Applied Soft Computing 48 (2016) 703–715

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

An agent-based fuzzy constraint-directed negotiation model for solving supply chain planning and scheduling problems

Chia-Yu Hsu [a,c], Bo-Ruei Kao [b], Van Lam Ho [b], Lin Li [b], K. Robert Lai [b,c,*]

[a] Department of Information Management, Yuan Ze University, Chungli, Taiwan
[b] Department of Computer Science & Engineering, Yuan Ze University, Chungli, Taiwan
[c] Innovation Center for Big Data and Digital Convergence, Yuan Ze University, Chungli, Taoyuan 32003, Taiwan

ARTICLE INFO

Article history:
Received 18 July 2015
Received in revised form 14 July 2016
Accepted 18 July 2016
Available online 1 August 2016

Keywords:
Autonomous agents
Fuzzy constraints
Negotiation
Multi-agent system
Supply chain

ABSTRACT

Supply chain planning and scheduling problems require manufacturers (project managers) to determine product configurations, select suppliers (contractors) for task allocation, and schedule project tasks while considering various production constraints among manufacturers and suppliers. Most relevant studies have focused on finding an optimal solution based on complete information provided by each enterprise in a supply chain. However, the practical implementation of complete information sharing is difficult, if not impossible, because of the fully distributed nature of the supply chain process. This work proposes an agent-based fuzzy constraint-directed negotiation (AFCN) model to solve problems associated with supply chain planning and scheduling, which are modeled as a set of fuzzy constraint satisfaction problems (FCSPs) that are interlinked *via* inter-agent constraints. To accommodate the perspectives and interests of each enterprise in a supply chain, conflicts among FCSPs are resolved using the AFCN protocol through the iterative exchange of offers/counter-offers with limited information sharing and without privacy breaches. A proposed offer/counter-offer represents not only a set of acceptable solutions and preferences for an operational task but also the possibility of conflict in this area. For each FCSP, the incremental process of offer/counter-offer evaluation eliminates redundant and infeasible solutions. The sharing of limited non-strategic sensitive information among agents enables them to elucidate their opponents' intentions through iterative negotiations, such that the agents can reach an agreement while ensuring that the solution to a project planning and scheduling problem is satisfactory. The AFCN model is also sufficiently flexible to incorporate different negotiation strategies such as competitive, win-win, and collaborative strategies, for various production environments. Herein, a numerical study was conducted to examine the practical viability and effectiveness of the proposed AFCN model. The experimental results show that the proposed AFCN model not only can generate a schedule that is comparable to a near-optimal solution but also is time-efficient. This indicates that the proposed AFCN is a practical and effective method for solving supply chain planning and scheduling problems in fully distributed environments.

1. Introduction

Faced with global economic volatility and intense competition, modern enterprises must respond rapidly to market fluctuations. In a supply chain, the various business entities, involved including suppliers, manufacturers, distributors and retailers, are partners, whose integrated actions produce products and services for customers [1]. In a customer-driven supply chain, an enterprise responds to rapidly customer demand, and diverse products are produced to satisfy that demand. In addition, most supply chains include various manufacturers that concurrently produce multiple products. To ensure good customer service and low production costs, enterprises must select partners and coordinate with them during process planning and scheduling in a cost-effective manner, given various constraints such as those related to processing time and due date [2–5].

The manufacturing process that is required to meet customer demand comprises various processes for the production of raw materials or the assembly of parts into finished products [3]. Planning involves supplier selection and product configuration decisions regarding the sequence of manufacturing operations

* Correspondence to: Department of Computer Science & Engineering, Yuan Ze University, 135 Yuan Tung Road, Chungli 32003, Taiwan.
E-mail address: krlai@cs.yzu.edu.tw (K.R. Lai).

http://dx.doi.org/10.1016/j.asoc.2016.07.030
1568-4946/

based on the precedence relations among those operations. Scheduling involves the allocation of various jobs to different facilities to fill a customer's order in a timely manner.

This work aims to solve a supply chain planning and scheduling problem (SCPSP) involving multiple projects undertaken by various project managers and contractors who are distributed and autonomous agents in a supply chain network. A project comprises a set of tasks or operations for a particular product under precedence constraints. Project managers in make-to-order enterprises coordinate the production sequences of these tasks. Each task is selected and performed by a firm or contractor from among a set of candidate partners. Each contractor has its own processing time, resources, and operating costs. The enterprises involved in a supply chain have diverse intentions. The objective of the project manager is to minimize the operating costs associated with a project by defining an optimal sequence of tasks given the precedence of the task operations. The goal of the contractors is to maximize their profits from the projects in which they participate, given their limited capacity to complete contracted tasks [6]. Furthermore, each enterprise in the supply chain must make decisions based on limited information about production operations obtained from either project managers or contractors, such as the processing costs and resource capacities of the contractors.

The two main classes of methods used to solve the SCPSP are centralized and distributed methods. In a centralized method, a single coordinator performs process planning and scheduling in a manner that integrates all operational information from all of the enterprises in the supply chain. Various centralized methods have been used to find an optimal or near-optimal solution to the SCPSP under conditions of full information sharing, including linear programming [4,7,8], fuzzy linear programming [9], fuzzy mixed integer-linear programming [10], genetic algorithm-based methods [11–14], and other meta-heuristic methods [15–20]. However, centralized approaches are difficult to apply to real industry problems because they typically require sensitive strategic information from business partners. Moreover, the centralized method lacks flexibility and is unable to adapt when the strategic information from each enterprise in the supply chain is updated.

Agent-based methods, which are characterized by distributed computation and information processing, are regarded as valid alternatives for SCPSP modeling [21–26]. A multiple-agent system comprises a set of inter-related agents in a distributed and heterogeneous environment [27], and such systems have recently been applied to solve various problems such as those involving cloud resource management [28–30], collaborative design [31], health care monitoring [32], manufacturing scheduling [33–35], transportation and logistics [36,37], and virtual enterprise negotiations [38,39]. However, the lack of global information for the coordination of production operations is a challenge in the agent-based approach. In this approach, agents make their decisions independently of each other and optimize their local objectives without considering the constraints of other agents or the global performance [6].

To ensure global performance with limited interaction among agents, a mediator or a third-party agent between the project manager and contractor can be introduced to coordinate task allocation [40]. This mediator facilitates the negotiation processes by assisting the negotiating parties in understanding their own needs, suggesting possible agreements, and supporting the parties in reaching a final agreement. However, in such situations, agents may be required to share sensitive strategic information that would otherwise not be revealed to opponents or even to a third-party mediator.

As alternatives to involving a third-party mediator, the contract net protocol (CNP) and the market-based iterative auction (MIA) protocol are used to support information privacy and enhance coordination in a fully distributed environment [6,15,41]. The CNP is used to facilitate negotiation for the distribution of subtasks among various agents [42–45]. In the CNP, a project manager agent (PA) sends a request to all qualified contractor agents (CAs) for bids to complete a task. After receiving bids from the contractors, the PA assigns the task to the best contractor based on his/her objective. However, negotiations using the CNP allow a contract with a CA to be established for only one project at a time. With limited interaction and limited information sharing, agents must make their decisions independently of each other, leading to the local optimization of the entire supply chain network. Lau et al. [6] proposed the modified contract net protocol (MCNP) to enable multiple project managers to simultaneously select contractors to undertake their operations. The MCNP also allows project managers to share the start-time window information for an operation with contractors to improve global performance. However, the CNP and MCNP support only limited interactions with single-shot negotiations and thus yield less-than-satisfactory results.

The MIA protocol provides a more sophisticated negotiation mechanism [46–50] by adopting a market-based approach for auctioning resources to contractors and enabling the contractor to receive maximum revenue. The MIA protocol also supports iterative bargaining between PAs and CAs to yield a task-performing price that reflects the global competition for resources. During the iterative bidding process, the project managers adjust their bids according to the prices quoted by the contractors, but they lack sufficient information regarding where any contention should be resolved. Hence, the bidding process may oscillate and not converge efficiently.

Accordingly, facilitating convergence is necessary to guarantee global performance in supply chain scheduling; moreover, both convergence and global performance are strongly affected by the degree of information sharing and remain critical challenges [51–53]. Therefore, this work investigates how to improve global performance in solving the SCPSP by increasing the interaction among enterprises without sacrificing convergence and by allowing more information to be shared among enterprises while still addressing the information privacy concerns of the enterprises involved.

This work proposes an agent-based fuzzy-constraint-directed negotiation (AFCN) method for solving the SCPSP in a fully distributed environment. In this AFCN, the SCPSP is modeled as a set of fuzzy constraint satisfaction problems (FCSPs) that are interlinked *via* inter-agent constraints. Each FCSP represents the interests and perspectives of one enterprise in the supply chain. These constraints are characterized by non-directional, declarative, and intuitive properties related to descriptions of real-world problems, such as the allocation capacity of a contractor or the production sequence of a product. Subjective, imprecise, and qualitative knowledge, including that associated with human cognition, preferences, or even opponents' perspectives, is frequently encountered in the business decision-making component of a supply chain and can easily be captured using fuzzy constraints at various consistency levels. In particular, fuzzy constraints are used to conveniently rank the candidate solutions by specifying the possibilities and prescribing to what extent the solutions are suitable. Furthermore, similarity measurements between offers and counter-offers are used to provide a basis for selecting a solution from among a set of feasible alternatives when making a multi-objective decision.

To accommodate the perspectives and interests of each enterprise in a supply chain, conflicts among FCSPs are resolved using an AFCN protocol through the iterative exchange of offers/counter-offers with limited information sharing and without privacy

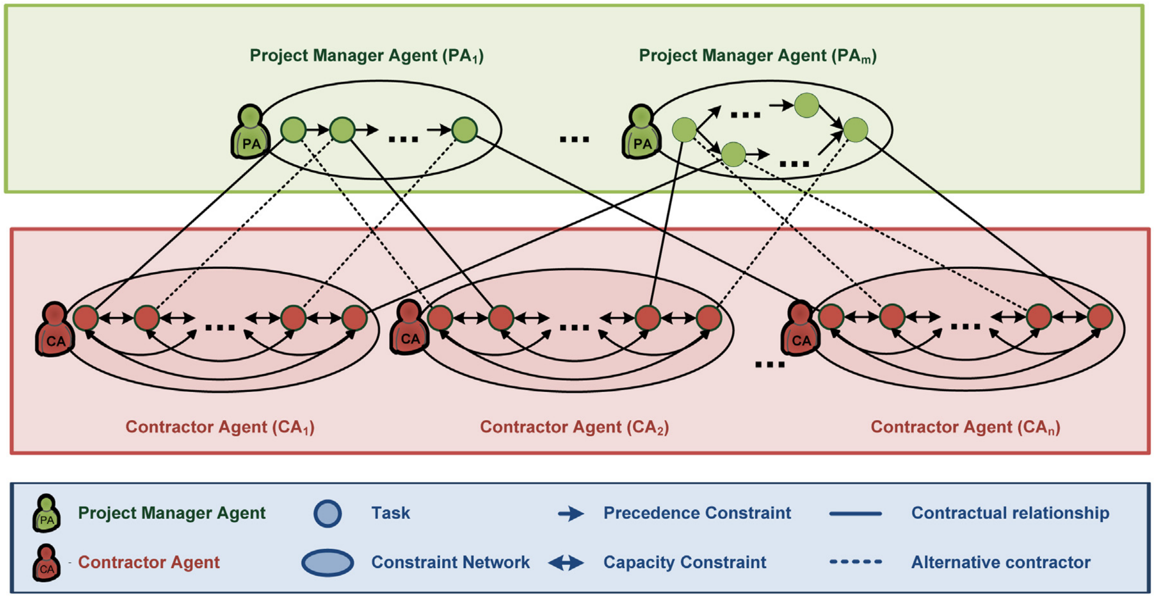

Fig. 1. Supply chain planning and scheduling with m PAs and n CAs.

breaches. When an enterprise makes an offer/counter-offer, it considers its own self-interest and preferences as well as its opponents' perspectives. A proposed offer/counter-offer represents not only a set of acceptable solutions and preferences for an operational task but also the possibility for conflict in this respect. For each FCSP, the incremental process of offer/counter-offer evaluation eliminates redundant and infeasible solutions. Based on the ranking of the solutions obtained through the application of fuzzy constraints, a set of feasible solutions can be further refined based on preferences through the application of an acceptable satisfaction value threshold, which enables an agent to respond rapidly to changes in the environment and to ensure that the proposed offers/counter-offers converge efficiently toward a satisfactory global solution.

The AFCN model is a practical and effective method for solving the SCPSP. The sharing of limited information (*i.e.*, fuzzy sets and preference functions) among agents enables them to elucidate their opponents' intentions through iterative negotiations, allowing the agents to reach an agreement while ensuring that the solution to a project planning and scheduling problem is satisfactory. The AFCN model is sufficiently flexible to incorporate different negotiation strategies such as competitive, win-win, and collaborative strategies, for various production environments. Experimental results show that the proposed AFCN model not only can generate a high-quality schedule (*i.e.*, comparable to those obtained using centralized methods) but also is time-efficient (*i.e.*, preserves the primary advantage of distributed and agent-based approaches) for solving a fully distributed SCPSP.

The remainder of this paper is organized as follows. Section 2 introduces the theoretical basis for the modeling of the SCPSP as a distributed fuzzy constraint satisfaction problem (DFCSP). Section 3 presents the AFCN model, including the process for generating offers and counter-offers, a communication protocol, and agent behavioral models that support effective interactions between PAs and CAs. Section 4 presents several experiments performed in various supply chain environments to verify the effectiveness and efficiency of the proposed method. Section 5 presents the conclusion to this paper and includes a discussion of the findings and directions for future research.

2. Modeling the SCPSP as a DFCSP

In a supply chain, a product passes through a variety of stages, from raw material and manufacturing to distribution, retail and delivery to end customers. This work focuses on the manufacturing stage of the supply chain. In this context, the SCPSP involves scheduling multiple projects across a network of manufacturers and suppliers. Manufacturers subcontract various tasks out to suppliers to complete their products. Suppliers differ from each other in how they perform various tasks in terms of available resources, processing time, and costs. Each project comprises a set of tasks or operations with complex precedence relationships. Typically, each task can be performed by more than one supplier. Thus, the process of project planning and scheduling involves allocating product completion tasks and selecting suppliers to minimize the total completion time and operating costs. If manufacturers are regarded as project managers and suppliers are regarded as contractors, then the SCPSP is defined as follows.

Definition 1. A supply chain planning and scheduling problem (SCPSP) can be modeled as a multi-agent system (MAS), ($\mathfrak{E}$, $\mathfrak{F}$, $\mathfrak{I}$), where

- $\mathfrak{E}$ is a set of m project manager agents (PAs) representing manufacturers, each of which is responsible for imposing temporal and precedence constraints on a set of tasks;
- $\mathfrak{F}$ is a set of n contractor agents (CAs) representing suppliers, each of which is responsible for imposing capacity constraints; and
- $\mathfrak{I}$ is a set of interrelations between the two classes of agents; each interrelation, $\mathfrak{I}_{i,j,t}$, specifies the jth CA, $\mathfrak{F}_j$, as a candidate for performing a task t that is governed by the ith PA, $\mathfrak{E}_i$.

According to Definition 1, an SCPSP is decomposed and converted into an MAS, which comprises PAs and CAs, as shown in Fig. 1. Each PA is responsible for imposing temporal and precedence constraints on a project. These constraints are further specified in terms of each task's processing time, due date, arrival (release) date, and tardiness cost. Each CA is responsible for imposing capacity constraints, which are specified by the corresponding supplier's processing cost and profit. Each interrelation, $\mathfrak{I}_{i,j,t}$, represents a contractual relationship. The i^{th} PA, $\mathfrak{E}_i$, and the j^{th} CA, $\mathfrak{F}_j$, must negotiate with each other to ensure task allocation and production at an acceptable processing time and cost.

Each agent who is responsible for some aspect of the SCPSP is described by an FCSP. The attributes of an agent consist of a set of objects, restrictions, and objectives, which are represented by fuzzy constraints and inter-relationships linking them with other related FCSPs. Therefore, the MAS is represented by a DFCSP, in which a solution that is mutually acceptable to the PAs and CAs is candidate solution that satisfies all of the constraints in a distributed fuzzy constraint network (DFCN) that specifies the fuzzy constraints of each individual agent and those among agents. Adapted from the work of Lai [54], a DFCN can be defined as follows.

Definition 2. A DFCN, $(\cup, \mathbf{X}, \mathbf{C})$, in an SCPSP, $(\mathfrak{E}, \mathfrak{F}, \mathfrak{I})$, can be defined as a set of $m+n$ fuzzy constraint networks (FCNs), $\{\mathfrak{N}^1, \mathfrak{N}^2, \cdots, \mathfrak{N}^{m+n}\}$, with $\mathfrak{N}^k = (\cup^k, \mathbf{X}^k, \mathbf{C}^k)$ representing the k$^{\text{th}}$ agent, where

- $\cup^k$ is the universe of discourse for the FCN $\mathfrak{N}^k$;
- $\mathbf{X}^k$ is a tuple of N_X non-recurring objects of the k$^{\text{th}}$ agent, $\mathbf{X}^k = (X_1^k, X_1^k, \cdots, X_{N_X}^k)$;
- $\mathbf{C}^k$ is a set of fuzzy constraints that includes a set of internal fuzzy constraints among objects in $\mathbf{X}^k$ and a set of external fuzzy constraints, $\mathfrak{I}^k$, between the k$^{\text{th}}$ agent and its opposing agents;
- $\mathfrak{N}^k$ is related to other FCNs by a set of external fuzzy constraints, $\mathfrak{I}^k$;
- $\cup$ is the universe of discourse for the entire DFCN;
- $\mathbf{X} = (\bigcup_{k=1}^{m+n} \mathbf{X}_k)$ is the tuple of all non-recurring objects; and
- $\mathbf{C} = (\bigcup_{k=1}^{m+n} \mathbf{C}_k)$ is the set of all fuzzy constraints.

According to Definition 2, the set of non-recurring objects $\mathbf{X}^k$ of the k^{th} agent represents his/her beliefs, knowledge of the environment (*e.g.*, tasks and resources), and any other attitudes (*e.g.*, the agent's desires, intentions, and concerns regarding his/her opponents' responses). The set of fuzzy constraints $\mathbf{C}_i$ for the i^{th} PA, $\mathfrak{E}_i$, corresponds to the set of restrictions (*e.g.*, precedence constraints), objectives (*e.g.*, total operating cost), and inter-agent constraints that apply between $\mathfrak{E}_i$ and his/her related CAs. Similarly, the set of fuzzy constraints $\mathbf{C}_j$ for the j^{th} CA, $\mathfrak{F}_j$, represents the set of restrictions (*e.g.*, capacity constraints), objectives (*e.g.*, revenue) and inter-agent constraints that apply between $\mathfrak{F}_j$ and his/her related PAs.

According to Definition 2, the solutions to $\mathbf{X}^k$, the FCN for the k^{th} agent, are regarded as the intention, $\prod_{\mathfrak{N}^k}$, of $\mathfrak{N}^k$, which is a fuzzy set of $\mathbf{X}^k$-ary tuples that satisfies all of the fuzzy constraints in $\mathbf{C}^k$. For simplicity, the network intention $\prod_{\mathfrak{N}^k}$ of the k^{th} agent is written as Π_k. Then, given a set of issues $\mathbf{Z}$ and a feasible solution $\mathbf{S}$ (*i.e.*, the value of an offer or counter-offer) such that $\mathbf{S} \in \Pi_k$, the aggregated satisfaction value of the solution $\mathbf{S}$ for the k^{th} agent, $\Psi^k(\mathrm{S})$, is defined as follows:

$$\Psi^k(\mathrm{S}) = \frac{1}{N_I}\sum_{l=1}^{N_I} Z_l(\mathbf{S}) \tag{1}$$

where $Z_l(\mathbf{S})$ is the l^{th} membership degree of solution $\mathbf{S}$ and N_I is the number of issues to be negotiated.

3. The AFCN model for the SCPSP

The AFCN model is proposed to solve project planning and scheduling problems in the context of a supply chain, in which offer generation and counter-offer evaluation are performed during task negotiation based on constraints, preferences and negotiation strategies. First, the process of generating offers and counter-offers is defined. Second, the inter-agent negotiation protocol for establishing and maintaining communication between agents is described. Third, agent behavioral models are used to address how an agent negotiates contracts with competing agents for the performance of particular tasks.

3.1. Generation of offers and counter-offers

The fuzzy-constraint-directed approach is known to be effective in agent negotiations [55–59]. During negotiations, each agent determines the solution to his/her own FCSP and exchanges offers with his/her partners (PAs and CAs) to resolve their inconsistencies subject to the time allowed to complete task allocation. If an opponent cannot accept an offer, then that opponent makes a counter-offer in a manner consistent with negotiation strategies that consider alternative solutions with a lower constraint satisfaction level. Offers and counter-offers are thus exchanged iteratively until a termination condition is met, causing the negotiation to end either in an agreement or in failure.

An agent's process of generating offers and counter-offers can be regarded as an inference procedure for solving an FCSP. Fig. 2 illustrates the process by which an agent generates an offer and a competing agent responds to that offer with a counter-offer. The process consists of six main steps: (i) evaluation of the opponent's state in response to a counter-offer, (ii) identification of the agent's internal state, (iii) determination of the agent's behavioral state, (iv) generation of a set of feasible solutions, (v) selection of a prospective solution, and (vi) generation of an offer and a subsequent counter-offer.

The process of generating an offer, $\mathbf{A}^*$, from the FCN of the k^{th} agent, $\mathfrak{N}^k = (\cup^k, \mathbf{X}^k, \mathbf{C}^k)$, to a contracting agent is described as follows. When the k^{th} agent receives a counter-offer, $\mathbf{B}$, it determines the behavioral state that it should adopt to present an alternative solution to the opposing agent or to offer a solution with a lower satisfaction level. The opponent's responsive state and the agent's own internal state, which represent the opponent's beliefs and the k^{th} agent's own perspectives, respectively, are both considered in the process of determining the agent's behavioral state.

The negotiation process begins once both an initial offer ($\mathbf{A}_0$) and the first counter-offer ($\mathbf{B}_0$) from the PA and CA, respectively, have been generated. The initial offer ($\mathbf{A}_0$) from the PA for a given task operation is computed according to Eqs. (11)–(19) and often represents the prospective solution with the highest possible degree of satisfaction. After receiving the initial offer $\mathbf{A}_0$ from the PA, the CA generates his/her first counter-offer ($\mathbf{B}_0$) according to Eqs. (11)–(19) and chooses the prospective solution with the highest possible degree of satisfaction for the CA to complete the corresponding task. The opponent's responsive state $\mathbf{Q}$ is the difference σ between the last offer $\mathbf{A}$ and the most recent counter-offer $\mathbf{B}$, which is defined as follows:

$$\sigma = 1 - (G(\mathbf{A}_0, \mathbf{B}_0) - G(\mathbf{A}, \mathbf{B}))/G(\mathbf{A}_0, \mathbf{B}_0) \tag{2}$$

where $\mathbf{A}_0$ and $\mathbf{B}_0$ are the initial offer and the first counter-offer received, respectively. The distance measure $G(\mathbf{A,B})$ is associated with an offer $\mathbf{A}$ and a counter-offer $\mathbf{B}$ over a set of negotiation issues $\mathbf{I} \in \mathbf{X}^k$, as shown below:

$$\mathrm{G(A,B)}\frac{1}{N_I}\sqrt{\sum_{l=1}^{N_I} \mathrm{L}(A_l, B_l)^2} \tag{3}$$

where L is the Euclidean distance between the two fuzzy sets [60,61] A_l and B_l, which represent the possibility distributions of offer $\mathbf{A}$ and counter-offer $\mathbf{B}$, respectively, over the issue $\mathbf{I}_l \in \mathbf{I}$. The

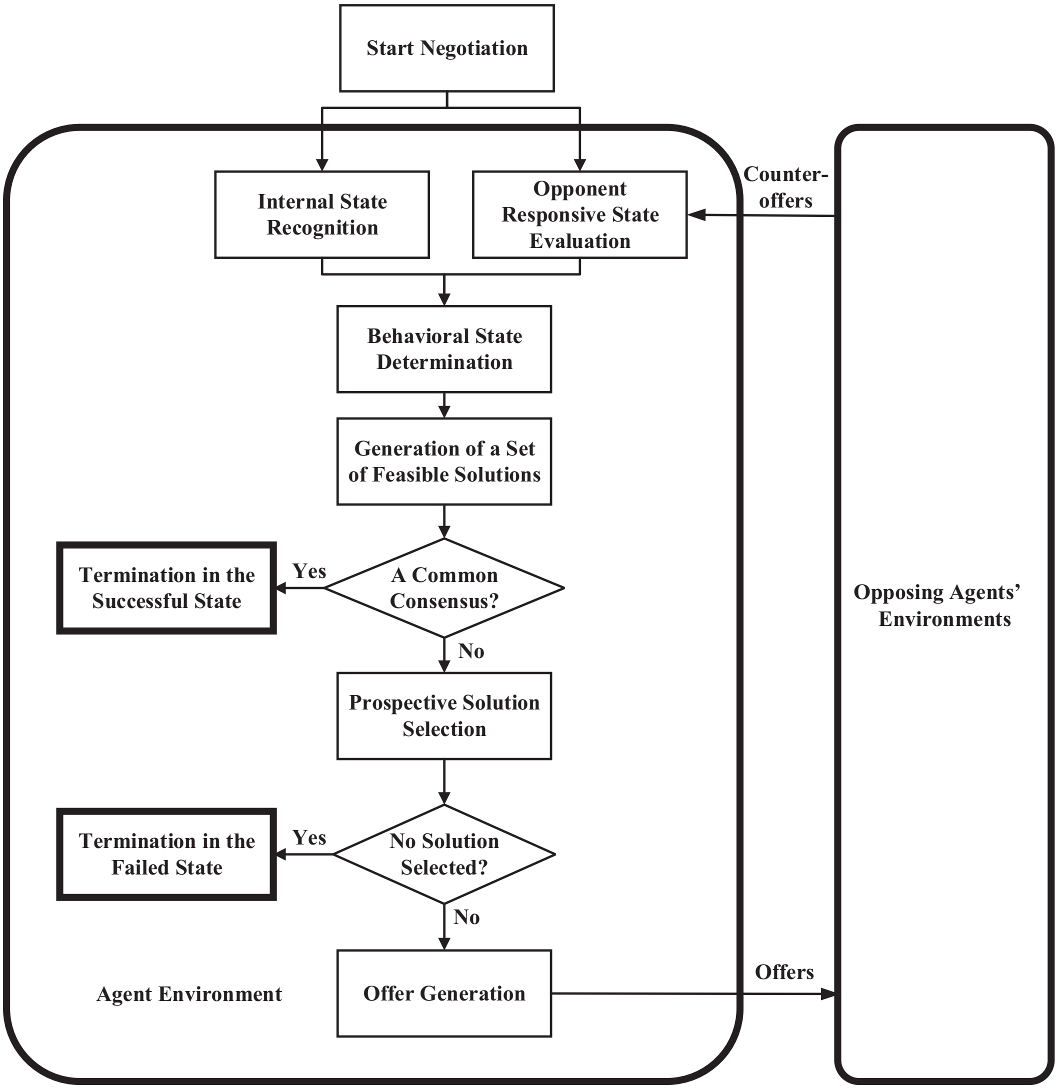

Fig. 2. The process of generating offers and counter-offers.

distance $L(A_l, B_l)$ is transformed from the fuzzy distance D_l^1 over issue $\mathbf{I}_l$ as follows:

$$L(A_l, B_l) = F_{df}(D_l^1) = \frac{\sum_z D_l^1(z) * z}{\sum_z D_l^1(z)} \tag{4}$$

where F_{df} is a defuzzification method and

$$D_l^1 = \sup_{d(x,y)} \min[A(x), B(y)] \tag{5}$$

where $d(x, y)$ is the Euclidean distance between x and y such that $x \in$ A and $y \in$ B.

The internal state $\mathbf{P}$ is related to the satisfaction level ρ associated with the most recent offer $\mathbf{A}$ and the tightness δ of a set of alternative solutions. Given the prospective solution $\mathbf{S}^*$ for the FCN $\mathfrak{N}^k$ in the last negotiation round and the threshold for the aggregated satisfaction value, ε, that should be obtained in the final negotiation round, the task of determining the internal state during offer evaluation involves estimating the satisfaction level ρ and the tightness δ of a set of alternative solutions, where these quantities are defined as follows:

$$\rho = \Psi^k(\mathbf{S}^*) \tag{6}$$

$$\delta = 1 - (\rho - \varepsilon) \tag{7}$$

Based on his/her current internal state $\mathbf{P}$ and the opponent's responsive state $\mathbf{Q}$, the k^{th} agent can decide whether to make a concession and how to specify the level of that concession. Then, the concession value $\Delta\varepsilon$ is transformed from the fuzzy concession set $\mathbf{V}$ using a τ level-cut as follows:

$$\Delta\varepsilon = F_{df}(V_\tau) \tag{8}$$

where F_{df} is a defuzzification method. The parameter τ for the desired concession level is determined by the satisfaction level $\mu_\rho(\rho)$, the tightness $\mu_\delta(\delta)$, and the difference between the last offer and the most recent counter-offer $\mu_\sigma(\sigma)$ as follows:

$$\tau = (\mathbf{P}, \mathbf{Q}) = (\mu_\rho(\rho) \wedge \mu_\delta(\delta) \wedge \mu_\sigma(\sigma))^{\omega_\tau} \tag{9}$$

where the parameter ω_τ is the weight that is associated with the applied negotiation strategy (for example, the competitive, win-win, or collaborative strategy), which affects the preferred level of concession. The new aggregated satisfaction threshold ε^* is defined as follows:

$$\varepsilon^* = \varepsilon - \Delta\varepsilon \tag{10}$$

Therefore, an agent must present a prospective solution from within his/her individual preference region, which is limited by the new aggregated satisfaction threshold ε^* for offer generation and counter-offer evaluation.

Definition 3. (Set of feasible solutions) Given the intention ${}_{\alpha}\Pi_k$ of the FCN $\mathfrak{N}^k$ at level α and the most recent aggregated satisfaction threshold ε^* of the kth agent, the task of generating a set of feasible solutions, $\mathfrak{P}$, is defined as follows:

$$\mathfrak{P} = \Gamma({}_{\alpha}\Pi_k, \varepsilon^*) \tag{11}$$

$$\Gamma({}_{\alpha}\Pi_k, \varepsilon^*) = \{\mathbf{S} | (\mathbf{S} \in {}_{\alpha}\Pi_k) \wedge (\Psi^k(\mathbf{S}^*) \geq \varepsilon *)\} \tag{12}$$

where $\Psi^k(\cdot)$ is the aggregated satisfaction value of the set of goals in the FCN $\mathfrak{N}_k$.

Given a counter-offer B and a feasible solution set $\mathfrak{P}$, the task of selecting a prospective solution to propose, $\mathbf{S}^*$, is defined as follows:

$$\mathbf{S}^* = arg(\max_{\mathbf{S} \in \mathfrak{P}} H(\mathbf{S}, \mathbf{B})) \tag{13}$$

where $H(\mathbf{S}, \mathbf{B})$ is a utility function that is used to evaluate the appropriateness of a feasible schedule $\mathbf{S} \in \mathfrak{P}$ and the counter-offer $\mathbf{B}$ and is defined as follows:

$$H(\mathbf{S}, \mathbf{B}) = \frac{1}{N_I} \sqrt{\sum_{l=1}^{N_I} (W_1(\mathbf{S}_l)^{\omega_1} \wedge W_2(\mathbf{S}_l, \mathbf{B}_l)^{\omega_2})^2} \tag{14}$$

where W_1 is a satisfaction function and W_2 is a similarity function based on the Euclidean distance between solution $\mathbf{S}$ and counter-offer $\mathbf{B}$ over issue $\mathbf{I}_l$. The distance used for the similarity function $W_2(\mathbf{S}_l, \mathbf{B}_l)$ is transformed from the fuzzy distance D_l^2 over the issue $\mathbf{I}_l$ as follows:

$$W_2(\mathbf{S}_l, \mathbf{B}_l) = F_{df}(D_l^2) = \frac{\sum_z D_l^2(z) * z}{\sum_z D_l^2(z)} \tag{15}$$

$$D_l^2 = \sup_{d(x,y)} \min[S(x), B(y)] \tag{16}$$

Depending on the adopted negotiation strategy, the parameters ω_1 and ω_2 are the weights associated with the satisfaction level and the similarity value, respectively, for a task.

Definition 4. (Offer) Given a feasible solution set $\mathfrak{P}$ and a prospective solution $\mathbf{S}^*$, the task of proposing an offer, $\mathbf{A}^* = (A_1^*, A_2^*, \cdots, A_{N_I}^*)$, over the set of issues $\mathbf{I} \in \mathbf{X^k}$ is defined as follows:

$$\mathbf{A}^* = \Lambda(\mathfrak{P}, \mathbf{S}^*) \tag{17}$$

Each A_l^* for issue $I_l \in \mathbf{I}$ is the marginal particularized possibility distribution of the value of S_l^* in the space $\mathbf{X}^k$ and is defined as follows:

$$A_l^* = Proj_{\mathbf{X}_l}\left(\mathfrak{P} \cap \bar{\Pi}_{\mathbf{X}_1^k} \cap \bar{\Pi}_{\mathbf{X}_2^k} \cdots \cap \bar{\Pi}_{\mathbf{X}_{l-1}^k} \cap \bar{\Pi}_{\mathbf{X}_{l+1}^k} \cap \cdots \cap \bar{\Pi}_{\mathbf{X}_{N_X}^k}\right) \tag{18}$$

where $\bar{\Pi}_{\mathbf{X}_l^k} = S_l^*$, $\bar{\Pi}_{\mathbf{X}_l^k}$ is the cylindrical extension of $\Pi_{\mathbf{X}_l}$ in the space $\mathbf{X}^k$, $\mathbf{X}_l$ is the object of the lth issue, and N_X is the total number of objects.

As determined by counter-offer $\mathbf{B} = (B_1, \ldots, B_l, \cdots, B_{N_I})$ over the set of issues $\mathbf{I}$, where $l = 1, \cdots, N_I$, the set of the opposing agent's preferred solutions $\hat{\mathfrak{P}}$ to the FCSP of the k^{th} agent, $(\cup^k, \mathbf{X}^k, \mathbf{C}^k)$, is defined as follows:

$$\hat{\mathfrak{P}} = (\bar{\mathbf{B}}_1, \cdots, \bar{\mathbf{B}}_l, \cdots, \bar{\mathbf{B}}_{N_I}) \tag{19}$$

where $\bar{\mathbf{B}}_l$ is the cylindrical extension of $\mathbf{B}_l$ in the space $\mathbf{X}^k$.

The negotiation processes involves the iterative exchange of offers and counter-offers until either the negotiation reaches an agreement or an offer/counter-offer fails to be proposed. Thus, the termination conditions for negotiation are defined as follows.

Definition 5. (Termination) Given a feasible solution set $\mathfrak{P}$ and a counter-offer $\mathbf{B}$ for the kth agent, the fuzzy-constraint-directed negotiation process of the kth agent terminates if the kth agent reaches an agreement when

$$(\mathfrak{P} \cap \hat{\mathfrak{P}}) \neq \phi \tag{20}$$

Alternatively, the kth agent fails in negotiation if

$$(\mathfrak{P} \cap \hat{\mathfrak{P}}) = \phi \tag{21}$$

The final solution is selected from the intersection of $\mathfrak{P}$ and $\hat{\mathfrak{P}}$. An agreement is reached when the k^{th} agent and the opposing agent find a solution with an acceptable satisfaction value that is present in both the current feasible solution set $\mathfrak{P}$ of the k^{th} agent and his/her opposing agent's preferred solution set, $\mathfrak{P}$. Alternatively, if the negotiation of the k^{th} agent does not reach an agreement with an acceptable satisfaction value, then a new prospective solution is generated. The negotiation results in failure when the aggregated satisfaction threshold drops below the acceptable level or when the maximum number of negotiation iterations has been reached.

3.2. Negotiation protocol and communication messages

A negotiation protocol is presented to prescribe common negotiation rules and to determine message sequences. The agents are connected *via* a communication network, and the passing of messages is considered to be the means of communication. During the negotiation process, agents may send or receive the following types of messages:

- ***Ask*** $((\mathfrak{C}_i, \mathfrak{F}), \mathbf{A})$: The i^{th} PA, $\mathfrak{C}_i$, proposes an offer $\mathbf{A}$ to a set of CAs, $\mathfrak{F}$, to request an operation time and price for the completion of a task.
- ***Tell*** $((\mathfrak{F}_j \mathfrak{C}_i), \mathbf{B})$: The j^{th} CA, $\mathfrak{F}_j$, proposes a counter-offer $\mathbf{B}$ to the i^{th} PA, $\mathfrak{C}_i$, in response to the PA's request.
- ***Accept*** $((\mathfrak{C}_i, \mathfrak{F}_j), \mathbf{B})$: The i^{th} PA, $\mathfrak{C}_i$, informs the j^{th} CA, $\mathfrak{F}_j$, that a successful deal $\mathbf{B}$ has been made and that the negotiation has been terminated.
- ***Reject*** $((\mathfrak{C}_i, \mathfrak{F}), \phi)$: The i^{th} PA, $\mathfrak{C}_i$, informs a set of CAs, $\mathfrak{F}$, that their counter-offers have been rejected and that the negotiation has been terminated.
- ***Agree*** $((\mathfrak{F}_j \mathfrak{C}_i), \mathbf{A})$: The j^{th} CA, $\mathfrak{F}_j$, informs the i^{th} PA, $\mathfrak{C}_i$, that offer $\mathbf{A}$ has been accepted by $\mathfrak{F}_j$.
- ***Abort*** $((\mathfrak{C}_i, \mathfrak{F}_j), \phi)$ or ***Abort***$(\left(F_j \mathfrak{C}_i\right), \phi)$: No feasible solution has been reached between the i^{th} PA, $\mathfrak{C}_i$, and the j^{th} CA, $\mathfrak{F}_j$, and they withdraw from the negotiation.

Fig. 3 illustrates the process of negotiation among PAs and CAs. A negotiation is initiated by a PA's announcement to corresponding CAs regarding the requirements of a task, such as a completion time and a price. In step 1, the initial offers generate "***Ask***" messages that are sent to CAs. When a CA receives an offer from a PA, he/she evaluates the offer using Eq. (20) to determine whether the offer satisfies the CA's constraints, such as capacities, restrictions, and preferences; this is step 2. If the offer does not satisfy these constraints (step 3), then the CA generates a counter-offer using Eq. (2) through Eq. (17) (step 4). The resources allocated to each task are then determined to enable the CA to respond to the PA with a counter-offer *via* a "***Tell***" message (step 5). By contrast, if an agent can generate no feasible solution in the counter-offer generation process, then he/she responds with an "***Abort***" message (step 17 or 18) and withdraws from the negotiation process.

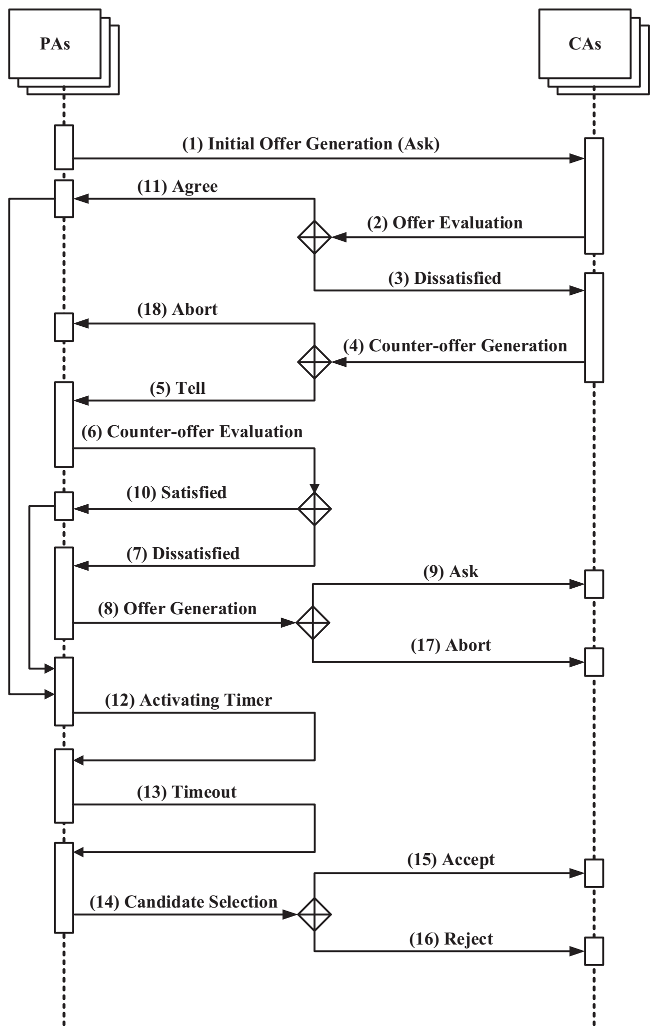

Fig. 3. The process of negotiation among PAs and CAs.

When a PA receives a counter-offer from a CA (step 6), this counter-offer is evaluated to determine whether the offer satisfies the task requirements. If the counter-offer does not satisfy the task requirements (step 7), then an offer is generated using Eq. (2) through Eq. (17) to adjust the task requirements (step 8). An "***Ask***" message with a new offer is then sent to the CA (step 9). During the negotiation process, the PAs and CAs iteratively exchange offers and counter-offers, as shown in steps 2 through 9.

However, if an offer satisfies a CA's constraints in step 2, then that CA sends an "***Agree***" message (step 11) to inform the PA that he/she accepts the offer and is awaiting a response. If the PA receives an "***Agree***" message from a CA or if a CA's counter-offer satisfies the PA's constraints (step 10), then the corresponding CA becomes a candidate for performing a project task, and a timer is activated (step 12). Next, the PA continues to negotiate with other CAs until a certain time limit is reached; this condition is also referred to as "timeout" in this paper. The PA selects the best counter-offer from among the set of candidate CAs (step 14). Then, an "***Accept***" message is sent (step 15) to inform the selected CA that a successful deal has been made and "***Reject***" messages are sent (step 16) to inform the other CAs that the negotiation has been terminated and that their counter-offers have been rejected.

The negotiation process terminates once all PAs have completed their negotiations. The contracting of each task is terminated when the corresponding PA reaches an agreement with one CA or when all of the potential CAs have withdrawn from negotiations. A PA continues contracting tasks until all tasks have been assigned or until one task has failed.

3.3. Agent behavioral models

Initially, a PA commences negotiations by proposing its ideal solution to all potential CAs. Subsequently, the PAs and CAs follow certain behavior models, introduced below, to determine how they will exchange offers and counter-offers to reach agreements regarding the scheduling of activities among agents.

3.3.1. PA behavior

Algorithm 1 describes the behavior of the i^{th} PA, $\mathfrak{E}_i$, as he/she contracts with the corresponding CAs, $\mathfrak{F}$, to perform a task. Initially, $\mathfrak{E}_i$ proposes his/her prospective solution $\mathbf{S}_i^*$, which is selected using Eq. (13) based on his/her feasible solutions in the first negotiation round. Then, the prospective solution $\mathbf{S}_i^*$ is translated into a set of offers, $\mathbf{A}_i^*$, which are incorporated into "**Ask**" messages and sent to a set of CAs, $\hat{\mathfrak{E}}_i$ (in lines 3 and 4). Next, $\mathfrak{E}_i$ continues collecting responses from the corresponding CAs in $\hat{\mathfrak{E}}_i$ within a particular period. These responses may include "***Tell***", "***Agree***", and "***Abort***" messages (in line 6).

If all of the received messages are of the "***Abort***" type, then all of the corresponding CAs in $\hat{\mathfrak{E}}_i$ have withdrawn from the negotiations because no feasible solution can be obtained. Then, PA $\mathfrak{E}_i$ will terminate the negotiations in the failure state (in lines 7 and 8). Otherwise, PA $\mathfrak{E}_i$ will adjust the set of feasible solutions using Eq. (14), based on his/her new behavioral state ε_i^*. The behavioral state ε_i^* is a function of the internal state $\mathbf{P}_i$ and the opponent's responsive state $\mathbf{Q}_i$ (in lines 10 and 11).

Then, a counter-offer $B_{j,i}$ is selected as a candidate solution $\mathbf{B}_i^*$ if the j^{th} CA, $\mathfrak{F}_j$, accepts this offer and sends an "***Agree***" message or if $\mathfrak{E}_i$ accepts the counter-offer $B_{j,i}$. Once a counter-offer has been selected as a candidate solution, a timer $\mathfrak{T}_i$ is activated (in lines 12–17).

If none of the counter-offers from the CAs satisfies a PA's constraints, then the prospective solution $\mathbf{S}_i^*$ is adjusted using Eq. (19), which considers the adjusted feasible solution set $\mathfrak{P}_i$ and the latest counter-offers $\mathbf{B}_i$. Then, the solution $\mathbf{S}_i^*$ is translated into offers A_i using Eq. (17), which are incorporated into "**Ask**" messages and sent to the corresponding CAs, $\hat{\mathfrak{E}}_i$. However, if $\mathfrak{E}_i$ is unable to propose a solution in response to counter-offer $B_{j,i}$, then an "***Abort***" message is sent to CA $\mathfrak{F}_j$ (in lines 18–23). If all of the received or sent messages are "***Abort***" messages, then PA $\mathfrak{E}_i$ terminates the negotiations in the failure state (in lines 24 and 25).

PA $\mathfrak{E}_i$ continues to negotiate with the corresponding CAs until timeout occurs. If timer $\mathfrak{T}_i$ reaches timeout, then the best counter-offer, $B_{j',i}$, is selected from among the set of candidate solutions, B_i^*. Thus, an agreement S_i^* regarding task allocation is generated using Eq. (13) and is sent to the $j'th$ CA, $\mathfrak{F}_{j'}$, *via* an "***Accept***" message. "***Reject***" messages are sent from PA $\mathfrak{E}_i$ to the other CAs, whose counter-offers have been rejected, and the negotiation process is terminated (in lines 27–32).

Algorithm 1. PA Behavior

Procedure PROJECT MANAGEMENT AGENT ($\mathfrak{E}_i$)
$status \leftarrow$ "*normal*";
$Generate\ initial\ offers\ \mathbf{A}_i^*\ for\ each\ \mathfrak{F}_j \in \widehat{\mathfrak{E}}_i$;
$\forall A_{i,j}^* \in \mathbf{A}_i^*, send\ M_{i,j} =$ "$Ask((\mathfrak{E}_i, \mathfrak{F}_j), A_{i,j}^*)$";
repeat
  $Receive\ \widehat{\mathbf{M}}_i = \{\widehat{M}_{j,i} | \widehat{M}_{j,i}$
    $=$ "$Tell((\mathfrak{F}_j, \mathfrak{E}_i), B_{j,i})$" **or** "$Agree((\mathfrak{F}_j, \mathfrak{E}_i), B_{j,i})$" **or** "$Abort((\mathfrak{F}_j, \mathfrak{E}_i), \emptyset)$", $\mathfrak{F}_j \in \widehat{\mathfrak{E}}_i\}$;
  if $\forall \widehat{M}_{j,i} \in \widehat{\mathbf{M}}_i, (\widehat{M}_{j,i}\ is$ "$Abort$"$)$ **then**
    $status \leftarrow$ "*failure*";
  else
    $Get\ counter\text{-}offer\ set\ \mathbf{B}_i\ from\ Tell\ messages$;
    $Generate\ new\ feasible\ solution\ set\ \mathfrak{P}_i$;
    for each $\mathfrak{F}_j \in \widehat{\mathfrak{E}}_i$ **do**
      if $\mathfrak{E}_i\ and\ \mathfrak{F}_j\ have\ reached\ a\ consensus$ **then**
        $Remove\ B_{j,i}\ from\ \mathbf{B}_i\ and\ add\ it\ to\ candidate\ set\ \mathbf{B}_i^*$;
        if $(Timer\ \mathfrak{T}_i = \emptyset)$ **then** $activate\ Timer\ \mathfrak{T}_i$;
      end if
    end for
    if $(\exists \mathfrak{F}_j \in \widehat{\mathfrak{E}}_i, \mathfrak{E}_i\ and\ \mathfrak{F}_j\ have\ not\ yet\ reached\ a\ consensus)$ **and** $(Timer\ \mathfrak{T}_i\ is\ counting)$ **then**
      $Generate\ offers\ \mathbf{A}_i^*\ for\ each\ B_{i,j}\ in\ \mathbf{B}_i$;
      for each $A_{i,j}^* \in \mathbf{A}_i^*$ **do**
        if $(A_{i,j}^* \neq \emptyset)$ **then** $Send\ =$ "$Ask((\mathfrak{E}_i, \mathfrak{F}_j), A_{i,j}^*)$";
        if $(A_{i,j}^* = \emptyset)$ **then** $Send\ =$ "$Abort((\mathfrak{E}_i, \mathfrak{F}_j), \emptyset)$";
      end for
      if $\forall \mathfrak{F}_j \in \widehat{\mathfrak{E}}_i, (\widehat{M}_{j,i}\ is$ "$Abort$"$)\ or\ (M_{i,j}\ is$ "$Abort$"$)$ **then**
        $status \leftarrow$ "*failure*";
      end if
    else if $Timer\ \mathfrak{T}_i\ is\ timeout$ **then**
      $Select\ the\ best\ count\text{-}offer\ B_{j',i} \in \mathbf{B}_i^*$;
      $Generate\ agreement\ S_i^*\ according\ to\ B_{j',i}$;
      $Send\ M_{i,j'} =$ "$Accept((\mathfrak{E}_i, \mathfrak{F}_{j'}), S_i^*)$";
      $\forall B_{j,i} \in \mathbf{B}_i^*, j \neq j', send\ M_{i,j} =$ "$Reject((\mathfrak{E}_i, \mathfrak{F}_j), \emptyset)$";
      $status \leftarrow$ "*success*";
    end if
  end if
until $status\ is$ "*success*" **or** "*failure*";

3.3.2. CA behavior

Algorithm 2 describes the behavior of CA $\mathfrak{F}_j$ when contracting to perform a set of tasks with the relevant PAs $\hat{\mathfrak{F}}_j$. The negotiations of $\mathfrak{F}_j$ begin when the CA receives messages from the relevant PAs $\hat{\mathfrak{F}}_j$. If all of the received messages are of the "***Abort***" or "***Reject***" type, then CA $\mathfrak{F}_j$ terminates the negotiations in the failure state (in lines 4 and 5).

An agreement $\mathbf{S}_j^*$ is achieved and the negotiation is terminated in the success state if none of the messages is an "***Ask***" message and at least one message is an "***Accept***" message (in lines 6–8). Otherwise, CA $\mathfrak{F}_j$ adjusts his/her set of feasible solutions using Eq. (11) based on his/her new behavioral state, ε_j^*. The behavioral state ε_j^* is determined based on the CA's internal state $\mathbf{P}_j$ and the opponent's response state $\mathbf{Q}_j$ (in lines 10 and 11). If CA $\mathfrak{F}_j$ accepts offers $\mathbf{A}_j$ after evaluating these offers, then solutions $\mathbf{S}_j^*$ are generated, and the CA sends corresponding "***Agree***" messages to the relevant PAs $\hat{\mathfrak{F}}_j$ (in lines 12–15).

If offers $\mathbf{A}_j$ do not satisfy the CA's constraints, then the prospective solution $\mathbf{S}_j^*$ is adjusted based on the adjusted feasible solution set $\mathfrak{P}_j$ and the latest offers $\mathbf{A}_j$. Then, solution $\mathbf{S}_j^*$ is translated into counter-offers $\mathbf{B}_j^*$, which are incorporated into "***Tell***" messages and sent to the relevant PAs in $\hat{\mathfrak{F}}_j$. However, if CA $\hat{\mathfrak{F}}_j$ cannot propose a solution in response to offer $A_{i,j}$, then an "***Abort***" message will be sent to PA $\mathfrak{E}_i$ (in lines 17–21). If all of the received messages are "***Abort***" messages, then CA $\hat{\mathfrak{F}}_j$ terminates the negotiation process in the failure state (in lines 22–24).

Algorithm 2. CA Behavior

Procedure CONTRACTOR AGENT ($\mathfrak{F}_j$)
repeat
  Receive $\widehat{\mathbf{M}}_j = \{\widehat{M}_{i,j} | \widehat{M}_{i,j} = \text{“}Ask((\mathfrak{E}_i, \mathfrak{F}_j), A_{i,j})\text{”}$ **or** $\text{“}Abort((\mathfrak{E}_i, \mathfrak{F}_j), \emptyset)\text{”}$ **or** $\text{“}Accept((\mathfrak{E}_i, \mathfrak{F}_j), A_{i,j})\text{”}$ **or** $\text{“}Reject((\mathfrak{E}_i, \mathfrak{F}_j), \emptyset)\text{”}, \mathfrak{F}_j \in \widehat{\mathfrak{E}}_i\}$;
  if $\forall \widehat{M}_{i,j} \in \widehat{\mathbf{M}}_j, (\widehat{M}_{i,j}\ is\ \text{“}Abort\text{”})$ **or** $(\widehat{M}_{i,j}\ is\ \text{“}Reject\text{”})$ **then**
    $status \leftarrow \text{“}failure\text{”}$;
  else if $\forall \widehat{M}_{i,j} \in \widehat{\mathbf{M}}_j, (\widehat{M}_{i,j}\ is\ \text{“}Accept\text{”})$ **then**
    Get agreements $\boldsymbol{S}_i^*$ *from Accept messages*;
    $status \leftarrow \text{“}success\text{”}$;
  else
    Get the offer set $\mathbf{A}_j$ *from Ask messages*;
    Generate new feasible solution set $\mathfrak{P}_j$;
    if $\forall \mathfrak{E}_i \in \widehat{\mathfrak{F}}_i, \mathfrak{F}_j$ *and* $\mathfrak{E}_i$ *have reached a consensus* **then**
      Generate agreement S_j^* *according to* $\boldsymbol{A}_j$;
      $\forall A_{i,j} \in \mathbf{A}_j, send\ M_{j,i} = \text{“}Agree((\mathfrak{F}_j, \mathfrak{E}_i), \mathbf{S}_j)\text{”}$;
      $status \leftarrow \text{“}success\text{”}$;
    else
      Generate counter-offsers $\mathbf{B}_j^*$ *for each* $A_{i,j} \in \mathbf{A}_j$;
      for each $\mathrm{B}_{j,i}^* \in \mathbf{B}_j^*$ **do**
        if $(\mathrm{B}_{j,i}^* \neq \emptyset)$ **then** $Send\ M_{j,i} = \text{“}Tell((\mathfrak{F}_j, \mathfrak{E}_i), B_{j,i}^*)\text{”}$;
        if $(\mathrm{B}_{j,i}^* = \emptyset)$ **then** $Send\ M_{j,i} = \text{“}Abort((\mathfrak{F}_j, \mathfrak{E}_i), \emptyset)\text{”}$;
      end for
      if $\forall \mathfrak{E}_i \in \widehat{\mathfrak{F}}_j, (\widehat{M}_{i,j}\ is\ \text{“}Abort\text{”})\ or\ (M_{j,i}\ is\ \text{“}Abort\text{”})$ **then**
        $status \leftarrow \text{“}failure\text{”}$;
      end if
    end if
  end if
until $status\ is\ \text{“}success\text{”}$ **or** $\text{“}failure\text{”}$;

4. Numerical experiments

To examine the effectiveness of the proposed AFCN model, numerical experiments were conducted using different numbers of PAs and CAs and various negotiation strategies. The performance of the AFCN model was compared with that of the centralized heuristic (CTR), which is based on fuzzy tabu search method [16,20], and two distributed and agent-based methods, the MCNP [6,42–45] and the MIA protocol [46–50], in terms of makespan, total operating cost, and computational efficiency. The CTR requires full information sharing and can generate an optimal or near-optimal solution; thus, the solution generated by this method was used as a baseline. The MCNP is a single-shot negotiation protocol in which start-time window information is shared between the PAs and CAs. The MIA protocol has the same level of information sharing, but it also supports an iterative auction-based negotiation process to improve the results. With some minor adaptations, these approaches are good representative methods for comparison with the proposed AFCN model.

In the numerical experiments, each project was defined by a manufacturer to comprise a linear sequence of tasks, and each task was specified in terms of its precedence constraints, processing time, and tardiness cost. Each supplier was specified in terms of its unique capacity and processing cost. Each PA contracted with CAs to perform a task within an allocated time for an agreed-upon price. The objective function of each PA served to minimize the operating cost of completing the corresponding project. The operating cost was defined to include the tardiness cost of the project and the sum of the production costs incurred to perform each task of the project. By contrast, the objective function of each CA served to maximize his/her total profit, which was computed by subtracting the total processing cost from the total revenue earned by the supplier for performing the contracted tasks.

In each experiment, 50 instances were randomly generated to measure the performance of each method. This repetition of the experiments enhanced the reproducibility of the performance comparisons between the proposed AFCN model and the CTR, MCNP, and MIA methods. The algorithms were implemented on a personal computer with a 2.0 GHz CPU and 1 GB of RAM, and the MATLAB environment was used to perform the numerical experiments.

4.1. Performance comparisons using different numbers of PAs

The proposed AFCN model was compared with the CTR, the MCNP and the MIA protocol in terms of makespan and total operating cost by considering different numbers of PAs from 2 to 20, with the number of CAs fixed at 15. Additionally, the number of tasks per project was fixed at 5 for all instances, and the number of candidate CAs considered to perform each task was fixed at 3.

Figs. 4 and 5 show the average makespans and average total operating costs, respectively, for the solutions found by the CTR, MCNP, MIA and AFCN methods. The CTR, with its complete information sharing, outperformed the other three methods. However, the computation time of the CTR grows exponentially with an increasing number of PAs, as shown in Table 1. By contrast, the MCNP, which is a single-shot, take-it-or-leave-it negotiation protocol, requires the least computation time. The MCNP coordinates the competition for resources with limited knowledge of other agents. Thus, compared with the other approaches, the MCNP results in larger makespans and total operating costs, which increase with an increasing number of PAs. The MIA method yields lower makespans and total operating costs than does the MCNP, as shown in Figs. 4 and 5. However, although the price of a bid can indicate the degree of contention of a demand, it does not indicate the direction of the available agreement area, and thus, the negotiation process may oscillate or may converge only slowly. The PAs can blindly increase the prices for tasks to acquire better resources to perform those tasks, but the total operating cost therefore does not improve.

Table 1
Computationtimes of the CTR, MCNP, MIA, and AFCN protocols for different numbers of PAs.

Total number of PAs	Computation time (unit of measure in seconds)			
	CTR	MCNP	MIA	AFCN
2	36	0.05	2.05	0.74
4	136	0.14	3.10	1.22
6	826	0.19	3.82	1.58
8	1,850	0.23	5.49	2.08
10	2,951	0.36	7.40	2.60
12	4,832	0.40	9.15	2.91
14	7,478	0.52	11.31	4.31
16	12,122	0.67	12.52	4.92
18	19,613	0.88	16.58	6.58
20	31,668	1.06	18.85	7.93

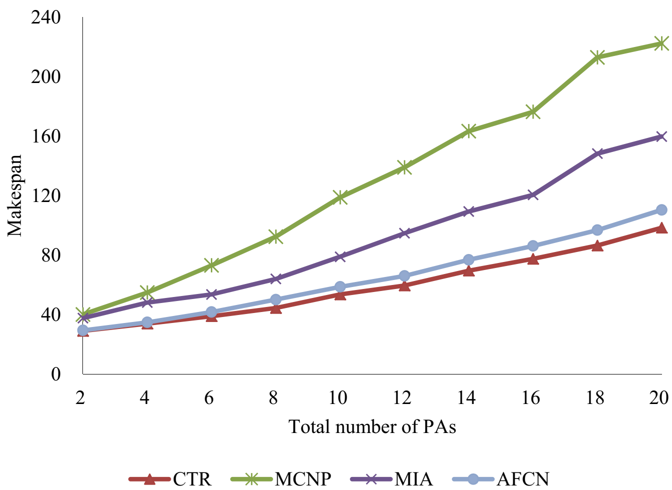

Fig. 4. Performance comparisons in terms of makespan for the CTR, MCNP, MIA, and AFCN methods.

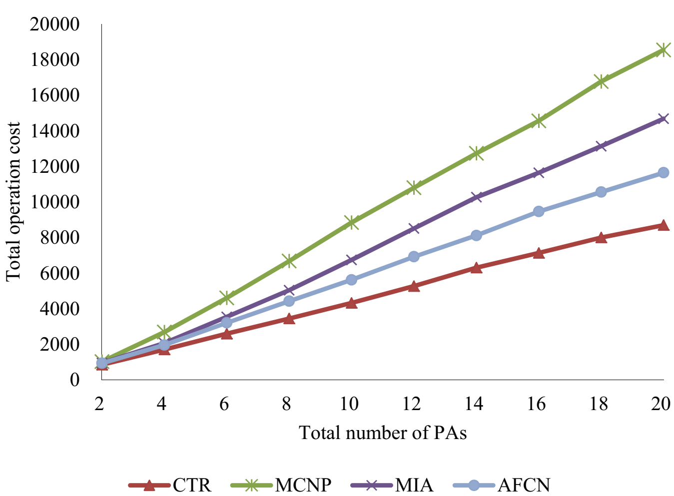

Fig. 5. Performance comparisons in terms of total operating cost for the CTR, MCNP, MIA, and AFCN methods.

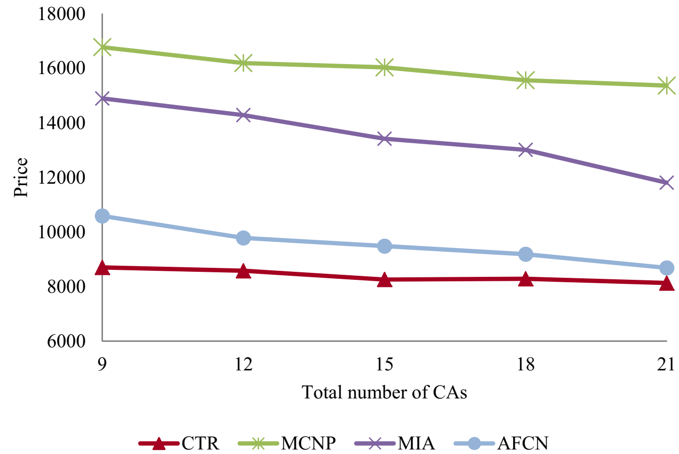

Fig. 6. Performance comparisons in terms of price for the CTR, MCNP, MIA, and AFCN methods.

The AFCN model outperforms the MIA protocol not only in terms of solution quality but also in computational efficiency. The differences between the AFCN and MIA methods in terms of makespan and total operating cost increase significantly with an increasing number of PAs. In the proposed AFCN model, the time within which a task must be performed is represented by a possibility distribution over a temporal region. That temporal region specifies the acceptable time that can be taken to perform the task, and the possibility distribution represents the preference function over the temporal region. The sharing of limited and fuzzified preference functions between the PAs and CAs enables them to interpret their opponents' intentions through iterative negotiation and to resolve potential conflicts, thereby accelerating convergence during negotiations.

4.2. Performance comparisons using different numbers of CAs

The AFCN model was compared with the CTR, MCNP and MIA approaches in terms of price and tardiness cost by varying the number of CAs from 9 to 21, with the number of PAs fixed at 20. The number of tasks per project was fixed at 5, and the number of candidate CAs to perform each task was fixed at 3.

As seen in Fig. 6, the MCNP yields the highest price relative to the other methods. The MIA protocol not only represents an overall improvement over the MCNP in terms of price but also yields a lower price as the number of CAs increases. However, the AFCN results in a lower price than either the MCNP or the MIA protocol. In the AFCN, the sharing of the preference function regarding the time required to perform a task helps the CAs avoid potential conflicts.

Fig. 7 reveals that the tardiness costs of all methods rapidly decrease as the total number of CAs increases. The tardiness cost decreases as the number of CAs approaches the number of PAs. The CTR, with its complete information sharing, yields the lowest tardiness cost. The proposed AFCN model results in a lower tardiness cost than either the MCNP or the MIA protocol.

Table 2
Computation times of the various investigated approaches for different numbers of CAs.

Total number of CAs	Computation time (unit of measure in seconds)					
	CTR	MCNP	MIA	AFCN (Competitive)	AFCN (Win-Win)	AFCN (Collaborative)
9	29,918	1.12	24.06	17.41	11.68	9.38
12	30,087	1.17	22.34	14.50	9.90	8.01
15	31,223	1.03	19.46	11.35	9.09	6.69
18	33,348	1.15	17.68	9.64	7.52	6.09
21	38,599	1.19	17.25	8.06	7.01	5.48

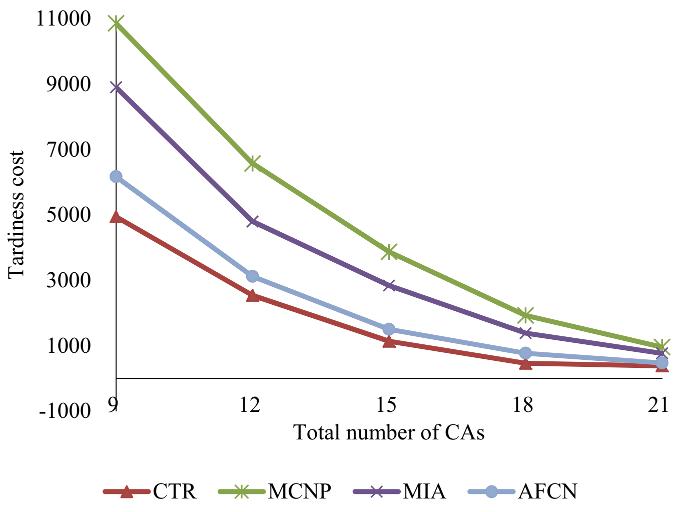

Fig. 7. Performance comparisons in terms of tardiness cost for the CTR, MCNP, MIA, and AFCN methods.

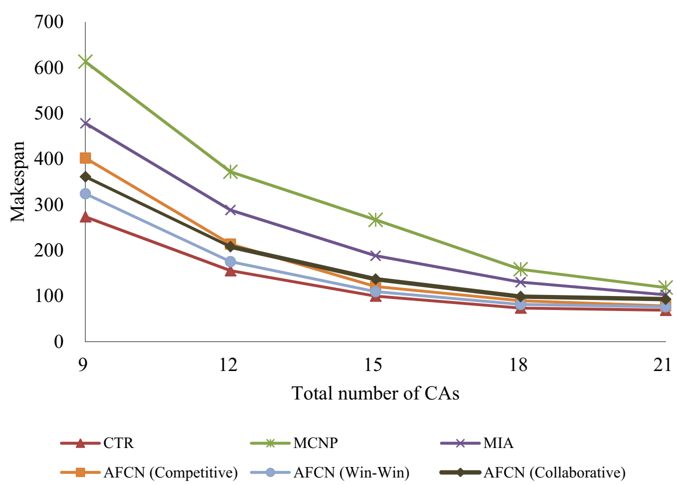

Fig. 8. Makespans for the solutions obtained using the CTR, MCNP, and MIA methods and using the AFCN model with three negotiation strategies.

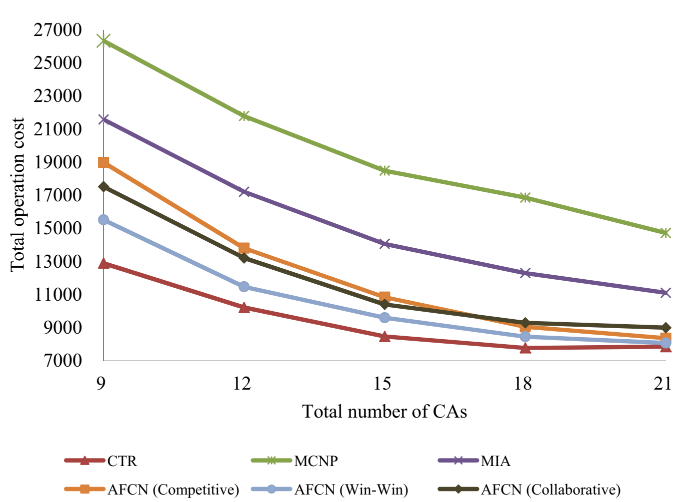

Fig. 9. Total operating costs for the solutions obtained using the CTR, MCNP, and MIA methods and using the AFCN model with three negotiation strategies.

Table 3
Number of failed instances and projects for different numbers of CAs.

Total number of CAs	CTR		MCNP		MIA		AFCN					
							Competitive		Win-Win		Collaborative	
	Number of failed instances	Number of failed projects	Number of failed instances	Number of failed projects	Number of failed instances	Number of failed projects	Number of failed instances	Number of failed projects	Number of failed instances	Number of failed projects	Number of failed instances	Number of failed projects
9	30.0	7.1	50.0	18.6	50.0	16.5	40.0	8.6	33.0	7.5	37.0	7.7
12	8.0	2.9	50.0	14.2	41.0	10.6	21.0	4.2	9.0	3.3	13.0	3.5
15	**0.0**	**0.0**	36.0	10.4	18.0	4.9	5.0	1.2	**0.0**	**0.0**	**0.0**	**0.0**
18	**0.0**	**0.0**	22.0	9.1	6.0	2.6	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
21	**0.0**	**0.0**	11.0	6.1	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

4.3. Performance comparisons using different negotiation strategies

The parameter settings for the experiments presented in this section were identical to those for the experiments reported in Section 4.2. The AFCN model has the flexibility to adopt different negotiation strategies, including competitive, win-win, and collaborative strategies. Figs. 8 and 9 represent the performances in terms of the makespan and total operating cost, respectively, for the CTR, MCNP, and MIA approaches as well as for the proposed AFCN model with each of these three different strategies. Regardless of the strategy adopted, the AFCN model outperforms the MCNP and the MIA protocol. Additionally, the win–win strategy exhibits the best performance among the three negotiation strategies because it tends to balance the benefits between the PAs and CAs.

Table 2 summarizes the computation times of the CTR, MCNP, and MIA methods and the AFCN model with the three different negotiation strategies. The CTR requires a much higher computation time because its time complexity is exponential. The MCNP is a single-shot negotiation protocol and thus has the lowest computation time. Regardless of the strategy adopted, the AFCN model requires less computational effort than the MIA protocol. Additionally, the collaborative strategy requires the fewest negotiation rounds to reach an agreement, whereas the competitive strategy requires the most computation time.

Table 3 reports the numbers of failed instances for the CTR, MCNP, and MIA methods and for the AFCN model with the three different negotiation strategies for various numbers of CAs. The results reveal that in terms of the numbers of failed instances and failed projects, the AFCN model exhibits comparable performance with respect to the CTR and outperforms the MCNP and the MIA protocol. Among the three different negotiation strategies, the win–win strategy results in the fewest failed instances and failed projects, whereas the competitive strategy demonstrates the worst performance.

5. Conclusions

This work proposed the AFCN model for solving the SCPSP through autonomous and interactive agent negotiations. Fuzzy-constraint-based modeling allows the SCPSP to be intuitively and effectively translated into a DFCSP. Instead of using a third-party agent as a moderator, the proposed AFCN model considers fuzzified preference functions in the generation of offers/counter-offers during negotiations. This method guides agents to restrict the solution space, avoid potential conflicts, and move toward a consistent agreement that benefits all agents. The presented experimental results demonstrate that with the sharing of limited, non-sensitive strategic information, the proposed AFCN model not only can obtain a schedule that is comparable to a near-optimal solution but is also time-efficient. This indicates that the proposed AFCN model is a practical and effective method of solving real-world SCPSPs.

Although the AFCN model appears promising, it can still be further improved. For example, the robustness and convergence of the proposed AFCN model in diverse supply chain environments remain to be investigated. Additionally, the proposed AFCN model can also be extended to other application domains, such as job shop scheduling, cloud resource management, transportation and logistics, and collaborative design.

Acknowledgment

This research was financially supported by the Ministry of Science and Technology, Taiwan (MOST 103-2221-E-155-029-MY2; MOST 104-3315-E-155-002; MOST 105-2218-E-155-010; MOST 104-2622-E-155-008-CC3).

References

[1] B.M. Beamon, Supply chain design and analysis: models and methods, Int. J. Prod. Econ. 55 (3) (1998) 281–294.
[2] A. Agnetis, N.G. Hall, D. Pacciarelli, Supply chain scheduling: sequence coordination, Discrete Appl. Math. 154 (15) (2006) 2044–2063.
[3] C. Moon, Y.H. Lee, C.S. Jeong, Y. Yun, Integrated process planning and scheduling in a supply chain, Comput. Ind. Eng. 54 (4) (2008) 1048–1061.
[4] T. Sawik, Coordinated supply chain scheduling, Int. J. Prod. Econ. 120 (2) (2009) 437–451.
[5] M. Wang, H. Wang, D. Vogel, K. Kumar, D.K.W. Chiu, Agent-based negotiation and decision making for dynamic supply chain formation, Eng. Appl. Artif. Intell. 22 (7) (2009) 1046–1055.
[6] J.S.K. Lau, G.Q. Huang, K.L. Mak, L. Liang, Agent-based modeling of supply chains for distributed scheduling: IEEE Trans, Syst. Man Cybern. Part A—Syst. Hum. 36 (5) (2006) 847–861.
[7] J. Spitter, C. Hurkens, A. Kok, J. Lenstra, E. Negenman, Linear programming models with planned lead times for supply chain operations planning, Eur. J. Oper. Res. 163 (3) (2005) 706–720.
[8] J. Mula, D. Peidro, M. Díaz-Madroñero, E. Vicens, Mathematical programming models for supply chain production and transport planning, Eur. J. Oper. Res. 204 (3) (2010) 377–390.
[9] D. Peidro, J. Mula, M. Jiménez, M. del Mar Botella, A fuzzy linear programming based approach for tactical supply chain planning in an uncertainty environment, Eur. J. Oper. Res. 205 (1) (2010) 65–80.
[10] D. Peidro, J. Mula, R. Poler, J.L. Verdegay, Fuzzy optimization for supply chain planning under supply, demand and process uncertainties, Fuzzy Sets Syst. 160 (18) (2009) 2640–2657.
[11] Y.H. Lee, C.S. Jeong, C. Moon, Advanced planning and scheduling with outsourcing in manufacturing supply chain, Comput. Ind. Eng. 43 (1) (2002) 351–374.
[12] D. Naso, M. Surico, B. Turchiano, U. Kaymak, Genetic algorithms for supply-chain scheduling: a case study in the distribution of ready-mixed concrete, Eur. J. Oper. Res. 177 (3) (2007) 2069–2099.
[13] A.D. Yimer, K. Demirli, A genetic approach to two-phase optimization of dynamic supply chain scheduling, Comput. Ind. Eng. 58 (3) (2010) 411–422.
[14] A. Mohtashami, M. Tavana, F.J. Santos-Arteaga, A. Fallahian-Najafabadi, A novel multi-objective meta-heuristic model for solving cross-docking scheduling problems, Appl. Soft Comput 31 (2015) 30–47.
[15] W. Shen, Distributed manufacturing scheduling using intelligent agents, IEEE Intell. Syst. 17 (1) (2002) 88–94.
[16] C. Li, J. Yu, X. Liao, Fuzzy tabu search for solving the assignment problem, Proceedings of the 2002 International Conference on In Communications Circuits and Systems and West Sino Expositions 2 (2002) 1151–1155.
[17] S. Kreipl, M. Pinedo, Planning and scheduling in supply chains: an overview of issues in practice, Prod. Oper. Manage. 13 (1) (2004) 77–92.
[18] J. Mula, D. Peidro, M. Díaz-Madroñero, E. Vicens, Mathematical programming models for supply chain production and transport planning, Eur. J. Oper. Res. 204 (3) (2010) 377–390.
[19] M. Ko, A. Tiwari, J. Mehnen, A review of soft computing applications in supply chain management, Appl. Soft Comput. 10 (3) (2010) 661–674.
[20] Y.J. Zheng, H.F. Ling, Emergency transportation planning in disaster relief supply chain management: a cooperative fuzzy optimization approach, Soft Comput. 17 (7) (2013) 1301–1314.
[21] T. Calosso, M. Cantamessa, M. Gualano, Negotiation support for make-to-order operations in business-to-business electronic commerce, Robot. Comput.—Integr. Manuf. 20 (5) (2004) 405–416.
[22] J. Jiao, X. You, A. Kumar, An agent-based framework for collaborative negotiation in the global manufacturing supply chain network, Robot. Comput.—Integr. Manuf. 22 (3) (2006) 239–255.
[23] F.R. Lin, H.C. Kuo, S.M. Lin, The enhancement of solving the distributed constraint satisfaction problem for cooperative supply chains using multi-agent systems, Decis. Support Syst. 45 (4) (2008) 795–810.
[24] A. Pan, S.Y.S. Leung, K.L. Moon, K.W. Yeung, Optimal reorder decision-making in the agent-based apparel supply chain, Expert Syst. Appl. 36 (4) (2009) 8571–8581.
[25] H.S. Kim, J.H. Cho, Supply chain formation using agent negotiation, Decis. Support Syst. 49 (1) (2010) 77–90.
[26] L.A. Santa-Eulalia, S. D'Amours, J.M. Frayret, Agent-based simulations for advanced supply chain planning and scheduling: the FAMASS methodological framework for requirements analysis, Int. J. Comput. Integr. Manuf. 25 (10) (2012) 963–980.
[27] M. Barbati, G. Bruno, A. Genovese, Applications of agent-based models for optimization problems: a literature review, Expert Syst. Appl. 39 (5) (2012) 6020–6028.
[28] K.M. Sim, Complex and concurrent negotiations for multiple interrelated e-markets, IEEE Trans. Cybern. 43 (1) (2013) 230–245.
[29] J.O. Gutierrez-Garcia, K.M. Sim, Agent-based cloud service composition, Appl. Intell. 38 (3) (2013) 436–464.
[30] K.M. Sim, Agent-based cloud computing, IEEE Trans. Serv. Comput. 5 (4) (2012) 567–577.

[31] Y.I. Lin, Y.W. Chou, J.Y. Shiau, C.H. Chu, Multi-agent negotiation based on price schedules algorithm for distributed collaborative design, J. Intell. Manuf. 24 (3) (2013) 545–557.
[32] C.J. Su, C.Y. Wu, JADE implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring, Appl. Sof Comput. 11 (1) (2011) 315–325.
[33] P. Leitão, F. Restivo, A holonic approach to dynamic manufacturing scheduling, Robot. Comput.—Integr. Manuf. 24 (5) (2008) 625–634.
[34] W. Xiang, H.P. Lee, Ant colony intelligence in multi-agent dynamic manufacturing scheduling, Eng. Appl. Artif. Intell. 21 (1) (2008) 73–85.
[35] V. Kaplanoğlu, Multi-agent based approach for single machine scheduling with sequence-dependent setup times and machine maintenance, Appl. Soft. Comput. 23 (2014) 165–179.
[36] B. Chen, H.H. Cheng, A review of the applications of agent technology in traffic and transportation systems, IEEE Trans. Intell. Transp. Syst. 11 (2) (2010) 485–497.
[37] R. Sprenger, L. Mönch, A decision support system for cooperative transportation planning: design, implementation, and performance assessment, Expert Syst. Appl. 41 (11) (2014) 5125–5138.
[38] T. Chen, R. Romanowski, Forecasting the productivity of a virtual enterprise by agent-based fuzzy collaborative intelligence-with facebook as an example, Appl. Soft Comput. 24 (2014) 511–521.
[39] G. Wang, T.N. Wong, X. Wang, A hybrid multi-agent negotiation protocol supporting agent mobility in virtual enterprises, Inf. Sci. 282 (2014) 1–14.
[40] Y. Chen, Y. Peng, T. Finin, Y. Labrou, S. Cost, B. Chu, R. Sun, B. Wilhelm, A negotiation-based multi-agent system for supply chain management, Proceedings of Workshop on Agent Based Decision-Support for Managing the Internet-Enabled Supply-Chain, at Third Conference on Autonomous Agents (1999).
[41] Q. Hu, A. Kumar, Z. Shuang, A bidding decision model in multiagent supply chain planning, Int. J. Prod. Res. 39 (15) (2001) 3291–3301.
[42] A. Karageorgos, N. Mehandjiev, G. Weichhart, A. Hämmerle, Agent-based optimisation of logistics and production planning, Eng. Appl. Artif. Intell. 16 (4) (2003) 335–348.
[43] P. Lou, Y.P. Chen, W. Ai, Study on multi-agent-based agile supply chain management, Int. J. Adv. Manuf. Technol. 23 (3–4) (2004) 197–203.
[44] J.X. Jiao, X. You, A. Kumar, An agent-based framework for collaborative negotiation in the global manufacturing supply chain network, Robot. Comput.-Integr. Manuf. 22 (3) (2006) 239–255.
[45] M. Owliya, M. Saadat, G.G. Jules, M. Goharian, R. Anane, Agent-based interaction protocols and topologies for manufacturing task allocation, IEEE Trans. Syst. Man Cybern. —Syst. 43 (1) (2013) 38–52.
[46] W.E. Walsh, M.P. Wellman, Decentralized supply chain formation: a market protocol and competitive equilibrium analysis, J. Artif. Intell. Res 19 (2003) 513–567.
[47] R.K. Dash, P. Vytelingum, A. Rogers, E. David, N.R. Jennings, Market-based task allocation mechanisms for limited-capacity suppliers, IEEE Trans. Syst. Man Cybern. Part A—Syst. Hum. 37 (3) (2007) 391–405.
[48] A. Gerber, C. Russ, M. Klusch, Supply web co-ordination by an agent-based trading network with integrated logistics services, Electron. Commer. Res. Appl. 2 (2) (2003) 133–146.
[49] M. Fan, J. Stallaert, A.B. Whinston, Decentralized mechanism design for supply chain organizations using an auction market, Inf. Syst. Res. 14 (1) (2003) 1–22.
[50] T. Kaihara, Multi-agent based supply chain modelling with dynamic environment, Int. J. Prod. Econ. 85 (2) (2003) 263–269.
[51] G.P. Cachon, M. Fisher, Supply chain inventory management and the value of shared information, Manage. Sci. 46 (8) (2000) 1032–1048.
[52] Y. Aviv, The effect of collaborative forecasting on supply chain performance, Manage. Sci. 47 (10) (2001) 1326–1343.
[53] P.J. Byrne, C. Heavey, The impact of information sharing and forecasting in capacitated industrial supply chains: a case study, Int. J. Prod. Econ. 103 (1) (2006) 420–437.
[54] K.R. Lai, Fuzzy constraint processing, in: PhD Thesis, NCSU, Raleigh, N.C, 1992.
[55] D. Dubois, H. Fargier, H. Prade, Propagation and satisfaction of flexible constraints, in: Fuzzy Sets, Neural Networks and Soft Computing, Van Nostrand Reinhold Company, New York, 1994, pp. 166–187.
[56] D. Dubois, H. Fargier, P. Fortemps, Fuzzy scheduling: modelling flexible constraints vs. coping with incomplete knowledge, Eur. J. Oper. Res. 147 (2) (2003) 231–252.
[57] K.R. Lai, M.W. Lin, Modeling agent negotiation via fuzzy constraints in e-business, Comput. Intell. 20 (4) (2004) 624–642.
[58] K.R. Lai, M. Lin, T. Yu, Learning opponent's beliefs via fuzzy constraint-directed approach to make effective agent negotiation, Appl. Intell. 33 (2) (2010) 232–246.
[59] C.H. Lan, S. Graf, K.R. Lai, Enrichment of peer assessment with agent negotiation, IEEE Trans. Learn. Technol. 4 (1) (2011) 35–46.
[60] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall Inc, New Jersey, 1995.
[61] H.J. Zimmerman, Fuzzy Set Theory and Its Applications, Kluwer Academic, Boston, MA, 1996.