Multi-agent Coalition via Autonomous Price Negotiation in a Real-Time Web Environment

Hoong Chuin LAU Wei Sian LIM
School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543 Tel: +65-8744589
lauhc@comp.nus.edu.sg limweisi@comp.nus.edu.sg

Abstract

In e-marketplaces, customers specify job requests in real-time and agents form coalitions to service them. This paper proposes a protocol for self-interested agents to negotiate prices in forming successful coalitions. We propose and experiment with two negotiation schemes: one allows information sharing while the other does not.

1. Introduction

Agent-mediated negotiation has received considerable attention in the field of electronic commerce and supply-chain management ([1],[5],[6]). Negotiations among agents help automate the trading procedure in e-markets.

In this paper, we study automated negotiation for coalition formation In our model, each job has a fixed price that the customer is willing to pay and not subjected to negotiation. We assume that each job is complex in that it has to be satisfied by more than one agent. Participating agents will form coalitions dynamically by specifying their capabilities and bid prices. A coalition is *feasible* iff the combined capabilities of participating agents match the job requirements and the total bid prices of agents must not exceed the job's offer price.

Previous works on coalition formation for task allocation assume that coalitions can always be formed. However when coalitions *cannot* be formed because the agents bid prices are not profitable, these jobs will never be serviced. In order to maximize the interest of both the customers as well as service providers, there is a need to efficiently negotiate price reductions among self-interested agents via a neutral party host so that coalitions are still profitable and jobs will be serviced.

In our model, customers submit their job requests dynamically to the host. The consolidated jobs are then broadcast to agents for them to submit their bids for jobs. The host runs a coalition-formation algorithm (such as [7]) to form coalitions that satisfy the jobs. For coalitions that are *infeasible*, the host initiates a negotiation process to efficiently allow each agent to reduce her bid price such that the coalitions will become feasible, as follows. The host computes drop prices and proposes them to agents. Upon receiving the drop price, each agent either accepts it or re-proposes a new bid price. Two negotiation schemes are proposed. In the first scheme, each agent publicizes its bidding behaviour to the host, and the decision to accept the drop price is based on other agents' aggregate behaviours. In the second scheme, each agent is oblivious of others' behaviour and thus relies on its own behaviour to decide if the drop price is acceptable. Experimental results show that the information sharing scheme is superior over the second, both qualitatively and quantitatively. Due to space constraints, the experimental results will not be discussed in this version of the paper.

In our current model, customers are not involved in negotiation and this will be considered for future works.

2. Related Works

In [2], the notion of a round in continuous double auctions was introduced. A history list of past transaction prices is used to compute the reference price that guides agents in their subsequent bidding behaviour. Each agent is associated with a utility function that describes its attitude towards risk. Our work makes use of their ideas of a history list and negotiation round.

Recently, [3] raised issues concerning real-time coalition formation among rational agents. It was pointed out that classical game-theoretic notions of coalition and respective negotiation algorithms are generally not applicable in dynamic settings such as the environment considered in this paper. [4] analysed a number of problems related to resource allocation and task distribution among self-motivated, rational and autonomous agents, each having its own utility functions. Their approach emphasizes the importance of the passage of time during negotiation, and applies to problems where the agreement involves all agents.

The problem we study here is very similar to the problem in [7], except that agents are self-motivated (i.e. they have individual utility functions) and can be persuaded to reduce prices according to their prescribed behaviors, and that there is no relationship between the utility gained and the resource amount utilized.

3. Preliminaries

In this section, we will define the notations followed by the intuition behind our proposed negotiation process.

3.1. Notations

The set of n agents is represented by $A = \{A_1, A_2, \ldots, A_n\}$. Without loss of generality, we assume that each agent A_i has exactly one capability b_i

that quantifies its ability to perform a specific task. Requests are denoted by m independent jobs $J = \{ j_1, j_2, \ldots, j_m \}$. Each job j_l consists of r distinct sub-jobs $S_l = \langle s_1^l, s_2^l, \ldots, s_r^l \rangle$. Each sub-job s_k requires exactly one capability b_k and is serviced by exactly one agent with that capability. When a new batch of jobs arrives, the coalition formation algorithm computes a coalition to service each job. Each coalition $C_j \subseteq A$ is made up of agents that have the capabilities to fulfill the requirements of sub-jobs in job *j*. For each job *j*, the system computes the offer price P_j. When agent *i* bids for job *j*, it specifies an initial bid price ρ_{ij}. To be profitable, the total bid price for all agents must not exceed the offer price of each job. This pricing constraint is thus stated as $P_j \le \sum_{\forall A_i \in C_j} \rho_{ij}$. Coalitions which violate the pricing constraint are termed *infeasible* and there is a need to get agents in an infeasible coalition to reduce their individual bid prices. For simplicity, since we are dealing with only one job at a time, the index *j* is dropped and all notations henceforth will refer to the current job being negotiated.

3.2. Negotiation Process

3.2.1. Negotiation Protocol The negotiation host tries to get agents to drop their respective bid prices by a proposed drop price. Each agent performs its own computation and either accepts or rejects this amount. In the latter case, it either re-proposes another bid price or simply rejects the host. This decision depends on the scheme used. When the host receives rejections or re-proposals, it re-computes the drop price based on the current state and the history of the negotiation process. Subsequently, the agent is informed of this new drop price and does its computation to decide the next step. This series of exchanges will iterate until the pricing constraint is satisfied or when a predetermined number of rounds has elapsed.

3.2.2. Negotiation Schemes The two negotiation schemes proposed are Group-rational and Selfish. For the first scheme, we assume that every agent shares their bidding behaviour with the host. Upon receiving the proposed drop price from the host, agents compute their expected revenues based on the aggregate behaviour of other agents except itself and decide if an agreement is possible or a re-proposal of the bid price is required. In the second scheme, agents do not share information and is assumed that each agent will act rationally using solely its behaviour curve with no other information available.

3.2.3. Negotiation Round A negotiation round r ($r \ge 1$) is defined as the period from the time the host computes and sends the proposed drop prices to the agents, to the time when the agents sends their replies back to the host. Henceforth, the initial bid price is denoted by ρ_i^0 and re-proposed bid prices (for Group-rational scheme) during each round will be denoted by ρ_i^r.

4. Negotiation Host

We discuss how the host computes the proposed drop prices based on negotiation history and the current state.

4.1. Negotiation History

The negotiation history h_i of agent *i* is the probability of a successful negotiation. To derive h_i, we need the number of participated negotiations thus far (n_i); and the number of successful negotiations thus far (s_i) of each agent. h_i is set to 0.5 if $n_i < l$ and s_i / n_i otherwise, where *l* is a large number (e.g. 1000) so as to accurately predict each agent's likeliness of a successful negotiation.

Definition 1. An agent *i* is said to be *likely* if $h_i \ge \mu$ and said to be *non-likely* otherwise. In this paper, we conveniently set $\mu = 0.5$.

Definition 2. An agent *i* is said to be *active* if it has yet to drop its bid price by the proposed amount and said to be *non-active* otherwise.

4.2. Current State Information

The current state information $D^r = \langle d_1^r, d_2^r, \ldots, d_{|C|}^r \rangle$ is computed at the start of each round and d_i^r denotes agent *i*'s confirmed drop price during round *r*. Thus $d_i^r = \rho_i^0 - \rho_i^{r-1}$ for active and likely agents. For each non-active but likely agent *i*, d_i^r is the difference of its initial bid price and the final bid price (after accepting the proposed drop price).

4.3. Proposed Drop Price

Let $\delta^r = \sum_{\forall d_i^r \in D^r} d_i^r$ and let $\varepsilon^r = \sum_{\forall A_i \in C} \rho_i^0 - P - \delta^r$ denote the amount in which the sum of all agents' bid prices is in excess of the job offer price during round *r*. Let π^r denote the sum of all active and likely agents' bid prices at during round *r*. The proposed drop price of each active and likely agent *i* at round *r* is given by $\Delta_i^r = \left\lceil \varepsilon^{r-1} \cdot \frac{\rho_i^{r-1}}{\pi^{r-1}} \right\rceil$.

Intuitively, the proposed drop price is calculated by distributing the excess amount, ε^r, proportionally among active and likely agents with respect to their previous rounds' bid prices. Note that when $\varepsilon^r \le 0$, Δ_i^r will be set to zero regardless of the value of π^r. If $\Delta_i^r = \Delta_i^{r-1}$ for all active and likely agents *i* at round *r*, the negotiation process will terminate and deemed as failed because the drop price does not decrease for the next round and the process will never converge to an agreement. Clearly, if $\Delta_i^r = 0$ for all active and likely agents, the negotiation is successful because the pricing constraint is satisfied.

Observe that this method is "conservative" because unlikely agents are excluded from the distribution of the excess amount since they are assumed to fail. This approach is also "optimistic" in the sense that inexperienced agents ($n_i < l$) are assigned a probability of 0.5, considered to be "likely" in agreeing to a negotiation.

5. Agents

This section discusses each agent's maximum bid price discount, its behaviour representation and the two negotiation schemes in detail.

5.1. Maximum Bid Price Discount

To negotiate price reductions among agents, we assume each agent has a range of prices to offer as a bid price for each sub-job. The initial bid price ρ_i^0 is the maximum value in this range of bid prices. To determine the minimum bid price, each agent is associated with a *maximum bid price discount fraction,* $\alpha_i \; (0 \le \alpha_i < 1)$. This value specifies the maximum discount off the initial bid price that the agent is willing to offer and is based the prevailing market rate and its individual policy. The minimum bid price, ρ_i^∞, is thus given by $(1-\alpha_i).\,\rho_i^0$. Hence the range of bid prices that an agent can quote for each sub-job ranges from ρ_i^∞ to ρ_i^0.

5.2. Agent Behaviour

Each agent *i* is associated with a probability function $\Pr_i$ that captures its bidding behaviour, where $\Pr_i(z)$, $0 \le z \le \alpha_i$, gives the probability (willingness) that agent *i* will offer *z* percent discount off its initial bid price. The expected value of $\Pr_i$ is denoted by EPr_i. In this paper, $\Pr_i$ is represented as a monotonically decreasing step function with t_i number of intervals. We will investigate experimentally the effectiveness of our protocol under different step functions.

5.3. Group-rational Scheme

After an active agent receives the proposed drop price from the host, it computes the expected revenue to decide if this drop price is acceptable.

Intuitively, the expected revenue of each agent relies on the current state and other agents' aggregate behaviour. To compute the expected revenue, we first compute the current round's *active job price* which is given by deducting the sum of confirmed drop prices (current state) and the sum of initial bid prices of non-active or non-likely agents from the job offer price. Next, we find the set of all combinations of each active and likely agent's bid price (except the current agent) whose sum is less than or equal to the active job price minus the current agent's minimum bid price. Since each of these combinations do not consider the current agent, it is possible to determine all possible bid prices of this agent for the current round by deducting the sum of bid prices in each combination from the active job price. The probabilities of the current agent offering these bid prices are equivalent to the probabilities of all other agents offering their respective bid prices in each combination. Finally the expected revenue of this agent is computed by summing over all possible bid prices and the normalized probabilities of them offering these bid prices.

Definition 3 will be used to generate all possible bid prices (restricted by each agent's range of bid prices) that agents are willing to transact (see the paragraph following it for more details on each variable).

Definition 3. Suppose $\beta^{\min}$ and $\beta^{\max}$ are two *k*-tuples of positive integers and *m*, $\omega_{\min}, \omega_{\max}$, *k* and *n* are integers. A $(\beta^{\min}, \beta^{\max}, m, \omega_{\min}, \omega_{\max})$– constrained ***k*-integer partition of *n*** is a *k*-tuple of positive integers $\lambda = (\lambda_1, \lambda_2, ..., \lambda_k)$, subject to the following constraints: (a) $\lambda_1 + \lambda_2 + ... + \lambda_k \le n$; (b) $\lambda - \beta^{\min} \ge (0,...,0)$ and $\beta^{\max} - \lambda > (0,...,0)$; and (c) $\omega_{\min} \le \left(m - \sum_{i=1}^{k} \lambda_i \right) \le \omega_{\max}$.

Let X_i^r be the random variable that denotes the revenue of agent *i* during *r*; C_i^r be the ordered set of active and likely agents except agent *i* during round *r*; $k = |C_i^r| - 1$; $C_i^r[t]$ be the t^{th} agent in C_i^r respectively; $\sigma^\infty = (\rho_1^\infty, ..., \rho_k^\infty)$ and $\sigma^r = (\rho_1^r, ..., \rho_k^r)$ be the tuples of minimum and current round bid prices of all agents in C_i^r; θ^r denote the sum of initial bid prices of non-active or non-likely agents during round *r*; $\xi^r = P - \delta^r - \theta^r$ denote the active job price during round *r*; $Q_i^r = \xi^r - \rho_i^\infty$; G_i^r be the set of $(\sigma^\infty, \sigma^{r-1}, \xi^{r-1}, \rho_i^\infty, \rho_i^0)$-constrained-*k*-integer partitions of Q_i^r (see Definition 3); *g* be an arbitrary tuple in G_i^r and g_t be the t^{th} element of *g*. The expected revenue of agent *i* at round *r* is:

$$\mathrm{E}[X_i^r] = \sum_{\forall g \in G_i^r} \left[\xi^r - \sum_{1 \le t \le k} g_t \right] \cdot \left[\frac{\prod_{1 \le t \le k} \Pr_a\left(\frac{g_t}{v_a}\right)}{\sum_{\forall g \in G_i^r} \prod_{1 \le t \le k} \Pr_a\left(\frac{g_t}{v_a}\right)} \right],$$

where $a = C_i^r[t]$. If $\mathrm{E}[X_i^r] = \infty$ because no partitions can be formed ($G_i^r = \varnothing$), the agent will reject the proposed drop price and set $\rho_i^r = \rho_i^{r-1}$. For each agent to compute $\mathrm{E}[X_i^r]$, there must be at least 1 other active and likely agent during that round. Otherwise, the negotiation fails.

Lemma 1. $\mathrm{E}[X_i^r]$ is monotonically decreasing over *r* if $G_i^r \neq \emptyset$.

***Proof*:** Omitted due to space constraints.

The negotiation process will converge, since the host proposed bid price (derived from the proposed drop price) will increase monotonically over rounds, while the agent expected revenue (or re-proposed bid price) as indicated by Lemma 1 will decrease monotonically over rounds.

It should be noted that enumerating integer partitions is a well-studied problem in combinatorics (see, for example [8]). Although it takes $O(n^k)$ time to enumerate all *k*-integer partition of *n*, we can drastically overcome the time complexity by generating a good enough subset of partitions for our purpose through scaling and dynamic pruning during enumeration.

5.4. Selfish Scheme

In this scheme, agents do not share information with one another and they have to rely on their respective expected behaviour (EPr_i) to decide whether to accept the proposed drop price Δ_i^r. To do this, each active and likely agent *i* computes $w_i^r = (\rho_i^0 - \Delta_i^r) / \rho_i^0$ which is the discount value assuming it accepts the proposed bid price. If $w_i^r \leq \mathrm{EPr_i}$, the proposed drop price is accepted. Otherwise the agent rejects the negotiation host and no new bid price is proposed.

6. The Negotiation Protocol

The detailed negotiation protocol for each scheme is explained below for an arbitrary round *r*. The following will be iterated until an agreement is reached or when a predetermined number of rounds has elapsed.

6.1. Protocol for Group-rational Scheme

Host:

i. Compute Δ_i^r for each active and likely agent *i*.

ii. If $(\Delta_i^r = 0)$ the negotiation is successful. Increase s_i and n_i by 1 for all agents.

iii. If $(\Delta_i^r = \Delta_i^{r-1})$ for all active and likely agents or there are less than 2 active and likely agents, the negotiation fails. Increase n_i by 1 for all agents.

iv. Send Δ_i^r to agent *i*, for all *i*.

For each active and likely agent *i*:

i. Compute $\mathrm{E}[X_i^r]$.

ii. If $\mathrm{E}[X_i^r] = \infty$, set $\rho_i^r = \rho_i^{r-1}$.

iii. If $\left(\rho_i^{r-1} - \mathrm{E}[X_i^r]\right) \geq \Delta_i^r$, accept Δ_i^r.

iv. Otherwise set $\rho_i^r = \mathrm{E}[X_i^r]$.

6.2. Protocol for Selfish Scheme

Host:

Same as above.

For each active and likely agent *i*:

i. Compute w_i^r.

ii. If $w_i^r \leq \mathrm{EP}r_i$, accept Δ_i^r. Otherwise reject Δ_i^r.

7. Conclusions

From this study, we conclude with two important (albeit intuitive) lessons on agent negotiation: (a) the importance of information sharing: agents who share information will be more profitable than those who do not since in the latter case, most fail to negotiate successfully to form coalitions, (b) the importance of co-operation in negotiating price reductions: as the number of agents who are willing to cooperate (likely agents) increases, the quality of the results also increases.

References

[1] A. Gerber, C. Russ, M. Klusch and I. Zinnikus. Holonic Agents for the Coordination of Supply Webs. *In Proc. of the Fifth International Conference on Autonomous Agents*, 2001.

[2] M. He, H. Leung and N. R. Jennings A Fuzzy Logic Based Bidding Strategy for Autonomous Agents in Continuous Double Auctions. *IEEE Transactions on Knowledge and Data Engineering*, 2003 (to appear).

[3] M. Klusch and A. Gerber, Issues of Dynamic Coalition Formation Among Rational Agents*. In Proc. of the Second International Conference on Knowledge Systems for Coalition Operations*, 2002.

[4] S. Kraus, J. Wilkenfeld, G. Zoltkin, Multiagent Negotiation Under Time Constraints, *Artificial Intelligence, Vol. 75, No. 2*, 1995.

[5] A. Lomuscio, M. Wooldridge, and N. Jennings. A Classification Scheme for Negotiation in Electronic Commerce". In Dignum, Sierra (eds), *Agent-mediated Electronic Commerce: A European Agentlink Perspective*, Springer Verlag, 2001.

[6] T. Sandholm. Agents in Electronic Commerce: Component Technologies or Automated Negotiation and Coalition Formation, *Autonomous Agents and Multi-Agent Systems, Vol. 3, No. 1*, 2000.

[7] O. Shehory and S. Kraus. Methods for Task Allocation via Agent Coalition Formation, *Artificial Intelligence, Vol. 101, No. 1–2*, 1998.

[8] D. Stanton and D. White, Constructive Combinatorics, Springer Verlag, 1986.