Multi-agent Negotiation for Distributed Production Scheduling Problems

Lei Duan
Department of Mechanical and Industrial Engineering
University of Toronto, Toronto, Canada
lduan@mie.utoronto.ca

Mustafa K. Doğru
Alcatel-Lucent Bell Labs
Blanchardstown Industrial Park, Dublin 15, Ireland
mustafa.dogru@alcatel-lucent.com

Ulaş Özen
Alcatel-Lucent Bell Labs
Blanchardstown Industrial Park, Dublin 15, Ireland
ulas.ozen@alcatel-lucent.com

J. Christopher Beck
Department of Mechanical and Industrial Engineering
University of Toronto, Toronto, Canada
jcb@mie.utoronto.ca

ABSTRACT

We tackle the challenge of applying automated negotiation to multiple self-interested agents with local but linked combinatorial optimization problems. Using distributed scheduling problems in the context of supply chain management, we propose two negotiation strategies for making concessions in a joint search space of agreements. The first strategy concedes on utility, an approach commonly used in the negotiation literature; the second strategy concedes in a metric space while maximizing an agent's local utility. Experimental results show that, on small local problem instances, the metric-space negotiation strategy outperforms its utility-based counterpart on both agreement quality and computational effort. This paper presents one of the first studies of applying automated negotiation to self-interested agents each with a local combinatorial optimization problem.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; K.4.4 [**Computers and Society**]: Electronic Commerce

General Terms

Algorithm, Experimentation, Management

Keywords

negotiation, multiagent planning and scheduling, combinatorial optimization, distributed problem solving

1. INTRODUCTION

Multi-agent negotiation is widely used in resolving conflicts and distributing profits among different participants [12]. However, there has been little work in literature on negotiations among agents who have complex utility functions based on local combinatorial optimization problems. In a supplier-manufacturer relationship, the agents frequently negotiate on the delivery schedules, i.e., timing and quantities, for a variety of components and products. These schedules form an integral part of an agent's local optimization problem which typically has a combinatorial nature. Making decisions on these problems can be complex but critical to the overall efficiency of the entire supply chain, which has long been held as a key objective in supply chain coordination [4]. As a concrete example, consider a computer manufacturer (e.g., Dell) and its supply chain including its suppliers and customers. The manufacturer purchases components (e.g., CPUs, RAM, and hard drives) from various suppliers, assembles them according to different product configurations, and sells them to customers worldwide. Although optimization of production and inventory decisions in large corporations are supported by software tools [7], negotiation of delivery schedules between a manufacturer and a supplier often relies on human interactions. The goal of this research is to apply automated negotiation to achieve better overall efficiency of the supply chain. This will free human planners from lengthy communications as well as help them create value with high-quality agreements.

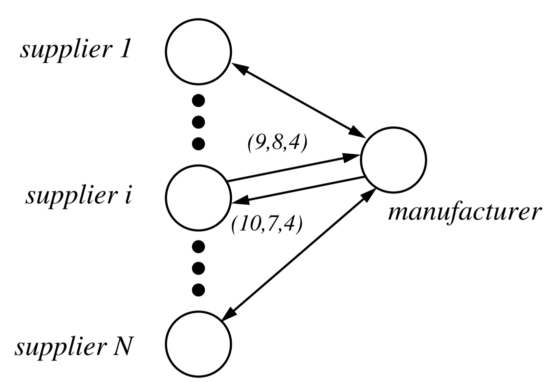

Figure 1: A supply chain consisting of one manufacturer and N suppliers.

In this paper, we investigate a multi-agent supply chain consisting of one manufacturer and multiple suppliers (Figure 1). The manufacturer solves a production scheduling problem to determine production quantities in each period in order to satisfy customer demand (e.g., 10 units in the first period, 7 in the second, and so on). The objective of the manufacturer is to maximize its profit subject to both local and external capacity constraints. The manufacturing of a product requires N different components, each from a different supplier. Each supplier has a similar scheduling problem to determine the number of components to be produced in each period based on the manufacturer's requirements as well as its local resource capacity. Similarly, every supplier seeks to maximize its profit.

In order to establish a delivery schedule for each of the components required, the manufacturer negotiates with each supplier since no product can be assembled on schedule if there is a disagreement on the component delivery schedule. To support automated negotiation, we use the negotiation protocol of Lai & Sycara [10]. To make concessions in a shared search space of agreements for our distributed production scheduling problems, we propose two negotiation strategies. The first strategy concedes on an agent's utility, an approach widely used in the literature; the second strategy concedes in a metric space while maximizing an agent's local utility. To assess the quality of agreements produced by the two negotiation strategies, we implement a centralized solver that generates optimal solutions corresponding to the best possible agreement that agents can achieve through negotiation. Experimental results on small local problem instances show that the metric-space strategy obtains agreements closer to the best ones than its utility-based counterpart and is overall more computationally efficient.

The rest of the paper is organized as follows. In Section 2, we present related work on multi-agent negotiations with a special attention of their applications to distributed scheduling. In Section 3, we present our negotiation protocol and the two negotiation strategies, followed by definitions and models of the production scheduling problems in Section 4. Experimental results are shown in Section 5. Finally, we conclude in Section 6.

2. RELATED WORK

In the negotiation framework of Ito et al. [8], each agent uniformly samples from its utility space to generate bids in a joint search space of agreements in which issues are interdependent and utility functions are nonlinear. These bids are then sent to a mediator which selects the winning proposal that maximizes the social welfare of all the participating agents. Although this approach significantly outperforms previous methods on agreement quality, it does require a mediator for completing the negotiation. In a supply chain, however, it may not be very cost-effective to coordinate distributed decision making through a mediator.

Turning to negotiation on scheduling problems, in [15], Zhang and Lesser study the multi-linked negotiation in which an agent negotiates with others in completing multiple tasks. These tasks contain operations with precedence constraints among them. Reasoning on these precedence relations based on a partial order schedule proposed by the authors enables an agent to manage the negotiations on the start and finish time of an operation much more effectively with other agents, resulting in more tasks being completed by their deadlines than ignoring the precedence relations and simply negotiating on the tasks separately. Although an agent uses heuristics to sequence its tasks locally, none of the agents has its utility based on solving a local combinatorial optimization problem.

Smith et al. [14] address the problem of a team of collaborative agents each of which executes part of a pre-computed global schedule in an uncertain environment. The objective of an agent is to maximize the total quality of the execution of the given global schedule. When deviations from the established schedule occur due to certain unexpected events, agents will locally reschedule their activities to restore feasibility or to maintain the quality of their partial solutions. Agents also collaborate on improving the quality of the global schedule through joint schedule changes. The key difference in our setting is that there is no global schedule existed before negotiation takes place. Each agent only knows its own production schedule. To make matters worse, our supply-chain agents are self-interested: each agent is solely interested in maximizing its own gain, even at the cost of other agents' interest. Therefore, conflicts are not only caused by uncertainties but also by each agent maximizing the objective function of its own problem disregarding others, a phenomenon known as incentive conflict [4]. The conflict needs to be resolved by agents negotiating with each other aiming to find a common solution to one another's local problem.

Crawford and Veloso [5] introduce the Semi-Cooperative Extended Incremental Multiagent Agreement Problem with Preferences in which an agent's utility is based on its own preferences discounted by the negotiation round (an exchange of an offer and a counter offer). In their agreement problem, variables arrive over time, and the assignment must be agreed by all the parties involved. Although an agent's preferences on variable assignment are private, the domain of the variables are public knowledge (e.g., all the possible time slots between 9AM and 5 PM in a meeting scheduling). As utility is discounted by time, an agent can learn from history and predict the round in which a value will be offered by other agents. Thus using this learned information, for example, by proposing offers in the descending order of estimated round, will help an agent obtain better utility in an agreement. In contrast, in our case, each agent has a combinatorial optimization problem, and the set of common values grows exponentially, making it impossible to propose every solution in a particular order based on learning.

Lastly, a closely related work is Burke et al. [3] in which distributed problem solving takes place among a manufacturer and several suppliers each with a production scheduling problem, a setting similar to ours. Agents form a component delivery schedule using the ADOPT algorithm [11]. The ADOPT algorithm in this context has several drawbacks. First, it does not truly realize a negotiation in which all of the agents can propose offers. The algorithm only enables the manufacturer to do so while the suppliers just revise their schedules based on the manufacturer's proposal, thus essentially blocking them from bargaining. Second, only one agent, corresponding to the root node in a depth-first search tree in ADOPT, can generate the optimal agreement and end the negotiation whereas in our negotiation protocol, every agent can exit a negotiation by accepting an offer. Lastly, the privacy of the agents in ADOPT is severely compromised as they need to communicate the values of their utility functions during negotiation: in our protocol, agents never disclose any information other than offers.

3. NEGOTIATION PROTOCOL AND STRATEGIES

In our supply chain, a manufacturer negotiates with N suppliers to establish a component delivery schedule with each supplier (see Figure 1). The manufacturer initiates the negotiation by offering to each supplier a schedule, e.g., to supplier i ($i \in [1, \ldots, N]$), 10 units for the first period, 7 for the second, and 4 for the third. Each supplier then individually evaluates the offer to decide whether to accept or reject it. In case of acceptance, the negotiation ends between the supplier and the manufacturer with the manufacturer's offer as the agreement; in case of rejection, the negotiation continues with the supplier counter offering another schedule, e.g., $(9, 8, 4)$ from supplier i back to the manufacturer. The manufacturer receives response (counter offers or confirmations of acceptance) from all the suppliers before another round of proposing offers to those suppliers who have not agreed. This exchange of offers and counter offers is repeated until either the manufacturer has reached an agreement with every supplier or a pre-specified maximum number of rounds has passed, where a round is defined as an exchange of the offers from the manufacturer to those suppliers who have not agreed in the previous round and the counter offers or confirmations of acceptance from those suppliers (Figure 2).

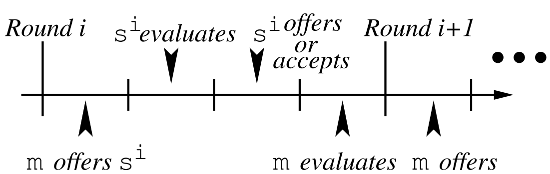

Figure 2: The sequence of events in round r. m represents the manufacturer while s^i denotes a supplier, i, who has not agreed in round $r-1$.

3.1 A Utility-based Strategy

For each agent, the manufacturer or a supplier, the decision of whether to accept or to reject an offer is controlled by a *negotiation strategy* [9]. A utility-based negotiation strategy requires a reservation utility as a threshold of acceptance for each round: if an offer is no worse than the current reservation utility, the offer will be accepted; otherwise, a counter offer will be made. We assume that an agent's utility is discounted by time and an outcome of agreement is better than no-agreement. Thus, similar to other works [1, 10], we use a time-dependent concession function [6] to calculate a reservation utility:

$$U(r) = U(0) - (U(0) - U(R)) \left(\frac{r}{R}\right)^{1/\beta} \qquad (1)$$

where R is the index of the final round, β is the rate of concession, and $U(r)$ is the reservation utility in round r ($r = 0, 1, \ldots R$). Through a series of reservation utilities, $\{U(r)\}$, an agent starts with its best utility, $U(0)$, and gradually concedes to its worst utility, $U(R)$, in the final round. Without loss of generality, we describe a utility-based negotiation strategy from the manufacturer's perspective. The manufacturer will accept offers from the suppliers if its utility is no worse than the reservation utility in a particular round. Otherwise, reject them and generate counter-offers with a utility equal to the reservation utility.

Before this utility-based strategy can be applied, two issues must be dealt with: one is to set the worst utility, $U(R)$, and the other is to find a solution that yields an optimization objective value of $U(r)$ in each round. Since establishing the worst utility prior to negotiation can be difficult, we allow it to be dynamically updated, an idea similar to An et al [1]. Specifically, in the beginning, $U(R)$ is set to be the local utility of the first offer(s) from the other agent(s). In the subsequent rounds, if the most recent offer(s) from the other agent(s) renders a utility worse than $U(R)$, then $U(R)$ is updated to the worse utility. Updating an agent's worst utility has the effect of increasing the concession step size while still guaranteeing that the agent will accept an offer with its worst observed utility in the last round. To tackle the second issue, we apply a heuristic proposing method [10] which would propose a counter-offer with a utility no less than $U(r)$ and with the maximum distance from the current offers of the other agents minimized. To measure the distance between two offers, we use Manhattan distance in a discrete space. In the rest of the paper, this strategy will be referred to as strategy U, for "utility".

3.2 A Metric-Space Strategy

Strategy U is based on calculating the utility that will be conceded to and then finding a solution of that utility as close as possible to the other agent's current offer. From an optimization perspective, an agent reformulates its local optimization model with a constraint placed on the utility function (no less than a reservation utility) while changing the objective to minimize the maximum distance between the most recent offer and the counter offer being searched for. We can invert this approach by placing a constraint on the distance: rather than giving up utility, an agent can explicitly move closer to an agreement while maximizing its utility. The time-dependent concession function we use in the metric-space strategy is similar to (1):

$$D(r) = D(r-1)\left[1 - \left(\frac{r}{R}\right)^{1/\beta}\right] \qquad (2)$$

where R and β are as defined in (1), and $D(r)$ is the distance threshold in round r ($r = 1, \ldots, R$). $D(0)$ is the maximum distance between the first offer-counter offer pair, corresponding to the two agents' best utilities, respectively. Different from (1), there is no need for an agent to explicitly set the worst utility.

We define a metric-space strategy (referred to as M hereafter, for "metric") as follows. In the initial round ($r = 0$), find the solution with the best utility (the same as in strategy U). In any subsequent round, find the solution that maximizes an agent's utility in a region bounded by $D(r)$. If this distance-constrained solution is no better than the offer from the other agent, then accept the offer and end the negotiation. Otherwise, reject the offer and counter offer the best solution bounded by $D(r)$.

4. THE PROBLEM DEFINITIONS AND MODELS

Consider the supply chain consisting of a manufacturer and N suppliers. The manufacturer sells a *product*, which is produced using N different *components*, each from a different supplier. Each agent determines its production schedule, the number of products/components to be manufactured

d_t	the customer demand in period t
M_t^m	the product production capacity in period t
I_t^m	the inventory level of product at the end of t
I_t^{m-}	the backlog level of product at the end of t
J_t^{mi}	the inventory level of component i at the end of t
P	the unit selling price of product
p^i	the unit selling price of component i
s_t^m	the setup cost of product production in t
c_t^m	the unit product production cost in t
H_t	the unit holding cost of product in t
B_t	the unit backlogging cost of product in t
h_t^{mi}	the unit holding cost of component i in t
M_t^{si}	the production capacity of component i in t
I_t^{si}	the inventory level of component i at the end of t
s_t^{si}	the setup cost of component i in t
c_t^{si}	the unit production cost of component i in t
h_t^{si}	the unit holding cost of component i in t

Table 1: Notation for the production scheduling problem.

during each period (e.g., day or week) over a fixed time horizon and maximizes its profit. The production costs include a setup cost (fixed and independent of the quantity produced) and a unit production cost (variable in the quantity produced). In addition, a unit inventory holding cost is charged for each product/component carried in stock from one period to the next. The quantity of products (components) that can be produced in a period by the manufacturer (supplier) is constrained by available capacity. If the manufacturer cannot fulfill the customer demand in a period, it will backlog the unfulfilled demand and deliver in a later period while paying a penalty cost to compensate the customer for the late delivery. In this case, the manufacturer uses a *backlogging* model [13] for its local production scheduling problem. Such a problem often arises in the context of supply chain management [13]. It is commonly known as the lot sizing problem and is NP-hard [2]. In our example, the manufacturer and the suppliers need to optimize their own problems while negotiating to establish common delivery schedules between them. In the following, we formulate the manufacturer's and the supplier's models for evaluating offers and making counter offers with strategy U and M.

4.1 The Manufacturer's Models for Negotiation

The planning horizon is divided into periods of equal length. Let $t = 1, \ldots, T$ be the index for periods and T denote the last period in the horizon. We use the superscript m to denote parameters and variables belonging to the manufacturer. The parameters are defined in Table 1. Without loss of generality, we assume that assembling each product requires one unit of a different component, i ($i \in \{1, \ldots, N\}$), and consumes one unit of production capacity.

Let y_t^i be the quantity of component i delivered in period t. The decision variables are δ_t^m, a 0-or-1 variable indicating whether to have a production set-up in period t; x_t^m, the quantity of the product to be manufactured in period t; g_t, the quantity of the product to be delivered to customers in period t; and z_t^i, the quantity of component i to order from supplier i in period t. The mathematical programming model is formulated as follows.

Maximize u^m:

$$\sum_{t=1}^{T} P g_t - \sum_{i=1}^{N}\sum_{t=1}^{T} p^i z_t^i - (\sum_{t=1}^{T} s_t^m \delta_t^m + c_t^m x_t^m + H_t I_t^m + B_t I_t^{m-} + \sum_{i=1}^{N} h_t^{mi} J_t^{mi}) \quad (3)$$

Subject to:

$$x_t^m \leq \delta_t^m M_t^m \qquad t = 1, \ldots, T \quad (4)$$
$$I_t^m = I_{t-1}^m + x_t^m - g_t \qquad t = 1, \ldots, T \quad (5)$$
$$J_t^{mi} = J_{t-1}^{mi} + z_t^i - x_t^m \qquad t = 1, \ldots, T; i = 1, \ldots, N \quad (6)$$
$$I_t^{m-} = I_{t-1}^{m-} + (d_t - g_t) \qquad t = 1, \ldots, T \quad (7)$$
$$z_t^i = y_t^i \qquad t = 1, \ldots, T; i = 1, \ldots, N \quad (8)$$
$$\delta_t^m \in \{0, 1\}; g_t, x_t^m, z_t^i \geq 0 \qquad t = 1, \ldots, T; i = 1, \ldots, N \quad (9)$$

where I_0^m and J_0^{mi} are the respective levels of product and component inventory at the beginning of the time horizon, which are assumed to be zero without loss of generality.

Given the complete customer demand in the time horizon, $\{d_t\}$, the objective function (3) maximizes the profit: the revenue from product sales minus the total purchasing, setup, production, inventory holding, and backlogging costs. Constraint (4) ascertains that there is a production set-up ($\delta_t^m = 1$) if the quantity produced is positive, and also enforces the production capacity. Constraints (5) and (6) are inventory balance equations for the product and the components, respectively. Constraint (7) ensures that the backlog level at the end of period t is the backlog level at the end of period $t-1$ plus the difference between demand and amount delivered in period t.

Constraint (8) specifies that the ordering quantity of component i in a period is equal to the quantity delivered from supplier i. When the manufacturer initiates an negotiation, there is no component delivery schedule from any supplier to refer to. So the manufacturer assumes that each supplier can provide as many components as needed and optimizes the model without Constraint (8). The resulting component ordering schedules, $\{z_t\}^i$, yields the maximum profit for the manufacturer, i.e., $U^m(0)$ for function (1), and is used as the manufacturer's first offer to all the suppliers. For subsequent rounds in which the manufacturer evaluates supplier i's counter offer of a different schedule, Constraint (8) will be enforced. It will be also enforced when the manufacturer reaches an agreement with a supplier during negotiation. Finally, Constraint (9) specifies the domains of the variables. We note that there is always a feasible solution to the backlogging model, i.e., to backlog all customer orders with hefty loss on profit.

Given all the counter offers, $\{\{y_t\}^i(r)\}$, in round r, the model for the manufacturer to offer new schedules, $\{\{z_t\}^i(r+1)\}$, in round $r+1$ using strategy U is given below.

Minimize:

$$\max_i \sum_{t=1}^{T} |z_t^i(r+1) - y_t^i(r)| \quad (10)$$

Subject to:

$$U^m(r+1) \le \sum_{t=1}^{T} Pg_t - \sum_{i=1}^{N}\sum_{t=1}^{T} p^i z_t^i(r+1) - \left(\sum_{t=1}^{T} s_t^m \delta_t^m + c_t^m x_t^m + H_t I_t^m + B_t I_t^{m-} + \sum_{i=1}^{N} h_t^{mi} J_t^{mi}\right) \quad (11)$$

and (4), (5), (6), (7), (9)

where $U^m(r+1)$ is the manufacturer's reservation utility in round $r+1$ according to concession function (1). The manufacturer will reject $\{\{y_t\}^i(r)\}$ and counter offer $\{\{z_t\}^i(r+1)\}$ if $u^m\left(\{\{z_t\}^i(r+1)\}\right) > u^m\left(\{\{y_t\}^i(r)\}\right)$. Also, the manufacturer initially sets its worst utility, $U^m(R)$, on the suppliers' first offers, $\{\{y_t\}^i(0)\}$.

The model for the manufacturer to find new component delivery schedules, $\{\{z_t\}^i(r+1)\}$, using strategy M is as follows.

Maximize: (3)

Subject to:

$$\max_i \sum_{t=1}^{T} \left| z_t^i(r+1) - y_t^i(r) \right| \le D(r+1) \quad (12)$$

and (4), (5), (6), (7), (9)

where $D(r+1)$ is the manufacturer's distance threshold in round $r+1$ according to concession function (2). ($D(0) = \max_i \sum_{t=1}^{T} |z_t^i(0) - y_t^i(0)|$) The objective function remains maximizing the profit. Similar to strategy U, strategy M is also guaranteed to converge to an agreement since in the last round, the manufacturer's distance threshold is decreased to zero and it will accept the suppliers' offers from round $R-1$, which always make the manufacturer's local problem feasible. In the initial round ($r = 0$), the manufacturer offers its best schedule, $\{z_t\}^i(0)$, to supplier i since it has no supplier's offer to compare on the utility yet.

As can be seen, strategy M takes a converse approach from its utility-space counterpart in integrating external information into the local optimization model. In strategy M, a constraint is placed on the distance in metric space of the new counter offer(s) from the current offer(s) while the optimization function remains the maximization of local utility as in the original combinatorial optimization problem.

4.2 The Supplier's Models for Negotiation

The supplier, unlike the manufacturer, is assumed to have unlimited supply of raw materials for manufacturing the components. Further, we assume that the cost of the raw material is negligible, so it is not considered in the model. Like in the manufacturer's model, we use the superscript s to denote parameters (Table 1) and variables that belong to the supplier.

The decision variables are δ_t^{si}, a 0-or-1 variable indicating whether to have a production set-up in period t, x_t^{si}, the quantity of the component to be manufactured in period t, and y_t^i. The mathematical programming model is given below.

Maximize u^{si}:

$$\sum_{t=1}^{T} p^i y_t^i - \left(\sum_{t=1}^{T} s_t^{si}\delta_t^{si} + c_t^{si} x_t^{si} + h_t^{si} I_t^{si}\right) \quad (13)$$

Subject to:

$$x_t^{si} \le \delta_t^{si} M_t^{si} \quad t = 1,\ldots,T; i = 1,\ldots,N \quad (14)$$
$$I_t^{si} = I_{t-1}^{si} + x_t^{si} - y_t^i \quad t = 1,\ldots,T; i = 1,\ldots,N \quad (15)$$
$$z_t^i = y_t^i \quad t = 1,\ldots,T; i = 1,\ldots,N \quad (16)$$
$$x_t^{si} \ge 0, y_t^i \ge 0, \delta_t^{si} \in \{0,1\} \quad t = 1,\ldots,T; i = 1,\ldots,N \quad (17)$$

where I_0^{si} is the inventory level of component i at the beginning of the planning horizon and is assumed to be zero. The objective function and the constraints resemble their counterparts in the manufacturer's model.

Strategy U requires that a supplier determines its best and worst utilities initially in round 0. After the manufacturer offers its first component delivery schedule, $\{z_t\}^i(0)$, the supplier solves (13)–(17) and sets its worst utility based on this offer: $U^{si}(R) = u^{si}\left(\{z_t\}^i(0)\right)$. (In case of infeasibility, we assign $U^{si}(R) = 0$.) For the calculation of its best utility, the supplier does not regard the manufacturer's delivery request as a set of constraints in which the specified quantity must be delivered in each period, but rather as a sequence of demand for which the total amount may be fulfilled within the planning horizon. Therefore, the supplier drops Constraint (16) and adds the following constraint

$$\sum_{t=1}^{T} y_t^i \le \sum_{t=1}^{T} z_t^i \quad (18)$$

to its model. The resulting schedule, $\{y_t\}^i(0)$, yields the best utility for supplier i, $U^{si}(0)$.

The model for supplier i to counter offer $\{y_t\}^i(r)$ in round r using strategy U is as follows.

Minimize:

$$\sum_{t=1}^{T} |z_t^i(r) - y_t^i(r)| \quad (19)$$

Subject to:

$$U^{si}(r) \le \sum_{t=1}^{T} p^i y_t^i(r) - \left(\sum_{t=1}^{T} s_t^{si}\delta_t^{si} + c_t^{si} x_t^{si} + h_t^{si} I_t^{si}\right) \quad (20)$$

and (14), (15), (17), (18)

where $U^{si}(r)$ is supplier i's reservation utility in round r according to concession function (1). The new objective function (19) minimizes the difference between the current offer on component delivery, $\{z_t\}^i(r)$, and a new proposal on delivery, $\{y_t\}^i(r)$. The original objective function (13) is revised as Constraint (20) according to strategy U. In round 0, the supplier will counter offer $\{y_t\}^i(0)$, its best schedule, if $u^{si}\left(\{y_t\}^i(0)\right) > u^{si}\left(\{z_t\}^i(0)\right)$.

The model for supplier i to counter offer $\{y_t\}^i(r)$ in round r using strategy M is given below.

Maximize: (13)

Subject to:

$$\sum_{t=1}^{T} |y_t^i(r) - z_t^i(r)| \le D^i(r) \quad (21)$$

and (14), (15), (17), (18)

where $D^i(r)$ is supplier i's distance threshold in round r according to concession function (2). Note that this model may fail to find a solution when the supplier has insufficient

p^i, the unit selling price of component i: 20
P, the unit selling price of the product: $2 \times \sum_i p_i$
s_t^m, the setup cost of product production: 5
c_t^m, the unit product production cost: 10
H_t, the unit holding cost of the product: $P \times 10\%$
B_t, the unit backlogging cost of the product: $P \times 40\%$
h_t^{mi}, the unit holding cost of component i: 2
s_t^{si}, the setup cost of component i (under capacity): 10
s_t^{si}, the setup cost of component i (over capacity): 100
c_t^{si}, the unit component production cost: 5
h_t^{si}, the unit holding cost of component i: 2

Table 2: Parameters for the production scheduling problems.

production capacity, in which case the supplier will counter offer the proposal from the previous round, $\{y_t\}^i(r-1)$, to the manufacturer. In the initial round ($r = 0$), the supplier, after finding its best schedule, $\{y_t\}^i(0)$, will accept $\{z_t\}^i(0)$ if $u^{si}(\{z_t\}^i(0)) \geq u^{si}(\{y_t\}^i(0))$, or counter offer $\{y_t\}^i(0)$ otherwise, since $\{y_t\}^i(0)$ is the best solution within $D^i(0)$, the distance between $\{z_t\}^i(0)$ and $\{y_t\}^i(0)$.

5. EXPERIMENTAL RESULTS

We now apply strategy U and M to the production scheduling problems in our one-manufacturer-N-supplier supply chain. We will compare these two approaches on criteria of convergence to agreement subject to a computation limit, agreement quality, and computational effort.

5.1 Parameter Settings

The index of the final round (R) and the concession rate (β) are set to 10 and 1 (a linear concession), respectively. Each strategy is restricted to one offer per round. The parameters for the production scheduling problems are set as follows.

The customer demand, d_t, $t = 1, \ldots, T$ ($d_0 = 0$), is uniformly sampled from $[\mu - \Delta, \mu + \Delta]$ with integer values, where μ represents the mean demand and Δ the maximum deviation from the mean. We set $\mu = 100$ and $\Delta = 30$.

The production capacity of an agent is set in a similar way to the customer demand: $[\mu^m - \Delta^m, \mu^m + \Delta^m]$ for the manufacturer and $[\mu^{si} - \Delta^{si}, \mu^{si} + \Delta^{si}]$ for supplier i. The manufacturer is set to have more than enough capacity to meet all the customer demand: $\mu^m = 500$, $\Delta^m = 50$. However, two different settings are experimented with for a supplier, "under capacity" in which a supplier, i, has barely sufficient capacity to supply the manufacturer: $\mu^{si} = 100$, $\Delta^{si} = 40$; and "over capacity" in which a supplier has more than enough capacity to satisfy the manufacturer's demand: $\mu^{si} = 400$, $\Delta^{si} = 40$. The number of suppliers with over capacity is controlled by a ratio, $\rho \in [0, 1]$: $\rho = 0$ indicates all under-capacity whereas $\rho = 1$ means all over-capacity. In the experiments, ρ is drawn from $[0, 0.25, 0.5, 0.75, 1]$.

The cost parameters are set in Table 2. The size of a problem is decided by the length of the planning horizon, T. In order for an agent to quickly solve its local problem, we chose a small value of T: $T = 5$. Lastly, the number of suppliers is drawn from $[2, 4, 16, 64]$.

5.2 Results and Analysis

Number of Suppliers	Strategy M	Strategy U
2	50	50
4	50	50
16	50	50
64	50	32

Table 3: The total number of times that a negotiation strategy converged to an agreement for different number of suppliers.

Experiments were conducted to test the performance of the two negotiation strategies on the underlying production scheduling problem. For each capacity ratio, ρ, 10 problem instances were randomly generated. A maximum CPU time of 12 hours is used to cut off a problem instance–exhausting this computational time is considered as failure to reach an agreement. The negotiation protocols were coded in C++, and the problem instances were solved by ILOG CPLEX 11.0. All the experiments were run on a Dual Core AMD 270 CPU with 4 GB main memory and Red Hat Enterprise Linux 4.

The two negotiation strategies are evaluated on the following three criteria. (i) The total number of times the negotiation converged to an agreement out of 50 problem instances for each number of suppliers across different ratios. (ii) The quality of the agreement. If an agreement yields for all the agents, the manufacturer plus all of the suppliers, a higher sum of their individual profits, then a strategy producing that agreement is deemed better. Also, we implemented a centralized solver for the whole supply chain, modelled in Appendix A, which yields the maximum possible profit that all of the agents can achieve together through negotiation. Thus, a negotiation strategy is better if it manages to reach an agreement with a smaller deviation from the best possible outcome. To measure the deviation on average, we calculate the mean relative error (MRE):

$$\text{MRE} = \frac{1}{n} \sum \frac{\xi^m + \sum_i \xi^{si} - \xi^{m+s}}{\xi^{m+s}}$$

where ξ^m, $\sum_i \xi^{si}$, and ξ^{m+s} are the final profit of the manufacturer, the sum of all the suppliers, and the centralized solver, respectively, and n is the number of agreements reached out of 10 problem instances for a given ratio. (iii) The computational effort, measured by the mean CPU time per negotiation round given a 12-hour computational time limit and a maximum number of 10 rounds.

Table 3 compares the two strategies on the total number of times a negotiation converged to an agreement for a given number of suppliers across different capacity ratios. Both strategies are able to scale with the number of suppliers when the number is small. However, when the number of suppliers becomes large (i.e., 64), strategy U can only finish about 60% of the negotiation instances by the computational time limit while strategy M is still able to finish all the negotiations with an agreement in time.

Figure 3 compares strategy M and U on the MRE for 2, 4, 16, and 64 suppliers, respectively. Strategy M consistently outperforms its utility-based counterpart across different capacity ratios. On the criterion of computational effort (Figure 4), strategy M is overall more computationally efficient than strategy U, especially when the number of suppliers reaches 64: strategy M spends about four orders of mag-

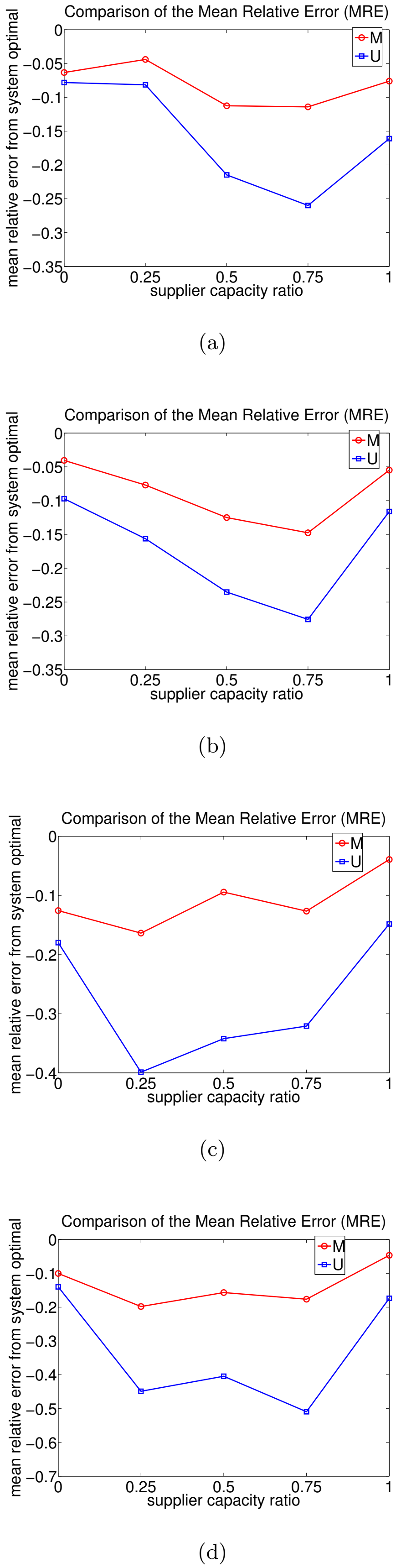

Figure 3: This set of figures shows the mean relative error on profit for 2 (3(a)), 4 (3(b)), 16 (3(c)), and 64 (3(d)) suppliers, respectively.

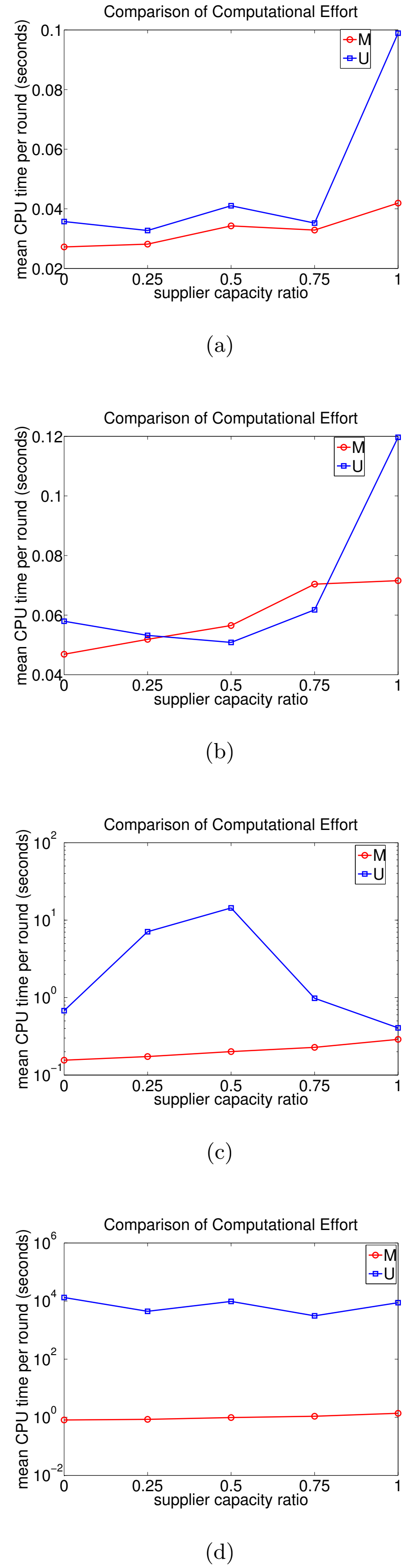

Figure 4: This set of figures shows the mean CPU time per negotiation round for 2 (4(a)), 4 (4(b)), 16 (4(c)), and 64 (4(d)) suppliers, respectively. Figure 4(c) and 4(d) are on $\log$ **scale.**

nitude less CPU time per round on average in finishing a negotiation.

6. CONCLUSION AND FUTURE WORK

In this paper, we tackle the challenge of applying automated negotiation to distributed production scheduling problems and make two main contributions: (1) the investigation of multi-agent negotiation in which each agent's utility depends on solving a local combinatorial optimization problem, and (2) the introduction of a novel concession strategy which constrains the distance in the metric space between successive offers while maximizing local utility. We experiment with two negotiation strategies. One strategy, based on a widely used approach in the negotiation literature, concedes on utility, and the other, the new one, takes a converse approach and concedes in a metric space while maximizing an agent's local utility. Experimental results show that, for small local problem instances, both strategies scale up to as many as 64 agents. When compared against each other, the metric-space strategy outperforms its utility-based counterpart on both agreement quality and computational effort.

Our next step is to apply the negotiation protocol with the strategies to a general supply network of negotiating agents each with a production scheduling problem and with more complex interactions among them. Our ultimate goal is to develop negotiation protocols and strategies for automated negotiation between software tools that solve combinatorial optimization problems.

7. ACKNOWLEDGMENTS

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada, the Canadian Foundation for Innovation, the Ontario Ministry of Research and Innovation, the Industrial Development Agency (IDA), Ireland, ILOG, S.A., and Microway, Inc.

8. REFERENCES

[1] B. An, V. Lesser, and K. M. Sim. Decommitment in multi-resource negotiation. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1553–1556, 2008.

[2] N. Brahimi, S. Dauzere-Peres, N. M. Najid, and A. Nordli. Single item lot sizing problems. *European Journal of Operational Research*, 168(1):1–16, 2006.

[3] D. A. Burke, K. N. Brown, M. Dogru, and B. Lowe. Supply chain coordination through distributed constraint optimisation. In *Proceedings of the 9th International Workshop on Distributed Constraint Reasoning*, 2007.

[4] G. P. Cachon. *Matching Supply with Demand: An Introduction to Operations Management.* McGraw-Hill/Irwin, 2008.

[5] E. Crawford and M. Veloso. Negotiation in semi-cooperative agreement problems. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 252–258, 2008.

[6] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *International Journal of Robotics and Autonomous Systems*, 24:159–182, 1998.

[7] B. Fleischmann and H. Meyr. *Supply Chain Management: Design, Coordination and Operation*, chapter 9: Planning Hierarchy, Modeling and Advanced Planning Systems. Elsevier, 2003.

[8] T. Ito, H. Hattori, and M. Klein. Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 1347–1352, 2007.

[9] S. Kraus. *Strategic Negotiation in Multiagent Environments.* MIT Press, 2001.

[10] G. Lai and K. Sycara. A generic framework for automated multi-attribute negotiation. *Group Decision and Negotiation*, 18(2):169–187, 2008.

[11] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. ADOPT: asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161:149–180, 2005.

[12] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, 2009.

[13] E. A. Silver, D. F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling.* John Wiley & Sons, 1998.

[14] S. F. Smith, A. Gallagher, T. Zimmerman, L. Barbulescu, and Z. Rubinstein. Distributed management of flexible times schedules. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 472–479, 2007.

[15] X. Zhang and V. Lesser. Multi-linked negotiation in multi-agent systems. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, 2002.

APPENDIX

A. THE CENTRALIZED SCHEDULING MODEL

The centralized model is formulated after incorporating all the parameters, variables, and constraints from all the manufacturer's and the suppliers' models. The decision variables are δ_t^m, x_t^m, g_t, δ_t^{si}, and x_t^{si}, as previously defined. The objective function is the sum of the objective functions of the manufacturer and all the suppliers. The mathematical programming model is formulated as follows.

Maximize:

$$\sum_{t=1}^{T} P g_t - (\sum_{t=1}^{T} s_t^m \delta_t^m + c_t^m x_t^m + H_t I_t^m + B_t I_t^{m-} + \sum_{i=1}^{N} (h_t^{mi} J_t^{mi} + h_t^{si} I_t^{si} + s_t^{si} \delta_t^{si} + c_t^{si} x_t^{si})) \qquad (22)$$

Subject to:

$$x_t^{si} = J_t^{mi} + I_t^{si} + x_t^m - J_{t-1}^{mi} - I_{t-1}^{si} \quad t = 1, \ldots, T; i = 1, \ldots, N \qquad (23)$$

$$x_t^m \geq 0, \delta_t^m \in \{0,1\}, g_t \geq 0, x_t^{si} \geq 0, \delta_t^{si} \in \{0,1\} \quad t = 1, \ldots, T; i = 1, \ldots, N \qquad (24)$$

and (4), (5), (7), (14)