UNIVERSITY OF
ILLINOIS LIBRARY
AT URBANA-CHAMPAIGN
BOOKSTACKS

Digitized by the Internet Archive
in 2011 with funding from
University of Illinois Urbana-Champaign

http://www.archive.org/details/fmsschedulingasc1326shaw

451-72

BEBR

FACULTY WORKING
PAPER NO. 1326

FEB

FMS Scheduling as Cooperative Problem Solving

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. 1326

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

February 1987

FMS Scheduling as Cooperative Problem Solving

Michael J. Shaw, Professor
Department of Business Administration

Abstract

This paper describes an FMS scheduling method that treats an FMS as a group of problem-solving agents cooperating to perform manufacturing jobs. The main thrusts of such a method include the ability to handle the dynamically changing production conditions, its taking into account the communication method, the improved reliability, and the use of distributed control. The paper emphasizes research issues associated with various aspects of the cooperative problem-solving method, including: (1) dynamic task assignments, (2) the coordination mechanism, and (3) knowledge-based scheduling as problem solving. A simulation study which compares the performance of the cooperative problem solving approach with that of the more traditional scheduling approaches is also reported.

I. Introduction

An emerging architecture for flexible manufacturing systems (FMS) is the cellular system where a group of flexible cells perform manufacturing tasks collectively (Bourne [1982], Cutkosky [1984], and Simpson et al. [1982]). Such cellular FMS, as shown conceptually in Figure I.1, have played an increasingly important role in the automated manufacturing technology for many reasons; among them are the reduced machine set-up time, simplified tooling requirements, the simplification of planning and control, reduced in-process inventory, the near-constant load time, and system modularity (McLean, et al. [1983], Green and Sadowski [1984], and Sikha and Hollier [1984]). This paper is concerned with the scheduling aspect of the cellular system. It presents a novel approach which essentially treats the scheduling problem by the multiagent problem-solving paradigm: because the whole scheduling task is large and complicated, the set of problem-solving agents--the cells--would carry out the task collectively. To emphasize the cooperation aspect, the method is characterized as "cooperative problem-solving." The cooperation among cells is achieved through exchanging information in an orderly manner, guided by a bidding mechanism.

Insert Figure I.1 Here

In the cellular FMS, as shown in Figure I.1, the cells communicate with each other through a local area network (LAN). Associated with such a networking environment, there are two possible control structures underlying the scheduling decisions: (1) the system uses a centralized scheduler in charge of job assignment and the scheduler keeps track of the whole cellular system by a global database; and (2)

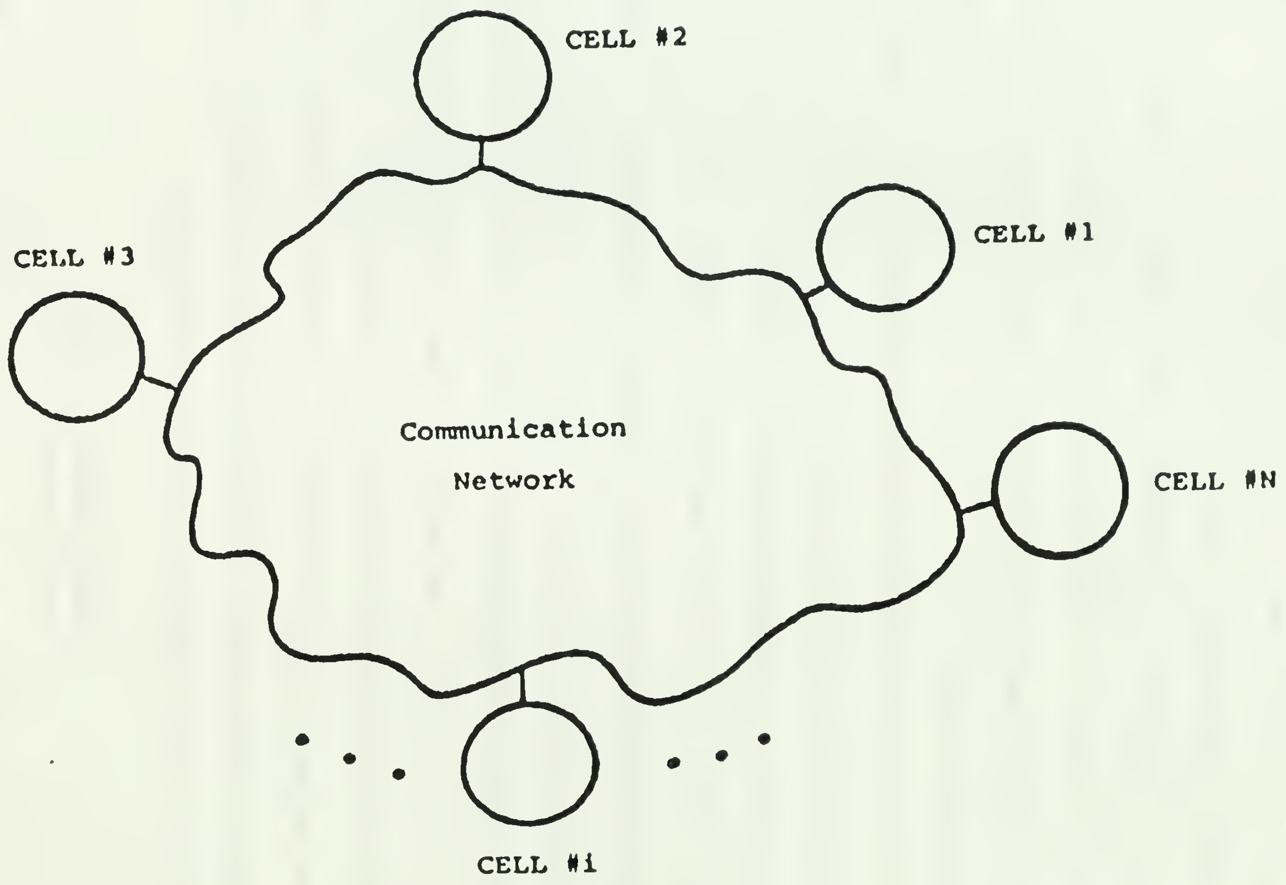

Figure I.1 A Conceptual Model of the Cellular CIM

the system uses 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 instead of centralized control has these advantages: (1) better reliability--the system degrades gracefully in the event 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 caused by the global scheduler; and (4) cost-effectiveness--it is more cost-effective because of the smaller processing requirements on the computers and less communication bandwidth requirements needed for global updating. The implications of distributed control structures to the scheduling method are summarized in Table I.1.

Insert Table I.1 Here

By treating FMS scheduling as cooperative problem solving, the scheduling approach presented in this paper has the following features: (1) it is a distributed scheduling technique; no cell has greater importance, as far as scheduling is concerned, than any other cell; (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

Table I.1 Implications of Control Structures to Scheduling

	Centralized System	Distributed 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

spending the cost of constant global updating; (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 cells with distributed control, packet-switching, communication delay, and the broadcasting capability; (5) the scheme can be represented by an augmented Petri net model and implemented in the multilayer protocol compatible with Manufacturing Automation Protocol (MAP); (6) on the cell level, the automatic problem-solving method is used to schedule the jobs, carried out by a knowledge-based system based on a heuristic searching procedure.

II. Scheduling as Cooperative Problem Solving

In the cellular FMS, the machines are grouped into flexible cells by group technology. For those operations in the same family, the corresponding workpieces will have similar shapes and can be made out by similar toolings. Each cell can have several set-ups for different families of operations; jobs entered into the system usually move between cells for completing several types of operations as specified. The devises responsible for transporting jobs between cells can take many forms, including conveyors, robots trucks, or automated guided vehicles (AGVs). When a new job arrives, the scheduler on the cell interacts with the scheduler on other cells in order to determine the most appropriate cell on which the job can be sent.

The jobs arrive at the system dynamically over time and the system behaves like a network of queues. The cellular FMS is a loosely coupled system of cooperating flexible cells in which each cell can be set up 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 prior scheduling research; assorted techniques have been used to solve 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], Stecke [1982, 1985], Kimemia and Gershwin [1985]), and artificial intelligence (Shaw [1984], Shaw and Whinston [1985a] [1985b]). Characteristics of the scheduling methods 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.

Although the importance of appropriately incorporating LAN technology in automated manufacturing systems has been pointed out by several researchers, such as McLean et al. [1983], Cutkosky et al.

[1984], Ranky [1985], and Keil and Dillon [1985], there has not been any work evaluating the impact of the LAN technology on the way the FMS scheduling is performed; nor is there any research that considers the networking environment in designing the scheduling method for FMS. The method presented in this paper can fill that void.

As previously described, the cellular FMS can be treated as a group of loosely-coupled, cooperating cells where each cell is an intelligent problem-solving agent. As such, coordinating the operations performed in a cellular FMS is analogous to organizing a group of specialists to accomplish the given set of tasks. Accordingly, organization models can be useful in analyzing cellular FMSs. Simon [1982] first drew the parallel between the structures of computerized systems and that of human organizations. He focused on the limitation on the processing capability of individual problem-solving agents and articulated the information processing model for analyzing organization structures. According to this school of thought, an organization can be viewed on an abstract level as consisting of: (1) a group of agents; (2) a set of activities specialized and performed by each agent; (3) a set of communications means among the agents; and (4) a set of performance goals by which the combined activities of the agents are evaluated. To organize, then, is to (1) establish the goal of organization; (2) segment the goals into separate activities; and (3) assign the activities to agent in such a way that the overall goals are achieved (Malone and Smith [1984]).

The same information processing model for analyzing human organizations can be applied to the scheduling of cellular FMS, where the host-computer of each cell can be treated as a problem-solving agent. Since the major problem-solving task in this case is the scheduling of jobs, the activities performed by each agent are confined by the set-up of the corresponding cell. Using such a paradigm, the scheduling of FMS is equivalent to solving problems by a group of agents, with each agent specialized in a given set of activities.

The configuration of such a cooperative problem-solving system and the effects of interactions between the cells can be modeled as a directed graph:

$$G = (E,I)$$

The graph G defines the information structure of the cellular system. The problem-solving activities in cell i may impact on the problem-solving activities in cell j through the interactions I_{ij}. Every cell in the graph represents a problem-solving agent E_i, corresponding to cell i. In the scope of this paper, the major problem-solving activity is the scheduling of manufacturing processes.

A job T is decomposed into tasks t_1, t_2, ..., t_n which are assigned to cells E_{e_1}, E_{e_2}, ..., E_{e_m} ($e_\ell \varepsilon I$ is the index of the corresponding cell). If the collection of tasks assigned to cell E_{e_i} is denoted by T_{e_i}, then

$$\bigcup_{\ell \,\varepsilon\, [1,m]} T_{e_\ell} = T$$

and

$$T_{e_i} \bigcap_{i \neq j} T_{e_j} = \phi.$$

We are mainly concerned with problems which can be sufficiently decoupled and the effects of one agent are largely independent of other agents. This is the case in the FMS environment where machining operations in different cells are mostly independent. The primary coordinating activities, then, are the assignments of tasks to appropriate cells. The process of cooperative problem solving in this situation can be algorithmically represented as follows:

```
Procedure DP (T)
Input:
        T:  the job to be achieved.
        E:  the set of scheduling cells.
Output:
        P:  a distributed schedule to achieve goal T.
Begin
(1)  T' ← -- DECOMPOSE(T)
        {T' is a partition of T}
(2)  A ← -- DISTRIBUTE(T',E)
        {A is the set of pairs $(e_i, T_{e_i})$}
     For all i Do
(3)  Begin
        If (OVERLOAD($e_i$)) Then DP($T_{e_i}$)
(4)     $P_i$ ← -- EXECUTE($T_{e_i}$)
(5)     P ← result($P_i$)
     End

End
```

The adoption of the cooperative problem-solving 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-assignment scheme among cells to ensure that all the resources can be efficiently utilized, and (2) the coordination mechanism exercised among the cells, so that the manufacturing tasks can be carried out cooperatively. The network-wide bidding scheme described in this paper can achieve these two functions.

In analyzing the information processing requirements of various forms of organizations, Simon [1982] singled out the market as a type of organization where only a small amount of information need to be transferred to achieve coordination. It has been shown that the bidding mechanism, an information-exchanging mechanism commonly used for allocating commodities or for establishing job contracts in the market, can achieve efficient allocation within an organization (Harris and Ravi [1981], Malone and Smith [1984]). To achieve the same type of information efficiency in the scheduling of an FMS, the bidding mechanism is used to regulate the coordination and task allocation among the agents--i.e., the cells. Specifically, the scheduling decision is made by collecting the price from each manufacturing cell for taking on the job. This paradigm for cooperative problem solving was first developed by research in artificial intelligence (Davis and Smith [1983], Shaw [1985]) and has been applied to such distributed systems as the sensor network (Smith [1980]) or computer networks (Malone [1983], Ramamritham and Stankovic [1984]). Davis [1981], Axelrod [1984], and Rosenschein and Genesereth [1984] presented formalisms for analyzing the cooperation between problem solvers.

III. The Distributed Scheme for Dynamic Task Assignment

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, i.e., 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. 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., the cell 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 III.1.

Insert Figure III.1 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 (discussed in Section V), while systematically coordinating with other cells through task sharing and bidding. By eliminating the necessity to collect dynamically changing

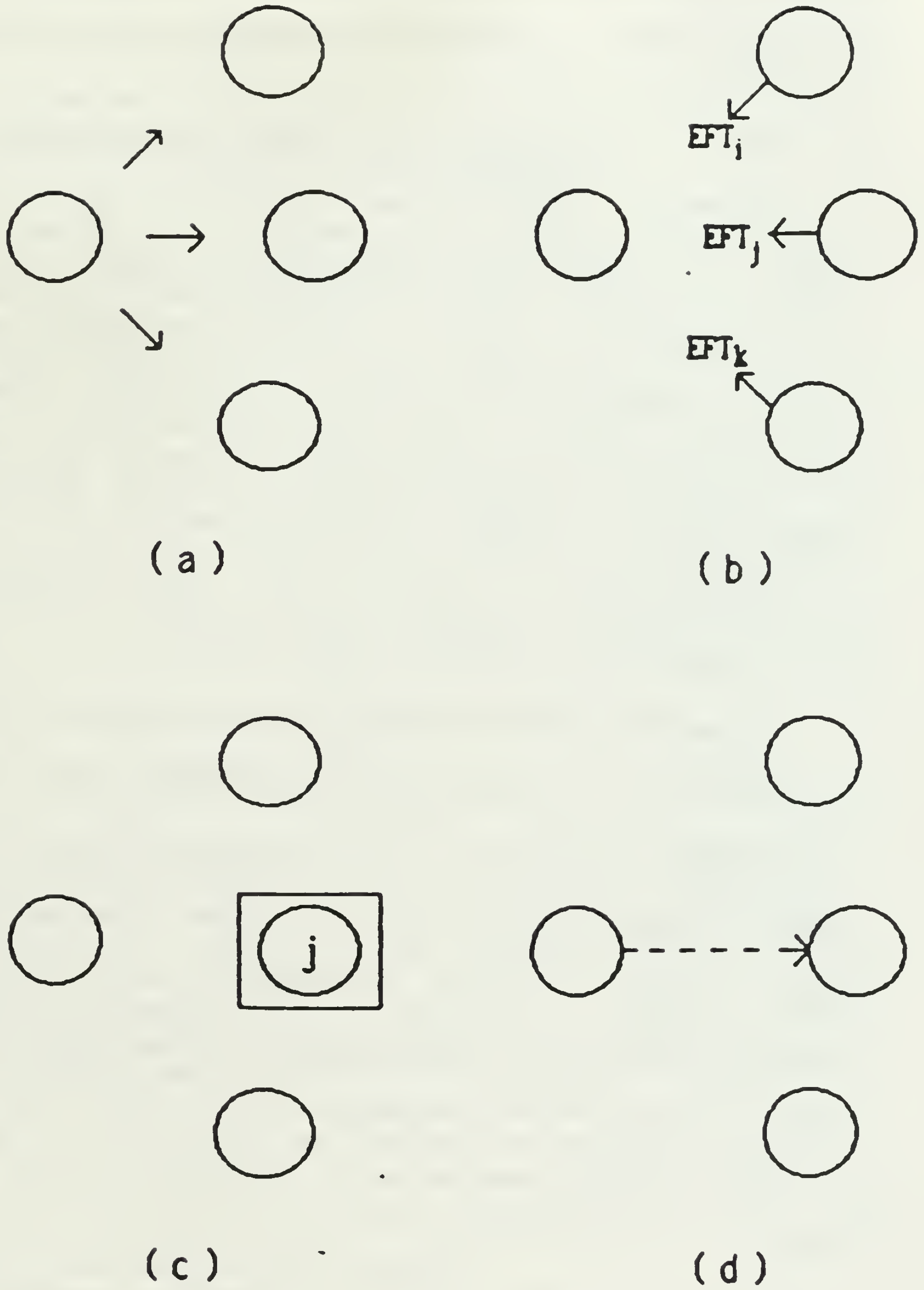

Figure III.1 The Bidding Scheme

system information in a global database, the possible bottleneck and the communication activities for constant updating are avoided.

IV. Modeling and Automating the Coordination Mechanism

This section concerns a mechanism that can carry out the bidding scheme in the distributed, networking environment. This mechanism can be activated by each cell in a decentralized fashion while keeping the execution of manufacturing tasks well coordinated. Three issues need to be considered in designing such a mechanism: (1) a model of the bidding scheme for dynamic, concurrent execution; (2) the execution of this bidding scheme in a decentralized, well-coordinated manner; and (3) a formalism for intercell communication.

The augmented Petri-net model, an integration of production rules and Petri-nets, is used to model the bidding scheme. The automation of this model, carried out by the corresponding Petri-net language, leads to a distributed algorithm for dynamic task assignment. The model includes a procedural representation of the interactions between cells and a declarative representation of the decision process within a cell. Let us review the components involved in the augmented Petri-net model and then describe using this model to carry out the bidding scheme.

Designed to model process concurrency and precedence relations, the Petri net model has been used to model, specify, and verify communication protocols (Peterson [1981], Nelson, et al. [1983]). The definition of the Petri net follows:

Definition 1 (Petri Net)

A Petri net, W, is a quadruple, $W = \langle P,T,I,O \rangle$, where P is the set of places, T is the set of transitions; $I:T \rightarrow P^*$ defines the input function, and $O:T \rightarrow P^*$ defines the output function.

A place is marked if it has one or more tokens; a transition is enabled if each of its input places are marked. The firing of an enabled transition removes one token from each of its input places and adds one token to each of its output places. A token distribution among the available places in a Petri net is called a marking of the net. Corresponding to each Petri net a labelling function for the transitions l: $T \rightarrow Z$, and an initial marking, λ, Petri net language is defined as:

$$L(l) = \{l(\beta) \in Z^* | \beta \in T^* \text{ and } \delta(\lambda,\beta)\}$$

where δ is the next-state function. For a sequence of transitions $t_{j1}, t_{j2}, \ldots, t_{jk}$, $\delta(\lambda, t_{j1}t_{j2}t_{j3}\ldots t_{jk})$ represents that the firing of the transition sequence, t_{j1}, t_{j2}, up to t_{jk}, is legal. $L(\ell)$ defines the set of all possible sequences of transition firings for a given Petri net. Thus, if one can represent a complicated process as a Petri-net, the corresponding Petri-net language can be used to regulate the correct execution of that process.

Definition 2 (Augmented Petri Nets)

An augmented Petri net is composed of seven elements:

$APN = \langle P,T,I,O,\lambda,AP,D \rangle$

where <P,T,I,O> is a Petri net as defined in Definition 1; λ is the initial marking of this net. The set of transitions, T, also defines the set of productions, with each transition corresponding to one production rule. D is the set of database elements in the production system and AP is the set of active productions whose conditions are satisfied by D.

A transition t in T is "firable" iff

(1) $t \epsilon AP$; and

(2) I(t) is marked; I(t) represents the set of input places of the transition t.

In the augmented Petri net model, since there is a production rule corresponding to every transition, one can label the transition and the associated production rule with the same labelling function. The Petri net language in the augmented Petri net can thus be seen as either the set of all possible sequences of transitions or, alternatively, as the set of all allowable sequences of production rule invocations. If each transition corresponds to a decision/activities pair, the Petri net language generates the correct sequence of making these decisions.

Task bidding for several tasks are usually executed concurrently. The manager cell may be ranking the incoming bids while the potential contractors at the same time are collecting task-announcements and deciding on whether to submit bids. Consequently, the transfer of messages (e.g., task-announcements, bids) from one cell to another requires synchronized activities among the cells involved. Augmented Petri nets can ensure the correct implementation of these synchronized activities.

To use the augmented Petri net model, the bidding process can be represented by two subsets: one (Figure IV.1(a)) models the necessary actions of the manager cell who announces a task to other cells, processing the incoming bids and awards the task to the selected cell; the other sub-net (Figure IV.1(b)) models the corresponding actions of the cells who receive the task-announcement (the contractor cells). This sub-net deals with the decision on submitting bids.

Insert Figure IV.1 Here

Each activity in the process is represented by a production rule, and the interactions among these activities are represented by the Petri net. Each transition in the Petri net (denoted by a bar in the figures) corresponds to one production rule. When a transition is enabled (i.e., all input places are marked), the corresponding rules will determine the firing condition. Figure IV.2 lists the set of production rules that correspond to the transitions in the two augmented Petri nets in Figure IV.1. At each step in the process, the augmented Petri nets guide the bidding process of all cells so that the task assignments are correctly carried out. The Petri net language can serve as the "control language" to regulate the invocation of production rules in the production system during its inference process. Such a production system whose control structure is represented explicitly is called a controlled production system (Georgeff [1982]).

Insert Figure IV.2 Here

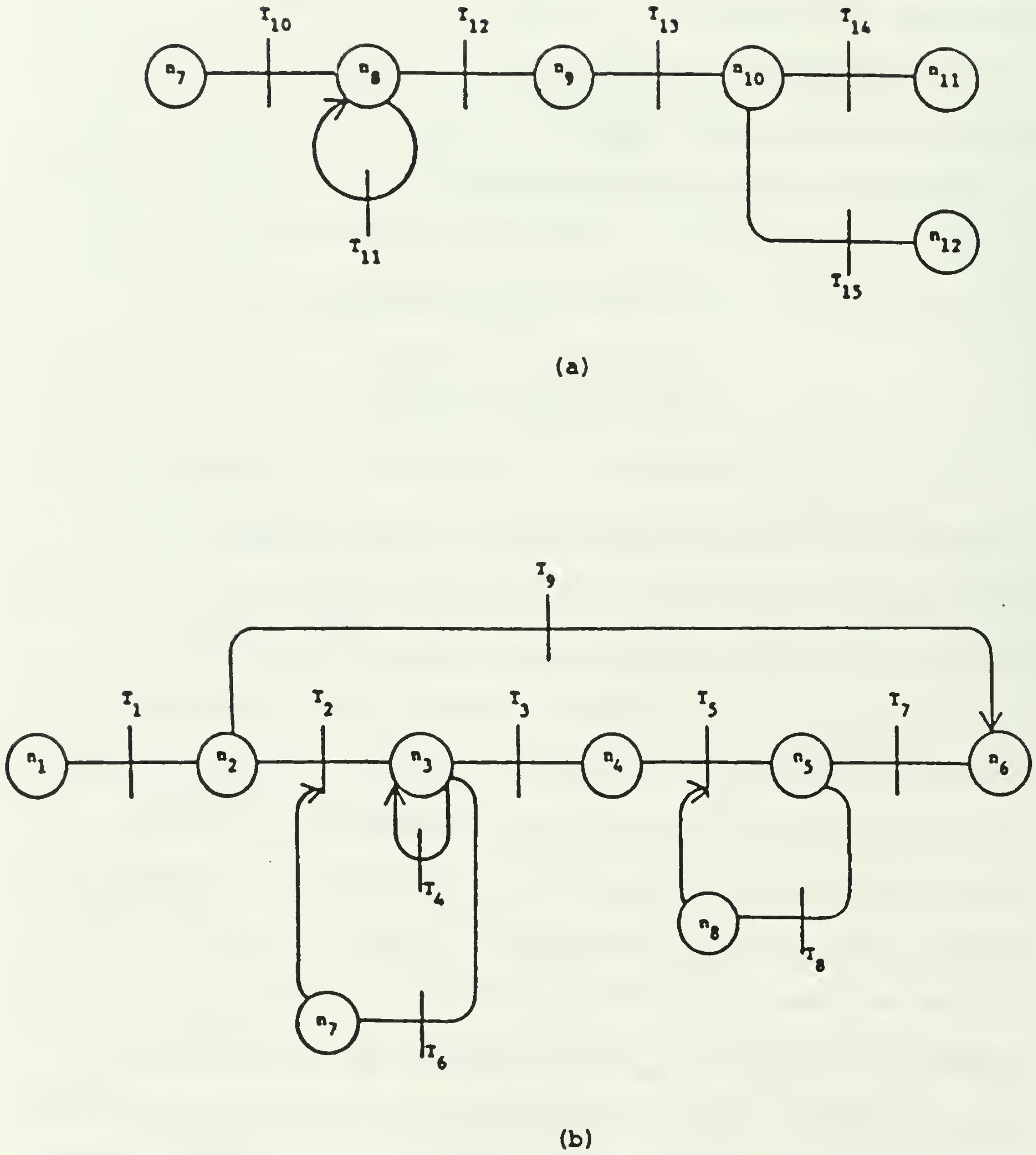

Figure IV.1 The APN Model for (a) Bidding and (b) Task Announcement

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

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

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

if (LEQ time-now deadline)
then

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

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

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

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

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

if (TASK-ANNOUNCED task)
then (TASK-RANKING task)
```

```
T11   if (EQ (PROCESSSOR-FOR-TASK task) busy)
      then (LIST-ACTIVE-TASK-ANNOUNCEMENT tas

T12   if (EQ (PROCESSOR-FOR-TASK task) idle)
      then (BID-REPLY (BID-SELECT a-t-a-1))

T13   if (LEQ time-now deadline)
      then (BIDDING task)

T14   if (BID-REPLY accept)
         AND (CELL-CONDITION normal)
      then (LIST-AGENDA task)
          AND (REPLY-TO-AWARD accept)

T15   if (BID-REPLY accept)
         AND (CELL-CONDITION not-normal)
      then (REPLY-TO-AWARD reject)

T16   if (BID-REPLY reject)
      then (RE-BIDDING (BID-SELECT a-t-a-1))
```

Figure IV.2. The Production Rules Used in the APN Model

The control language in effect guides the allowable sequences of production invocations, i.e., a production is applicable only if it is accepted by the control language. At each stage of the execution, the control language acts to focus the control on a subset of the productions, the applicable productions, and prohibits the other productions from being invoked. This isomorphism between (1) the augmented Petri net model and (2) a production system model with a separate control language enables each cell to deal with the task-assignment problem by executing the production rules listed in Figure IV.2 and using the Petri-net language corresponding to Figure IV.1 to guide the rule selection.

For executing correct communication activities in a network, a communication protocol is required so that each communicating node can follow the protocol to transmit data correctly through the network. Shaw [1986b] showed that the aforementioned coordination mechanism, executed by the controlled production system can be implemented in the MAP environment. In addition to the coordination mechanism, a common interface language is also required to enable cell-host computers to communicate their intentions and share information with one another. This parallels how people communicate. For this purpose, a formalism for expressing the messages needs to be specified so that the interface language for achieving coordination 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 IV.3. The format is based on phrase-structure grammar specified in Backus-Naur Form (Danthine [1980]).

```
<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 IV.3 The Syntax of the Interface Language

Insert Figure IV.3 Here

V. Knowledge-Based Problem Solving for Cellular Scheduling

Based on the cooperative problem solving paradigm, the FMS scheduling is carried out by a group of cooperating, loosely-coupled flexible cells, each cell specialized in specified areas of manufacturing expertise. The scheduling problem, then, becomes a two-level problem: the first level is the task assignment problem and the second level is the local scheduling of each cell (Shaw [1986a]). This two-level scheduling approach is illustrated in Figure V.1.

Insert Figures V.1. Here

Due to the flexibility of the machines, a given task assigned to a cell usually can be performed by a number of different ways; the decision of assigning a job to a machine on the cell level is thus dependent upon the cell status at that particular moment. In addition, sometimes there may be needs to cancel or reassign machines or other resources because of unexpected breakdowns. Consequently, the scheduling decision within each cell needs to be adaptive to dynamic changes of the FMS environment (Ranky [1986], McLean [1983]). In specifying the desirable functions of the cell controller for the Automated Manufacturing Research Facilities in National Bureau of Standards, McLean [1983] characterized them as: (1) state-space planning, (2) adaptive scheduling, (3) optimizing, and (4) learning. To achieve these functions in the FMS environment, the incorporation of artificial intelligence in the schedular becomes necessary.

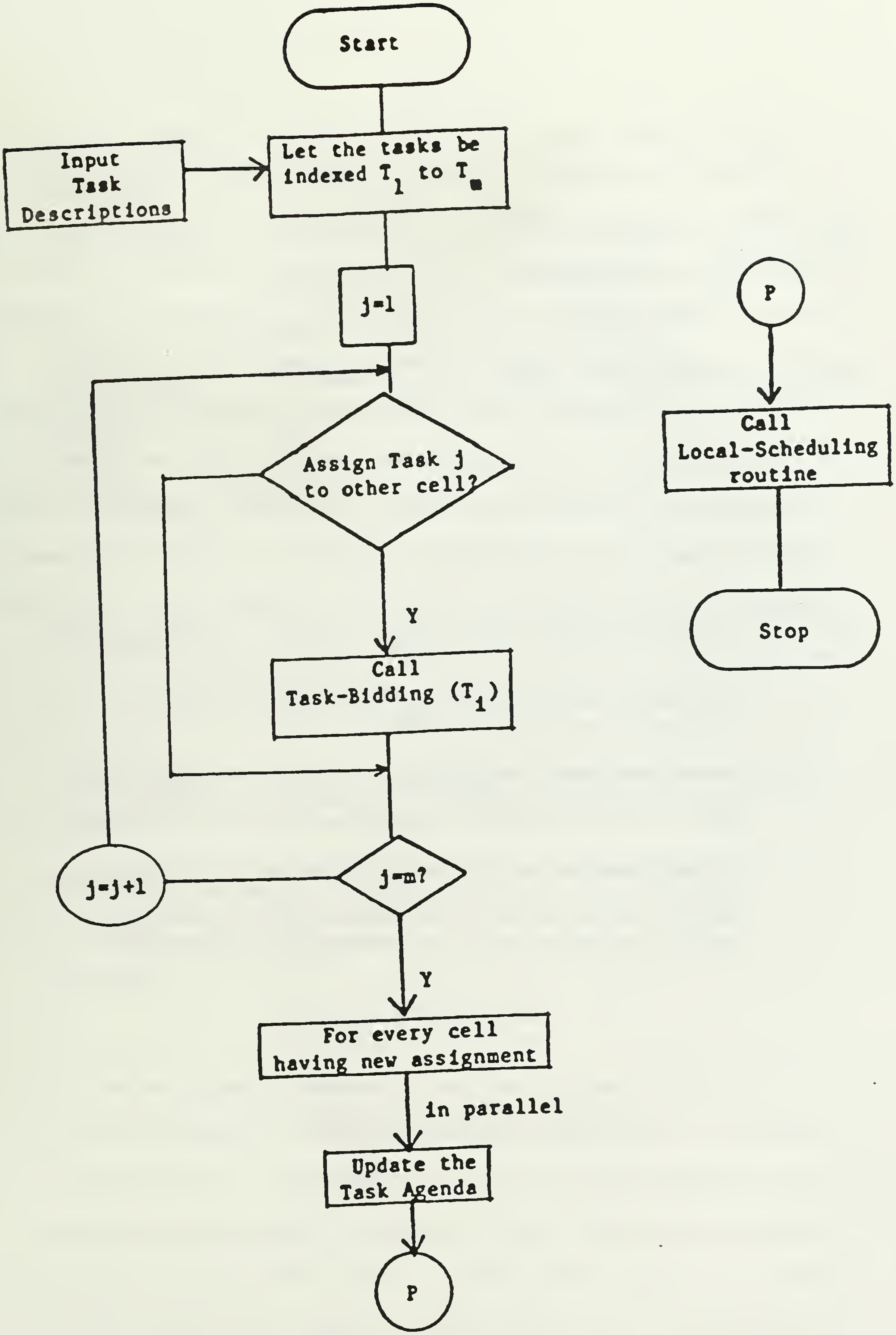
Start
Input Task Descriptions
Let the tasks be indexed T1 to Tm
j=1
Assign Task j to other cell?
Y
Call Task-Bidding (T1)
j=m?
j=j+1
Y
For every cell having new assignment
in parallel
Update the Task Agenda
P
P
Call Local-Scheduling routine
Stop

Figure V.1 The Flowchart for the Two-Level Scheduling Approach

From the AI perspective, problem solving is frequently described as finding a series of state-changing actions that will achieve a desired goal state given the initial state (Newell and Simon [1972]). Thus, the scheduling task can be interpreted as developing a course of action, a plan, for the agents to reach the goals desired. In a flexible cell, the agents--which may be robots, computerized machines, or the host computer of a manufacturing cell--can carry out a variety of operations, including various types of machining, workpiece routing, loading/unloading, and communication activities. Like most AI-based problem solver, the scheduling system for a flexible cell is organized as a knowledge-based system consisting of the following components:

1. A database storing state descriptions of the flexible cell, referred to as the world model.

2. A knowledge base, consisting of a set of operators, that modifies the descriptions in the database. These operators model the state-changing manufacturing activities (Figure V.2).

3. An inference engine that executes the control scheme--it decides at any given time the most appropriate operator to apply based on the state descriptions in the world model. This process continues until the scheduling is completed.

Insert Figure V.2 Here

Scheduling a flexible cell by the inference engine can be described as a state-space searching process or as exploration of a tree of possible action sequences. Consequently, the generation of schedules also suffers from the problem of combinatorial explosion. There are two ways, among others, by which this complexity problem can be alleviated: (1) by incorporating a decomposition method and (2) by employing effective heuristics (Shaw [1986d]). Presenting a

OPERATOR NAME:	TRANSFER /* Transfer a part from m_1 to m_2 */
ARGUMENTS:	($?m_1$ $?m_2$?pt)
PRECONDITIONS:	FINISH-OP($?m_1$?op ?pt) $\wedge$ PT-NEXTOP($?op_1$ $?op_2$?pt) $\wedge$ MACH-OP($?m_2$ $?op_2$) $\wedge$ DIFFERENT ($?m_1$ $?m_2$)
ADD-LIST:	MACH-PT($?m_2$ $?op_2$?pt) $\wedge$ IDLE($?m_1$)
DELETE-LIST:	FINISH-OP($?m_1$ $?op_1$?pt) $\wedge$ PT-NEXTOP($?op_1$ $?op_2$?pt)
RESOURCE:	$?m_2$
DURATION:	2

(a)

OPERATOR NAME:	GRASP /* take a part from the buffer by the robot arm */
ARGUMENTS:	(?pt arm buffer)
PRECONDITIONS:	POSITION (arm buffer) $\wedge$ READY-to-GRASP (?pt)
ADD-LIST:	IN-ARM (?pt)
DELETE-LIST:	IN-BUFFER (?pt) READY-to-GRASP (?pt)
RESOURCE:	arm
DURATION:	1

(b)

Figure V.2 Sample Operators

computation study of such a knowledge-based scheduling system, Shaw [1986b] described a heuristic searching procedure, called the A* algorithm, for expiditing the state-space search. The typical computation reduction resulting from heuristic searching is shown in Figure V.3, based on the computation study we have conducted on a LISP machine. Such a knowledge-based approach is also capable of performing dynamic scheduling and of refining scheduling skills by machine learning. As such, the AI-based approach appears promising for cell-level scheduling.

The ability for dynamic scheduling is due primarily to the data-driven nature of the knowledge-based system. A symbolic description of the FMS environment is recorded in the world model which uses sensorary data to update the current status of the cell. Since scheduling is viewed as a problem-solving process which establishes a plan of actions to achieve the goal (job completions) from the current state of the system, environment changes are taken into account automatically. Figure V.4 shows graphically an example of dynamic scheduling, where environmental changes are caused by new job arrivals, thus triggering dynamic scheduling.

Insert Figures V.3 and V.4 Here

V.1 Performance Analysis

To evaluate the performance of the cooperative problem solving method, a simulation study has been conducted on hypothetic cellular flexible manufacturing systems (Shaw [1986b]). The primary objective of the simulation study is (1) to compare the performance of the

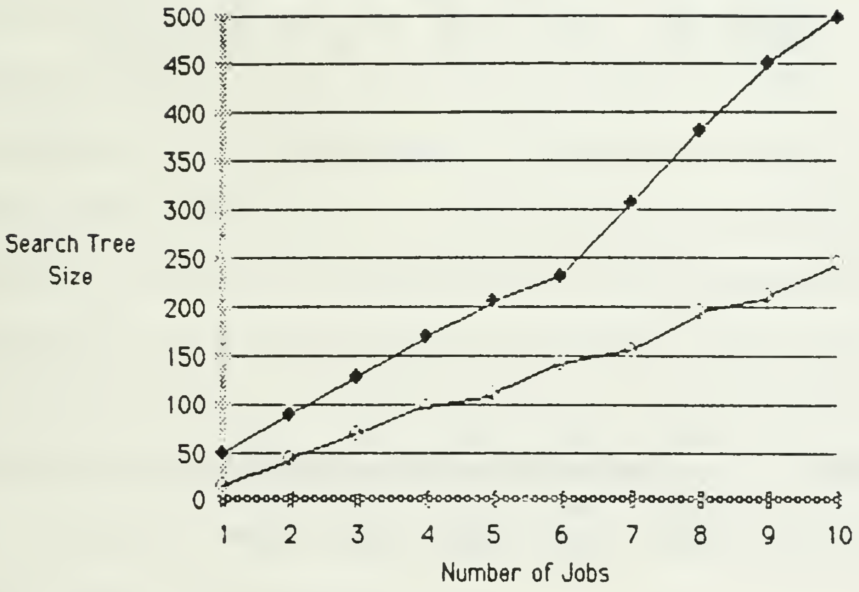

Figure V.3

Computation Reduction Due to Heuristic Searching

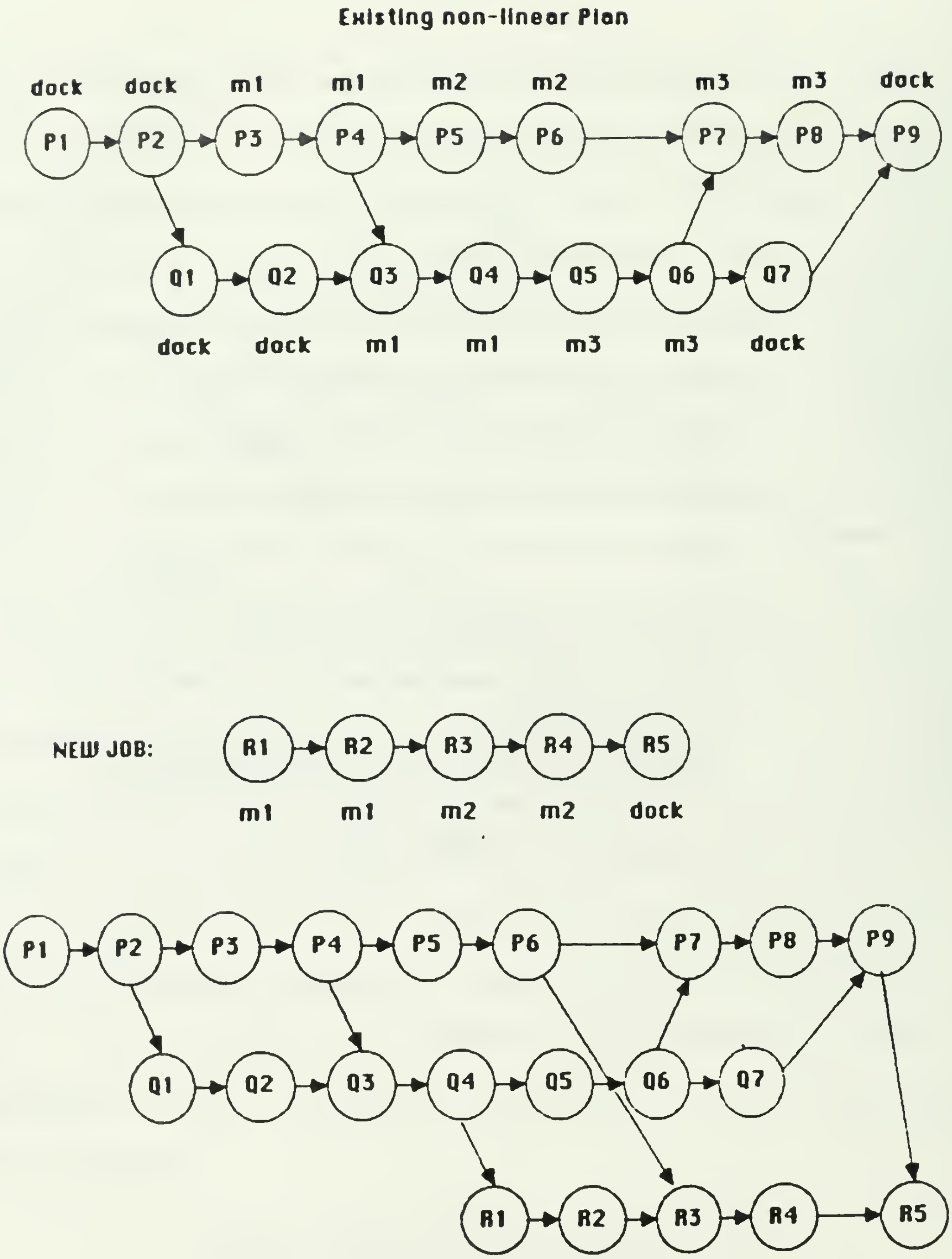

Figure V.4 Dynamic Scheduling for
Newly Arrived Jobs

bidding scheme with other approaches used for manufacturing task assignment in prior research. Specifically, we compared the bidding method with the centralized, dynamic dispatching method; and (2) to evaluate the performance of the bidding scheme using different heuristics in bidding. For this purpose, the shortest-processing-time (SPT) heuristic and the earliest-finishing-time (EFT) heuristic are evaluated. To this end, three coordination strategies are tested for performance comparison: (1) Myopic-SPT, a centralized scheme employing shortest-processing-time as the dispatching rule; (2) Bidding-SPT, a distributed scheme employing shortest-processing-times to calculate bids; (3) Bidding-EFT, a distributed scheme employing earliest-finish-times to calculate the bids.

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 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

$$\text{Due-date} = \text{TNOW} + (\text{estimated total processing time}) * 1.3 + (\text{no. of operations}) * \text{SD}.$$

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 Figures VI.1 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.

The simulation results for the three scheduling methods performed on the six-cell systems are shown in Figure VI.1. 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) The bidding-SPT method performs significantly better than the myopic-SPT method, also in terms of late jobs, in-process waiting time, tardiness, and mean flow-time.

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

arameter Set	I				II				III			
eplications	1	2	3	4	1	2	3	4	1	2	3	4
% 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	18.6
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	6.4
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	14.

(a)

Parameter Set	I				II				III			
Replications	1	2	3	4	1	2	3	4	1	2	3	4
% 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	39.
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	7.
Avg. lateness	4.90	5.23	6.38	5.24	2.32	2.54	3.41	2.78	1.42	1.44	1.91	1.
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	15.

(b)

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	39.
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	7.
Avg. lateness	6.18	6.35	7.17	6.89	2.92	3.47	5.00	3.65	1.65	1.87	2.32	1.
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	15.

(c)

Figure VI.1 Simulation Results of Using (a) Bidding-EFT, (b) Bidding-SPT and (c) Myopic-SPT Strategies

"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]). However, our results further show that having more up-to-date information still results in significantly 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 the reliability would improve as a result, 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.

VII. Conclusions

We have shown a cooperative problem-solving method for scheduling cellular FMS. The method has the following features: (1) it employs distributed control; no cell has greater importance, as far as scheduling is concerned, than any other cell; (2) the underlying scheduling heuristic is flexible, and can take into account such information as loading factor, unexpected break-downs, or resource constraints in the bidding scheme; (3) compared with the traditional manufacturing control methods, the bidding mechanism is characterized by its more accurate information contents without spending the cost of constant updating--the performance improvement by such information is verified by simulation results; (4) based on the cooperative problem-solving paradigm, we can show a scheduling method 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 distributed organization enables us to take a divide-and-conquer strategy to manage the whole flexible manufacturing system, that is, the knowledge-base in each cell is a stand-alone decision-making unit, thus greatly simplifying the complexity of information processing requirements; and (6) the scheduler at each cell uses heuristic procedure to generate schedules--it can adapt to changes of the manufacturing environment.

An interesting characteristic of the methodology is that an unified framework is used for cellular scheduling and for achieving inter-cell cooperation; both are treated as state-space problem-solving processes. We have also shown the implementation aspect of the cooperative problem-solving method, which uses the augmented Petri-net to model the bidding scheme, the controlled production system to execute the coordination mechanism, and a knowledge-base system to carry out cell-level scheduling.

Acknowledgements

This research is supported in part by grants from the Office of Information Management sponsored by IBM, the Research Board of the University of Illinois, and the Amoco Foundation Professorship. Thanks are due to J. Yu and H. Ho-Chen who ran the simulation programs, and to S. Park who ran the knowledge-based program for the computation study on cell-level scheduling.

References

[1] Axelrod, R., 1984, The Evolution of Cooperation (New York: Basic Books, Inc.).

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

[3] 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.

[4] 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.

[5] 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.

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

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

[8] 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.

[9] Danthine, A., 1980, Protocol Representation with Finite State Models, IEEE Transactions on Communications, pp. 632-642.

[10] Davis, R., 1981, A Model for Planning in a Multi-Agent Environment: Steps Toward Principles for Teamwork, Working Paper 217, MIT AI Lab.

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

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

[13] Georgeff, M., 1982, Procedural Control in Production Systems, Artificial Intelligence, Vol. 18, pp. 175-201.

[14] 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.

[15] Harris, M. and Raviv, A., 1981, Allocation Mechanisms and the Design of Auction," Econometrica, Vol. 49, No. 6, pp. 1477-1499.

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

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

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

[19] 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.

[20] Malone, T. and Smith, S., 1984, Tradeoffs in Designing Organizations: Implications for New Forms of Human Organizations and Computer Systems, Working Paper No. 1541-84, Sloan School of Management, Massachusetts Institute of Technology.

[21] 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.).

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

[23] 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.

[24] Nelson, R., Haibt, L., and Sheridan, P., 1983, Casting Petri Nets Into Programs, IEEE Trans. on Software Engineering, Vol. SE-9, No. 5, pp. 590-602.

[25] Newell, A. and Simon, H., 1972, Human Problem Solving, (Prentice-Hall: Englewood, New Jersey).

[26] Nilsson, N., 1980, Principles of Artificial Intelligence, (Tioga: Palo Alto, California).

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

[28] Peterson, J., 1981, Petri Net and the Modeling of Systems, Englewood Cliffs, NY: Prentice-Hall.

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

[30] Ranky, P., 1986, Computer Integrated Manufacturing, (Prentice-Hall: Englewood, New Jersey)

[31] Rosenschein, J. and Genesereth, M., 1984, Communication and Cooperation, Technical Report HPP-84-5, Computer Science Department, Stanford University.

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

[33] 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.

[34] Shaw, M., 1986a, A Two-level Approach to Scheduling in Computer Integrated Manufacturing, Proceedings of the NBS Symposium on Real-Time Optimization in Automated Manufacturing Facilities, National Bureau of Standards, U.S.A.

[35] Shaw, M., 1986b, A Distributed Scheduling Scheme for Computer Integrated Manufacturing: The Use of Local Area Network for Cellular Systems, BEBR Working Paper No. 1244, University of Illinois.

[36] Shaw, M., 1986c, A Pattern-Directed Approach to FMS Scheduling, Proceedings of 2nd ORSA/TIMS Conference on Flexible Manufacturing Systems, Ann Arbor, Michigan.

[37] Shaw, M., 1986d, Applying Artificial Intelligence to the Scheduling of Flexible Manufacturing Systems: A Pattern-Directed Approach, Faculty Working Paper No. 1301, College of Commerce and Business Administration.

[38] Shaw, M. and Whinston, A., 1985a, A Distributed Knowledge-Based Approach to Flexible Automation, Working Paper, Decision and Information Sciences, University of Illinois. A shorter version appeared in Proc. IEEE Conference on A.I. Application.

[39] Shaw, M. and Whinston, A., 1985b, Automated Planning and Flexible Scheduling, Proc. Int. Conference on Automation and Robotics, St. Louis, MO.

[40] Shaw, M. and Whinston, A., 1986, Application of Artificial Intelligence to Planning and Scheduling in Flexible Manufacturing, in Flexible Manufacturing System: Methods and Studies, A. Kusiak (Ed.), North-Holland.

[41] 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.

[42] Simon, H., 1982, The Sciences of the Artificial, (The MIT Press: Cambridge, Massachusetts).

[43] 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.

[44] 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.

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

[46] Stecke, K. and Schmeiser, B., 1982, Equivalent Representation of System Throughput in Closed Queueing Network Models of Multiserver Queues, Working Paper No. 324, Divison of Research, Graduate School of Business Administration, The University of Michigan, Ann Arbor, Michigan.

[47] Stecke, K. and Solberg, J., 1985, The Optiality of Unbalancing Both Workloads and Machine Group Sizes in Closed Queueing Networks of Multiserver Queues, Operations Research, 33, 4, pp. 882-910.

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

[49] 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).

HECKMAN
BINDERY INC.
DEC 95
Bound -To -Please® N. MANCHESTER,
INDIANA 46962

UNIVERSITY OF ILLINOIS-URBANA
3 0112 060296057