

Agent-Based Modeling of Supply Chains for Distributed Scheduling

Jason S. K. Lau, George Q. Huang, K. L. Mak, and L. Liang

Abstract—This paper considers a supply chain that comprises multiple independent and autonomous enterprises (project managers) that seek and select various contractors to complete operations of their project. Both the project managers and contractors jointly determine the schedules of their operations while no single enterprise has complete information of other enterprises. The centralized scheduling approach that can usually obtain good global performance but must share nearly complete information that is difficult or even impractical due to the distributed nature of real-life supply chains. This paper proposes an agent-based supply chain model to support distributed scheduling. A modified contract-net protocol (MCNP) is proposed to enable more information sharing among the enterprises than conventional CNP. Experimental simulation studies are conducted to compare and contrast the performances of the centralized [centralized heuristic (CTR)], conventional CNP, and MCNP approaches. The results show that MCNP outperforms CNP and performs comparably with CTR when project complexity is high in terms of the total supply chain operating cost. Moreover, it is found that although CTR is better than MCNP in terms of global performance, MCNP yields good schedule stability when facing unexpected disturbances.

Index Terms—Agent-based modeling, distributed scheduling, information sharing, project scheduling, supply chain.

I. INTRODUCTION

THIS paper considers a distributed scheduling problem in manufacturing supply chains. In such supply chains, there are multiple enterprises that offer different products to their customers. One example of the enterprises is original equipment manufacturer (OEM). Generally, there are two supply chain management (SCM) paradigms for the enterprises to meet the demands from the customers. The paradigms are make-and-sell and sense-and-respond [1]. Make-and-sell enterprises focus on forecasting demand of customers and managing production according to the target established by forecasting. The customers' demand is fulfilled by available inventory of the product. The enterprises can save costs due to the advantage of mass production. On the other hand, sense-and-respond en-

terprises aim at agilely responding to customers' demand. The production is tailored to satisfy the requirements from diverse customers. This characteristic is called mass customization. The make-and-sell paradigm is well-suited for the market with wide consumption, characterized by a very long product life cycle, a stable demand pattern, and low differentiation. On the other hand, products that have volatile demand and a short life cycle need a sense-and-respond supply chain paradigm. As global competition becomes fierce nowadays, providing differentiated products and services in a cost-effective way is the key for surviving in and, also, leading the market. Such differentiation is best captured by mass customization in the sense-and-respond paradigm. The way to achieve cost-effective mass customization is, therefore, of paramount importance for the enterprises that want to succeed in the market.

The rapid advancement of the Internet and web technologies enables enterprises to capture (or "sense") customers' requirements efficiently. The next problem is how the enterprises respond to the customers' requirements in a cost-effective way. In recent years, both industrial practitioners and academia advocate outsourcing as a tool for improving productivity and reducing operating cost [2]–[4]. The sense-and-respond enterprises can exploit the expertise of other enterprises (e.g., contract manufacturers) for performing production operations. A well-known example of sense-and-respond paradigm is Dell's supply chain. The configuration of personal computers (PCs) is determined by consumers on Dell's web site. Dell then arranges the contract manufacturers and suppliers for producing and supplying the components [e.g., central processing unit (CPU), hard disk, random access memory (RAM), etc.] of the PC according to the location and preferences of the consumer. All the suppliers and contract manufacturers need to join the Dell's trading network in order to be potential partners of Dell. The large PC market share (second largest in world's sales) and effective selling channels (e.g., online sales) justify Dell's investment on the information technology (IT) infrastructure. It is, however, not economically feasible for small- and medium-sized OEMs to invest in the IT infrastructure as complicated as Dell's. The OEM also does not have the power to force its supply chain partners to join its IT platform. A new modeling approach is needed to enable the small- and medium-sized OEMs to find suitable manufacturing partners that are distributed in different locations.

This paper proposes a new modeling approach of configuring and scheduling multiple projects in a network of enterprises or virtual enterprises, which consist of project managers and contractors. The project manager coordinates and manages the process of producing a product on a make-to-order (MTO)

Manuscript received March 19, 2003; revised November 8, 2003 and February 5, 2004. This work was supported in part by the Committee on Research and Conference Grants (CRCG) of The University of Hong Kong and the National Natural Science Foundation of China (NSFC) under Grant 70371023. This paper was recommended by Associate Editor L. Fang.

J. S. K. Lau, G. Q. Huang and K. L. Mak are with the Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Hong Kong (e-mail: fractals@graduate.hku.hk; gquang@hkucc.hku.hk; makkl@hkucc.hku.hk).

L. Liang is with the Business School, University of Science and Technology of China, Hefei 230026, China.

Digital Object Identifier 10.1109/TSMCA.2005.854231

basis. The process of realizing the product is described by a project. Each project consists of many tasks or operations with complex precedence relationship. Each operation can be performed by a set of alternative partnering enterprises or contractors. Each contractor is capable of performing one or more types of operation. Contractors differ from each other in terms of resource capacity, processing times, and costs of performing operations. Since the contractors are dispersed in different geographical areas, the transportation cost and duration between different contractors are significant when compared with the processing time of operations. The configuration of a project refers to selection of contractors for performing operations of the project. The scheduling of a project refers to determining when the operations start. Decisions of the project configuration are made by project managers while decisions of scheduling operations are jointly determined by project managers and contractors. The objective of the configuration and scheduling is to minimize the costs of performing operations, transportation, and tardiness. The resulting schedule of a project manager should respect the precedence constraints between the operations. On the other hand, a contractor wants to get as many operations as possible from the project managers to maximize its revenue without violating its capacity constraints. In general, we use the term “supply chain scheduling problem” (SCSP) to describe the above-mentioned problem.

The supply chain considered in this paper has the following three characteristics.

- 1) *Information privacy*. There is no single enterprise that has global information about the supply chain. For example, a project manager does not know about the capacity and current operations of the contractor. Furthermore, both project managers and contractors are not willing to share their operating information to others due to privacy reason. This characteristic is manifested by the present global and interorganizational supply chain.
- 2) *Multiobjectives*. Participants in the supply chain have different objectives. The objective of project managers is to minimize operating costs of the project. The objective of contractors is to maximize the revenue earned by performing operations.
- 3) *Distributed decision making*. Each of the project managers and contractors makes its operating decision independently and autonomously. No single enterprise can control the decision making of the others.

The purpose of this research is “not” to develop an optimal or near-optimal algorithm to solve SCSP. As the problem is complicated and the agents lack of global knowledge in the problem solving process, finding an optimal or near-optimal solution is virtually impractical in real supply chains. Instead, we aim at modifying a well-known distributed problem solving approach, contract-net protocol (CNP), to address SCSP. The modification allows multiple project managers simultaneously select contractors for performing their operations. We also propose to share time window information by project managers to contractors in order to improve the performance of the modified contract-net protocol (MCNP). We will evaluate MCNP

with the proposed level of information sharing by comparing its performance with a centralized heuristic (CTR), which requires full information sharing, and the conventional CNP, which requires minimal information sharing. This research is unique because effects of different levels of information sharing on the supply chain performance in SCSP have not been investigated in literature. Moreover, there are few studies that compare agent-based coordination approaches with centralized approaches in supply chain problems. The objectives of this paper are as follows: 1) to establish a mathematical model of SCSP; 2) to develop an agent-based model for SCSP; 3) to propose an MCNP to coordinate agents to solve SCSP; and 4) to evaluate the global performance of MCNP with the centralized approach and conventional CNP for the distributed project scheduling problem.

The remainder of this paper is structured as follows. In Section II, a literature review on centralized approaches and CNPs in supply chain planning and scheduling problems is presented. In Section III, a mathematical formulation of the scheduling problem will be described. In Section IV, an agent-based model of the supply chain is presented. In Section V, the MCNP is described. In Section VI, a CTR to solve the distributed project scheduling problem is presented. In Section VII, the performance of MCNP is evaluated. In Section VIII, the paper will be concluded.

II. LITERATURE REVIEW

There are two general approaches to supply chain planning and scheduling: 1) centralized; and 2) distributed. The centralized approach in which computation of the planning and scheduling are performed by a centralized authority has been investigated for many years [5]–[9]. The information required for planning may be centralized or distributed depending on the nature of problem. Erenguc *et al.* [10] gave an excellent review of centralized planning. The centralized approach requires input of production information (e.g., lead time, operating costs, etc.) from all of the firms involved in a supply chain. An optimal plan is generated by one of the firms that acts as a coordinator and sent back to other firms (e.g., contractors, component suppliers). One of the established methods of finding the optimal plan is mathematical programming. Gaonkar and Viswanadham [11] formulated a linear programming model for solving supply chain planning problem under information sharing. In their model, each firm in the supply chain shares all of its production information to other supply chain partners through an Internet-based platform. The centralized approach, however, has several barriers in implementation despite of its reported benefits. One of the major barriers is due to reluctance for a firm to share its private and sensitive information like operating costs with other firms, which may well be its competitors. Another worry stems from the security and reliability of sharing sensitive information through the Internet. Even though each supply chain partner agrees to freely share its production information to all the others, the computational time of solving a real industrial problem is tremendous. Moreover, the integrated approach makes it inefficient to react to random events (e.g., transportation delay) in the supply chain. The solution or

schedule needs to be recomputed completely whenever there is an update of local information of each supply chain firm.

The distributed approach, on the other hand, requires limited sharing of key information between supply chain partners. This approach is characterized by decentralization of computation and information in a supply chain. The rationale behind the distributed approach is that each enterprise makes its scheduling decision based on its local information and objective. The enterprises exchange their scheduling results and try to find a common agreement on the global schedule. The process continues until some termination conditions (e.g., a feasible schedule is obtained) are met. Durfee [12] discussed several methods in distributed problem solving. By using distributed approaches, the enterprises need not share all local information with each other. The computational efficiency of problem solving can be improved as the local problem of each enterprise is less complicated than the centralized one and the local problems can be solved concurrently by a network of computers in the supply chain. Moreover, as enterprises can act autonomously, they respond to local disturbances more quickly in the centralized approach. The distributed approach is, therefore, more reactive than the centralized approach.

A celebrated distributed approach to planning and scheduling is CNP. CNP specifies a bidding approach that enables task allocation among multiple agents [13]. CNP consists of two types of decision makers: 1) managers; and 2) contractors. A manager seeks a contractor to complete a task. It sends a request to all eligible contractors for getting bids of the task. After receiving the bids from the contractors, the manager awards the task to the best contractor based on certain criteria. CNP has been adopted extensively in job shop scheduling because of its simplicity and intuitiveness. Job and machine are represented by manager and contractor in CNP, respectively. Vancza and Markus [14] applied market mechanism (based on job price) for resource allocation in job shop scheduling. Macchiaroli and Riemma [15] also proposed a price mechanism in CNP to assign parts to resources in flexible manufacturing systems (FMSs). Sousa and Romas [16] presented a negotiation protocol based on CNP for scheduling in a holonic manufacturing system. They proposed forward and backward propagation of scheduling information between resource holons. A similar backward propagation of contract in CNP can be found in [17]. The manager starts task announcement from the last operation of the job. Temporal constraints of the operation are propagated to preceding operations for further announcements. Baker [18] demonstrated the viability of CNP in a real job shop environment. Saad *et al.* [19] proposed an FMS scheduling policy called production reservation. Each job is scheduled only when it arrives. Xue *et al.* [20] combined constraint-based search and agent-based collaboration for production scheduling. CNP is used to collect resource information and assign operations to resources. In addition to manufacturing scheduling, Fischer *et al.* [21] also reported a successful application of CNP in transportation domain. Most of the studies on application of CNP in manufacturing scheduling consider one job at a time. If there are several jobs arrive, the jobs are

scheduled sequentially by applying CNP to each job. This leads to unsatisfactory global performance. Moreover, in manufacturing domains, agents can use some global information of the system in their decision making. For example, part agents proposed in [15] know the slack information of all parts in the system. In the holonic manufacturing system described in [16], resource holons need to know the downstream and upstream resources in order to propagate scheduling information. Availability of global information may not be possible in realistic supply chains as discussed in Section I. Recently, Naso and Turchiano [22] have proposed an intelligent decision making and coordination strategies for part and workstation agents to solve dynamic part routing problems in automated manufacturing systems. Like previous research, the multiagent system requires a supervisory agent to resolve conflicts. Sousa *et al.* [23] proposed a new version of CNP called “contract net with constraint propagation protocol” (CNCPP) for scheduling manufacturing systems. CNCPP allows multiple task agents to concurrently negotiate with resource agents. The main advantage of CNCPP is the existence of explicit cooperation between the resource agents in order to guarantee feasible schedules. This advantage, however, may not be appealing in supply chains in which contractors or suppliers are not supposed to frequently and truthfully share local information.

Agent-based modeling has been considered as a suitable approach to modeling distributed scheduling problem in supply chains. Zhang [24] used a PC supply chain example to illustrate their negotiation protocol. They investigated the effects of varying attitudes from self-interested and completely cooperative on social welfare of the agent society (i.e., sum of rewards of all agents). Collins [25] developed a bidding system for multiagent contract negotiation. The system facilitates customers to contract a network of tasks to suppliers based on CNP. Their work mainly focused on presenting a bid selection algorithm for the customers to determine an optimal selection of the suppliers. Reis *et al.* [26] proposed a coordination mechanism for scheduling a network of jobs in an extended enterprise. In their multi-agent system (MAS) model, there is only one supplier for each job and only one customer order is processed at a time. The product structure is an in-tree structure. Lee *et al.* [27] developed a market-based control mechanism for controlling distributed multiple projects in supply chains. Agents are used to represent projects and resources. The market-based approach, however, does not guarantee a feasible project schedule. Wagner *et al.* [28] examined the production scheduling problem of a small volume manufacturing supply chain that produces build-to-order goods. Their model did not consider alternative component manufacturers and capacity constraint of resources. Most of the abovementioned agent-based supply chain models considered only a single customer order and did not address the competition of resources among customers. The models did not support alternative resources with multiple capacities. There also is a lack of comparison between the agent-based and centralized modeling approach for supply chain problems.

The agent-based approach is, however, not without limitations. A major drawback of such approach is that the feasibility

and optimality of the solution are sacrificed when compared with the centralized approach [29]. Due to the lack of global information, agents make their decisions independently and optimize their local objectives without considering other agents' constraints and global performance. We are motivated to investigate how to improve the global performance of CNP to solving the distributed project scheduling problem by sharing more information among the enterprises.

III. PROBLEM FORMULATION

In the formulation, N and M represent the sets of project managers and contractor, respectively. The project manager and contractor are indexed by n and m , respectively. The scheduling time horizon is T with index t ranging from 1 (the current period) to T . Each project manager aims at completing a project with a given due date and tardiness cost. In the rest of this paper, subscripts n and m will be omitted when there is no ambiguous interpretation of symbols.

A project P is defined as a directed acyclic graph $\langle J, E \rangle$ where J is a set of operations (nodes) in a project and E is a set of ordered pair (edges) (i, j) where $i, j \in J$. An ordered pair of operations (i, j) represents a precedence relationship, namely operation i immediately precedes operation j where $i, j \in J$. The index of operation is labeled by the following rule: $(i, j) \Rightarrow i < j$. For example, operation 1 precedes operation 2. The index of an operation must be greater than the indexes of its preceding operations and smaller than the indexes of its succeeding operations. A set of beginning operations is denoted by BJ . A beginning operation $i \in BJ \subset J$ satisfies $(\phi, i) \in E$. The last operation is denoted by f , which satisfies $(\phi, i) \in E$ where ϕ is a null operation. It is assumed that there is one and only one last operation in P . Notice that $f = \max_{j \in J}(j)$.

The project information of a project manager n is denoted by J_n and E_n . The project manager needs to complete the project before a due date D_n given by its customer. When the due date is violated, the project manager is penalized by a unit tardiness cost tc_n . In this sense, the due date is, therefore, a "soft" due date. When an operation is completed earlier (i.e., a positive slack time with its succeeding operation), a unit earliness cost ec_i is incurred. Each operation i of the project can be started only after a known release time rp_i . The release time may represent the available date of some documents or financial arrangement that enables realization of the operation.

A set of contractors that are capable of performing operation i is denoted by AC_i . Contractor m has a single resource, which can be a group of machines or workers. The capacity of the resource in each time period t is denoted by CP_{mt} . The contractor can perform more than one operation at a time. Processing time, resource requirement, and cost of performing operation i are denoted by l_{im} , q_{im} , and c_{im} , respectively. The lead time and cost of transporting the output of operation i from contractor agent (CA) h to CA k are lt_{ihk} and lc_{ihk} , respectively. It is assumed that, in the current model, the transportation task can be performed by some logistic service providers that have sufficient capacity to perform the task. Operation i can only be started after a release time rc_i given

by the contractor. The release time indicates the availability of materials and tooling for performing the operation.

A global schedule of a supply chain is defined as $S = \{S_1, \dots, S_n\}$. $S = \langle X_n, Z_n \rangle$ is a local schedule of project manager n . $X_n = \{X_1, \dots, X_i, \dots, X_f\}$ is a vector of start times of operations. $Z_n = \{Z_{im}\}$ represents selection of contractors for operations. $Z_{im} = 1$ indicates contractor m is selected for operation i , $Z_{im} = 0$ indicates that contractor m is "not" selected for operation i . Only one contractor can be selected for an operation. X_n and Z_n represent scheduling and configuration decisions of project agent (PA) n , respectively.

A mathematical programming formulation of SCSP is shown as

$$\begin{aligned} \min \\ \text{TC} = \sum_{n \in N} \left[\sum_{i \in J_n} \left(\sum_{m \in AC_i} Z_{im} c_{im} + EC_{ni} \right) \right. \\ \left. + \sum_{(i,j) \in E_n} \sum_{h \in AC_i, k \in AC_j} Z_{ih} Z_{jk} lc_{ihk} + dc_n(X_f) \right] \end{aligned} \quad (1)$$

subject to

$$\begin{aligned} X_i + \sum_{h \in AC_i} Z_{ih} l_{ih} + \sum_{h \in AC_i, k \in AC_j} Z_{ih} Z_{jk} lt_{ihk} \leq X_j \\ \forall n \in N, \forall i, j \in J_n, \forall (i, j) \in E_n \end{aligned} \quad (2)$$

$$\begin{aligned} \sum_{n \in N} \sum_{i \in J_n} \sum_{X_i \in [t-l_{im}+1, t]} q_{im} Z_{im} \leq CP_{mt} \\ \forall m \in M, \forall t \in [1, T] \end{aligned} \quad (3)$$

$$\sum_{m \in M} Z_{im} = 1 \quad \forall n \in N, \forall i \in J_n \quad (4)$$

$$\begin{aligned} X_i \in \left[\max \left(rp_i, \sum_{m \in AC_i} Z_{im} rc_{im} \right), T \right] \\ \forall n \in N, \forall i \in J_n \end{aligned} \quad (5)$$

$$Z_{im} \in \{0, 1\} \quad \forall n \in N, \forall i \in J_n, \forall m \in M. \quad (6)$$

The objective of the problem is to minimize the total operating cost (TC) of the supply chain (1). The first part of TC is the sum of the costs of performing the operations on the selected contractor and the earliness cost of each operation as defined by

$$\begin{aligned} EC_{ni} = ec_i \sum_{(i,j) \in E_n} \left(X_j - \sum_{h \in AC_i, k \in AC_j} Z_{ih} Z_{jk} lt_{ihk} \right. \\ \left. - X_i - \sum_{h \in AC_i} Z_{ih} l_{ih} \right). \end{aligned}$$

The earliness cost penalizes idle time between consecutive operations. The second part is the transportation cost between

two operations. The transportation cost depends on the choices of contractors. The last part is the tardiness–earliness cost function of the project as defined by

$$dc_n(t) = \begin{cases} tc_n(t + l_f - D_n), & t + l_f > D_n \\ ec_f(D_n - t - l_f), & t + l_f \leq D_n \end{cases}$$

where $l_f = \sum_{m \in AC_f} Z_{fm} l_{fm}$.

Constraint (2) represents the precedence constraints between operations of each project. It means that the start time of an operation must be greater than the arrival times of the outputs of its preceding operations. The arrival time is the left part of constraint (2). The constraint includes the transportation lead time between contractors, shown in the last term of the left part. The capacity constraints of contractors in each period are represented by constraint (3). Constraint (4) restricts that only one contractor can be selected for each operation. Operations cannot be divided into batches for separate processing by different contractors. Constraint (5) indicates that operations must start equal to or later than the release dates. Constraint (6) is the domain constraint of decisions variable of contractor selection. A global schedule S is feasible if and only if X and Z satisfy constraints (2) to (6).

IV. AGENT-BASED INFORMATION MODELING FOR SUPPLY CHAIN SCHEDULING

In this study, we model a supply chain as a multiagent system. An agent is used to model the scheduling process of an enterprise. It is implemented as a software program that is located at the enterprise and aims at building a feasible schedule by coordinating with other agents in the supply chain. The two types of enterprises, project manager and contractor, are represented by PA and CA, respectively. Fig. 1(a) shows an instance of an SCSP that is modeled as a multiagent system. Each PA aims at completing a project that consists of a network of operations. A contractual relationship between a PA and a CA is said to be established when the PA selects the CA to perform one of its operations [see the solid lines in Fig. 1(a)]. A PA determines the schedule of the operations with respect to the precedence constraints and transportation constraints between the CAs. Fig. 1(b) shows the transportation constraints between CAs. The CA constructs its local schedule of operations obtained from PAs based on its capacity constraint. Each agent has its own scheduling objective and acts independently to each other. Since the capacity of CA is limited, there is resource competition among PAs. The schedule of operations is jointly determined by PAs and CAs. This is done by exchanging bids that contain proposed start time and related information of operations. Moreover, no agent has global information of other agents and the whole supply chain.

In addition to PA and CA, there is a middle agent (MA) that facilitates the scheduling process collectively performed by PAs and CAs. Specifically, MA has four core functionalities: 1) registering PAs that use resources in the supply chain; 2) allowing CAs to advertise their capabilities of performing

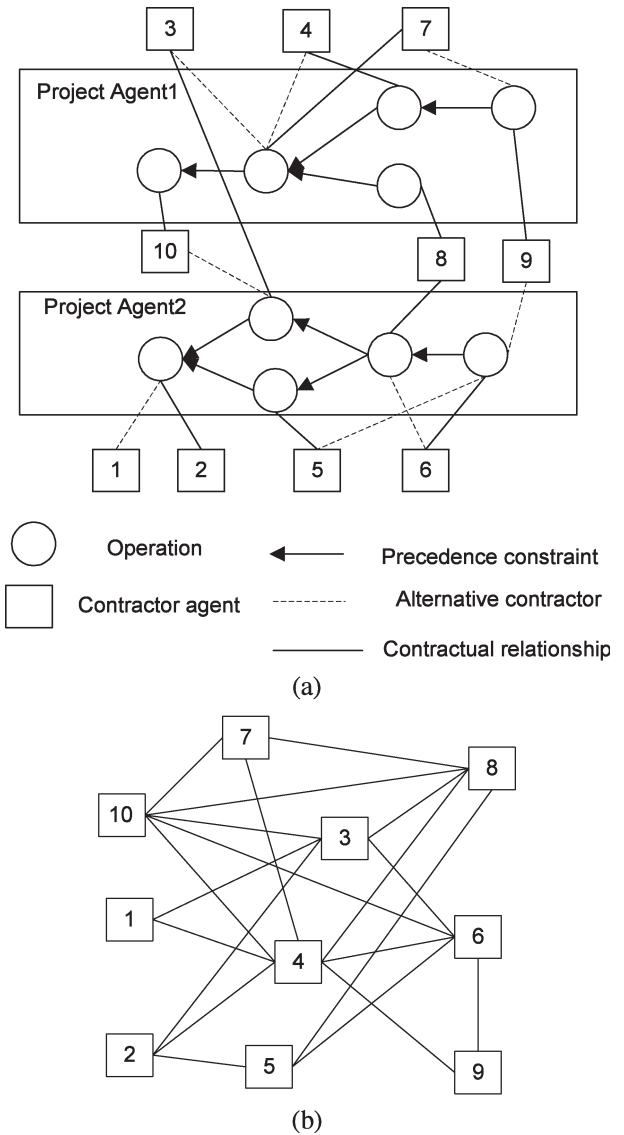


Fig. 1. (a) Instance of a supply chain composed of two project agents and ten contractor agents. (b) Transportation constraints between contractors.

operations; 3) providing directory of CAs’ capability and services for PAs; and 4) coordinating the scheduling process. Fig. 2 depicts the relationship among PAs, CAs, and MAs. There are many types of MA proposed in the MAS literature [30]. Different types of MA have different levels of participation in the interaction among the agents and require different levels of information sharing from the agents. In the MAS proposed in this paper, the MA does not know about schedule information of and contractual relationships among PAs and CAs. The level of participation of MA in the scheduling process varies with the level of information sharing. Section V will discuss the role of MA in the scheduling process under different levels of information sharing.

A. Project Agent’s Model

There are two objectives of a PA n . One objective is to find a feasible local schedule $S_n = \langle X_n, Z_n \rangle$ that satisfies all

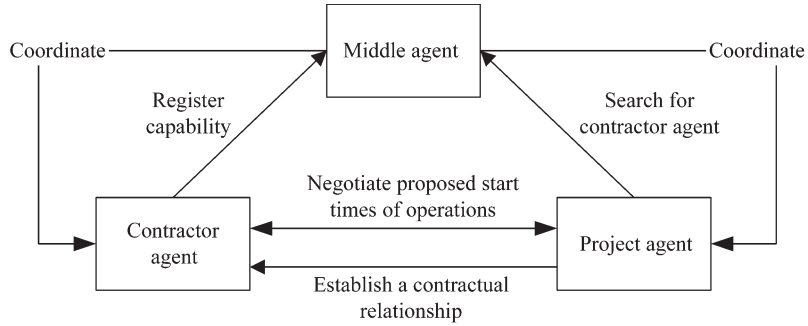


Fig. 2. Agent-based modeling for supply chain scheduling.

precedence constraints between operations. Another objective is to minimize the operating cost of the project.

PA's knowledge is defined as $KN = \{P, D, tc, \{ec_i, rp_i, l_i\}, \{AC_i\}, \{\{lt_{ihk}, lc_{ihk}\}\}, \{B_i\}\}$, which are the project information, contractor capability, transportation lead times and costs, and bids information (B_i) of CAs, respectively. \hat{l}_i is an estimated processing time of operation i . We assume that a PA can estimate the process times of its operations by the experience of the project manager. The capability of contractors can be obtained from the directory service offered by the MA. The transportation lead times and costs between two contractors can be obtained from logistic service providers. A bid (b) of operation (i) submitted by a CA (m) is defined as a tuple $\langle i, m, p_{im}, Y_{im}, l_{im} \rangle$ where p_{im} is the price of performing the operation, Y_{im} is the proposed start time, and l_{im} is the proposed processing time.

States of PA is defined as $ST = \{S_n, \{es_i, ls_i, lbs_i\}, C(S_n), SchST\}$, which are the local schedule, temporal information of the schedule, operating cost of the project, and states of scheduling process (SchST), respectively. In a feasible schedule, we define an alternative notation of selected contractor of operation i as $z(i)$. That is, $Z_{im} = 1 \Leftrightarrow z(i) = m$, where m is the selected contractor. Given a feasible local schedule, a set of temporal information of every operation i can be derived using (7) shown at the bottom of the page.

ls_i is the latest start time and es_i is the earliest start time. A continuous time window of operation i is described by $[es_i, ls_i]$. The start time of operation i can be changed within the time window without affecting other operations. lbs_i is the lower bound of feasible start time. It is obtained by propa-

gating the release time for beginning operations to the last operation. It is obvious that $lbs_i \leq es_i \leq ls_i$. The time window can be violated, but the lower bound of start time is a hard constraint. The operating cost of PA's project is defined as

$$C(S_n) = \sum_{i \in J_n} \left(\sum_{m \in AC_i} Z_{im} c_{im} + EC_{ni} \right) + \sum_{(i,j) \in E_n} lc_{i,z(i),z(j)} + dc_n(X_f).$$

SchST will be described in Section V.

B. Contractor Agent's Model

The objectives of a CA are to find a capacity-feasible schedule for performing operations for PAs and maximize the revenue obtained by performing operations for PAs. Knowledge of CA m is defined as $KN = \{\{CP_{mt}\}, \{c_{im}, l_{im}, q_{im}\}, RB\}$, which consists of capacity in each time period, parameters of performing operations i (i.e., cost, processing time, and capacity requirement), and a set of bids for performing operations of PAs (RB).

States of CA is defined as $ST = \{\{Y_b, ces_b, cls_b\}, R_m, ACP_{mt}, SchST\}$ where $\{Y_b\}$ is a set of start times of operations in RB. It also represents the local schedule of the CA. ACP_{mt} denotes the available capacity in period t . The available capacity is obtained by deducting capacity requirement of operations in period t from total capacity CP_{mt} . A set of

$$\begin{aligned} es_i &= \begin{cases} \max \left(\max_{(j,i) \in E_n} (X_j + l_{j,z(j)} + lt_{j,z(j),z(i)}), rp_i \right), & i \in \frac{J_n}{B J_n} \\ rp_i, & i \in \frac{J_n}{B J_n} \end{cases} \\ ls_i &= \begin{cases} \min_{(i,j) \in E_n} (X_j - lt_{i,z(i),z(j)} - l_{i,z(i)}), & i \neq f \\ \max (X_j, D_n - l_{i,z(i)}), & i = f \end{cases} \\ lbs_i &= \begin{cases} \max \left(\max_{(j,i) \in E_n} (lbs_j + l_{j,z(j)} + lt_{j,z(j),z(i)}), rp_i \right), & i \in \frac{J_n}{B J_n} \\ rp_i, & i \in \frac{J_n}{B J_n} \end{cases} \end{aligned} \quad (7)$$

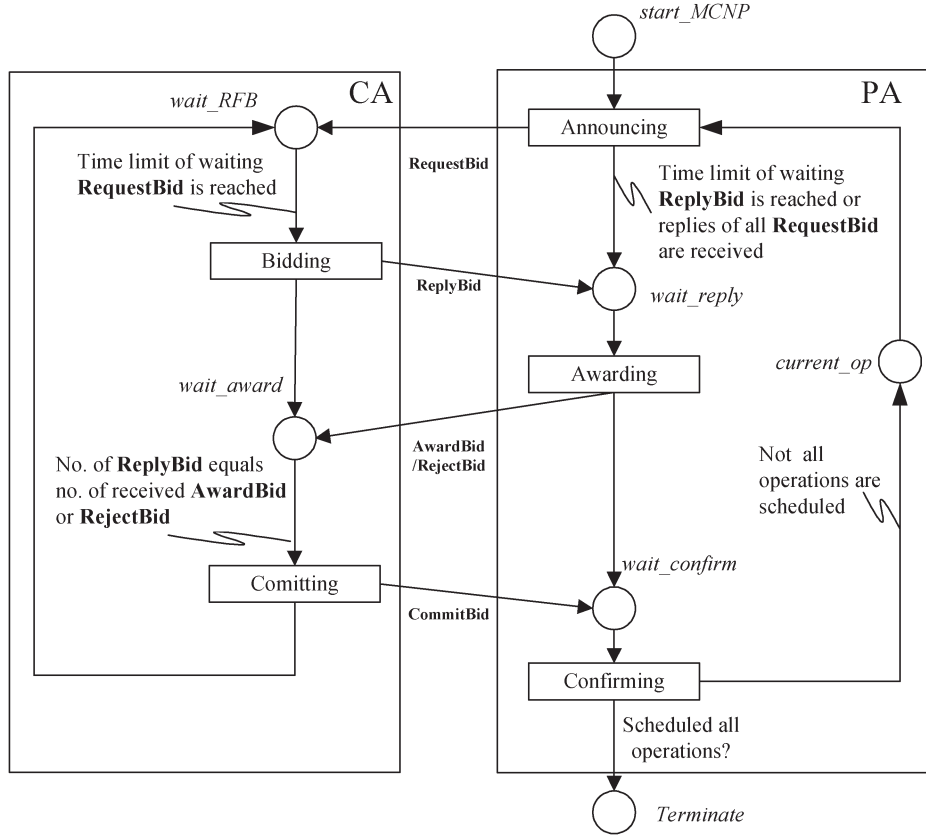


Fig. 3. State transition diagram of MCNP.

temporal information that can be obtained from the local schedule is given as

$$ces_b = \max_{rc_{i(b),m} \leq v < Y_b} \left(v + 1 | q_{i(b),m} > ACP_{mt} \right)$$

$$\forall t \in [v, v + l_{i(b),m} - 1]$$

$$cls_b = \min_{Y_b < v \leq T} \left(v - 1 | q_{i(b),m} > ACP_{mt} \right)$$

$$\forall t \in [v, v + l_{i(b),m} - 1].$$

ces_b is the earliest start time and cls_b is the latest start time. A continuous time window of operation i in bid b is described by $[ces_b, cls_b]$. The start time of the operation can be changed within the time window without affecting other operations. It is obvious that we have $ces_b \leq Y_b \leq cls_b$. The subscript of temporal information of CA is the index of bid (b). It is because the domain of local scheduling of CA is a set of bids RB. Using bid as a subscript is equivalent to using operation (as in PA) as a bid contains one and only one operation in CAs. When a bid of CA is accessed by a PA, the subscript of information in the bid is the index of operation and CA (im). The revenue of CA m is $R_m = \sum_{b \in RB_m} p_{i(b)}$. SchST is a set of states related to scheduling process that will be described in Section V.

V. MODIFIED CONTRACT-NET PROTOCOL

In this section, we present a distributed approach based on an MCNP to solve the SCSP. The conventional CNP [13] is based on the scenario in which one manager announces a task to multiple contractors. In SCSP, this implies that only one PA can contract tasks to CAs at any time. We modify the conventional CNP so that multiple managers (PA) can simultaneously select contractors for interrelated tasks. Like the conventional CNP, there are four phases of contracting an operation in MCNP.

- 1) *Announcing*. Each PA selects an operation that is ready for scheduling, i.e., the operation's preceding operations are scheduled. The PA announces the preferred time window of the operation to eligible CAs by sending request-for-bid (RFB) or call-for-bid messages.
- 2) *Bidding*. After receiving RFBs from PAs, the CA allocates resource to the operations in the RFBs and sends bids of the operations to the PAs.
- 3) *Awarding*. The PA then selects the best bid for the operation and sends an award message to the selected CA.
- 4) *Committing*. When a CA receives an award message of its bid from a PA, it considers improving the awarded bids by reducing the violation of the bid's time window. Resource is reserved for the bid and a commitment message is sent to the PA. The PA confirms the bid and starts to announce another operation as described in (1). A PA finishes its scheduling process when all of its operations are assigned to CAs.

Unlike the conventional CNP, each PA conducts the abovementioned process independently and asynchronously. Moreover, CAs do not know which and when the PAs will announce operations. MCNP is schematically shown in Fig. 3. The figure represents scheduling states (circles) and transitions (rectangles) of MCNP. SchSTs are shown in italic. Texts linked to arrows describe the conditions of directed states or transitions. Bolded labels on arrows between PA and CA are the types of message. Transitions correspond to the four phases of MCNP. Details of the phases including the states and transitions will be described in Section V.

A message *msg* is represented as **msgType** (senderID, receiverID, msgContent). *msgType* indicates the type of message *msg*. *senderID* is the index of agent that sends the message. *receiverID* is the index of agent that receives the message. *msgContent* is the content of the message, which is a set of information that is conveyed by the message. Each message has a unique identifier (not shown in the representation) in the agent. The message type and content depends on the problem domain. It is assumed that agents understand the meaning of message types and information contained in message content. That is, there exists a common ontology of interpreting the meaning conveyed in messages. We also assume that the communication channel through which the agents exchange message is reliable. It means that a message sent by an agent will be received by the receiving agent in a finite time. Messages received by an agent are stored in the agent's message list before processing.

A. Detailed Description of MCNP

MA initializes protocol and informs PAs to start announcing operations by sending **StartSch** messages to the PAs. When a PA receives a **StartSch** message, it starts to announce the first operation to CAs. At the beginning of the protocol, CAs wait for RFB messages in state *wait_RFB*.

Announcing Phase: The PA selects an operation of which the precedent operations have been scheduled for announcing. If there are more than one operations that are ready, the operation with the smallest index is selected. The selected operation is called current operation (*current_op*). A request for bid or call for proposal message (**RequestBid**) is sent to CAs, which are capable of performing the operation. In the message, the PA shares time window information $\{es_i, \hat{ls}_i\}$ of the operation with the CAs. The PA computes the earliest start time es_i . \hat{ls}_i is the estimated latest start time of the operation beyond which the due date of the project could be violated. It is given by

$$\hat{ls}_i = \begin{cases} \min_{(i,j) \in E_n} (\hat{ls}_j - \min_{h \in AC_i, k \in AC_j} lt_{ihk} - \hat{t}_i), & i \neq f \\ \hat{ls}_i = D_n - \hat{t}_i, & i = f \end{cases}$$

After sending the **RequestBid** messages, the PA waits for bids from the CAs in the awarding phase. State *wait_reply* then becomes true.

Bidding Phase: A CA collects **RequestBid** messages when state *wait_RFB* is true. The CA stops collecting the messages when a time limit at which a period *time_waitRFB* is passed after receiving the first RFB is reached. When the time limit

of collecting **RequestBid** messages is passed, the state the *wait_RFB* becomes false and the CA starts assigning start times to the operations in the received RFBs. Notice that if a **RequestBid** message arrives to the CA, i.e., stored in a message list, when the CA is performing local scheduling, i.e., *wait_RFB* is false, the **RequestBid** message is not considered in the local scheduling and will be considered when the state *wait_RFB* becomes true later. The order of scheduling operations in the **RequestBid** messages is based on a priority. Higher priority should be given to an operation (i) of which the possibility of violating the latest start time \hat{ls}_i is higher and the price of performing the operation is higher. The priority is, therefore, represented as the ratio of the difference between \hat{ls}_i and capacity-feasible earliest start time ces_i over the price p_i of performing the bid. Formally, the priority value of operation i is defined as $(ls_i - ces_i)/p_i$. An operation with the smallest priority value is scheduled first. This rule tries to schedule operations within the time windows given by PAs in order to increase the chance of being selected by the PAs and, hence, the expected revenue. Moreover, operations with higher price can directly contribute to maximize the revenue. When an operation is selected, the proposed start time is the capacity-feasible earliest start time. After assigning the operation, the CA sends a **ReplyBid** message to the PA. The message contains price, start time, and processing time of the operation. The assignment process continues until all requests are replied. The CA waits for awards (i.e., state *wait_award* is true) from the PAs in the committing phase after all RFBs are replied.

Awarding Phase: When a PA receives the first **ReplyBid** message, it starts to collect other **ReplyBid** messages within a time limit (*time_waitReply*) counted from the time it receives the first message when state *wait_reply* is true. If the PA receives any **ReplyBid** message of scheduled operations, it will reject the bid. When the time limit is passed, the PA stops collecting bids and starts to select the best CA m for performing the current operation. The best bid is the bid that minimizes operating costs, which consists of three terms. The first term is the price of the bid. The second term (BC_{im}) is the earliness and transportation costs with the immediate preceding operations of the current operation. It is defined as $BC_{im} = \sum_{(j,i) \in E_n} (ec_j(Y_{im} - X_j - l_{j,z(j)} - lt_{j,z(j),m}) + lc_{j,z(j),m})$. The third term (FC_{im}) is the minimum transportation cost with the immediate succeeding operations and tardiness cost of the project if the proposed start time of the CA is later than the estimated latest start time \hat{ls}_i . This term is given by

$$FC_{im} = \sum_{(i,g) \in E_n} \min_{k \in AC_g} lc_{imk} + \begin{cases} (Y_{im} - \hat{ls}_i)tc_n, & Y_{im} > \hat{ls}_i \\ 0, & \text{otherwise} \end{cases}$$

This rule of selecting CA reflects the PA's goal of minimizing the operating cost of the project. After selecting a CA, the PA sends an **AwardBid** message to the selected CA and **RejectBid** messages to other CAs. The PA then waits for commitment of the bid (state *wait_confirm* is true) from the selected CA in the committing phase.

Committing Phase: A CA keeps collecting **AwardBid** or **RejectBid** messages when state *wait_award* is true. When the

number of the collected messages is the same as the number of received RFBs, the CA will commit the awarded bids (i.e., *wait_award* becomes false). The CA checks if the start time of an awarded operation could be improved by scheduling closer to the earliest start time and earlier than the latest start time shared by the PA without affecting the scheduled operations in the local schedule of the CA. The improvement is possible if some bids, which were scheduled earlier than the awarded bid in the bidding phase, are not awarded by PAs. Some idle time slots will be available for such improvement. The awarded bids are considered in the order of the proposed start time. The purpose is to increase the possibility of accepting future bids within their time window. It also reduces the possibility of violating the due date of the PAs. Once the start time of the operation is confirmed, the available capacity is updated and a **CommitBid** message is sent to the PA. When all awarded operations are committed, the CA will wait for new requests and state *wait_RFB* becomes true.

When a PA receives a **CommitBid** message of the current operation, it confirms the bid from the selected CA and the start time of the operation. The PA then starts seeking bids for the next operation, which will become the current operation. If all operations are scheduled, the scheduling process of the PA terminates, i.e., state *Terminate* is true.

B. Differences Between MCNP and Conventional CNP

Differences between MCNP and the conventional CNP are highlighted as follows.

MCNP supports multiple PAs to contract operations to multiple CAs while CNP only supports one manager to assign a task. In MCNP, PAs can execute the process of contracting operations simultaneously (the right part of Fig. 3). The scheduling states of a PA do not depend on the states of other PAs. In the conventional CNP, only one PA can contract operations at a time. Another PA can start after the PA finishes selecting PAs for all its operations.

In the committing phase, when a PA rejects a bid of a CA, the CA could improve the bids accepted by other PAs by modifying the start times within the time windows given by the PAs. In the conventional CNP, CAs cannot improve bids as there is only a single operation for bidding at a time.

MCNP allows PAs and CAs to work concurrently and asynchronously. PAs schedule their operations independently from each other. There is no global synchronization among the agents as required in the conventional CNP [13].

In the bidding phase, local scheduling decision of CAs considers competition of resource among PAs' operations. With the knowledge of time window information of the PAs' operations, CAs can make better decisions on resource allocation. This can lead to improvement on the global objective. In the conventional CNP [13], a CA can only submit one bid, which is the most recently received announcement, at a time.

As agents work asynchronously, MCNP needs to guarantee that the agents would not wait indefinitely for receiving messages. This liveness property is fundamental and essential in any distributed algorithms [31]. Lau [32] shows that CAs and PAs will not stay in intermediate states *wait_award* and

wait_reply, respectively. Lau also proves the correctness of MCNP by showing that MCNP can find a feasible solution of SCSP [32].

VI. CENTRALIZED HEURISTIC FOR SCSP

In this section, we propose a CTR to solve SCSP. The purpose is to evaluate the performance of MCNP. From an information sharing perspective, CAs only need to share start times, prices, and processing times of operations, and PAs only need to share the time windows of operations in MCNP. In the CTR, however, every agent (PA and CA) needs to share all of its local information including states and knowledge to the MA, which performs the centralized scheduling. CTR can be viewed as a best-first search heuristic without backtracking.

The heuristic starts scheduling from the beginning operations, which have no preceding operations. The operations are put in a ready-for-scheduling list (EJ). An operation is said to be scheduled if a CA is selected for performing the operation and a start time of the operation is determined. An operation is ready for scheduling if all its preceding operations are scheduled. The heuristic selects an operation in EJ for scheduling. It then updates EJ by adding operations that are ready for scheduling. The selection of operations is based on a look-ahead cost function. The look-ahead cost of an operation (i) function returns a start time st and a CA m for performing the operation such that the look-ahead cost of scheduling the operation at start time st by CA m is minimized. The look-ahead cost of operation i is the sum of operating costs of temporary scheduling the unscheduled operations given that operation i is scheduled at start time st and performed by CA m .

The look-ahead cost of operation i scheduled at st and performed by CA m is calculated as follows. 1) Update operating cost of current schedule of operation i . 2) Select an unscheduled operation (j) in EJ for temporary scheduling. The operation with the minimum slack time is selected. The slack time of an unscheduled operation j is calculated by the latest start time minus the earliest precedence- and capacity-feasible start time among all possible CAs. The latest start time is the time beyond which the project will become tardy. A CA (m) that gives the minimum operating cost of performing operation j is selected. The earliest precedence- and capacity-feasible start time of CA m becomes the temporary start time of operation j . New ready-for-scheduling operations are added to EJ and operation j is removed from EJ. 3) Step 2) is repeated until EJ is empty, i.e., all unscheduled operations are temporarily scheduled.

The look-ahead cost function then searches a start time st and CA m of operation i that minimizes the resulting operating cost by the abovementioned procedure. After scheduling operation i , available capacity of the selected CA is updated. New ready-to-scheduling operations are added in EJ and operation i is removed from EJ. The heuristic stops when all operations are scheduled.

VII. COMPUTATIONAL STUDY

This section presents a computation study that evaluates performances of MCNP and CTR. The computation consists of two parts. The first part evaluates MCNP by comparing

TABLE I
FACTORS OF EXPERIMENT

Factor	Name	Description	Level 1	Level 2
PS	Project structure	Number of tasks and precedence relationship	Type I	Type II
PN	Number of PA	Number of project agents	4	8
DD	Due date factor	Tightness of completing the project	10%	20%
CL	Capacity level	Number of CAs that have 2 units of capacity	2	3
AC	Alternative CA	Average number of contractor agents of each operation	1–2	2–3

with CTR and the conventional CNP in terms of the global performance (i.e., total operating cost, TC). The second part aims at evaluating the reactive performance of MCNP and CTR under different types of unexpected disturbance. In both parts of the experimental study, the algorithms are implemented by C++ and are executed on a single PC (Pentium III 450 MHz, 512 MB RAM). It is due to the fact that control of the experiments and collection of data from the computer in centralized implementation are easier than that in distributed environment. Moreover, we do not have sufficient computing resources for performing the experiments in a truly distributed computer network.

A. Experiments—Part I

In order to evaluate the advantage of MCNP over conventional CNP, we implement the conventional CNP that allows PAs to announce their operations sequentially. The implementation of the conventional CNP is described as follows. The MA informs the PAs for scheduling in ascending order of the PAs' indexes. Only one PA can schedule one of its operations at a time. The **RequestBid** message only contains the earliest start time (precedence-feasible) of the operation. Each of the CAs replies the earliest capacity-feasible start time of the operation in a **ReplyBid** message. The PA then selects a CA that minimizes the operating cost of performing the operation and sends an award of bid to the selected CA. After the PA has found a feasible schedule of its project, another PA is informed by the MA for scheduling. The abovementioned process continues until all PAs have scheduled their projects. In order to have a fair comparison between CNP and MCNP, we assume that messages **RequestBid** and **ReplyBid** messages are received within the time limits of CAs and PAs, respectively.

Table I shows the factors studied in the experiments. Two types of project structures (PS) are considered. Type I is less complicated than Type II. The average number of operations in Type I and Type II are 5 and 12, respectively. Three instances of each type are randomly generated. Two sets of PAs that belong to the same type are considered in experiments. The number of PAs (PN) is either 4 or 8. Due dates of projects are obtained by multiplying the due date factor (DD) (110% or 120%) on the expected lead time of the project. Capacity level (CL) indicates the number of CAs that have 2 units of capacity. AC indicates the average number of alternative CA that is capable of performing an operation. There are two levels of AC. The low and high levels indicate that operations have one

TABLE II
CASES OF THE EXPERIMENTAL STUDY

Case	PS	PN	DD	AC	CL
1	1	2	1	1	1
2	1	2	1	2	1
3	1	2	2	1	2
4	1	2	1	1	2
5	1	2	2	1	1
6	1	2	2	2	1
7	1	2	1	2	2
8	1	2	2	2	2
9	2	1	1	1	1
10	2	1	1	2	1
11	2	1	2	1	2
12	2	1	1	1	2
13	2	1	2	1	1
14	2	1	2	2	1
15	2	1	1	2	2
16	2	1	2	2	2

to two and two to three alternative CAs, respectively. Different combinations of the levels of these factors constitute different cases in the experiments. The cases are shown in Table II.

Fig. 4 shows the performances of CTR, MCNP, and CNP on TC of supply chain under different cases (see Table II). It is clear that CTR outperforms MCNP and CNP in most of the cases. The performance gap is due to the amount of information used in scheduling. In CTR, before selecting an operation for assignment, a strong look-ahead heuristic value is computed for each schedulable operation. The value accounts for the effect of assigning an operation on other potential operations as the assignment would occupy some capacity that could have been used by other unassigned operations. Therefore, the heuristic value gives a good indication on how well the consequence (i.e., total supply chain cost) is when a schedulable operation is assigned. Such global information is very helpful to optimize the global objective. In MCNP and CNP, however, agents have no information to guide them to improve the global objective. They only maximize their own utilities when receiving bid information from other agents. For example, a PA selects a bid from CA so that the cost of assigning the current operation is minimized. However, it does not, and cannot, consider the impact on its future assignment or even other PA in reality. We also observe that MCNP outperforms CNP in all cases.

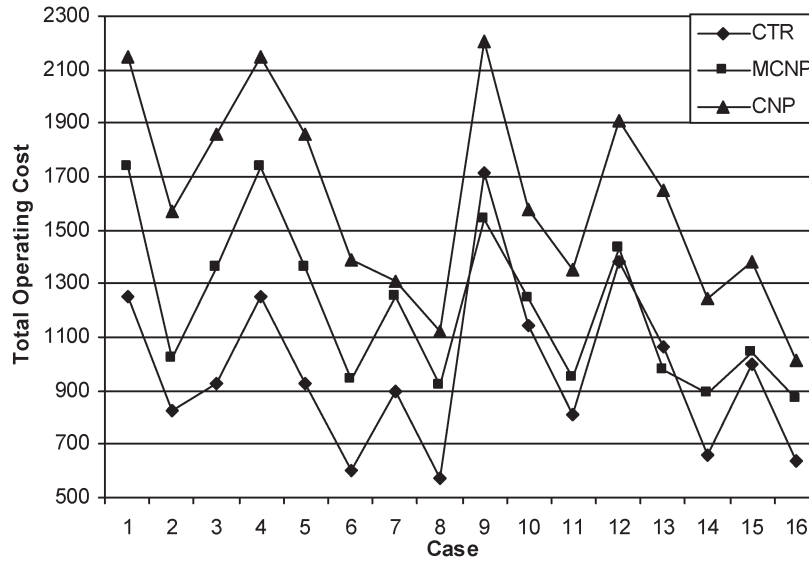


Fig. 4. Experimental results on global performance.

TABLE III
COMPUTATIONAL TIME (SECOND) OF CTR, CNP, AND MCNP

Case	CTR	CNP	MCNP	Case	CTR	CNP	MCNP
1	30.2	0.23	0.08	9	77.21	0.32	0.15
2	48.51	0.3	0.14	10	170.54	0.38	0.22
3	36.46	0.23	0.07	11	77.84	0.31	0.17
4	29.71	0.23	0.09	12	73.98	0.32	0.15
5	36.45	0.22	0.08	13	95.45	0.31	0.15
6	62.73	0.29	0.15	14	282.25	0.39	0.23
7	50.11	0.28	0.15	15	132.15	0.39	0.23
8	59	0.28	0.141	16	248.96	0.37	0.22

It is because, in MCNP, CAs can consider several bids of PAs at a time. The CA can balance the operating costs of PAs by minimizing violation of the time window information shared by the PAs when constructing its local schedule.

When DD is longer, the operating cost is lower as tardiness of projects is smaller (e.g., cases 10 and 14). The TC is higher when AC is smaller. More alternative CAs allows PAs to explore less expensive way to perform operations (e.g., cases 1 and 2). Higher CL leads to lower operating cost (e.g., cases 14 and 16). The difference of TC between high and low CLs is, however, not very significant especially in cases 1–8. Moreover, we observe that, in cases 9–16, MCNP performs very close to CTR (see Fig. 4). When the complexity of the project is higher, CTR behaves more heuristically as the problem becomes more combinatorial with more interrelated operations.

In addition to the performance of total operating cost, we are also interested in the computational efficiency of CTR, CNP, and MCNP. Table III shows the computational time in CPU second taken by CTR, CNP, and MCNP to solve each test case in a single computer. It is obvious that both CNP and MCNP are much more efficient than CTR. When alternative CAs are more (e.g., cases 1, 6–8, 9, and 14–15), CTR, CNP, and MCNP take more time to find solutions. It is obvious that MCNP is much more efficient than CTR and is slightly faster than CNP. In solving larger problems of cases 9–16, CTR takes

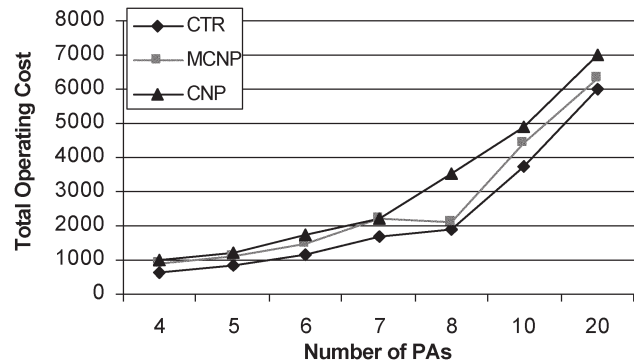


Fig. 5. Performance under large problems.

TABLE IV
CPU TIME (SECOND) UNDER LARGE PROBLEMS

Number of PAs	CTR	MCNP	CNP
4	162.96	0.12	0.34
5	503.53	0.24	0.40
6	800.00	0.28	0.46
7	2272.50	0.47	0.56
8	2596.06	0.50	0.77
10	6363.66	0.51	1.06
20	3014.10	1.07	1.94

considerably more time than solving smaller problems of cases 1–8. The increase in computational time in MCNP for solving larger problems is negligible.

In order to evaluate the performance of MCNP under large problems, we increase the number of PAs in case 16. When the number of PAs is between 4 and 7, CL is 2 as described in Table I. When the number of PAs is 8 and 10, there are six and two CAs with 2 and 1 units of capacity, respectively. When the number of PAs is 20, there are four and three CAs with 3 and 4 units of capacity. Fig. 5 shows the performance of CTR, CNP, and MCNP on total operating cost when problem size becomes

TABLE V
SETTING OF PART II EXPERIMENT

Disturbance	Example	Severity Parameters	Levels		
			1	2	3
New projects arrival	New customer order	Number of new PAs = 2, Due date factor =	1.2 (1)	1.4 (2)	1.6 (3)
Resource breakdown	Machine breakdown	Number of CAs = 2, Downtime =	8 (4)	10 (5)	12 (6)
Increase in processing times of operations	Rework due to quality problem of operation's outputs	Number of disturbing operations = 3, Length of increased processing time =	4 (7)	5 (8)	6 (9)

larger. The computational times are shown in Table IV. It is expected that CTR outperforms CNP and MCNP in terms of total operating cost. MCNP performs better than CNP. On the other hand, from the perspective of computational efficiency, MCNP and CNP are superior to CTR. Computational time of CTR grows exponentially with the problem size. However, the increase in computational time of MCNP and CNP is insignificant. We also observe that MCNP can still improve CNP under large problems.

B. Experiments—Part II

Due to the dynamic nature of supply chains, unexpected disturbances always occur after a global schedule is established by the agents. In this experiment, CTR and MCNP are evaluated against three types of unexpected disturbances: arrival of new project, resource failure, and increase of operation's processing time. The initial global (local) schedule is called global (local) preschedule. An agent that detects disturbances is called disturbing agent. On the other hand, an agent that is affected by the change of schedule of the disturbing agent is called disturbed agent. When a disturbance occurs, some constraints (precedence or capacity) of the disturbing agent are violated. The disturbing agent has to reschedule its local schedule in order to restore the feasibility of its schedule. Such local rescheduling may affect the local schedules of other agents. For example, when a new project is launched, the new PA needs to select CAs to perform its operations. This may affect the operations that have been scheduled in the CAs as the capacities that have been assigned to the operations would be assigned to the new operations. The objectives of the rescheduling problem are to minimize the total operating cost of the new schedule, deviation of start times, and change of contractual relationships of the new schedule from the preschedule. The schedule deviation is the sum of absolute differences between start times of operations in the new schedule and the preschedule. The contractual relationship is changed if a PA decommits from a bid of a CA confirmed in the preschedule and selects another CA.

The rescheduling process in MCNP is described as follows. When a disturbing agent detects a disturbance, the agent reschedules its operations and sends the modified start times to the affected (i.e., disturbed) agents. The disturbed agents perform rescheduling if their preschedules become infeasible after receiving the changes from the disturbing agents. The disturbed agents then send the new start times of their operations to other

affected agents. The agents propagate the changes of schedules until no more changes are detected. During the rescheduling process, PAs cannot change the selected CAs.

When a new PA is registered by MA, the PA announces their operations to CAs. The CAs reply a start time that minimizes the schedule deviation of committed operations. The PA selects a bid that minimizes its operating cost. After the selected CA commits the bid, it initializes the rescheduling process as described in the previous paragraph where the CA becomes the disturbing agent.

The centralized approach to rescheduling is simply a complete rescheduling of the global schedule. The disturbing agents notify the MA about the detected disturbances. The MA then collects the latest schedule information from other agents. A "reduced" SCSP is formulated with less to-be-schedule operations and updated parameters. The CTR is then applied to solve the "reduced" SCSP by the MA.

The disturbances are tested on case 16 (see Table II). The preschedule is generated by using CTR. Disturbances are introduced at a time instance called disturbance time (DT). In this experiment, we set DT to 20. All operations that are scheduled to be finished before DT are not affected by rescheduling. We also evaluate the algorithms by testing different levels of severity of the disturbances. Severity of a disturbance means the "size" of the disturbance. If the severity of the disturbance is higher, impacts of the disturbance on the supply chain would be larger. The severity of disturbance is specified by parameters. Table VI shows the setting of severity parameters of the three disturbances in our experiments. We test two new projects with due dates equal to DT plus expected project lead time multiplied by the DD. Two CAs are selected for simulating resource breakdown. Duration of the breakdown is indicated by downtime. Three operations are selected to have increased processing time. Due date factor, downtime, and length of increased processing time are varied at three levels (Table V). The number in the bracket at each level is the case number.

Fig. 6 shows the total operating cost of the new schedule. Schedule deviation after rescheduling is shown in Fig. 7. It is obvious that CTR generally performs better than MCNP in terms of total operating cost. Schedule deviation of MCNP is, however, smaller than that in CTR in most of the cases. When the due date of new PAs is less tight (i.e., DD is larger), both the total operating cost and the schedule deviation get smaller (cases 1–3). When facing resource breakdown, the total operating cost and the schedule deviation increase with the downtime (cases 4–6). Performance on total operating cost of

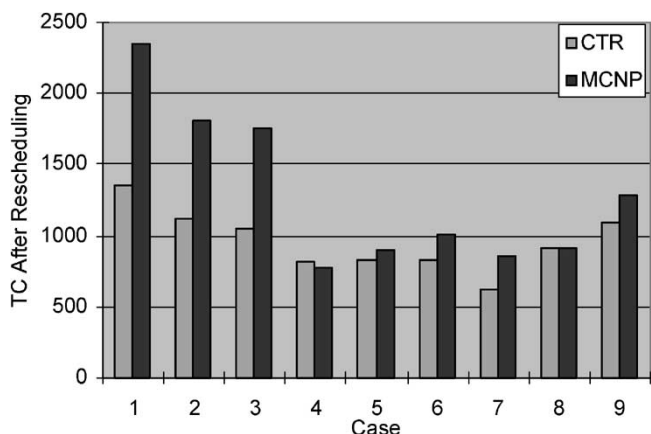


Fig. 6. Total operating cost.

CTR is slightly better than that of MCNP under disturbance of increased operation’s processing time (cases 7–9). On the other hand, MCNP outperforms CTR in terms of schedule deviation. We observe that the performance gap on the total operating cost between CTR and MCNP is largest under the arrival of new PAs. In other cases, the performance on total operating cost of MCNP is quite close to that of CTR. It is because the number of constraint violations under new projects arrival is higher than that of resource breakdown and increased processing time of operations. More search efforts are needed to reduce the total operating cost. On the other hand, when the number of constraint violations is small, simple propagation of constraints among agents in MCNP may produce a fairly good result as the increase of tardiness due to the propagation is small. Therefore, the performance of CTR under new projects arrival is better than that under the other two types of disturbances. Moreover, the average number of changes of contractual relationship between CAs and PAs in CTR is about 14 while MCNP does not require any such changes. The change of contractual relationship implies the costs of withdrawing the contract and establishing the new one.

C. Comparison of CTR, CNP, and MCNP

Table VI summaries a comparison between the centralized (CTR) and decentralized (CNP, MCNP) approaches. The first three criteria are related to the three characteristics of the supply chains studied in this paper. The criteria concern the requirements of distributed problem solving by a set of independent and autonomous enterprises. CNP and MCNP are certainly better than CTR. Such advantages of CNP and MCNP inherit from the agent-based modeling features as discussed in Section IV.

According to the results from part I of the experimental study (Section VII), the performance of CTR on total operating cost is better than that of CNP and MCNP as there is complete information sharing between agents/enterprises in CTR. However, the computational efficiency of CNP and MCNP is much better than CTR. The numbers in Table VI represent the ranking of the performances. From the experimental results (Table III), the computational times of CNP and MCNP are less than 1 s when they are performed on a single PC while CTR takes several

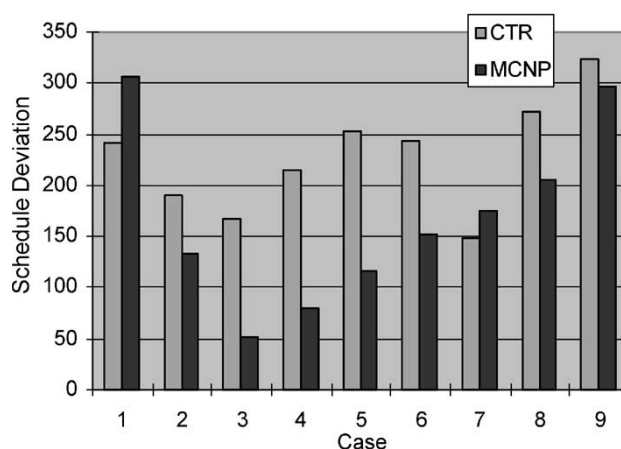


Fig. 7. Schedule deviation.

minutes to complete the computation. Moreover, when problem size gets larger, computation efficiency of CNP and MCNP is not much affected. However, the computational time of CTR grows exponentially with the problem size.

In terms of concurrency, both CNP and MCNP allow agents to make decisions concurrently. However, concurrency is not relevant to CTR as only one agent (MA) makes decisions. In MCNP, agents can perform asynchronously. In CNP, agents are synchronized by MA. In terms of modularity, the agent-based approach is certainly better than the centralized approach [33]. Agents can dynamically join or leave the supply chain with the aid of MA. Decision rules of an agent can be changed without affecting other agents. In CTR, MA needs to maintain changes of knowledge or states of individual agents as it is the only decision maker in the supply chain.

According to part II of the experimental study, we observe that CTR performs better than MCNP in terms of the total operating cost under expected disturbances. In terms of schedule stability, MCNP performs better than CTR. Less operations need to be changed, and no changes of contractual relationship between PAs and CAs are required in MCNP. This property is very desirable when changes on the established schedule are very costly in supply chains. Moreover, as MCNP is much faster than CTR, MCNP can offer a very quick response to unexpected disturbances.

VIII. CONCLUSION

This research has discussed an agent-based approach to solving distributed supply chain scheduling problems (SCSPs). The contributions of this paper can be observed from several aspects. First, we propose a mathematical model of an SCSP that includes the following features. 1) A multiproject environment in which projects are held by a set of independent and autonomous enterprises. 2) A set of contractors that can perform different operations of the projects and every operation can be performed by a set of alternative contractors. 3) Contractors have multiple capacities and different operations have different requirements on capacity, processing times, and costs. 4) Lead times and costs of transportation between two contractors are different. 5) Decisions of the model are selection

TABLE VI
SUMMARY OF COMPARISON BETWEEN CTR, CNP, AND MCNP

Criteria	CTR	CNP	MCNP
Information sharing	Agents need to share all information	Agents share start times of operations	PAs shares time windows of operations
Multiobjectives	Only global objective		Yes
Independency and autonomy of decision making	No (MA makes all decisions for PAs and CAs)	PAs and CAs can make their own decisions	
Total operating cost	1	3	2
Computational efficiency	3	2	1
Concurrency and asynchrony	No	Synchronous	Asynchronous
Modularity	Low		Yes
Total operating cost under disturbances	Good		Not good
Stability of schedule	Low		High

of contractors (or configuration of projects) and scheduling of operations. Each configured project represents a supply chain with a network structure. 6) The objective of the model is to minimize the sum of tardiness costs of the projects, costs of selected contractors, transportation costs, and slacks between operations to achieve just-in-time philosophy. To the best of our knowledge, these features have not been investigated in the literature of project scheduling under a supply chain context. Second, an agent-based model has been proposed to address the distributed project scheduling problem. Information privacy and distributed decision making of agents are respected in the model. Third, we develop a modified contract-net protocol (MCNP) to solve SCSP in a distributed manner based on the agent-based model. MCNP allows agents to work concurrently and asynchronously. A project agent needs to share time window information of an operation with contractor agents when the project agent seeks a contractor agent for performing the operation. Finally, we compare the proposed MCNP with the conventional CNP and a CTR through computational experiments. CNP and CTR represent minimal and full information sharing between PAs and CAs. We show that MCNP outperforms CNP in terms of total operating cost and computational efficiency. It also performs closely to CTR when project complexity is high in terms of the performance on total operating cost of the supply chain. Moreover, MCNP is much more efficient than CTR. The result is encouraging as the proposed distributed approach with appropriate level of information sharing can achieve comparable performance to the centralized approach, which requires full information sharing, when problem size is large. We also evaluate MCNP with CTR under unexpected disturbances in the supply chain. We show that although CTR is better than MCNP in terms of total operating cost, MCNP yields good schedule stability after rescheduling.

In theory, the centralized approach is generally superior in terms of global performance over the distributed approach. The main reason is that complete information sharing in the centralized approach allows better search for optimal or near-optimal solution. The requirement of complete information sharing in the centralized approach is, however, virtually impractical in a real-life supply chain that comprises a set of independent enterprises. In contrast, we discuss some attractive characteristics (e.g., concurrency, modularity, and stability) of the

distributed approach, which are suitable for solving scheduling problem in supply chains. A compromise must be made under the circumstance where only limited information sharing is possible for practical reasons. This paper reports our attempt on investigating what kind of information should be shared and how the shared information aids the distributed scheduling process. Our research interest is now focused on the impact of the increased information sharing among the enterprise agents on global performance of the supply chain scheduling. Study on information sharing has been a growing interest in supply chain management [34]. However, issues like what kind of and how information should be shared among agents to improve the performance of the distributed approach of the supply chain scheduling have not been investigated in literature. The agent-based information modeling and the distributed algorithm proposed in this paper will be enhanced to deal with this question in our next research work, which will be reported separately.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for critical but constructive comments.

REFERENCES

- [1] S. Bradley and R. Nolan, *Sense and Respond: Capturing Value in the Network Era*. Boston, MA: Harvard Bus. Sch. Press, 1998.
- [2] M. A. Razaque and C. C. Sheng, "Outsourcing of logistics functions: A literature survey," *Int. J. Phys. Distrib. Logist. Manage.*, vol. 28, no. 2, pp. 89–107, 1998.
- [3] N. Kakabadse and A. Kakabadse, "Critical review—Outsourcing: A paradigm shift," *J. Manage. Dev.*, vol. 19, no. 8, pp. 670–728, 2000.
- [4] S. J. Mason, M. H. Cole, B. T. Ulrey, and L. Yan, "Improving electronics manufacturing supply chain agility through outsourcing," *Int. J. Phys. Distrib. Logist. Manage.*, vol. 32, no. 7, pp. 610–620, 2002.
- [5] M. A. Cohen and H. L. Lee, "Strategic analysis of integrated production-distribution systems: Models and methods," *Oper. Res.*, vol. 36, no. 2, pp. 216–228, 1988.
- [6] K. Ertogral, S. D. Wu, and L. I. Burke, "Coordination production and transportation scheduling in the supply chain," Dept. IMSE, Lehigh Univ., Bethlehem, PA, Tech. Rep. #98T-010, 1998.
- [7] E. Sabri and B. M. Beamon, "A multi-objective approach to simultaneous strategic and operational planning in supply chain design," *Omega*, vol. 28, no. 5, pp. 581–598, 2000.
- [8] V. Jayaraman and H. Pirkul, "Planning and coordination of production and distribution facilities for multiple commodities," *Eur. J. Oper. Res.*, vol. 133, no. 2, pp. 394–408, 2001.
- [9] Y. H. Lee, C. S. Jeong, and C. Moon, "Advanced planning and scheduling with outsourcing in manufacturing supply chain," *Comput. Ind. Eng.*, vol. 43, no. 1–2, pp. 351–374, 2002.

- [10] S. S. Erenguc, N. C. Simpson, and A. J. Vakharia, "Integrated production/distribution planning in supply chains: An invited review," *Eur. J. Oper. Res.*, vol. 115, no. 2, pp. 219–236, 1999.
- [11] R. Gaonkar and N. Viswanadham, "Collaboration and information sharing in global contract manufacturing networks," *IEEE/ASME Trans. Mechatronics*, vol. 6, no. 4, pp. 366–376, Dec. 2001.
- [12] E. H. Durfee, "Distributed problem solving and planning," in *Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA: MIT Press, 1999, ch. 3.
- [13] R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," *IEEE Trans. Comput.*, vol. COM-29, no. 12, pp. 1104–1113, Dec. 1980.
- [14] J. Vancza and A. Markus, "An agent model for incentive-based production scheduling," *Comput. Ind.*, vol. 43, no. 2, pp. 173–187, 2000.
- [15] R. Macchiaroli and S. Riemma, "A negotiation scheme for autonomous agents in job shop scheduling," *Int. J. Comput. Integr. Manuf.*, vol. 15, no. 3, pp. 222–232, 2002.
- [16] P. Sousa and C. Ramos, "A distributed architecture and negotiation protocol for scheduling in manufacturing systems," *Comput. Ind.*, vol. 38, no. 2, pp. 103–113, 1999.
- [17] D. Ouelhadj, C. Hanachi, B. Bouzouia, A. Moualek, and A. Farhi, "A multi-contract net protocol for dynamic scheduling in flexible manufacturing systems (FMS)," in *Proc. IEEE Int. Conf. Robotics and Automation*, Detroit, MI, 1999, pp. 1114–1119.
- [18] A. D. Baker, "Metaphor or reality: A case study where agents bid with actual costs to schedule a factory," in *Market-Based Control: A Paradigm for Distributed Resource Allocation*, S. H. Clearwater, Ed. River Edge, NJ: World Scientific, 1996, ch. 8, p. 184.
- [19] A. Saad, K. Kawamura, and G. Biswas, "Performance evaluation of contract net-based hierarchical scheduling for flexible manufacturing systems," *Intell. Autom. Soft Comput.*, vol. 3, no. 3, pp. 229–248, 1997.
- [20] D. Xue, J. Sun, and D. H. Norrie, "An intelligent optimal production scheduling approach using constraint-based search and agent-based collaboration," *Comput. Ind.*, vol. 46, no. 2, pp. 209–231, 2001.
- [21] K. Fischer, B. Chaib-draa, J. P. Muller, M. Pischel, and C. Gerber, "A simulation approach based on negotiation and cooperation between agents a case study," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 29, no. 4, pp. 531–545, Dec. 1999.
- [22] D. Naso and B. Turchiano, "A coordination strategy for distributed multi-agent manufacturing systems," *Int. J. Prod. Res.*, vol. 42, no. 12, pp. 2497–2520, 2004.
- [23] P. Sousa, C. Ramos, and J. Neves, "The Fabricare system: A multi-agent-based scheduling prototype," *Prod. Plan. Control*, vol. 15, no. 2, pp. 156–165, 2004.
- [24] X. Q. Zhang, "Sophisticated negotiation in multi-agent systems," Ph.D. thesis, Dept. Comput. Sci., Univ. Massachusetts, Amherst, 2002.
- [25] J. Collins, "Solving combinatorial auctions with temporal constraints in economic agents," Ph.D. thesis, Comput. Sci. Dept., Univ. Minnesota, Minneapolis, 2002.
- [26] J. Reis, N. Mamede, and H. O'Neill, "Locally perceiving hard global constraints in multi-agent scheduling," *J. Intell. Manuf.*, vol. 12, no. 3, pp. 223–236, 2001.
- [27] Y. H. Lee, S. R. T. Kumara, and K. Chatterjee, "Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism," *J. Intell. Manuf.*, vol. 14, no. 5, pp. 471–484, 2003.
- [28] T. Wagner, V. Guralnik, and J. Phelps, "TAEMS agents: Enabling dynamic distributed supply chain management," *Electron. Commer. Res. Appl.*, vol. 2, no. 2, pp. 114–132, 2003.
- [29] N. R. Jennings and M. Wooldridge, "Applications of intelligent agents," in *Agent Technology: Foundations, Applications, and Markets*. Berlin, Germany: Springer-Verlag, 1998, pp. 1–27.
- [30] K. Sycara, "Multi-agent infrastructure, agent discovery, middle agents for web services and interoperation," in *Multi-Agents Systems and Applications*, M. Luck *et al.*, Eds. New York: Springer-Verlag, 2001, pp. 17–49. ACAI 2001, LNAI 2086.
- [31] G. Tel, *Introduction to Distributed Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [32] J. S. K. Lau, "Impacts of sharing production information on supply chain dynamics," Ph.D. dissertation, Dept. Ind. Mfg. Sys. Eng, Univ. Hong Kong, Hong Kong, 2004.
- [33] Y. Yuan, T. P. Liang, and J. J. Zhang, "Using agent technology to support supply chain management: Potentials and challenges," Working Paper, Hamilton, ON, Canada: School of Business, McMaster University, 2001.
- [34] G. Q. Huang, J. S. K. Lau, and K. L. Mak, "The impacts of sharing production information on supply chain dynamics: A review of the literature," *Int. J. Prod. Res.*, vol. 41, no. 7, pp. 1483–1517, 2003.

Jason S. K. Lau received the B.Eng. and Ph.D. degrees in industrial engineering from the University of Hong Kong, Hong Kong, in 2000 and 2004, respectively.

Since 2004, he has been a Research Assistant in the Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong. His research interests are agent-based manufacturing, production planning and control, meta-heuristics optimization techniques, and enterprise resources planning (ERP). He has published five journal articles and five proceedings.

Dr. Lau is a member of the Institution of Electrical Engineers (IEE) and committee member of the Specialized Section in Manufacturing and Systems Engineering, IEE (HK).

George Q. Huang received the B.Eng. and Ph.D. degrees in mechanical engineering from Southeast University, Nanjing, Jiangsu, China, and Cardiff University, Cardiff, U.K., in 1983 and 1991, respectively.

Dr. Huang joined the Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Hong Kong, in 1997 after a few years of researching and teaching at British universities. His main research interests include agent-based collaborative environments for complex engineering and business systems and computational game theory. He has published extensively in these topics, including over 60 journal papers, two monographs entitled *Cooperating Expert Systems in Mechanical Design and Internet Applications in Product Design and Manufacturing* respectively, and an edited reference book entitled *Design for X: Concurrent Engineering Imperatives*.

Dr. Huang is a Chartered Engineer and a member of the Institution of Electrical Engineers (IEE), HKIE, IIE, and ASME. He is the Editor-in-Chief of the *International Journal of Mass Customization*.

K. L. Mak received the M.Sc. degree in manufacturing engineering and the Ph.D. degree in systems engineering from the University of Salford, Salford, U.K.

He is a Professor in the Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong. He has accumulated substantial experience in industry by working in several U.K. engineering enterprises including the Pilkington Brothers Limited, the T.S. Harrison and Sons Limited, as well as in some enterprises in Hong Kong. Indeed, he has wide exposure to industry and is well connected with enterprises in the local and overseas communities. His current research interest focuses mainly on production and operations management, and manufacturing systems design and control, and has published extensively in these areas.

Prof. Mak is a Chartered Engineer, a fellow of HKIE, and a Member of IMechE. He also serves on the editorial boards of a number of journals in engineering.

L. Liang received the M.Sc. degree in computer science from Hefei Polytechnic University, Hefei City, Anhui Province, China, in 1987 and the Ph.D. degree in systems engineering from the Southeast University of China, Nanjing, China, in 1991.

He is a Professor in Business School, The University of Science and Technology of China. His main research interests include multiple objective optimization, data envelopment analysis, and supply chain theory and practice. He has served many local enterprises in management diagnosis and consultation, and published more than 80 journal papers, and two monographs entitled, *Evaluation Theory and Application* and *Modern Management Methods and Its Practice*.

Prof. Liang also serves on the editor boards of the *International Journal of Mass Customization* and *Asian-Pacific Business Reviews*.