or layers. Each layer provides a set of communication-related services to the layer above; the top layer, the application layer, supports the necessary communication activities with other stations for the user's programs in that station.

Associated with such a networking environment, there are two possible control structures underlying the scheduling decisions: (1) to use a centralized scheduler in charge of job assignment. The scheduler keeps track of the whole cellular system by a global database; and (2) to use a distributed scheduling scheme and let the set of cells perform scheduling based on local information (Schoeffler [1984)]. By way of comparison, scheduling with distributed control has these advantages: (1) better reliability--the system degrades gracefully in the face of scheduler breakdown; (2) upward extensibility--the control structure remains the same with additions of new cells to the extent that the network is not saturated; (3) improved performance--the scheduling performance can be improved because the scheduling is achieved by parallel processing and also because of the elimination of the bottleneck associated with global scheduler; and (4) cost-effectiveness--it is more cost-effective because of the smaller processing requirements on the computers and less communication activities needed for global updating. The implications of control structures to the scheduling method are summarized in Table 1.

Insert Table 1 Here

The adoption of distributed scheduling method implies the need for a new type of information-control mechanism for coordinating manufacturing activities. Since there is no centralized master controller directing the activities of individual cells, it becomes essential that the cells have to be able to reach scheduling decisions by collective, concerted efforts. Two major issues warrant attention: (1) an effective task allocation scheme among cells to ensure that all the resources can be efficiently utilized, and (2) the coordination mechanism exercised among the cells, carrying out manufacturing tasks cooperatively. The network-wide bidding scheme described in this paper can achieve these two functions.

	Centralized System	Diatributed Net
Control Structure	centralized	decentralized
Execution of Scheduling	a master scheduler	a scheduler in each cell
Control Mechanism for Scheduling	master-slave control with unidirectional message-passing	coordination through exchanging messages
Vulnerability to Scheduler's Failure	entire system would stop	only that particular cell would be disrupted
Manufacturing Database Management	a global database	distributed databases
Maintaining Dynamic System Information	constant updating through communication messages	local updating without communication activities

Table 1. Implications of Control Structures to Scheduling

Such an approach essentially treats the scheduling problem by a multiagent problem-solving paradigm: because the whole scheduling problem is too complicated, the set of problem-solving agents--the cells--carry out the tasks collectively. Just as in human organizations, bidding is employed as a mechanism for coordinating the execution of tasks among the cells. This paradigm was developed by research in artificial intelligence (Davis [1983] and Shaw [1985]) and has been applied to various types of distributed systems such as the sensor network (Smith [1980]) or computer networks (Malone [1983] and Ramamritham [1984]).

3. The Distributed Scheme for Dynamic Scheduling

In the network-wide bidding scheme, when a cell needs to initiate the task assignment algorithm for one of its jobs, it begins with broadcasting a task-announcement message through the LAN to other cells and takes on the role as the manager cell of the job. Those cells that receive this message will, in turn, transmit a bidding message which contains its estimation of the earliest finish time, the surrogate for the "price" of the job if assigned. When all the bids have returned, the manager cell then selects the cell which can finish the job the earliest to perform the task. The corresponding workpiece is then transferred to the cell selected, or the contractor cell.

Task Announcement

When a job finishes its operations in a cell, the cell's control unit will check to see if there are any remaining operations to be done. If all operations have been completed, the workpiece is sent to the storage area; otherwise, the cell's control unit would have to make the decision regarding which cell the job should go to next. Keeping the job in the same cell is also a valid decision, but this has to be made after the performance data from other cells are collected and compared through bidding.

Associated with each task announcement packet would be a deadline before which the bid must be submitted. To make sure the deadline for bid return is set in such a fashion that all the qualified cells have enough time to evaluate the task and return the bid, the bidding interval Δt enforced by the deadline should be postulated to satisfy a lower-bound condition: $\Delta t \geq 2 \times t_1 + t_2$, where t_1 is the communication delay and t_2 is the estimated time necessary for task evaluation.

In the cellular manufacturing system, three types of manufacturing cells may exist: (1) flexible cells, where general-purpose machines are used and the set-up is flexible for performing a wide-ranging family of operations; (2) product-oriented cells, where a certain type of product is manufactured, e.g., gear cell for producing gears; and (3) robot assembly cells, where robots are used for putting subassemblies together. Depending on the set-up of a flexible cell or a robot assembly cell, the cell's control unit would give different performance estimates at different moments. The product-oriented cells, on the other hand, have relatively more static functions in terms of the set of operations they perform. For a job requesting an operation that can be performed in these product-oriented cells, the task-announcement message can be directly addressed to the destination cell. The scheduling of jobs can be accelerated by such "focused addressing."

Bidding

When a cell receives a task-announcement message from the communication network, it first matches the task description with its capability-list and checks whether the required operations are within its capabilities. A bid for the task is returned only if the cell can perform the task. The cell then proceeds to calculate the bidding function which has the following three components: (1) The estimated processing time, which is calculated by a routine based on the machining parameters specified in the task-announcement packet, such as cutting speed, raw material, depth of cut, surface finish requirement, cutting tools' wearing condition, current set-up, and lubrication temperature; (2) the estimated waiting time, which is calculated by adding up the estimated processing time of the jobs in the queue; and (3) the estimated travel time, which is calculated based on the travel distance between the two cells.

This particular bidding function implies that each flexible cell submits its estimation on the earliest time it can finish the task if assigned. By assigning the task to the lowest bidder, the manager cell essentially is executing the earliest-finishing-time (EFT) heuristic for dynamic scheduling (Baker [1974]). Other dispatching heuristics can also be incorporated. For example, if the bidding function is determined by the estimated processing time of each cell, then the scheduling is essentially based on the decentralized version of shortest-processing-time (SPT) dispatching, which has been shown to give good scheduling performance to dynamic job shop (French [1982]) and flexible manufacturing systems (Chang [1984]). This flexibility enables the bidding scheme to integrate very well with the traditional scheduling methods. The simulation study in Section 5 will examine the performance implications of different bidding functions.

If jobs arrive at the system in clusters, then there is a possible flaw in the way the waiting time is estimated. That is, when a cell is granted more than one job simultaneously, the actual waiting time will be greater than the estimated waiting time, since the estimation is calculated disregarding the other jobs, some of which may end up in the same cell. For dealing with such an environment, the distributed algorithm needs to be modified so that a cell will rank the announced tasks and only bid on the most preferred task and the bidding algorithm will be executed in periodical cycles. Such an arrangement, however, would prolong the time taken for making the assignment decision.

Bid Evaluation and Task Awarding

When the deadline for bid submission is due, a bid-evaluation procedure is carried out by the cell that originally announced the task. All the bids submitted for this task have been put in a list, ranked by the value of each bid. In our algorithm, the bid of cell i is calculated based on the earliest finish time of each task if the task is assigned to cell i. The scheduler of the manager cell then chooses the cell with the smallest bid, i.e., which can finish processing the task the earliest.

Once bid-evaluation is completed, an award message is sent to the best bidder, informing the awardee of the pending job so that the cell which has been awarded the task will take this new task into consideration in the subsequent calculation of earliest-finish-time in bidding for future jobs. This task-awarding information also enables the awardee cell to start loading part programs for the new task. The local scheduler of the awardee cell will take the newly assigned job into consideration in the next scheduling cycle. The bidding scheme is schematically shown in Figure 2.

Insert Figure 2 Here

The bidding scheme is appealing for scheduling because (1) the bidding procedure known to be an efficient allocation mechanism (for example, see Oren [1975]), (2) different dispatching heuristics can be incorporated to carry out varying scheduling objectives, (3) it can be executed dynamically by message passing (essentially, the bid submitted by each cell reflect the "price" for cell to embark on the task and therefore, the same scheme can be applied to resource allocation as well and if any necessary manufacturing resources, e.g., fixtures, tools, part programs, etc. are not readily available, then the bid should include the price for getting the resources, i.e., the time it will take for the supporting resources to arrive); and (4) it is a distributed mechanism and can be implemented on top of the seven-layer network environment, as will be described in Section 4. The bidding scheme and the corresponding information flows are shown in Figure 3.

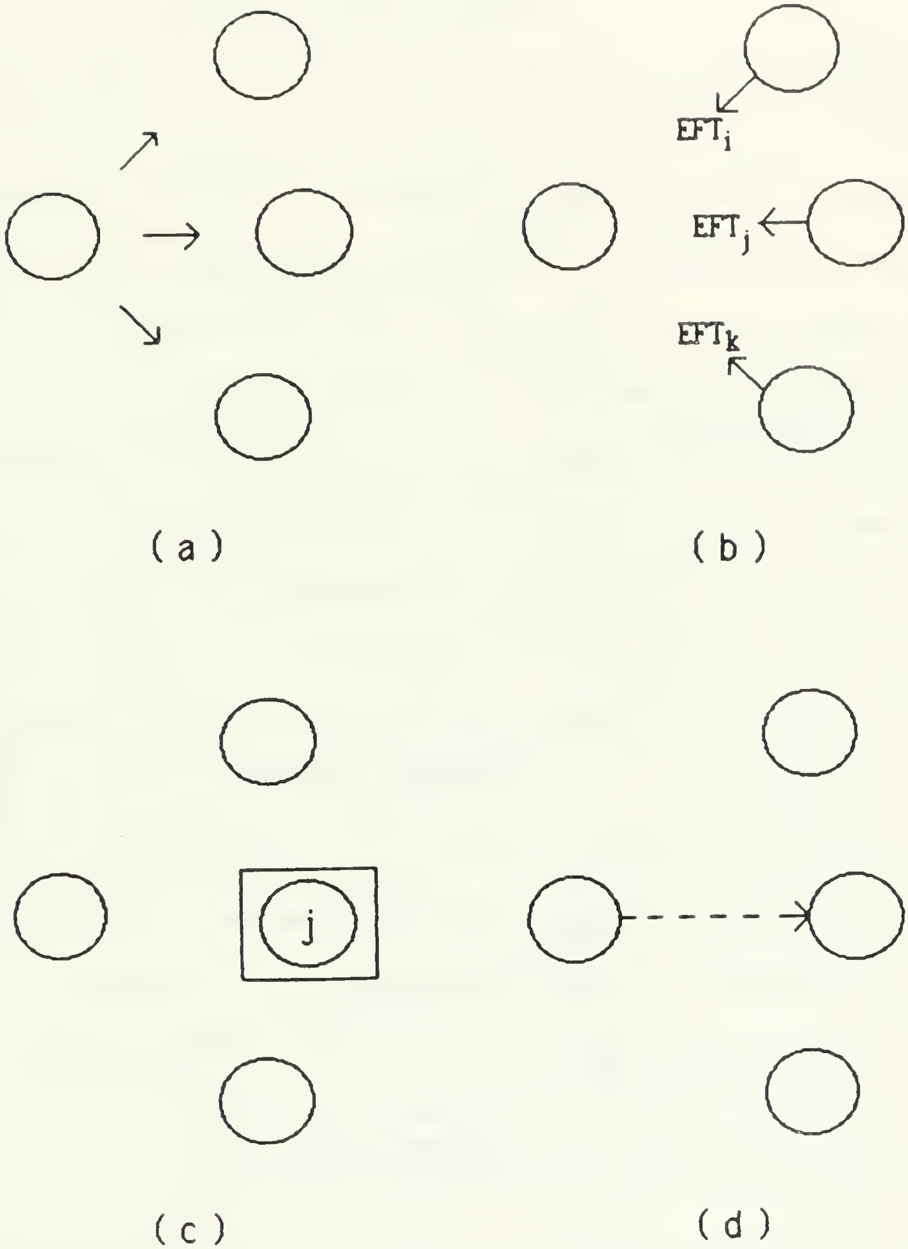


Figure 2. The Bidding Scheme

Insert Figure 3 Here

Under the distributed control scheme, the dynamic system information such as cell status, location of parts, position of tools, progress of jobs, etc., is managed by a distributed database system. Each cell maintains its own local world model, while systematically coordinating with other cells through task sharing and bidding. By eliminating the necessity to collect dynamically changing system information in a global database, the possible bottleneck and the communication activities for constant updating are avoided.

It is necessary to add that, based on the distributed problem solving paradigm discussed in Section 2, the bidding scheme is part of a two-level scheduling approach for cellular CIM systems. The first-level problem is the task assignment problem described in this paper, and the second-level problem is the local scheduling problem within each cell. Shaw and Whinston [1985a] presented a knowledge-based system for handling the scheduling problem within each cell. The flowchart of the two-level scheduling approach is shown in Figure 4.

Insert Figure 4 Here

---

4. Modeling and Implementing the Distributed Scheduling Scheme

To implement the distributed scheduling method in the cellular manufacturing system, three issues should be addressed:

(1) the design of a network interface language that enables effective communication among cell-host computers;

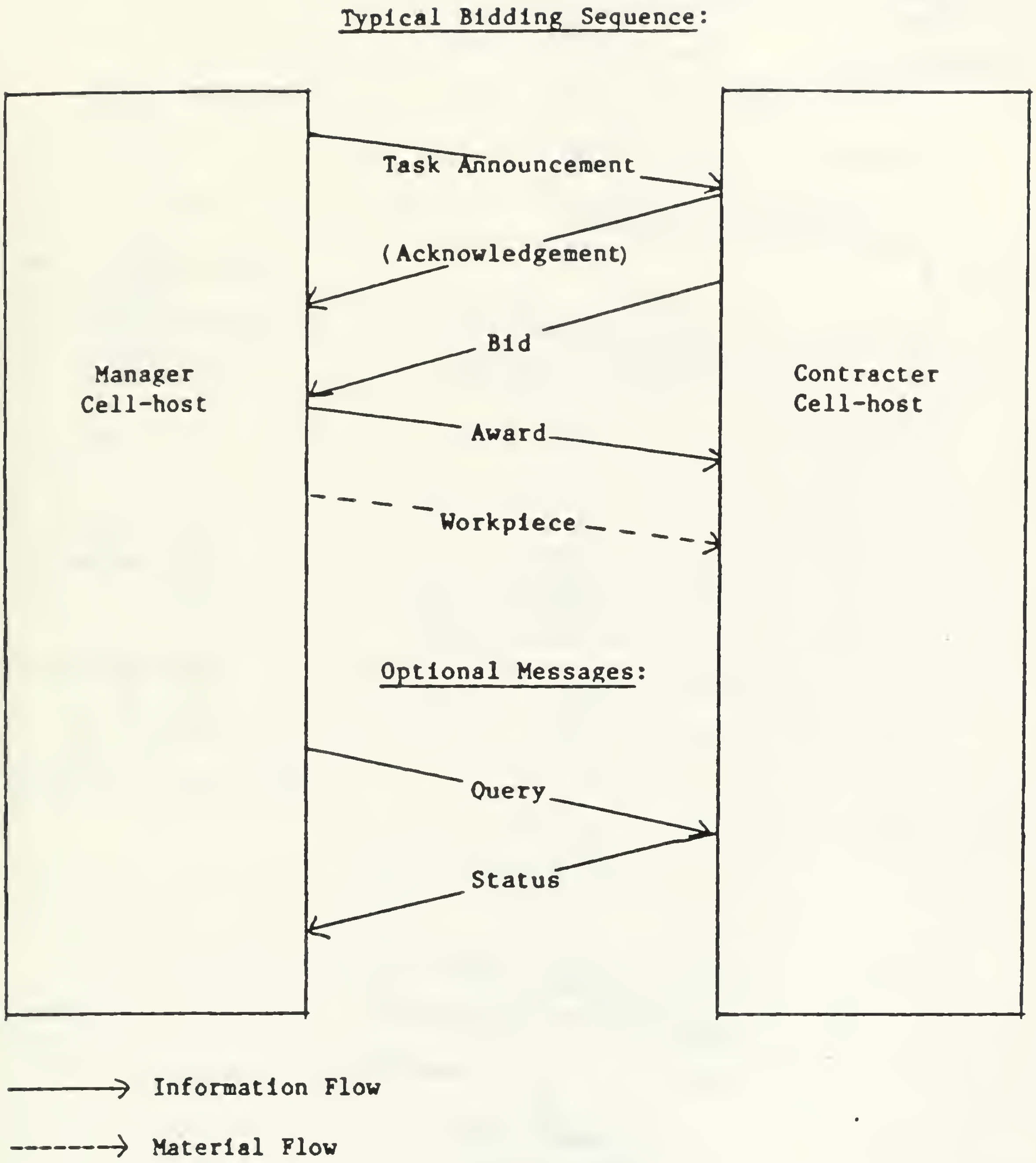


Figure 3. Information Flows in the Bidding Sequence

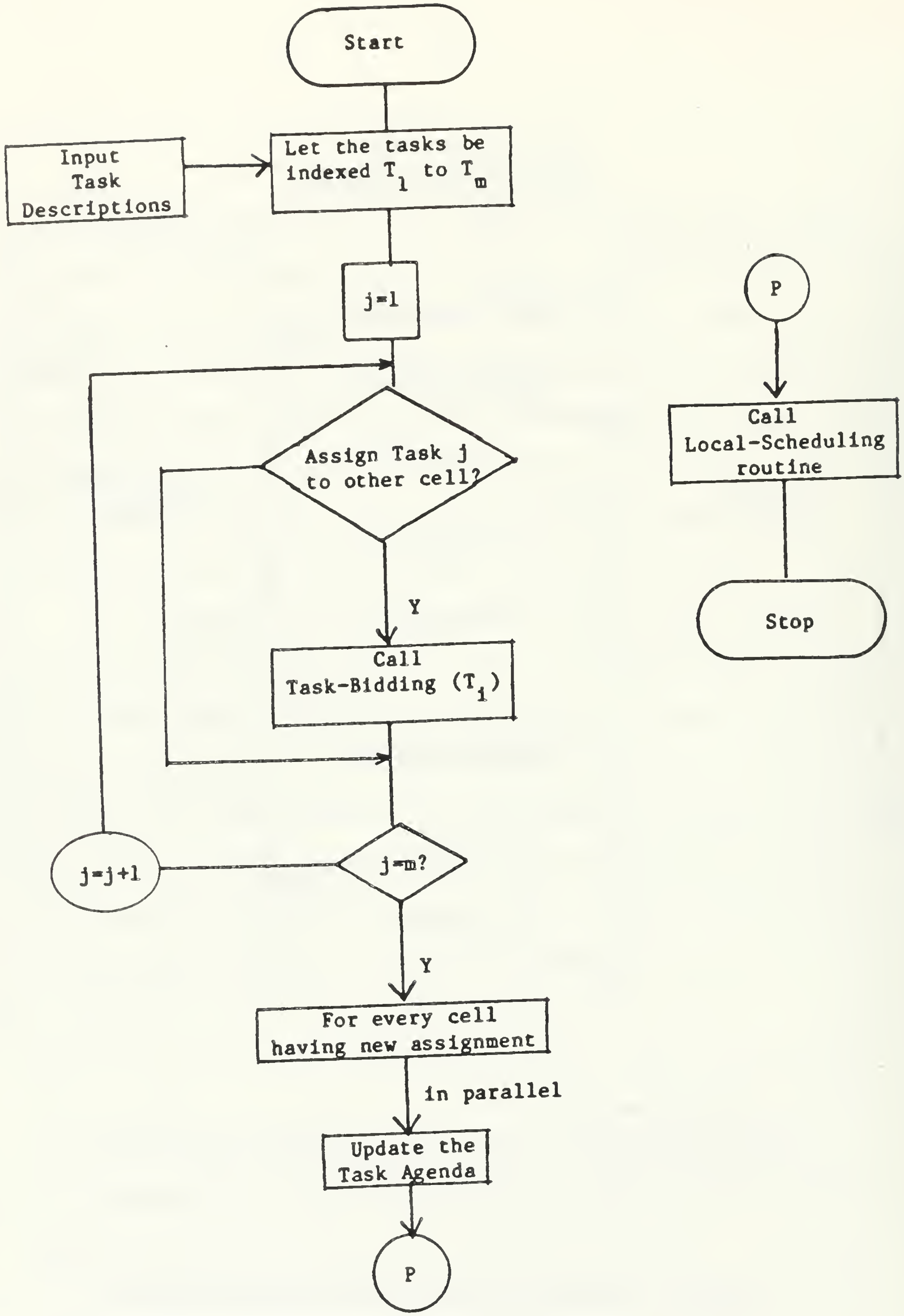


Figure 4. The Flowchart for the Two-Level Scheduling Approach

(2) a model for the bidding scheme based on which the individual cell can perform task bidding correctly; and

(3) the implementation of the bidding scheme based on the language design in (1) and the model in (2). Such an implementation should consider the characteristics of the local area network employed in the system.

A common interface language is required to enable cell-host computers to communicate their intentions and share information with one another. This parallels how people communicate in human organizations. For this purpose, a formalism for the messages needs to be specified so that the interface language is consistently used and should be recognizable to all host computers. The format for the messages used in the distributed scheduling method is shown in Figure 5. The format is based on phrase-structure grammar specified in Backus-Naur Form (BNF).

Insert Figure 5 Here

A model for the bidding scheme is needed to specify the proper sequencing of actions to carry out task bidding. Such a model must be able to represent asynchronous parallel processes, since the bidding of several different tasks may occur concurrently. Because the cells are asynchronous, loosely coupled units, there are strict requirements for communication and coordination between cells. Thus a good formal model for the bidding process should describe two aspects of the decisions and activities involved:

```
<MESSAG> :== <ADDRESSEE><ORIGINATOR><TEXT>

<ADDRESSEE> :== [NET-ADDRESS]|[SUBNET-ADDRESS]|[NODE-ADDRESS]

<ORIGINATOR> :== [NET-ADDRESS]|[SUBNET-ADDRESS]|[NODE-ADDRESS]

<TEXT> :== <TASK-ANNOUNCEMENT>|<BID>|<ACKNOWLEDGEMENT>|<AWARD>|

           <QUERY>|<STATUS>

<TASK-ANNOUNCEMENT> :== TASK-ANNOUNCEMENT[TASK-ID][ELIGIBILITY]

                        [TASK-ABSTRACTION][DEADLINE]

<BID> :== BID[TASK-ID][EARLIEST-FINISHING-TIME]

<ACKNOWLEDGEMENT> :== ACK[TASK-ID]

<AWARD> :== AWARD[TASK-ID][EXPECTED-ARRIVAL-TIME]

<QUERY> :== QUERY[TASK-ID]

<STATUS> :== STATUS[TASK-ID][STARTING-TIME][COMPLETION-TIME]
```

Figure 5. The Syntax of the Interface Language

(1) A procedural representaton of the communication and coordination mechanisms between the cells; and

(2) A declarative representation of the local decision-making process when a cell receives messages.

Shaw [1984] used the augmented Petri net (APN) to model the bidding scheme. An augmented Petri net is an integration of two representational models: production rules are used to represent the decisions involved in distributed scheduling and the Petri net is used to model the procedural knowledge of the bidding scheme. The APN model has been proven effective in modeling asynchronous concurrent processes where the combination of state variables grows exponentially (Zisman [1978]). The APN model for the distributed scheduling scheme is shown in Figure 6, where each transition, represented by a vertical bar in the graph, corresponds to a production rule. In essence, the Petri net in the model regulates interactions between production rules. Shaw [1984] showed that the APN model is isomorphic to a rule-based system with Petri net language as the control language. Thus the bidding scheme, as represented in the APN model, can be implemented by a rule-based system with explicit procedural control. The set of production rules used in the APN model is shown in Figure 7.

Insert Figures 6 & 7 Here

For executing correct communication activities in a network, a communication protocol is required. Conceptually, a protocol is a set of rules for each communicating node to follow to transmit data through the network. In a CIM system, to ensure that processes at

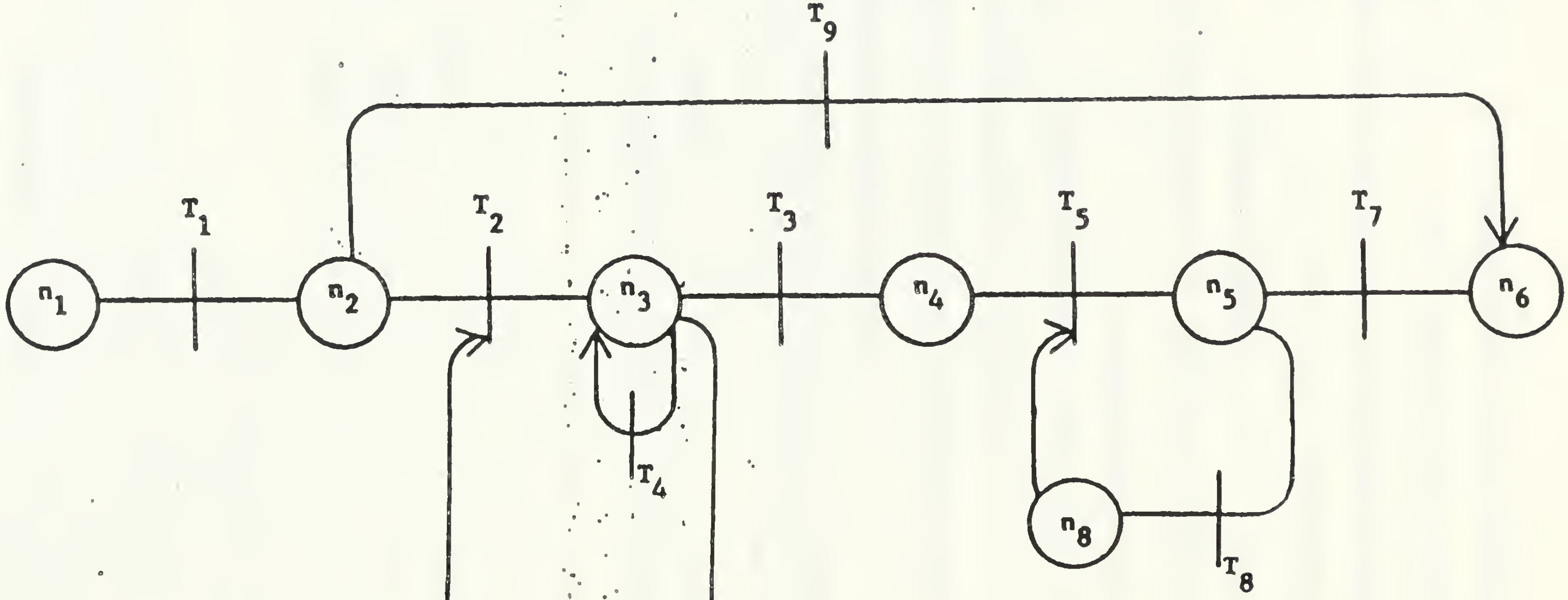


Figure 6(a). The APN Model

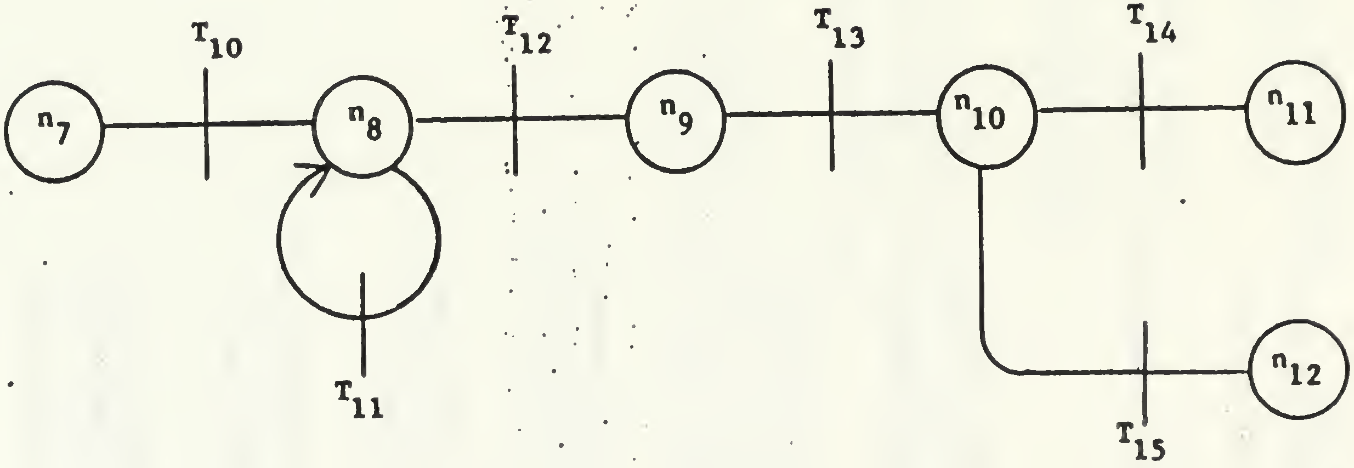


Figure 6(b). The APN Model

T_1

```
if (NEW-TASK task)
then (TASK-INITIALIZATION task)
```

T_2

```
if (TASK-EVALUATE task)
then (TASK-ANNOUNCEMENT task)
```

T_3

```
if (BID-RETURN bid)
   AND (LEQ time-now deadline)
then (BID-PROCESSING bid)
```

T_4

```
if (LEQ time-now deadline)
then □
```

T_5

```
if (GT time-now deadline)
   AND (NE bid-list blank)
then (BID-AWARD bid-list)
```

T_6

```
if (GT time-now deadline)
   AND (EQ bid-list blank)
then (REANNOUNCE task)
```

T_7

```
if (REPLY-TO-AWARD accept)
then (LIST-ASSIGNMENT task)
```

T_8

```
if (REPLY-TO-AWARD reject)
then (RE-AWARD task)
```

T_9

```
if (NOT(TASK-EVALUATE task))
then (LIST-AGENDA task)
```

T_{10}

```
if (TASK-ANNOUNCED task)
   AND (BID-EVALUATE task)
then (TASK-RANKING task)
```

Figure 7. Production Rules in the APN Model

```
$T_{11}$    if (EQ(PROCESSOR-FOR-TASK task)busy)
            then (LIST-ACTIVE-TASK-ANNOUNCEMENT task)

$T_{12}$    if (EQ(PROCESSOR-FOR-TASK task)idle)
            then (BID-REPLY(BID-SELECT a-t-a-1))

$T_{13}$    if (LEQ time-now deadline)
            then (BIDDING task)

$T_{14}$    if (BID-REPLY accept)
               AND (CELL-CONDITION normal)
            then (LIST-AGENDA task)
               AND (REPLY-TO-AWARD accept)

$T_{15}$    if (BID-REPLY accept)
               AND (CELL-CONDITION not-normal)
            then (REPLY-TO-AWARD reject)

$T_{16}$    if (BID-REPLY reject)
            then (RE-BIDDING(BID-SELECT a-t-a-1))
```

Figure 7 - Continued

different cells are correctly communicating and that the necessary message transmissions for scheduling are properly carried out, the protocol must incorporate the aforementioned common interface language and the APN model. The network on which this distributed scheduling protocol is implemented can be modelled as a three-layer structure (Figure 8). The distributed scheduling protocol is a high-level, problem-oriented protocol governing the communication between cell hosts for task-sharing. The host-to-host protocol, or the transport protocol, is to provide reliable communication between processes in cell-host computers. This layer is often implemented by the program called transport stations which is part of the cell-host's operating system. The lowest level of the protocol, the transmission protocol, is responsible for the transmission, packeting, and routing of data between cells; the transmission layer actually incorporates the functions of the physical layer, the data-link layer and the network layer in the OSI multilayer protocol model, as defined in Tanenbaum [1981].

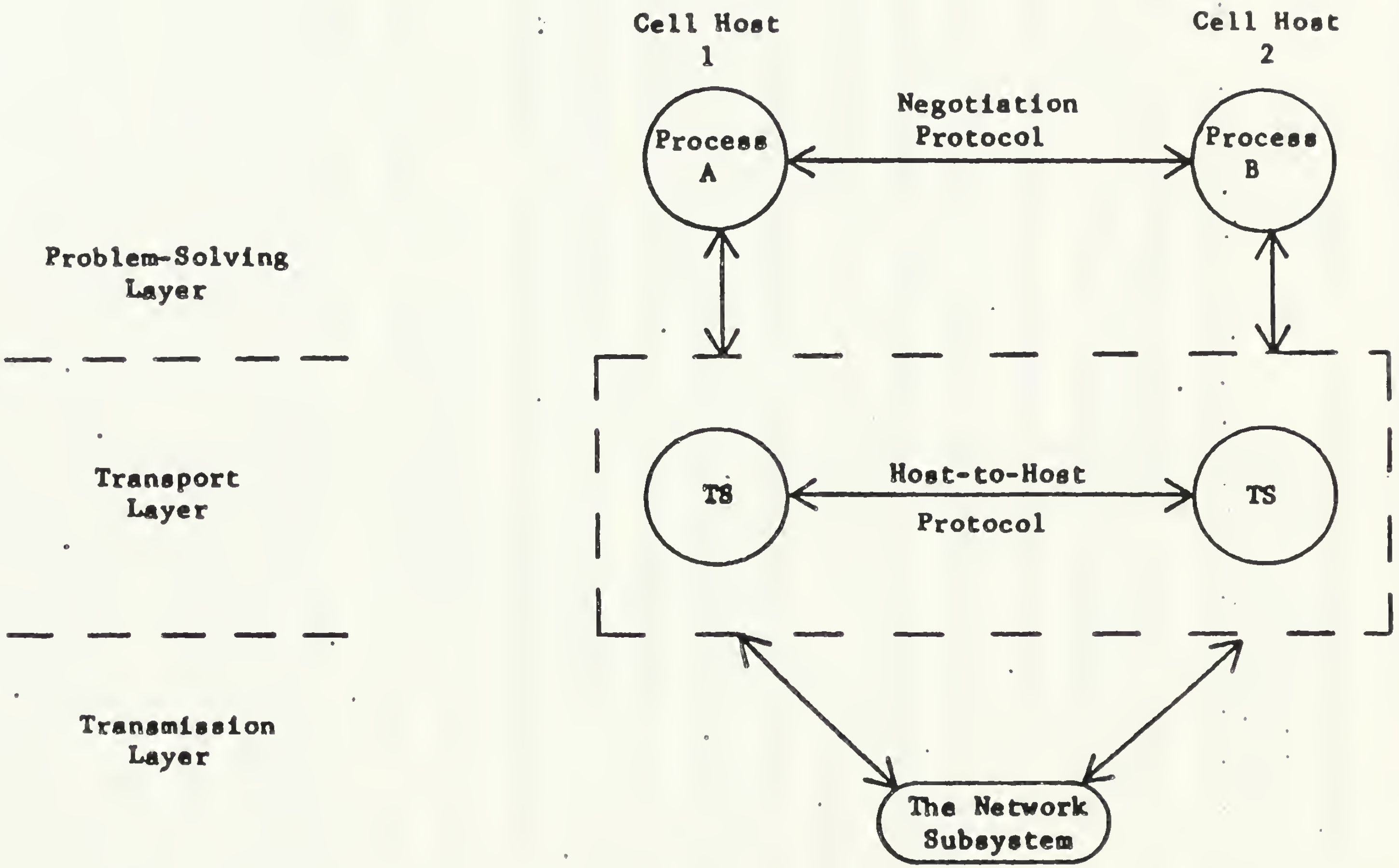
Insert Figure 8 Here

5. Evaluating the Distributed Scheduling Scheme: A Simulation Study

To evaluate the performance of the network-wide bidding scheme as a dynamic scheduling algorithm, I have conducted a simulation study on hypothetic cellular flexible manufacturing systems. The primary objective of the simulation study is (1) to compare the performance of the bidding algorithm with other approaches used in prior scheduling research. Specifically, I compared the bidding algorithm with the dynamic dispatching method based on shortest-processing-time heuristic; and (2) to evaluate the performance of the bidding algorithm with different bidding functions. For this purpose, the SPT heuristic and the EFT heuristic are evaluated.

It is known that in a single-machine environment the shortest-processing-time (SPT) rule is optimum with respect to certain measures of performance. The SPT rule, sometimes referred to as the shortest imminent operation rule, selects for processing the operation which can be completed in the least amount of time. In the single-machine environment studies have shown that SPT minimizes mean flowtime and minimizes mean lateness (Conway and Maxwell [1962], Baker [1976], and Blackstone [1982]). Conway and Maxwell further extended the concept and applied the SPT rule in the job-shop environment with m machines. Among the results, they found that (1) the SPT rule performed the best relative to mean flowtime, mean lateness, and the average number of tardy jobs; and that (2) imperfect information about processing times had little effect on the operation of the SPT rule. Recently, Chang et al. [1984] compared the SPT rule with other dispatching rules in the flexible manufacturing environment. They concluded that SPT rule performs better, in terms of throughput, than such rules as first-come-first-serve (FCFS), most work remaining (MWKR), and least work remaining (LWKR).

In this simulation study, in order to isolate the effects of the underlying dispatching heuristics, i.e., the shortest-processing-time rule versus the earliest-finish-time rule, and the effects of the use of scheduling methods, i.e., the centralized dispatching scheme versus


Problem-Solving Layer
Transport Layer
Transmission Layer
Cell Host 1
Cell Host 2
Negotiation Protocol
Process A
Process B
TS
Host-to-Host Protocol
TS
The Network Subsystem

the bidding scheme, three scheduling methods are tested for performance comparison: (1) Myopic-SPT, a centralized scheduling scheme employing shortest-processing-time as the dispatching rule; (2) Bidding-SPT, a distributed scheduling scheme employing shortest-processing-times to calculate bids; (3) Bidding-EFT, a distributed scheduling scheme employing earliest-finish-times to calculate the bids.

In the cellular system based on which the simulation study is conducted, the machines are grouped into flexible cells by group technology (GT). Each cell can have several different set-ups for different families of operations and jobs can be moved between cells. The devices responsible for transporting jobs between cells can take many forms, including conveyors, robot trucks, or automated guided vehicles (AGV). When a new job arrives, the scheduler on the cell interacts with the scheduler on other cells in order to determine the particular cell on which the job can be sent.

When a job arrives at the system, the first attribute specified is the sequence of operations determined by process planning; similar operations have already been made adjacent by applying GT so that these operations will be manufactured at the same cell. For those operations in the same family, the corresponding workpieces will have similar shapes and be made out of the same material by similar tooling set-ups. In estimating the processing time for certain operations, the processing time is to be based on the total processing requirements of each batch of jobs.

In effect, the scheduling problem of the cellular system is partitioned into two decisions:

(1) the assignment of jobs to the appropriate manufacturing cells; and

(2) the sequencing and scheduling of jobs within each cell.

The simulation study was conducted on the cellular FMS with different configurations, each configuration determined by the set of parameters randomly selected. For each job arrival, the interarrival time is exponentially distributed; the set of operations required by a job is randomly selected from a set of 10 operations. The processing time for each operation is exponentially distributed. In the case of myopic-SPT simulation, the actual processing time differs with the corresponding estimation by a deviation generated by normal distribution with mean zero. In order to account for the time taken for reaching the scheduling decision, we have incorporated a duration estimation, denoted by SD, between the time when the job arrives and the time when the job is assigned to a cell. This duration represents the time taken for reaching a given scheduling decision. For scheduling with the bidding-EFT method, this duration is

$$SD_E = \text{communication-delay} * 2 + \text{task-evaluation time.}$$

The SD value assigned to simulation runs for the bidding-SPT method is shorter because less information needs to be collected. The SD value assigned to myopic-SPT is the shortest due to the saving on communication delay. The time taken for a station in the token-bus network to broadcast a packet to every other station is assumed to be constant, independent of the load of the communication network.

The response variables gathered from the simulation runs are the following:

(1) job flow time statistics;

(2) proportion of jobs failing to meet the due date;

(3) job lateness and tardiness statistics; and

(4) average in-process waiting time.

The due-date for each job is calculated by

Due-date = TNOW + (estimated total processing time) * 1.3 + (no. of operations) * SD.

The performance of each scheduling approach was evaluated by 12 simulation runs, using the combination of 3 sets of configuration parameters and 4 sets of random-number seeds in generating various distributions. The simulation programs are written in SLAM, a Fortran-based simulation language, on CYBER 175.

Insert Exhibits 1, 2, and 3 Here

As described in the objectives of the simulation study, we are especially interested in comparing the performance between bidding-SPT and bidding-EFT to evaluate the two scheduling heuristics incorporated in the bidding function. Furthermore, by comparing the performance of the bidding-SPT and myopic-SPT, we can evaluate the characteristics of distributed scheduling with the bidding mechanism against centralized scheduling with myopic dispatching rules.

Parameter Set	I				II				III		
Replications	1	2	3	4	1	2	3	4	1	2	3
% of jobs late	9.55	15.28	26.97	14.49	11.567	14.61	29.98	13.93	21.28	20.19	46.89
Avg. waiting time	7.65	8.24	10.40	8.15	7.58	7.38	8.70	7.38	6.89	6.17	7.32
Avg. lateness	3.56	4.69	6.62	4.90	4.05	2.83	3.16	2.32	.87	.91	1.36
Mean Flow Time	25.75	27.06	29.88	27.25	19.23	19.94	21.71	19.75	14.47	14.27	15.77

Exhibit 1. Simulation Results of Using the Bidding-EFT Strategy

Parameter Set	I				II				III		
Replications	1	2	3	4	1	2	3	4	1	2	3
% of jobs late	15.85	21.80	32.56	22.05	23.25	28.78	41.20	23.75	43.28	47.43	67.34
Avg. waiting time	9.53	9.99	11.56	9.98	9.26	9.29	10.34	8.96	8.60	8.15	8.98
Avg. lateness	4.90	5.23	6.38	5.24	2.32	2.54	3.41	2.78	1.42	1.44	1.91
Mean Flow Time	28.13	28.30	30.67	28.60	20.33	21.07	22.70	20.68	15.53	15.51	16.79

Exhibit 2. Simulation Results of Using the Bidding-SPT Strategy

Parameter Set	I				II				III		
Replications	1	2	3	4	1	2	3	4	1	2	3
% of jobs late	24.11	29.89	37.27	27.28	25.56	29.15	43.46	26.54	44.13	46.95	66.83
Avg. waiting time	10.22	10.62	11.99	10.52	9.37	9.28	10.92	9.12	8.55	8.20	9.09
Avg. lateness	6.18	6.35	7.17	6.89	2.92	3.47	5.00	3.65	1.65	1.87	2.32
Mean Flow Time	27.81	28.69	30.94	28.85	20.35	21.80	23.28	20.79	15.48	15.50	16.92

Exhibit 3. Simulation Results of Using the Myopic-SPT Strategy

The simulation results for the three scheduling methods performed on the six-cell systems are shown in Exhibits 1, 2, and 3. Among the performance data, two particular results stand out: (1) bidding-EFT clearly has the best performance in terms of mean flow-time, tardiness, and in-process waiting-time measures. (2) In 10 out of the 12 simulation runs, the bidding-SPT method performs better than the myopic-SPT method, also in terms of mean flow-time, tardiness, and in-process waiting-time measures.

The distributed scheduling method performs better than the centralized counterpart primarily because, by executing the bidding mechanism, the scheduling decision is achieved by cells collectively based on purely local information stored within each cell. If the scheduling was to be done with centralized control, then there must be a global database and thereby a large amount of communication activities are needed to keep the dynamic information in the database up-to-date. In contrast, by letting each individual cell estimate its "price" for performing the announced tasks, all the estimation and calculation can be done based on information stored within the cell, and message-passing is carried out only to announce task or submit bid. Therefore, the distributed scheduling scheme utilizes more accurate information for estimating scheduling heuristics.

It is shown that the SPT dispatching rule, while performing well in many situations, is relatively insensitive to the accuracy of the estimation on processing times; i.e., it degrades gracefully with incorrect information on processing time (Conway [1962] and Baker [1976]). Such conclusions can help explain the 2 deviate cases where the myopic-SPT method performs better than the bidding-SPT method. However, our results further show that having more up-to-date information still results in better performance overall and the effort to obtain such information at the expense of communication overhead is well worthwhile.

In addition, the distributed scheduling scheme has much greater flexibility in taking into account additional information such as the estimated waiting time or estimated transporting time because decisions are made locally and these data are readily available. No extra communication messages are necessary. This additional information, constituting the major difference between bidding-SPT and bidding-EFT schemes, significantly improves the scheduling performance.

The distributed scheduling scheme also introduces parallel processing into the scheduling decision, since the bidding mechanism implies that the scheduling heuristics are estimated concurrently by the bidding cells, rather than letting a central scheduler do all the calculation. Parallel processing not only increases scheduling efficiency, it also helps avoid the possible communication bottleneck associated with any central scheduler. The other implication is that reliability would improve, since the scheduling performance would degrade gracefully if any cell-scheduler breaks down. Such reliability improvement, however, is not explicitly shown in the simulation results.

5. Conclusions

I have shown a distributed method for dynamic scheduling in cellular flexible manufacturing systems. The method has the following features:

(1) It is a distributed scheduling technique; no node has greater importance, as far as scheduling is concerned, than any other node.

(2) The algorithm is flexible, and can take into account such information as loading factor, unexpected breakdowns, or resource constraints in the bidding scheme.

(3) Compared with dynamic dispatching rules previously used, the bidding algorithm is characterized by its more accurate estimation of processing times, without spending the cost of constant updating. The improvement by such information is verified by simulation results.

(4) This is the only scheduling algorithm in the manufacturing area to date that considers the characteristics of the communication network, i.e., loosely coupled nodes with distributed control, packet-switching, communication delay, and the broadcasting capability.

(5) The bidding scheme can be represented by an augmented Petri net model and implemented in the ISO multilayer protocol compatible with MAP.

References

Baker, K., 1974, Introduction to Sequencing and Scheduling (New York: John Wiley & Sons).

Blackston, J. H., Phillips, D. T., and Hogg, G. L., 1982, A State-of-the-art Survey of Dispatching Rules for Manufacturing Job Shop Operations, International Journal of Production Research, Vol. 20, No. 1, p. 27.

Bourne, D. and Fussell, P., 1982, Designing Programming Languages for Manufacturing Cells, Technical Report CMU-Rl-Tr-82-5, The Robotics Institute, Carnegie Mellon University.

Box, W., Closs, F., Jauson, Pl, Kummerle, K., Muller, H., and Rothauser, E., 1982, A Local-Area Communication Network Based on Reliable Token-Ring System, IFIP International Symposium on Local Area Networks.

Chang, Y. L., Sullivan, R. S., Bagchi, U., 1984, Experimental Investigation of Quasi-Realtime Scheduling in Flexible Manufacturing, Proc. First ORSA/TIMS Conference on FMS.

Conway, R., Johnson, B., and Maxwell, W., 1962, An Experimental Investigation of Priority Dispatching, J. of Ind. Enging, 11, p. 221.

Conway, R., and Maxwell, W., 1962, Network Dispatching by Shortest Operation Discipline, Operations Research, 10, p. 51.

Cutkosky, M., Fussell, P., and Milligan, R., 1984, Precision Flexible Machining Cells Within a Manufacturing Systems, Technical Report CMU-RI-TR-84-12, The Robotics Institute, Carnegie-Mellon University.

Davis, R. and Smith, R., 1983, Negotiation as a Metaphor for Distributed Problem Solving, Artificial Intelligence, Vol. 20, pp. 63-109.

French, S., 1982, Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop (New York: John Wiley).

Greene, T. and Sadowski, R., 1984, A Review of Cellular Manufacturing Assumptions, Advantages, and Design Techniques, Journal of Operations Management, Vol. 4, No. 2, pp. 85-97.

Keil, R. and Dillon, S., 1985, Manufacturing Automation Protocol (MAP) Specification, General Motors Technical Center (Warren, Michigan).

Kimemia, J. and Gershwin, S., 1985, Flow Optimization in Flexible Manufacturing Systems, Int. J. Prod. Res., Vol. 23, No. 1, pp. 81-96.

Krutsch, T., 1981, A User Speaks Out: Broadband or Baseband for Local Nets?, Data Communications, p. 105-112.

Maira, A., 1986, Local Area Networkds - The Future of the Factory, Manufacturing Engineering, Vol. 96, No. 3.

Malone, T. W., Fikes, R. E., and Howard, M. T., 1983, Enterprise: A Market-like Task Schedular for Distributed Computing Environments, Working Paper, Xerox Palo Alto Research Center.

McLean, C. R., Bloom, H. M., and Hopp, T. H., 1982, The Virtual Manufacturing Cell, Information Control Problems in Manufacturing Technology, (McGregor & Werner: Washington, D.C.).

McLean, C., Mitchell, M. and Barkmeyer, E., 1983, A Computer Architecture for Small-Batch Manufacturing, IEEE Spectrum, p. 59.

Metcalfe, R. and Boggs, D., 1976, Ethernet: Distributed Packet Switching for Local Computer Networks, Comm. ACM, 19:7, pp. 395-404.

Mosier, C. T., Elvers, D. A., and Kelly, D., 1984, Analysis of Group Technology Scheduling Heuristics, Internation Journal of Production Research, Vol. 22, No. 5, p. 857.

Myers, W., 1982, Toward a Local Network Standard, IEEE Micro, p. 28.

Oren, M. E. and Williams, A. C., 1975, On Competitive Bidding, Operations Research, Vol. 23, p. 1072-1079.

Ramamritham, K. and Stankovic, J., 1984, Dynamic Task Scheduling in Hard Real-time Distributed Systems, Proc. Distributed Computing Systems, IEEE Computer Society Press.

Schoeffler, J. D., 1984, Distributed Computer Systems for Industrial Process Control, Computer, Vol. 17, No. 2, p. 11.

Shanthikumar, J. G. and Sargent, R. G., 1980, A Hybrid Simulation/ Analytical Model of a Computerized Manufacturing System. Working Paper 80-017, Dept. of Industrial Engineering and Operations Research, Syracuse University.

Shaw, M., 1984, A Distributed Knowledge-based Approach for Intelligent Manufacturing Information Systems, Ph.D. Dissertation, Purdue University, (W. Lafayette, Ind.).

Shaw, M., and Whinston, A., 1985a, Automatic Planning and Flexible Scheduling, Proc. Int. Conference on Automation and Robotics, (St. Louis, MI).

Shaw, M. and Whinston, A., 1985b, Task Bidding, Distributed Planning, and Flexible Manufacturing, Proc. of IEEE Conference on Artificial Intelligence Applications (Miami, Florida).

Sikha, R. K. and Hollier, R. H., 1984, A Review of Production Control Problems in Cellular Manufacturing, International Journal of Production Research, Vol. 22, No. 5, p. 773.

Simpson, J. A., Hocken, R. J., Albus, J. S., 1982, The Automated Manufacturing Research Facility of the National Bureau of Standards, Journal of Manufacturing, Vol. 1, No. 1.

Smith, R. G., 1980, The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, IEEE Trans. Computers, Vol. C-29, No. 12.

Solberg, J. J., 1977, A Mathematical Model of Computerized Manufacturing Systems, Proc. of 4th International Conference on Production Research (Tokyo, Japan).

Tanenbaum, A., 1981, Computer Networks, (Prentice-Hall: New Jersey).

Zisman, M., 1978, Use of Production Systems for Modelling Asynchronous Concurrent Processes, in Pattern Directed Inference Systems, D. Waterman and F. Hayes - Roth (Eds.), (Academic Press, New York).