Scheduling via Agent Negotiation : A Fuzzy Constraint-based Approach

K. Robert Lai
Yuan Ze University
Department of Computer Science & Engineering
135 Yuan-Tung Road, Chung-Li
Taiwan 32026

Menq-Wen Lin
Ching Yun University
Department of Information Management
229, Chien-Hsin Road, Chung-Li
Taiwan, 32026

Bo-Rue Kao
Yuan Ze University
Department of Computer Science & Engineering
135 Yuan-Tung Road, Chung-Li
Taiwan 32026

Abstract: This work presents a general framework for modeling a distributed scheduling problem via fuzzy constraint-based agent negotiation. Fuzzy constraints, in this way, are used not only to represent the requirements that jobs being scheduled must satisfy, but also to specify the possibilities prescribing to what extent the solutions are suitable for scheduling to rank the solutions. Furthermore, fuzzy constraint-based agent negotiation provides a systematic method to gradually relax the requirements to generate a local schedule, and then utilizes possibility functions to select an alternative that is subject to the others' acceptability. Each agent, who is in charge of different aspects of the scheduling problem, not only distributively solves its problems to maximize its local objectives, but also iteratively proposes its local schedules with other agents to gradually move toward a satisfactory and globally beneficial schedule. Experimental results suggest that the proposed approach is focused not only on the minimization of parameters such as makespan and tardiness, but also on the economical effects to maximize the profits of the enterprise.

Key–Words: Planning and Scheduling, Distributed Problem Solving, Multi-Agent Systems, Fuzzy Constraints.

1 Introduction

Scheduling is a process to allocate limited resources over time to satisfy a collection of jobs. But, real-world scheduling problems are often inherently distributed, that is, constraints and preferences of each entity in the problem might not be accessible to each other. Thus, two major methodologies, distributed problem solving (DPS) and multi-agent systems (MAS), have been proposed to model distributed scheduling problems.

In DPS approaches, the task of schedule is decomposed into a set of sub-problems and solved by the individual entities to achieve a global interest. However, the coordination strategy and information sharing in DPS are usually incorporated into an integral part of the system [1, 13]. Thus, it does not have the sophisticated reasoning required for social interaction and cannot deal with the autonomous nature of the components.

In contrast to DPS, MAS approaches solve the scheduling problem via autonomous agents interaction to maximize their own welfare. Based on some negotiation strategies, agents act from their individual perspectives to negotiate with others to obtain a compromise schedule. Yet they are not assumed to achieve the common goal cooperatively. Several negotiation models of MAS have been proposed for the scheduling problems [3, 10]. Among them, contract net protocol, a commonly used negotiation model, involves a process of task announcement, bidding, and awarding to establish a deal among agents [12]. Relying on this protocol, several bidding-based or auction-based approaches have demonstrated a flexible manner for resources selection and allocation [4, 8, 10]. While these negotiation models are proved to be effective and easy to implement, a drawback of these approaches is myopic, i.e. the choice of a resource is usually based on a local evaluation, and it could fail to support for more complex and less structured negotiations.

This paper presents a general framework for modeling a distributed scheduling problem via fuzzy constraint-based agent negotiation. Fuzzy constraints, in this way, are used not only to represent the requirements that jobs being scheduled must satisfy, but also to specify the possibilities prescribing to what extent the solutions are suitable for scheduling to rank the solutions [2, 5]. Furthermore, fuzzy constraint-based agent negotiation provides a systematic method to gradually relax the requirements to generate a proposal, and then utilizes possibility functions to select an alternative that is subject to the others' acceptability [6, 7, 9]. Each agent, on behalf of each entity involved in the scheduling, iteratively proposes its offers in order to gradually move toward a satisfactory schedule. The iterative nature of agent negotiation process forces the convergence between demand and offer. Thus, each agent, who is in charge of different aspects of the problem, not only distributively solves its problems to maximize its local objectives, but also works together with other agents to attain a globally beneficial schedule. Experimental results suggest that the proposed approach is focused not only on the minimization of parameters such as makespan and tardiness, but also on the economical effects to maximize the profits of the enterprise.

The remainder of this paper is organized as follows. Section 2 introduces the theoretical basis of modeling distributed scheduling as agent negotiation. Section 3 presents the negotiation process for obtaining the scheduling solutions. Section 4 demonstrates the effectiveness of the proposed approach followed by some conclusions in Section 5.

2 Distributed scheduling via Agent Negotiation

Planning a schedule among a set of entities can be modeled as agent negotiation in that finding a satisfactory scheduling solution in a distributed environment is the same as reaching an acceptable agreement in agent negotiation. Furthermore, fuzzy constraints have also been used to represent the requirements that jobs being scheduled must satisfy [2, 5]. Thus, a distributed scheduling problem can be formulated as a distributed fuzzy constraint satisfaction problem (DFCSP) and its graphical representation, a distributed fuzzy constraint network (DFCN) adapted from [7], can be defined as below.

Definition 1 ***(Distributed fuzzy constraint network) :*** *A distributed fuzzy constraint network* $(\mathcal{U}, \mathbf{X}, \mathbf{C})$ *can be defined as a set of fuzzy constraint networks* $\{\mathfrak{N}^1, ..., \mathfrak{N}^L\}$, $\mathfrak{N}^k = (\mathcal{U}^k, \mathbf{X}^k, \mathbf{C}^k)$ *being an FCN k, where*

- $\mathcal{U}^k$ *is a universe of discourse for an FCN k;*
- $\mathbf{X}^k$ *is a tuple of* n^k *non-recurring objects* $X^k_{1^k}, \ldots, X^k_{n^k}$*;*
- $\mathbf{C}^k$ *is a set of* $m^k \geq n^k$ *fuzzy constraints, which is the union of a set of internal fuzzy constraints* $\mathbf{C}^{k_i}$ *existing among objects in* $\mathbf{X}^k$ *and a set of external fuzzy constraints* $\mathbf{C}^{k_e}$ *referring to at least one object in* $\mathbf{X}^k$ *and another not in* $\mathbf{X}^k$*;*
- $\mathfrak{N}^k$ *is connected to other FCNs by* $\mathbf{C}^{k_e}$*;*
- $\mathcal{U}$ *is a universe of discourse;*
- $\mathbf{X} = \left(\cup_{k=1}^{L} \mathbf{X}^k\right)$ *is a tuple of all non-recurring objects;*
- $\mathbf{C} = \left(\cup_{k=1}^{L} \mathbf{C}^k\right)$ *is a set of all fuzzy constraints.*

That is, each individual fuzzy constraint network $\mathfrak{N}^k = (\mathcal{U}^k, \mathbf{X}^k, \mathbf{C}^k)$ in a DFCN can be the representation of job, resource, or some other forms of agents in a distributed scheduling problem. Then, the task of distributed scheduling is to attain a schedule that can satisfy all the fuzzy constraints in $\mathbf{C}$ simultaneously. The job and resource agents can be further defined as follows.

Definition 2 ***(Fuzzy constraint network for a job agent) :*** *A job agent* k_i^J*, which involves a set of activities required by job* J_i *and concerns with temporal, precedence, required resource, and problem-specific constraints, can be represented as a fuzzy constraint network* $\mathfrak{N}^{k_i^J} = (\mathcal{U}^{k_i^J}, \mathbf{X}^{k_i^J}, \mathbf{C}^{k_i^J})$*, where*

- $\mathbf{X}^{k_i^J}$ *is a tuple of the objects of job agent* k_i^J*, including start time* s_i *and end time* e_i *associated to job* J_i *as well as start time* s_{ij}*, end time* e_{ij}*, and required resource* r_{ij} *associated to activity* $a_{ij} \in J_i$*.*
- $\mathbf{C}^{k_i^J}$ *is a set of the fuzzy constraints of job agent* k_i^J*, in which*

  $\mathbf{C}_{tmp}$ *represents the temporal constraint which the job has to be started after the release date and finished before the deadline. For job* J_i*,* $\mathbf{C}_{tmp(i)}$ *implies start time* s_i *has to be later than release date* $\widetilde{R_i}$*, and end time* e_i *has to be earlier than due-date* $\widetilde{D_i}$*. Release date and due-date are often subject to preference and are modeled by fuzzy number(fuzzy duration/fuzzy constraint).*

$$\begin{aligned} \mathbf{C}_{tmp(i)} :& s_i \in [\widetilde{R_i}, +\infty), s_i = \min s_{ij}, \forall a_{ij} \in J_i, \\ & e_i \in (0, \widetilde{D_i}], e_i = \max e_{ij}, \forall a_{ij} \in J_i \end{aligned} \tag{1}$$

  $\mathbf{C}_{pre}$ *represents the precedence constraint which defines the preceding restriction between two activities.* $\mathbf{C}_{pre(ij->iq)}$ *implies that activity* a_{ij} *has to be performed before activity* a_{iq}*.*

$$\mathbf{C}_{pre(ij->iq)}\text{: } e_{ij} \leq s_{iq} \tag{2}$$

$\mathbf{C}_{req}$ represents the required resource constraint which defines the set of possible resources are required by an activity. $\mathbf{C}_{req(ij,H)}$ implies that activity a_{ij} can be performed by a set of alternative resources H_{ij}. For each candidate resource $R_h \in H_{ij}$ is held by resource agent k_h^R.

$$\mathbf{C}_{req(ij,H)}\text{: } r_{ij} = R_h,\ R_h \in H_{ij} \tag{3}$$

Definition 3 ***(Fuzzy constraint network for a resource agent) :*** *A resource agent k_h^R, which holds resource R_h and concerns with processing time, capacity, and problem-specific constraints, can be represented as a fuzzy constraint network $\mathfrak{N}^{k_h^R} = (\mathcal{U}^{k_h^R}, \mathbf{X}^{k_h^R}, \mathbf{C}^{k_h^R})$, where*

- *$\mathbf{X}^{k_h^R}$ is a tuple of the objects of resource agent k_h^R, including start time s_{hj}, processing time p_{hj}, and end time e_{hj} associated to activity a_{jh} which requires resource R_h.*
- *$\mathbf{C}^{k_h^R}$ is a set of the fuzzy constraints of resource agent k_h^R, in which*

  $\mathbf{C}_{pro}$ represents processing time constraint which defines the possible duration of processing time for an activity. Durations are determined by tuning the machine or allocating the amount of resources. $\mathbf{C}_{pro(hj)}$ implies that the processing time p_{hj} of activity a_{hj} are bounded by possible duration $\widetilde{P_{hj}}$ which is represented as a fuzzy number.

$$\mathbf{C}_{pre(hj)}\text{: } e_{hj} = s_{hj} + p_{hj},\ p_{hj} \in \widetilde{P_{hj}} \tag{4}$$

  $\mathbf{C}_{cap}$ represents the capacity constraint which limits the available capacity of resource over time. $\mathbf{C}_{cap(hj,hq)}$ implies that the processing times of activities a_{hj} and a_{hq} which are performed on the resource R_h, cannot overlap on times.

$$\mathbf{C}_{cap(hj,hq)}\text{: } s_{hj} \geq e_{hq} \text{ or } s_{hq} \geq e_{hj} \tag{5}$$

When planning a schedule, both the job agents and resource agents govern activity by maintaining the consistency of inter-constraints $C_{jr(ijh)}$, which requires that activity a_{ijh}, performed by a resource agent k_h^R, has to start and finish within a specified time window $[s_{ij}, e_{ij}]$ assigned by a job agent k_i^J, where

$$\mathbf{C}_{jr(ijh)}\text{: } s_{hj} \geq s_{ij} \text{ and } e_{hj} \leq e_{ij} \tag{6}$$

For evaluating and find more satisfactory schedules, the aggregated satisfaction value of the local schedule is defined as follows.

Definition 4 *Given an local schedule $\mathbf{U}$ involving a number of objects $(x_1, \ldots, x_n)$, the aggregated satisfaction value of an local schedule $\mathbf{U}$ to agent k, denoted by $\Psi^k(\mathbf{U})$, can be defined as a function of the values of satisfaction with the objects as follows:*

$$\Psi^k(\mathbf{U}) = \frac{1}{n}\sum_{j=1}^{n} \mu_{C_j^k}(x_j), \tag{7}$$

where $\mu_{C_j^k}(\cdot)$ is the satisfaction degree of the constraint C_j^k of agent k over objects j.

$\Psi^k(\mathbf{U})$ can be viewed as a constraint among the objects to represent the preference over the combination of objects, and determine whether to accept or reschedule conflicting events. Thus, $\Psi^k(\mathbf{U})$ is used by agent k to make trade-offs among objects for local schedules, and also regarded as an objective function to measure the quality of the local schedule $\mathbf{U}$ produced by the agent k.

For each job and resource agent, which is modeling as fuzzy constraint satisfaction problem (FCSP), the local schedule $\mathbf{U}$ is sequencing of activities with minimizing violation degrees of constraint satisfying. For solving FCSP to generate the feasible local schedules $\mathfrak{B}_{\mathbf{u}}^k$ for each agent, a branch and bound algorithm is adopted. Pruning will occur when constraints violated degree of partial sequencing $\mathbf{u}$ is lower than the α-cut. And only the local schedule $\mathbf{U}$ which satisfaction degree is greater than or equal to threshold Ψ_{th}^k should be selected into $\mathfrak{B}_{\mathbf{u}}^k$.

However, maintaining the consistency of activities by job agents may incur constraint violations for resource agents and vice-versa. Thus, negotiation mechanism is employed to resolve the conflicts among the agents. Yet, how does the agent negotiate with other agents to decide its local scheduling solution to reach an agreement that benefits all agents with a high satisfaction degree of fuzzy constraints, and move toward the deal more quickly?

To that end, the negotiation strategies are adopted by agents to determine the negotiation process in scheduling. These strategies determine how agents evaluate and generate local schedules to reach an agreement that is most in their self-interest or perform global goals. Agents exchange local schedules throughout the negotiation according to their own negotiation strategies. Whenever an local schedule is not acceptable by other agents, they make counter-offers by making concessions or by finding new alternatives to move toward an agreement. Hence, a concession strategy is presented, and a trade-off strategy is proposed to find alternatives.

In a scheduling process, agents employ the concession strategy to compromise their private schedules which are movable. Agents attempt to entice one another into agreement by manipulating the ranges associated with a given constraint in a scheduling problem. Hence, the set of feasible concession scheduling proposals for agent k at a threshold α_i^k is defined as follows.

Definition 5 ***(Set of feasible concession scheduling proposals)*** *: Given the latest scheduling offer* $\mathbf{u}$ *and a threshold* α_i^k *of agent* k*, the set of feasible concession scheduling proposals at the threshold* α_i^k *for the next offer of agent k, denoted by* ${}_{\alpha_i^k}\mathfrak{C}_{\mathbf{u}}^k$*, can be defined as*

$$ {}_{\alpha_i^k}\mathfrak{C}_{\mathbf{u}}^k = \left\{ \mathbf{v} \mid \left(\mu_{\mathbf{C}^k}(\mathbf{v}) \geq \alpha_i^k \right) \wedge \left(\Psi^k(\mathbf{v}) = \Psi^k(\mathbf{u}) - r \right) \right\}, \tag{8} $$

where r is the concession value.

The agent's concession value r for its next offer may be determined from the agent's mental state and the opponent's responsive state.

Besides, agents employ the trade-off strategy to reschedule the private schedules without reducing satisfactions. Agents attempt to entice one another into agreement by reconciling their constraints. An alternative solution, which is above or equal to a certain threshold, is proposed to the opponent from a certain solution space. Hence, the set of feasible trade-off scheduling proposals is defined as follows.

Definition 6 ***(Set of feasible trade-off scheduling proposals)*** *: Given the latest scheduling offer* $\mathbf{u}$ *and a threshold* α_i^k *of agent* k*, the set of feasible trade-off scheduling proposals at the threshold* α_i^k *for the alternatives of agent k, denoted by* ${}_{\alpha_i^k}\mathfrak{T}_{\mathbf{u}}^k$*, is defined as*

$$ {}_{\alpha_i^k}\mathfrak{T}_{\mathbf{u}}^k = \left\{ \mathbf{v} \mid \left(\mu_{\mathbf{C}^k}(\mathbf{v}) \geq \alpha_i^k \right) \wedge \left(\Psi^k(\mathbf{v}) \geq \Psi^k(\mathbf{u}) \right) \right\}. $$

A normalized Euclidean distance can be applied in establishing a trade-off strategy to measure the similarity between alternatives, and thus generate the best possible scheduling offer. This function tends to distinguish options whose satisfaction values are relatively close. Hence, a similarity function is defined as follows.

Definition 7 ***(Similarity function)*** *: Assuming that* $\mathbf{U}' = (\mathbf{u}'_1, \dots, \mathbf{u}'_n)$ *is the set of offers proposed by* n *other agents, and* $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_n)$ *is a feasible trade-off scheduling proposal of agent* k *for* n *other agents, the similarity function between* $\mathbf{V}$ *and* $\mathbf{U}'$ *on the negotiated issues for agent k, denoted by* $\Theta^k(\mathbf{V}, \mathbf{U}')$*, is defined as*

$$ \begin{aligned} &\Theta^k(\mathbf{V}, \mathbf{U}') \\ &= 1 - \frac{1}{n} \sum_{j=1}^{n} \left(\frac{1}{m} \left(\sum_{i=1}^{m} \left(\mu_{C_i^k}(\mathbf{v}_j) - \mu_{C_i^k}(\mathbf{u}'_j) + p_{C_i^k}(\mathbf{u}'_j) \right)^2 \right)^{\frac{1}{2}} \right), \end{aligned} \tag{9} $$

where m is the number of fuzzy constraints of agent k on issues, $\mu_{C_i^k}(\mathbf{v}_j)$ and $\mu_{C_i^k}(\mathbf{u}'_j)$ denote the satisfaction degree of the i^{th} (weighted) fuzzy constraint associated with the $\mathbf{v}_j$ and the $\mathbf{u}'_j$ for agent k to agent j, and $p_{C_i^k}(\mathbf{u}'_j)$ denotes the penalty from the i^{th} dissatisfied (weighted) fuzzy constraint associated with the offer $\mathbf{u}'_j$ made by agent k.

For each feasible trade-off scheduling proposal $\mathbf{v}$ of an agent, a fuzzy similarity between any $\mathbf{v}$ and the scheduling offer $\mathbf{u}'$ proposed by the opponent can be defined as a fuzzy set in which the membership grade of any particular $\mathbf{v}$ represents the similarity between $\mathbf{v}$ and $\mathbf{u}'$. Hence, the expected trade-off scheduling proposal $\mathbf{U}^*$ that benefits all parties can be defined as follows.

Definition 8 ***(Expected trade-off scheduling proposal)*** *: Assuming that agent k proposes a scheduling offer $\mathbf{U}$ to its opponents, and that the opponents subsequently proposes a set of scheduling counter-offer $\mathbf{U}'$ to agent k, the expected trade-off scheduling proposal $\mathbf{U}^*$ for the next scheduling offer by agent k is defined as*

$$ \mathbf{U}^* = \arg_{\mathbf{V}} \left(\max_{\mathbf{v} \in {}_{\alpha_i^k}\mathfrak{B}_{\mathbf{u}}^k} \Theta^k(\mathbf{V}, \mathbf{U}') \right), \tag{10} $$

where α_i^k is the highest possible threshold such that ${}_{\alpha_i^k}\mathfrak{B}_{\mathbf{u}}^k \neq \{\}$ and $\Theta^k(\mathbf{V}, \mathbf{U}') > \Theta^k(\mathbf{U}, \mathbf{U}')$.

The constraint $\Theta^k(\mathbf{V}, \mathbf{U}') > \Theta^k(\mathbf{U}, \mathbf{U}')$ is used to ensure that the next scheduling solution is better than the previous solution. Thus, based on the fuzzy similarity, an agent can use a trade-off strategy to generate a scheduling proposal that may benefit all parties without lowering the agent's requirements. Thus, by trade-off negotiation in a scheduling problem, agents can reallocate their initially assigned resources whenever timing of the jobs is undesirable.

Different combinations of strategies can be applied to cooperative or competitive situations. Hence, the trade-off strategy and/or concession strategy can be further meshed and ordered into a meta strategy $\mathfrak{M}$ over the whole scenario of negotiation.

3 Negotiation Process

A solution of distributed scheduling can be obtained via fuzzy constraint-based agent negotiation by maintaining the satisfiability of both inter-agent and intra-agent constraints. Agents take turns to propose local schedules to explore potential global schedules, thereby moving the negotiation toward a consensus.

```
01  i = 1; Ψ^k_th = 1.0; α^k_i = 1.0; Deal ← False; Failure ← False;
02  repeat
03    if Receive "Tell(U',K',k)" then
04      if Ψ^k(U') ≥ Ψ^k(U) and μ_{C^k}(U') ≥ α^k_i then
05        Deal ← True;
06      else
07        𝔐 ← PS^k_n;
08        while(True)
09          if _{α^k_i}𝔅^k_u ≠ {} then
10            U* ← Sel_offer(_{α^k_i}𝔅^k_u, 𝔐);
11            if U* ≠ {} then
12              exit; end if;
13          else
14            if Chk_com(𝔐) = True then
15              Ψ^k_th = Ψ^k_th − r;
16              _{α^k_i}𝔅^k_u = LocalSch(Ψ^k_th, α^k_i, U', ∅); end if;
17            if Chk_tra(𝔐) = True then
18              _{α^k_i}𝔅^k_u = LocalSch(Ψ^k_th, α^k_i, U', ∅); end if;
19            if _{α^k_i}𝔅^k_u = {} then
20              α^k_i ← α^k_{i+1};
21              if α^k_i < δ^k then
22                Failure ← True;
23                exit; end if; end if;
24        end while;
25        if Ψ^k(U') ≥ Ψ^k(U*) and μ_{C^k}(U') ≥ α^k_i then
26          Deal ← True;
27        else
28          Tell(U*,K,K');end if;end if;
29  until Deal = True or Failure = True;
```

Figure 1: Agent behavior for scheduling.

The process of each agent's behavior for scheduling is shown in Fig. 1.

Given the local schedule (time interval of activities) $\mathbf{U}' = \{\mathbf{u}_{k^1}, ..., \mathbf{u}_{k^J}\}$ from agents K', each agent k find the solution concurrently and independently for obtaining the feasible solution. Using Definition 8, a local schedule $\mathbf{U}^* = \{\mathbf{u}^*_{k^1}, ..., \mathbf{u}^*_{k^J}\}$ would be selected from the feasible schedules $\mathfrak{B}^k_{\mathbf{u}}$ and proposed to the corresponding agents K' (in line 10). To ensure that the next local schedule solution $\mathbf{U}^*$ is better than the previous solution $\mathbf{U}$ for gradually converge, the constraint $\Theta^k(\mathbf{U}', \mathbf{U}^*) > \Theta^k(\mathbf{U}', \mathbf{U})$ has to be satisfied. If no solution found (in lines 13 to 23), agent k will relax the constraint to the next acceptable threshold Ψ^k_{i+1} to create a new feasible solution space $\mathfrak{B}^k_{\mathbf{u}}$ ($\mathfrak{B}^k_{\mathbf{u}} = \mathfrak{C}^k_{\mathbf{u}}$, in lines 14 to 16) by concession strategy (Definition 5); or will create a new alternative solution space $\mathfrak{B}^k_{\mathbf{u}}$ ($\mathfrak{B}^k_{\mathbf{u}} = \mathfrak{T}^k_{\mathbf{u}}$, in lines 17 and 18) by trade-off strategy (Definition 6). Solution space $\mathfrak{B}^k_{\mathbf{u}}$ is obtained from $LocalSch$ which adopt a branch and bound algorithm. Strategies will be decided along the meta strategy $\mathfrak{M}$ of agent k (in line 7). If agent k faces no feasible proposal that matches the expected satisfaction value at the threshold α^k_i, with the capability of self-relaxation, the agent lowers its threshold of acceptability to the next threshold α^k_{i+1} until it generates an expected offer $\mathbf{U}^*$ or the threshold is less than δ^k (in line 19 or 23) in which case the negotiation fails and terminates.

4 Experiments

In what follows, we conduct several experiments to compare the effectiveness of our model with contract net protocol-based negotiation (CNP), extended contract net protocol negotiation (ECNP), and a well-known priority rule, longest processing time (LPT) [4, 11]. The manufacturing problem composed of five shops, each equipped with predefined resource and processing time is deterministic. The number of orders (jobs) is varied from 5 to 15 to examine the effects of the conflicts which arise between the demands of orders and the production capacities in the resources. Results are averaged over 100 different randomly generated data sets in which consist of the requirements from orders, including the unit price and delivery date. For simplicity, all agents employ a fixed concession strategy with 0.1 urgency value.

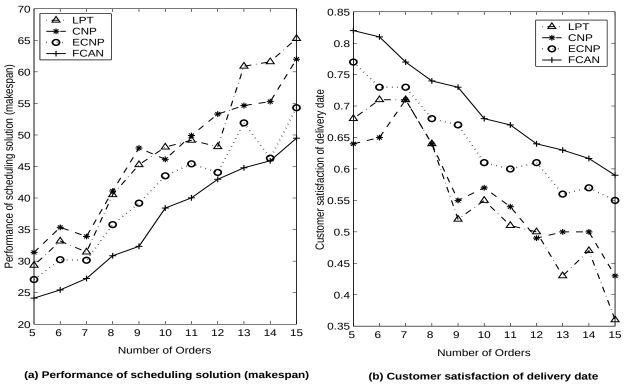
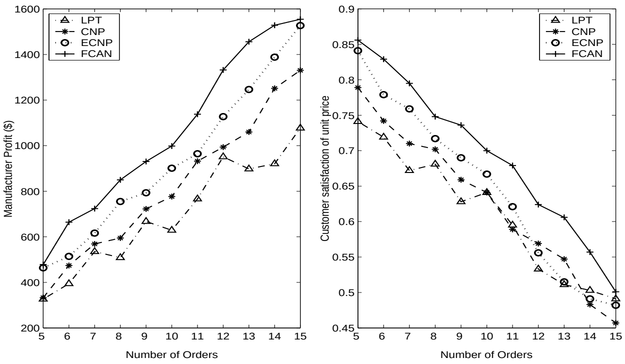


Figure 2: Performance comparison in makespan and customer satisfaction of delivery date.

In the experiment, the proposed approach along with LPT, CNP, and ECNP approaches are evaluated by the scheduling performance and customer satisfaction. We first evaluate the minimize makespan and maximize customer satisfaction of delivery date when the cost is limited, and the results are shown in Fig. 2. On the other hand, Figure 3 illustrates the results of an experiment to balance the profit and customer satisfaction when minimizing the production cost. Both in Figures 2 and 3 show that LPT and CNP have inferior performance and higher disturbance in all criteria when number of orders (demand conflicts) is increasing. Through iterative bidding, ECNP is more aware about resource contention and performs better than LPT and CNP. However, these approaches with local decision cannot guarantee the overall system perfor-



(a) Manufacturer Profit (b) Customer satisfaction of unit price

Figure 3: Comparison in manufacturer profit and customer satisfaction of unit price.

mance. On the other hand, the results demonstrate that our approach not only yields a shorter makespan (as shown in Fig. 2 (a)), also respects the tardiness of orders to improve the customer satisfaction in delivery date (as indicated in Fig. 2 (b)). Furthermore, it also show that the proposed approach is more profitable (as indicated in Fig. 3 (a)), and maintains customer satisfaction in unit price as higher as possible (as indicated in Fig. 3 (b)). In other words, through interactive negotiation, the proposed approach can reflect the differences in the constraint violations to other agents. Meanwhile, by evaluating the similarity of schedule and demands, it also provides a guideline for generating counteroffer to improve the convergence and solution quality.

5 Conclusions

This paper has presented a novel framework to planning a schedule via fuzzy constraint-based agent negotiation. The gradual relaxation and evaluation method with iterative negotiation process enables participants in distributed scheduling to progressively move toward a globally satisfactory schedule. The experiments have been utilized to demonstrate the effectiveness of our model. While the proposed model yielded some promising results, considerable work remains to be done, such as designing a learning model, applying to other forms of planning/scheduling problems, and studying coherence of negotiation strategies in various scheduling problems.

References:

[1] F. T. Chan, S. J. Zhang, and P. Li, Modelling of integrated, distributed and cooperative process planning system using an agent-based approach, *Proc. Inst. Mech. Eng., Part B—J. Eng. Manuf.*, vol.215(B10), 2001, pp.1437-1451.

[2] D. Dubois, H. Fragier, and Philippe Fortemps, Fuzzy scheduling: Modeling flexible constraints vs. coping with incomplete knowledge, *European Journal of Operational Research*, vol.147, 2003, pp.231-252.

[3] P. Gu, S. Balasubramanian, and D. H. Norrie, Bidding-based process planning and scheduling in a multi-agent system, *Comput. Ind. Eng.*, vol.32(2), 1997, pp.477-496

[4] N. Krothapalli, and A. Deshmukh, Design of negotiation protocols for multi-agent manufacturing systems, *Int. J. Prod. Res.*, vol.37, 1999, pp.1601-1624.

[5] K. R. Lai, *Fuzzy Constraint Processing*. Ph.D. thesis, NCSU, Raleigh, N. C., 1992.

[6] K. R. Lai, and M. W. Lin, Agent negotiation as fuzzy constraint processing, *FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, vol.2(12-17), 2002, pp.1021-1026

[7] K. R. Lai, and M. W. Lin, Modeling Agent Negotiation via Fuzzy Constraints in e-Business, *Computational Intelligence*, vol.20(4), 2004, pp.624-642

[8] Y. Lee, and S. R. Kumara, and K. Chatterjee, Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism, *J. Intell. Manuf.*, vol.14(5), 2003, pp.471-484

[9] Xudong. Luo, R. Jennings. Nicholas, Nigel Shadbolt, Ho-fung Leung, and Jimmy Ho-man Lee, A fuzzy constraint based model for bilateral multi-issue negotiations in semi-competitive environments, *Artificial Intelligence*, vol.148, 2003, pp.53-102

[10] P. McDonnell, Smith, S. G. Joshi, and S. R. T. Kumara, A cascading auction protocol as a framework for integrating process planning and heterarchical shop floor control, *Int. J. Flexible Manuf. Syst.*, vol.11(1), 1999, pp.37-62

[11] R. Macchiaroli, and S. Riemma, A Negotiation Scheme for Autonomous. Agents in Job Shop Scheduling, *Int. J. Comput. Integr. Manuf.*, vol.15(3), 2002, pp.222-232

[12] R. G. Smith, The contract net protocol: High-level communication and control in a distributed problem solver, *IEEE Trans. Comput.*, vol.C-29(12), 1980, pp.1104-1113

[13] L. Wang, and W. Shen, DPP: An agent-based approach for distributed process planning, *J. Intell. Manuf.*, vol.14(5), 2003, pp.429-440