A heuristic model for concurrent bi-lateral negotiations in incomplete information settings

Thuc Duong Nguyen and Nicholas R. Jennings*

Abstract

Bi-lateral negotiations represent an important class of encounter in agent-based systems. To this end, this paper develops and evaluates a heuristic model that enables an agent to participate in multiple, concurrent bi-lateral encounters in competitive situations in which there is information uncertainty and deadlines.

1 Introduction

Automated negotiation is a key form of interaction in agent-based systems. Such negotiations exist in many different forms including one-to-one, one-to-many and many-to-many. Generally speaking, however, the latter two cases are dealt with using some form of auction protocol (be it single-sided or double-sided, respectively), while the former are often tackled using some form of heuristic method. Here we focus on the one-to-one case, in which one agent wants to purchase a service[1] from another. Moreover, we consider competitive situations in which the agents have *no a priori knowledge about the preferences of their opponents.* In such cases, the agents exchange proposals, representing acceptable solutions, until either an agreement is reached or the negotiation terminates with a failure.

To date, one of the inherent drawbacks of bi-lateral negotiation models is that the agent has to a priori identify a single partner to interact with. However, this is inefficient in an uncertain setting where there are multiple providers of the service that each have different characteristics. In this case, there are two alternatives: (1) negotiate sequentially with all the providers or (2) negotiate concurrently with them. The former has the disadvantage that it may result in lengthy negotiation encounters, but has the advantage that it is comparatively easy to use the outcome of one negotiation to dictate behavior in subsequent ones. The latter case has the advantage of taking less time, but the disadvantage that coordinating behaviors among the various negotiation threads is more difficult. Since we are interested in situations in time-constrained domains (such as e-commerce and grid computing), we concentrate on the concurrent case and develop a coordinated bidding model in which the various negotiation threads *mutually influence* one another. By mutually influence, we mean that the progress and agreement in one negotiation thread is used to alter the behavior of the agent in another thread for the same service. For example, having obtained a good deal in one thread, the agent may adopt a tougher stance in its other threads, to see if it can get an even better deal than the one it already has[2].

2 The concurrent negotiation model

The agent that wishes to purchase the service is called the *buyer* and the agents that offer the service are called the *sellers.* Service agreements (contracts) are assumed to be multi-dimensional (covering issues such as *price, quality, quantity,* etc.). The buyer has a deadline t_{max} by when it must conclude its negotiations for the service. Similarly, each seller *i* has its own negotiation deadline $t_{i_{max}}$. All the agents have their own preferences about the service and this information is private (as are the strategies the individual agents follow). The agents follow an alternating sequential protocol, in which the illocutions are *offer* (a proposal made by one agent to the other), *counter-offer* (a counter proposal from an agent in response to a proposal it received), *accept* (accept a proposed offer), *finalize* (secure a deal with the chosen seller), *decline* (reject the previously accepted offer) and *withdraw* (terminate the negotiation thread). The difference between an *accept* and *finalize* is necessary in this work to deal with the problem of concurrent encounters. If the buyer accepts an offer from a seller then this is viewed as binding on the seller (for a specified period of time that is assumed to be longer than t_{max}). However, it is not binding on the buyer. Thus, the buyer may accept several offers from multiple sellers in any one negotiation episode. However, when it has completed all the negotiations, the buyer will *finalize* one of the accepted deals with one of the sellers and *decline* the others (thus freeing them

* Department of Electronics and Computer Science, University of Southampton, UK. Email: {tdnOlr, nrj}@ecs.soton.ac.uk

[1] A service is here viewed an abstract representation of the capability of an agent.

[2] This model differs from a onc-to-many auction in that it allows *direct* interaction between the agent requiring the service and the providers offering it. This ability to exchange unmediated counter-offers enables the participants to indicate their preferences and constraints directly to one another.

from their commitment to the proposal). This two phase process is necessary so that the buyer can use accepted deals as a base line for the subsequent concurrent negotiations.[3]

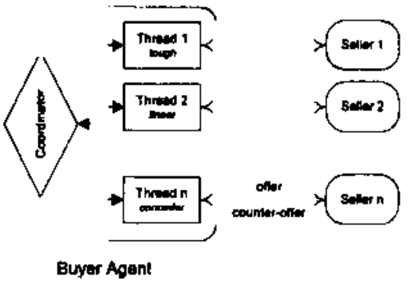

Figure 1: System architecture

In more detail, the model for the buyer agent consists of two main components: a *coordinator* and a number of *negotiation threads* (see figure 1). The negotiation threads deal directly with the various sellers (one per seller) and are responsible for deciding what counter-offers to send to them and what proposals to accept. For maximum flexibility, we assume that the buyer agent may adopt different strategies in each of its threads. We adopt separate semi-independent threads for reasons of modularity and coherence. The alternative of having every single negotiation move centrally coordinated and intercepting all the bids received from all the sellers, is viewed as a computational bottleneck for the time-constrained environments we are targeting.

Each negotiation thread inherits the preferences from the main buyer agent, including the acceptable ranges of values for each negotiation issue, the deadline of the negotiation and the current reservation value (the lowest utility value of an offer that the agent considers acceptable). The coordinator decides the negotiation strategies for each thread. After each negotiation round, the threads report back their status to the coordinator. If a thread reaches a deal with a particular seller, it terminates its negotiation. Based on the coordination schema it is using (see section 2.1 for more details), the coordinator will then notify all other negotiation threads of the new reservation value and it may change the negotiation strategy of some of them. The detailed working of the two components are described below.

2.1 The coordinator

The coordinator is responsible for coordinating all the negotiation threads and choosing an appropriate negotiation strategy for each thread.

As a first step, the coordinator acts like a blackboard for shared information about the ongoing encounters. It receives the current status from the various negotiation threads (including the proposal's values), keeps a list of agreements reached and notifies the threads about any changes in their reservation values. Second, and more importantly, the coordinator decides the negotiation strategy for each thread initially and whether this should change over time to reflect any agreements that have been made to date.

To ground our model, at this time we consider the set of strategies S to be composed of the class of time dependent strategies advocated in [Faratin, 2001] for bi-lateral negotiations in uncertain environments with time constraints. These strategies fall into three categories, namely: *conceder* (S_c), *linear* (5/) and *tough* (S_t) where $S = S_c \cup S_l \cup S_t$. All of the strategies start with the same initial value that is generated in relation to the deadline and the reservation value. The *conceder* strategy quickly lowers its value until it reaches its reservation value. The *linear* strategy drops to its reservation value in a steady fashion. Finally, the *tough* strategy keeps its value until the deadline approaches, then it rapidly drops to its reservation value.

In his empirical analysis of the behavior of negotiating agents that adopted these strategies, Faratin showed that if it is possible to approximate the type of the opponent then the agent can alter its strategy to be more effective. Given this observation, the coordinator attempts such a classification. Specifically, at time t: $2 < t \leq t_{max}$, called the *analysis time,* the coordinator tries to determine if a given seller is a *conceder* or a *non-conceder.* In particular, assume $O^i j$ is the value of the offer that seller agent i made at time $j : 0 \leq j \leq t$. Then seller i is considered a *conceder* if $\forall k \in [2, t] : \frac{O^i_k - O^i_{k-1}}{O^i_{k-1} - O^i_{k-2}} > \alpha$ where α is the threshold value set on concessionary behavior. There is a similar characterization of non-conceders and if the agent falls into neither category, it is judged not classified.

Let the set of *conceder* and *non-conceder* agents be represented by A^c and A^n, respectively. Now, given the set of strategies $S = \{S_c, S_l, S_t\}$ and the set of agents $A = \{A^c, A^n\}$, the coordinator changes the strategy for each negotiation thread based on the type of the agent it is negotiating with, in order to try and obtain better outcomes. Agents belonging to the set A^c are willing to concede in order to end up with agreements. Therefore, if the agent negotiates toughly with some of them (keeping its offer consistent), it may obtain a deal that has higher value than if it continues negotiating in its present manner. However, if the agent negotiates in this way with all the agents, it may not obtain any deals at all. Therefore, for reasons of balance, the agent will negotiate in a tough manner with a subset of the agents in A^c, specifically with a percentage $(P^c_t\%)$ of them. For the remainder of the agents in A^c, the strategy remains unchanged. Similarly, if the agent believes a particular agent is in the set A^n then in order to make sure it obtains a deal with some of them, it makes some of its own strategy more conciliatory. Specifically, for the agents belonging to the set A^n, a fixed percentage $(P^n_c\%)$ of them will have their behavior made conciliatory, while the remainder have their strategies unchanged. There is no change to agents whose behavior cannot be classified.

2.2 The negotiation threads

An individual negotiation thread is responsible for dealing with an individual seller agent on behalf of the buyer. Each

---

[3] This model is obviously biased in favor of the buyer and future work will look at relaxing this constraint so that sellers can also renege on deals.

such thread inherits its preferences from the buyer agent and has its negotiation strategy specified by the coordinator.

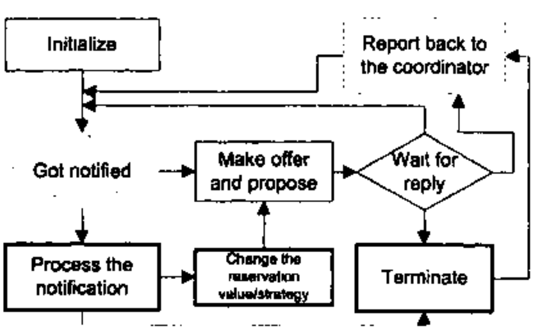

Figure 2: A single negotiation thread

In each thread (see figure 2), there are three main subcomponents; namely *communication* (represented by the dotted lines), *process* (represented by the bold lines) and *strategy.* The *communication* subcomponent is responsible for communicating with the coordinator. Before each round, it checks for incoming messages from the coordinator and if there are any, it passes them to the *process* subcomponent. After each round, it reports the status of the thread (including proposed proposals and the deal's value if an agreement is reached) back to the coordinator. The *process* subcomponent processes messages from the *communication* subcomponent. This can either be changing the reservation value or changing the strategy. The *strategy* subcomponent is responsible for making offers/counter-offers, as well as deciding whether or not to accept the offer made by the seller agent. It uses the reservation value as the basis for deciding whether to accept the seller's offer; in this case any offer with a value greater than this is accepted, otherwise a counter-proposal is made (unless the deadline has passed in which case a decline is sent).

3 Empirical evaluation

Having defined the model, the next step is to evaluate it. Given the aims of our work, we are interested in operational performance and so we decided to evaluate it empirically. In particular, we posed a number of hypotheses and evaluated them in different types of environment.

Our concurrent model is compared against a sequential negotiation model based on [Faratin, 2001]. In this model, all the agents' preferences, as well as the allocation of the strategies, are drawn from the same distributions as the concurrent ones. The only difference with [Faratin, 2001] is that if the buyer agent reaches an agreement of value *p* in negotiation i, then in all subsequent negotiations, *p* will be its new reservation value.

We now turn to specific hypotheses. Due to the limitation of space, we cannot show the corresponding graphs to support our claims (see [Nguyen and Jennings, 2003] for more details), but, nevertheless we can summarize our main findings:

Result 1 *The time to complete the negotiation is less for the concurrent model than for the sequential one.*

Result 2 *The number of proposals that are made in the concurrent model is less than the number made in the sequential model.*

Result 3 *To realize the benefits of concurrent negotiations, the buyer agent's deadline must not be too short.*

Result 4 *The final agreements reached by the concurrent model have, on average, higher or equal utility'for the buyer than those of the sequential model (assuming the deadline is not too short).*

Result 5 *Changing the strategy in response to the agent's assessment of the ongoing negotiation is equal or better than not doing so.*

Result 6 *To improve the performance of the concurrent model, the analysis time should be moderately early (to have time to have some effect) but not too early (so it is reasonably accurate).*

Result 7 *When dealing with sellers in* A^c*, the tougher the buyer negotiates the better the overall outcome it obtains.*

4 Conclusions and future work

This paper has developed a heuristic model for managing concurrent negotiations in time-constrained settings where agents have no prior knowledge of their opponents or their types. Through empirical evaluation, we showed how the model leads to better deals, more quickly than its sequential counterpart. We also highlighted the importance of the time when the opponents' negotiation strategies are classified and on the response to this assessment in terms of the degree of toughness adopted.

There are, however, a number of areas that still require further work. First, the means by which negotiation opponents are classified as being conceder or non-conceder needs to be refined so that this monitoring can be made on an ongoing basis (rather than as a one-off decision). Second, we need to allow for the possibility of sellers decommitting and then having these commitment changes feedback into the buyer's negotiation behavior. Third, we wish to extend the implementation so that the coordinator and the negotiation threads also embody fundamentally different models of bi-lateral negotiation (e.g. based on constraint-satisfaction, game-theory or any other method that is likely to be effective). In this case, the key challenge is in designing the coordinator so that it can select, monitor and modify these strategies in line with the agent's overarching negotiation objectives.

References

[Faratin, 2001] P. Faratin. *Automated Service Negotiation Between Autonomous Computational Agents, PhD Thesis.* Queen Mary College, London, England, 2001.

[Nguyen and Jennings, 2003] T. D. Nguyen and N. R. Jennings. Concurrent bi-lateral negotiation in agent systems. *Proceedings of the 4th DEXA Workshop on e-Negotiations, Prague, Czech Republic,* 2003.