Metric-Space Negotiation for Distributed Scheduling Problems

Student: Lei Duan[1] **Supervisor: J. Christopher Beck**[1] **Collaborators: Mustafa K. Doğru**[2], **Ulaş Özen**[2]
[1]Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Canada
{lduan,jcb}@mie.utoronto.ca
[2]Alcatel-Lucent Bell Labs, Blanchardstown Industrial Park, Dublin 15, Ireland
{mustafa.dogru,ulas.ozen}@alcatel-lucent.com

Abstract

We tackle the challenge of applying automated negotiation to self-interested agents with local but linked combinatorial optimization problems. Using distributed production scheduling problems in the context of supply chain management, we first present a negotiation protocol built on existing work in the multi-agent negotiation literature. Then, 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 this novel metric-space negotiation strategy outperforms its utility-based counterpart and is able to obtain close-to-optimal solutions. This paper presents the first study of applying automated negotiation to self-interested agents, each with a local combinatorial optimization problem.

1. Introduction

Multi-agent negotiation has played a key role in resolving conflicts and distributing profits among different participants. However, the academic interest on negotiation among agents who have complicated utility functions that are represented by local combinatorial optimization problems has been limited. In a typical supplier-manufacturer relationship in a supply chain, the agents negotiate on a frequent basis (e.g., daily or weekly) on delivery schedules (timing and quantities of replenishments). These schedules form an integral part of an agent's local optimization problem which generally has a combinatorial nature. Although optimization of production and inventory decisions in large corporations are supported by software tools, negotiation of material deliveries between the manufacturer and the suppliers still rely on human interaction. Automated negotiation can alleviate the burden on human planners and facilitate inter-system interactions for reaching higher-quality agreements.

In this paper, we investigate a two-agent supply chain consisting of a supplier and a manufacturer (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 local resource capacity constraints. The 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, the supplier seeks to maximize its profit.

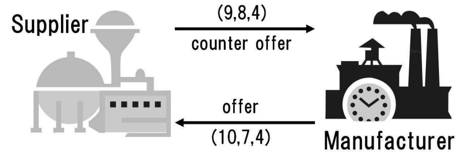

Figure 1: A two-agent supply chain consisting of a supplier and a manufacturer.

Since decision making is distributed, the manufacturer and the supplier negotiate to establish a common component delivery schedule. The negotiation protocol that we adopt is based on the framework of Lai and Sycara (2008). We introduce two negotiation strategies for making concessions in our distributed scheduling problems: one strategy, based on a method frequently used in the literature, concedes on an agent's local utility; the other concedes in a metric space while maximizing an agent's local utility. Experimental results show that this new metric-space strategy obtains better agreements overall compared to the utility-based approach and high-quality solutions compared to the optimal solutions when the whole supply chain is optimized centrally. The key contributions we make in this paper are: (1) The investigation of multi-agent negotiation in which each agent's utility depends on solving a local combinatorial optimization problem. (2) A novel concession strategy which constrains the distance in the metric space between successive offers while maximizing local utility.

2. Negotiation Protocol and Strategies

The negotiation protocol is known as the Rubinstein's alternating-offer game (Rubinstein 1982) and works as follows. The manufacturer initiates the negotiation by offering the first schedule to the supplier, e.g., 10 units for the first period, 7 for the second, and 4 for the third. The supplier then evaluates the offer to decide whether to accept or reject it. In case of acceptance, the negotiation ends with the manufacturer's offer as the agreement; in case of rejection, the negotiation continues with the supplier counter offering another schedule, e.g., $(9, 8, 4)$. This exchange of offer and counter offer is repeated until either an agreement or a prespecified maximum number of rounds is reached, where a round is defined as an exchange of an offer and a counter offer.

The decision of whether to accept or to reject an offer is controlled by a *negotiation strategy*. 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 use a time-dependent concession function (Faratin, Sierra, and Jennings 1998) 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 ($r = 0, \ldots, R$). 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. Thus, a utility-based negotiation strategy can be described as follows. An agent will accept an offer if its utility is no worse than the reservation utility for a particular round. Otherwise, reject it and generate a counter-offer that has a utility no worse than the reservation utility and is closest to the offer. In the rest of the paper, we will refer to this strategy as strategy U, for "utility".

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 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. Similar to (1), we introduce a time-dependent function for concession in a metric space:

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

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 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 set the worst utility.

We define a metric-space strategy (referred to as strategy M, 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, s, 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 s.

P	unit selling price of product
p	unit selling price of component
d_t	customer demand in period t
M_t^m	product production capacity in period t
I_t^m	inventory level of the product at the end of period t
J_t^m	inventory level of the component at the end of period t
s_t^m	setup cost of product production in period t
c_t^m	unit product production cost in period t
H_t	unit holding cost of the product in period t
h_t^m	unit holding cost of the component in period t

Table 1: Notation for the production scheduling problem.

3. The Production Scheduling Problem

In the supply chain, the manufacturer sells a *product*, which is produced using a certain type of *component* from the supplier. Each agent determines its production schedule, the number of products/components to be manufactured during each period over a fixed planning horizon and maximizes its own 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 abandon any portion of unmet demand in that period. Such a production scheduling problem often arises in the context of supply chain management. It is commonly known as the *lot sizing problem* and is NP-hard (Brahimi et al. 2006). The manufacturer and the supplier need to optimize their own problems while negotiating to establish a common delivery schedule. We formulate the manufacturer's models for evaluating offers and making counter offers with strategy U and M, respectively; the supplier's models can be formulated likewise.

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 each product uses one unit of the component and consumes one unit of production capacity. Let y_t be the quantity of component 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, the quantity of the component to order from the supplier in period t. The mathematical programming model is formulated as follows.

Maximize u^m:

$$\sum_{t=1}^{T} Pg_t - \sum_{t=1}^{T} pz_t - \sum_{t=1}^{T}(s_t^m\delta_t^m + c_t^m x_t^m + H_t I_t^m + h_t^m J_t^m) \quad (3)$$

Subject to:

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

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

Given the complete customer demand in the planning horizon, $\{d_t\}$, as an input, the objective function (3) maximizes the profit: the revenue from product sales minus the total purchasing, setup, production, and inventory holding 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 component, respectively. Constraint (7) ensures that quantity delivered does not exceed the customer demand. Constraint (8) specifies that the quantity of components ordered in a period is equal to the quantity that is delivered from the supplier. Finally, Constraint (9) specifies the domains of the variables.

When the manufacturer initiates a negotiation, there is no component delivery schedule from the supplier to refer to in Constraint (8). Thus, the manufacturer assumes that the supplier can provide as many components as needed and optimizes the model without Constraint (8). The resulting component ordering schedule, $\{z_t\}$, 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 the supplier. For subsequent rounds in which the manufacturer evaluates a supplier's counter offer, Constraint (8) is enforced.

Let $\{y_t\}(r)$ denote the supplier's counter offer in round r (the negotiation round number is shown in the parentheses). The manufacturer accepts supplier's offer $\{y_t\}(r)$ if solving (3)–(9) with $\{y_t\}(r)$ yields a profit greater than or equal to the threshold utility. Otherwise, the manufacturer rejects the offer and makes a counter offer. The model using strategy U is given below.

Minimize:

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

Subject to:

$$U^m(r+1) \le u^m$$
and (4), (5), (6), (7), (9)

where $U^m(r+1)$ is the manufacturer's reservation utility for round $r+1$ according to the concession function (1). Conceding in strategy U also requires $U^m(R)$ be determined. The manufacturer *initially* sets its worst utility on the supplier's first counter offer, $U^m(R) = u^m(\{y_t\}(0))$, provided that $\{z_t\}(0)$ is not accepted by the supplier. Then $U^m(R)$ is updated if the counter offer of the supplier yields a worse utility in any subsequent rounds.

The model for a new component delivery schedule using strategy M is as follows.

Maximize: (3)

Subject to:

$$\sum_{t=1}^{T} |z_t(r+1) - y_t(r)| \le D^m(r+1) \quad (11)$$
and (4), (5), (6), (7), (9)

where $D^m(r+1)$ is the manufacturer's distance threshold in round $r+1$ according to concession function (2). The objective function remains maximizing the profit while (11) constrains the counter offer to be within $D^m(r+1)$ distance from $\{y_t\}(r)$.

4. Experimental Results

The experiment is set up as follows. The index of the final round (R) and the concession rate (β) are set to 20 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 similar to the customer demand: $[\mu^m - \Delta^m, \mu^m + \Delta^m]$ for the manufacturer and $[\mu^s - \Delta^s, \mu^s + \Delta^s]$ for the supplier; $\mu^m = 110$, $\Delta^m = 50$, $\mu^s = 90$, $\Delta^s = 40$. The size of a problem, T, is set from 10 to 100 with an increment of 10. Other parameters are in Table 2. For each problem size, 20 problem instances are randomly generated. A maximum CPU time of 5 hours is used to cut off a problem instance. The negotiation protocols are coded in C++, and the problem instances are solved by ILOG CPLEX 11.0. All the experiments are run on a Dual Core AMD 270 CPU with 4 GB main memory and Red Hat Enterprise Linux 4.

The two negotiation strategies are compared on: (i) the total number of times the negotiation converged to an agreement out of 20 problem instances for each problem size and the total number of system-optimal agreements (an agreement is deemed system-optimal if the manufacturer and the supplier obtain a combined total profit equal to that found by a centralized solver, which incorporates all the parameters, variables, and constraints from both the manufacturer's and the supplier's models.) (ii) the mean relative error (MRE) of the sum of the profits of the manufacturer and supplier from the profit found by a centralized solver, defined as MRE $= (1/n)\sum(\xi^m + \xi^s - \xi^{m+s})/\xi^{m+s}$ where ξ^m, ξ^s, and ξ^{m+s} are the final profits of the manufacturer, the supplier, and the centralized solver, respectively, and n is the

Parameter	Value
P: unit selling price of the product	100
p: unit selling price of the component	20
s_t^m: setup cost of product production	10
c_t^m: unit product production cost	20
H_t: unit holding cost of the product	10
h_t^m: unit holding cost of the component	4
s_t^s: setup cost of component production	20
c_t^s: unit component production cost	10
h_t^s: unit holding cost of the component	4

Table 2: Parameters for the production scheduling problems.

number of agreements reached out of 20 problem instances for a given problem size, and (iii) the computational effort, measured by the mean CPU time per negotiation round in instances converged to an agreement given the 5-hour CPU time limit.

The advantage of strategy M is evident (Figure 2): it converges on far more instances than its utility-based counterpart. It also reaches the system-optimal on 17 agreements out of a total of 200 problem instances where strategy U fails to find a single optimal agreement. Moreover, strategy M outperforms U on the mean relative error. Compared to the centralized solver, strategy M is less than 1% from the system-optimal on average. On computational efficiency (the bottom graph in Figure 2), strategy M spends one to four orders of magnitude less CPU time per round on average in reaching an agreement.

5. Conclusion and Future Work

This paper presents the first result of applying an automated negotiation protocol to distributed scheduling problems in which each agent maximizes its own utility based on a local combinatorial optimization problem. We developed two negotiation strategies: one concedes on utility, and the other concedes in a metric space while maximizing an agent's local utility. Experimental results have shown that the new metric-space strategy outperforms its utility-based counterpart and is able to achieve agreements close to the system-optimal. The next step is to investigate how the two strategies can be applied to negotiations with more than two agents all of which must solve combinatorial optimization problems, and this work is underway.

References

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

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

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

Rubinstein, A. 1982. Perfect equilibrium in a bargaining model. *Econometrica* 50:97–110.

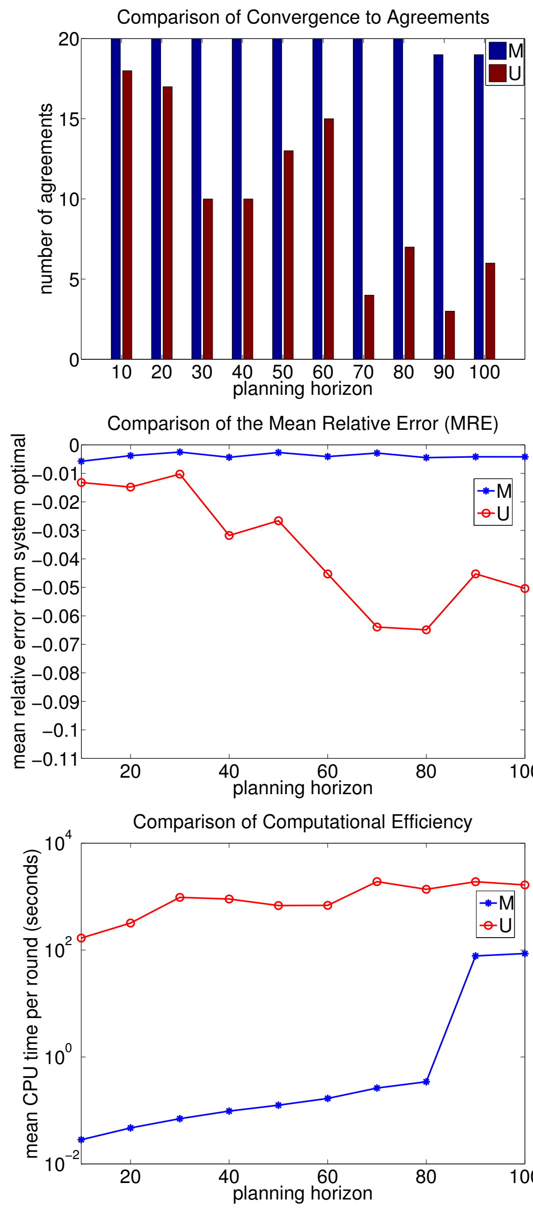

Figure 2: This set of figures shows, from the top to the bottom, (i) the total number of times that a negotiation strategy converged to an agreement out of 20 problem instances for each problem size, (ii) the mean relative error of the sum of the profits of the manufacturer and supplier from the system-optimal, and (iii) the mean CPU time per negotiation round in instances converged to an agreement ($\log$ scale).