Bilateral Negotiation in a Multi-agent Supply Chain System

Fernando Lopes[1] and Helder Coelho[2]

[1] LNEG – National Research Institute
Estrada do Paço do Lumiar 22, 1649-038 Lisbon, Portugal
fernando.lopes@ineti.pt
[2] University of Lisbon, Department of Computer Science
Bloco C6, Piso 3, Campo Grande, 1749-016 Lisbon, Portugal
hcoelho@di.fc.ul.pt

Abstract. A supply chain is a set of organizations directly linked by flows of services from suppliers to customers. Supply chain activities range from the ordering and receipt of raw materials to the production and distribution of finished goods. Supply chain management is the integration of key activities across a supply chain for the purposes of building competitive infrastructures, synchronizing supply with demand, and leveraging worldwide logistics. This paper addresses the challenges created by supply chain management towards improving long-term performance of companies. It presents a multi-agent supply chain system composed of multiple software agents, each responsible for one or more supply chain activities, and each interacting with other agents in the execution of their responsibilities. Additionally, this paper presents the key features of a negotiation model for software agents. The model handles bilateral multi-issue negotiation and incorporates an alternating offers protocol, a set of logrolling strategies, and a set of negotiation tactics.

Keywords: Autonomous agents, Multi-agent supply chain system, Automated negotiation, Bargaining.

1 Introduction

Multi-agent systems (MAS) are ideally suited to represent problems that have multiple problem solving entities and multiple problem solving methods [3]. The major motivations for the increasing interest in MAS research include the ability to solve problems in which data, expertise, or control is distributed, the ability to allow inter-operation of existing legacy systems, and the ability to enhance performance along the dimensions of computational efficiency, reliability, and robustness. Agent technology has been used to solve real-world problems in a range of industrial and commercial applications, including manufacturing, process control, telecommunications, air traffic control, information management, electronic commerce, and business process management (see, e.g., [11]).

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 195–206, 2010.

A supply chain is a set of organizations directly linked by flows of services from suppliers to customers. Supply chain activities range from the ordering and receipt of raw materials to the production and distribution of finished goods. Supply chain management (SCM) is the integration of key activities across a supply chain for the purpose of improving long-term performance. SCM encompasses the planning and management of all activities involved in sourcing, procurement, conversion, and logistics. It also includes the crucial components of collaboration and coordination with channel partners (e.g., suppliers, intermediaries, and customers). In essence, SCM integrates supply and demand management within and across companies. The main objectives include building competitive infrastructures, leveraging worldwide logistics, synchronizing supply with demand, and measuring performance globally.

Supply chain management in general and multi-agent supply chain systems in particular have received some attention lately (see, e.g., [1,2]). However, despite the prominent models proposed in the literature, most challenges created by SCM are still waiting to be addressed more thoroughly. At present, there is a need to develop computational tools to help manage the complexity of SCM. Against this background, the purpose of this paper is twofold:

1. *to present a multi-agent supply chain system* – the system is composed of a collection of software agents, each responsible for one or more supply chain activities, and each interacting with other agents in the execution of their responsibilities;
2. *to present the key features of a negotiation model for software agents* – the model handles bilateral multi-issue negotiation and incorporates an alternating offers protocol, a set of logrolling strategies, and a set of negotiation tactics.

Logrolling strategies are computationally tractable functions that define the tactics to be used both at the beginning and during the course of negotiation. The words "computationally tractable functions" presume that agents are able to compute the strategies in a reasonable amount of time. Furthermore, at every period of negotiation, the strategies state whether bargaining should continue or terminate. Negotiation tactics are functions that specify the individual moves to be made at each point of the negotiation process.

This paper builds on our previous work in the area of automated negotiation [6,7,8]. In particular, it introduces precise definitions for logrolling strategies. It also lays the foundation for performing an experiment to investigate the performance of agents operating in a supply chain system and equipped with our negotiation model.

The remainder of the paper is structured as follows. Section 2 is devoted to negotiation between software agents. Section 3 describes a multi-agents supply chain system and illustrates how software agents equipped with our model operate in a negotiation setting. Finally, related work and concluding remarks are presented in sections 4 and 5 respectively.

2 Multi-agent Negotiation

Negotiation is a discussion among conflicting parties with the aim of reaching agreement about a divergence of interests. Negotiation may involve two parties (bilateral negotiation) or more than two parties (multilateral negotiation) and one issue (single-issue negotiation) or many issues (multi-issue negotiation). This section presents the key features of a model for software agents that handles two-party and multi-issue negotiation.

2.1 Pre-negotiation

Pre-negotiation is the process of preparing and planning for negotiation and involves mainly the creation of a well-laid plan specifying the activities that negotiators should attend to before actually starting to negotiate [5]. Let $\mathcal{A}=\{a_1, a_2\}$ be the set of autonomous agents (negotiating parties). Both the number of agents and their identity are fixed and known to all the participants. Let $\mathcal{I}=\{x_1, \ldots, x_n\}$ be the negotiating agenda – the set of issues to be deliberated during negotiation. The issues are quantitative in nature and defined over continuous domains. Let $\mathcal{D}=\{D_1, \ldots, D_n\}$ be the set of issue domains. For each issue x_k, the range of acceptable values is represented by the interval $D_k=[min_k, max_k]$. The issues are also known to all the participants.

Effective pre-negotiation requires that negotiators prioritize the issues and define the targets. Priorities are set by ranking-order the issues, *i.e.*, by defining the most important, the second most important, and so on. The priority prt^i_k of an agent $a_i \in \mathcal{A}$ for an issue $x_k \in \mathcal{I}$ is a number that represents the importance of x_k. The weight w^i_k is a number that represents the preference for x_k. The resistance point or limit lim^i_k is the ultimate fallback position for x_k, the point beyond which a_i is unwilling to concede on x_k. The level of aspiration or target point trg^i_k is the point at which a_i is satisfied with the value of x_k. The asking price or optimistic point opt^i_k is the most preferred or ideal value for x_k.

Additionally, effective pre-negotiation requires that negotiators agree on an appropriate protocol that defines the rules governing the interaction. The negotiation literature describes several protocols that vary significantly depending on the type and amount of information exchanged between agents (see, e.g., [9,15]). Simple protocols allow agents to exchange only proposals, *i.e.*, solutions to the problem they face. Richer protocols allow agents to provide feedback on the proposals they receive. This feedback often takes the form of critiques, *i.e.*, comments on which parts of proposals are acceptable or unacceptable. Sophisticated protocols allow agents to provide arguments to support their negotiation stance.

Most complex protocols make, however, considerable demands on any implementation, mainly because they appeal to very rich representations of the agents and their environments. Therefore, we consider a simple alternating offers protocol [10]. Two agents or players bargain over the division of the surplus of $n \geq 2$ distinct issues. The agents determine an allocation of the issues by alternately submitting proposals at times in $\mathcal{T} = \{1, 2, \ldots\}$.

The negotiation process starts with an agent, say $a_i \in \mathcal{A}$, submitting a proposal $p^1_{i\rightarrow j}$ to the other agent $a_j \in \mathcal{A}$ in period $t=1$. The agent a_j receives $p^1_{i\rightarrow j}$ and can either accept the offer (`Yes`), reject it and opt out of the negotiation (`Opt`), or reject it and continue bargaining (`No`). In the first two cases the negotiation ends. Specifically, if $p^1_{i\rightarrow j}$ is accepted, negotiation ends successfully and the agreement is implemented. Conversely, if $p^1_{i\rightarrow j}$ is rejected and a_j decides to opt out, negotiation terminates with no agreement. In the last case, negotiation proceeds to the next time period $t=2$, in which a_j makes a counter-proposal $p^2_{j\rightarrow i}$. The tasks just described are then repeated. Once an agreement is reached, the agreed-upon allocations of the issues are implemented.

The negotiation procedure, labelled the "joint-offer procedure", involves bargaining over the allocation of the entire endowment stream at once. A proposal $p^t_{i\rightarrow j}$ submitted by an agent $a_i \in \mathcal{A}$ to an agent $a_j \in \mathcal{A}$ in period $t \in \mathcal{T}$ is a vector $(v_1, \ldots, v_n)$ of issue values. An agreement is a proposal accepted by all the agents in $\mathcal{A}$. The set of possible agreements is $\mathcal{S} = \{(v_1, \ldots, v_n) \in \mathbb{R}^n\}$, where $v_k \in D_k$, for $k = 1, \ldots, n$, is a value of an issue $x_k \in \mathcal{I}$.

The players' preferences are modelled by defining a utility function over all possible outcomes. More specifically, we consider that each agent $a_i \in \mathcal{A}$ has a continuous utility function: $U_i : \{D_1 \times \ldots \times D_n\} \cup \{\texttt{Opt}, \texttt{Disagreement}\} \rightarrow \mathbb{R}$. Accordingly, when the utility for a_i from one outcome is greater than from another outcome, we assume that a_i prefers the first outcome over the second. The outcome `Opt` is interpreted as one of the agents opting out of the negotiation in a given period of time. Perpetual disagreement is denoted by `Disagreement`.

Now, the additive model is probably the most widely used in multi-issue negotiation − the parties assign numerical values to the different levels on each issue and add them to get an entire offer evaluation [16]. This model is simple and intuitive, and therefore well suited to the purposes of this work. The utility function U_i of a_i to rate offers and counter-offers takes the form:

$$U_i(x_1, \ldots, x_n) \; = \; \sum_{k=1}^{n} w^i_k \times V^i_k(x_k)$$

where w^i_k is the weight of a_i for an issue $x_k \in \mathcal{I}$ and $V^i_k(x_k)$ is the scoring (or marginal) utility function of a_i for x_k, *i.e.*, the function that gives the score a_i assigns to a value of an issue x_k.

Negotiation may end with either agreement or no agreement. Failure to agree can occur in two ways: (i) either party decides to opt out unilaterally, or (ii) the two do not agree to any proposal. The resistance points or limits play a key role in reaching agreement when the parties have the ability to unilaterally opt out of the negotiation − they define the worst agreement for a given party which is still better than opting out. For each agent $a_i \in \mathcal{A}$, we will denote this agreement by $\hat{s}_i \in \mathcal{S}$. Hence, $\hat{s}_i$ will be the least-acceptable agreement for a_i, *i.e.*, the worst (but still acceptable) agreement for a_i. The set of all agreements that are preferred by a_i to opting out will be denoted by S_i. Perpetual disagreement is the least-preferred or worst outcome, *i.e.*, disagreement is even worse than opting out.

2.2 Actual Negotiation

Actual negotiation is the process of moving toward agreement (usually by an iterative exchange of offers and counter-offers). The negotiation protocol defines the states (e.g., accepting a proposal), the valid actions of the agents in particular states (e.g., which messages can be sent by whom, to whom, at what stage), and the events that cause states to change (e.g., proposal accepted). It marks branching points at which agents have to make decisions according to their strategies. Thus, at each step of negotiation, agents often need to follow their strategies to choose among different possible actions to execute.

Negotiation strategies are often implemented through a variety of tactics [13,14]. The line between strategies and tactics often seems indistinct, but one major difference is that of scope. Tactics are short-term moves designed to enact broad (or high-level) strategies – they are structured, directed, and driven by strategic considerations [5]. Accordingly, in this work strategies are computationally tractable functions that define the tactics to be used both at the beginning and during the course of negotiation. The words "computationally tractable functions" presume that agents are able to compute the strategies in a reasonable amount of time. Also, at every period of negotiation, strategies state whether bargaining should continue or terminate. Tactics, in turn, are functions that specify the short-term moves to be made at each point of negotiation.

Negotiation strategies can reflect a variety of behaviours and lead to strikingly different outcomes. However, logrolling is commonly discussed in the behavioral negotiation literature – two parties agree to exchange concessions on different issues, with each party yielding on issues that are of low priority to himself and high priority to the other party [17]. Accordingly, logrolling will receive the preponderance of our attention in this paper.

Logrolling Strategies. Most well-intended negotiators tend to believe that, above all, success depends on the creativity to devise agreements that yield considerable gain to both negotiating parties. They see the essence of negotiation as expanding the "pie" of available resources, as pursuing joint gains. They are essentially value creators – they attempt to probe below the surface of the other party's true needs to locate mutually superior solutions [12].

Logrolling is possible only when several issues are under consideration and the parties have different priorities among these issues. The parties then agree to exchange concessions on (part or all) of the issues, each party winning on the issues he places greater emphasis. In this way, each party gets the fraction of his demands that he deems most important. Clearly, a theory of logrolling in complex agendas is of particular importance to both human and automated negotiation. However, there are important questions still waiting to be addressed more thoroughly. We highlight the following: which issues will be grouped for the exchange of concessions? Relevant efforts to answer this questions include the theory of appropriate exchange and the principle of equivalence [14]. But it is clear that much more research still needs to be performed. In this work, we consider the following three subsets of the agenda for each agent $a_i \in \mathcal{A}$:

- a subset I_i^+, containing the issues of higher priority to a_i (and are also believed to be of lower priority to his opponent a_j);
- a subset I_i^-, containing the issues of lower priority to a_i (and are also believed to be of higher priority to a_j);
- a subset $I_i^\pm$, containing the remaining issues of the agenda ($\mathcal{I} = I_i^+ \cup I_i^\pm \cup I_i^-$).

Negotiators have frequently something to offer that is relatively less valuable to them than to their opponent, and thus, the subsets I_i^+ and I_i^- are typically non-empty. These two subsets contain the logrolling issues, *i.e.*, the issues that can be logrolled to make profitable trade-offs. By contrast, the subset $I_i^\pm$ contains both the distributive issues (the parties' interests are directly opposed) and the compatible issues (the parties have coordinated interests). A formal definition of a generic logrolling strategy follows.

Definition 1 (Logrolling Strategy). *Let $\mathcal{A}$ be the set of negotiating agents, $\mathcal{I}$ the negotiating agenda, $\mathcal{T}$ the set of time periods, and $\mathcal{S}$ the set of possible agreements. Let $a_i \in \mathcal{A}$ be the first agent to submit a proposal, T_i his set of tactics, and $a_j \in \mathcal{A}$ his opponent. Let I_i^+ be the set of issues that are of higher priority to a_i (and are believed to be of lower priority to a_j), I_i^- the set of issues that are of lower priority to a_i (and are believed to be of higher priority to a_j), and $I_i^\pm$ the remaining issues of the agenda. A logrolling strategy $L_i : T_i \times T_i \times T_i \times T_i \times \mathcal{T} \rightarrow \mathcal{S} \cup \{\texttt{Yes}, \texttt{No}, \texttt{Opt}\}$ for a_i is a function with the following general form:*

$$L_i = \begin{cases} \textit{apply } O_i(x_k) \textit{ and offer } p^1_{i\rightarrow j}\,, & \textit{if } a_i\textit{'s turn and } t=1 \\ \textit{reject } p^{t-1}_{j\rightarrow i} \textit{ and quit}, & \textit{if } a_j\textit{'s turn and } U_i(p^{t-1}_{j\rightarrow i}) < U_i(\hat{s}_i) \\ \textit{apply } Y_i^+(x_k, f_k^i) \textit{ to } I_i^+ & \textit{if } a_j\textit{'s turn and } U_i(p^{t-1}_{j\rightarrow i}) \geq U_i(\hat{s}_i) \\ \textit{apply } Y_i^\pm(x_k, f_k^i) \textit{ to } I_i^\pm & \\ \textit{apply } Y_i^-(x_k, f_k^i) \textit{ to } I_i^- & \\ \textit{prepare } p^t_{i\rightarrow j} & \\ \textit{if } U_i^* \geq 0 \textit{ accept } p^{t-1}_{j\rightarrow i} \textit{ else reject}, & \\ \textit{offer logrolling solution } p^t_{i\rightarrow j}\,, & \textit{if } a_i\textit{'s turn and } t>1 \end{cases}$$

where:

(i) $p^1_{i\rightarrow j}$ *is the opening offer of* a_i, $p^{t-1}_{j\rightarrow i}$ *is the offer of* a_j *for time period* $t-1$ *of negotiation, and* $p^t_{i\rightarrow j}$ *is the offer of* a_i *for the next time period* t *of negotiation;*

(ii) *for each issue* $x_k \in \mathcal{I}$, $O_i(x_k)$ *is an opening negotiation tactic;*

(iii) $Y_i^+(x_k, f_k^i)$, $Y_i^\pm(x_k, f_k^i)$ *and* $Y_i^-(x_k, f_k^i)$ *are concession tactics, and* f_k^i *is the concession factor of* a_i *for* x_k *(see below);*

(iv) $U_i(\hat{s}_i)$ *is the utility of the least-acceptable agreement for* a_i*;*

(v) $U_i^* = U_i(p^{t-1}_{j\rightarrow i}) - U_i(p^t_{i\rightarrow j})$ ∎

Two explanatory and cautionary notes are in order here. First, notation is being abused somewhat, by using L_i rather than $L_i(O_i(x_k), Y_i^+(x_k, f_k^i), Y_i^\pm(x_k, f_k^i), Y_i^-(x_k, f_k^i), t)$. The abuse helps improve readability, however, and meaning will always be clear from context. Second, tactics are functions of a single issue rather than a vector of issues. This permits great flexibility, since it allows agents to model a wide range of negotiation behaviors.

Logrolling can be insightful or simply emerge from concession making. Typical strategies that lead to logrolling solutions include:

1. *starting high and conceding strategically* – negotiators adopt an optimistic opening position, slightly reduce their low-priority demands (and they believe are of high priority to their opponent), and hold firm on their high-priority demands (and they believe are of low priority to their opponent);
2. *starting high and negotiating creatively* – negotiators adopt an optimistic opening position, substantially reduce their low-priority demands (and they believe are of high priority to their opponent), and hold firm on their high-priority demands (and they believe are of low priority to their opponent).

The definition of these and other relevant strategies involves basically the specification of particular tactics. For instance, the strategy "`starting high and negotiating creatively`" is defined by considering the opening negotiation tactic "`starting optimistic`" and the concessions tactics "`moderate`" and "`stalemate`" (but see below).

Opening Negotiation Tactics. Opening negotiation tactics are functions that specify the demands to be made at the outset of negotiation. The following three tactics are commonly discussed in the behavioral negotiation literature [5,14]:

1. *starting optimistic* – specifies a value for an issue close to the optimistic point;
2. *starting realistic* – specifies a value for an issue in the range defined by the target and the optimistic points;
3. *starting pessimistic* – specifies a value for an issue in the range defined by the target and the resistance points.

A formal definition of the tactic "`starting optimistic`" follows (the definition of the other two tactics is essentially identical, and is therefore omitted).

Definition 2 (Starting Optimistic). *Let $\mathcal{A}=\{a_1, a_2\}$ be the set of negotiating agents and $\mathcal{I}=\{x_1, \ldots, x_n\}$ the negotiating agenda. Let $\mathcal{D}=\{D_1, \ldots, D_n\}$ be the set of issue domains. The tactic starting optimistic of an agent $a_i \in \mathcal{A}$ for an issue $x_k \in \mathcal{I}$ takes the form:*

$$O_i(x_k) = opt_k^i + \epsilon$$

where:

(i) $\epsilon > 0$ *is small;*
(ii) opt_k^i *is the optimistic point of a_i for x_k.* ∎

Concession Tactics. Concession tactics are functions that model the concessions to be made throughout negotiation. Practically speaking, negotiators may consider strikingly different patterns of concessions as negotiation unfolds. However, the following three levels of concession magnitude are commonly discussed in the negotiation literature [13,17]: large, substantial, and small. To this we would add two other levels: null and complete. Accordingly, we consider the following five concession tactics:

1. *stalemate* – models a null concession on an issue x_k at stake;
2. *tough* – models a small concession on x_k;
3. *moderate* – models a substantial concession on x_k;
4. *soft* – models a large concession on x_k;
5. *accommodate* – models a complete concession on x_k.

A formal definition of a generic concession tactic follows (without loss of generality, we consider that $a_i \in \mathcal{A}$ wants to maximize $x_k \in \mathcal{I}$).

Definition 3 (Concession Tactic). *Let $\mathcal{A}=\{a_1,a_2\}$ be the set of negotiating agents, $\mathcal{I}=\{x_1,\ldots,x_n\}$ the negotiating agenda, and $\mathcal{D}=\{D_1,\ldots,D_n\}$ the set of issue domains. A concession tactic $Y_i\colon D_k \times [0,1] \rightarrow D_k$ of an agent $a_i \in \mathcal{A}$ for an issue $x_k \in \mathcal{I}$ is a function with the following general form:*

$$Y_i(x_k, f_k^i) = x_k - f_k^i(x_k - lim_k^i)$$

where:

(i) *f_k^i is the concession factor of a_i for x_k;*
(ii) *lim_k^i is the limit of a_i for x_k.* ∎

The five tactics are defined by considering specific values for the concession factor f_k^i. In particular, the "`stalemate`" tactic is defined by $f_k^i = 0$ and the "`accommodate`" tactic by $f_k^i = 1$. The other three tactics are defined by considering values for f_k^i in different ranges (e.g., the "`tough`" tactic by $f_k^i \in]0.00, 0.05]$, the "`moderate`" tactic by $f_k^i \in]0.05, 0.15]$, and the "`soft`" tactic by $f_k^i \in]0.15, 0.20]$).

3 Agents for Supply Chain Management

Multi-agent systems have generated lots of excitement in recent years because of their promise as a new paradigm for conceptualizing and implementing complex software systems. Central to the design and effective operation of a multi-agent system are a core set of problems and research questions, notably [3]:

1. *the design problem* – how to formulate, describe, decompose, and allocate different problems and synthesize results among a group of intelligent agents?
2. *the coordination problem* – how to ensure that agents act coherently in making decisions or taking action, accommodating the non-local effects of local decisions and avoiding harmful interactions?

The design problem is focused on the domain the system is intended to solve in a distributed manner. This problem consists mainly in distributing different supply chain activities across a number of agents. A typical distribution involves at least the following agents:

1. *sales agent* – responsible for acquiring orders from customers, negotiating with customers, and handling customer requests for modifying or canceling orders;
2. *logistics agent* – responsible for coordinating the plants and distribution centers of a manufacturing enterprise: it manages the movement of materials and products across the supply chain, from the suppliers of raw materials to the customers of finished goods;
3. *scheduling agent* – responsible for scheduling and rescheduling the activities of a manufacturing enterprise;
4. *resource management agent* – responsible for dynamically managing the availability of resources in order to execute the scheduled activities;
5. *supplier agents* and *customer agents* – the suppliers sell raw materials and the customers buy finished goods.

The agents are essentially computer systems capable of flexible autonomous action in order to meet their design objectives.

The coordination problem is focussed on ensuring that agents act in a tightly coordinated manner in order to effectively achieve their objectives. This problem is addressed, at least in part, by designing agents that are able to coordinate their activities through negotiation. Specifically, for the case of a supply chain system, the agents are charged with executing actions towards the achievement of their private goals and, thus, conflicts inevitably occur among them. Negotiation is the predominant process for resolving conflicts.

Let us introduce a specific scenario involving interaction between the sales agent and the logistics agent:

> David, the director of Sales, has lined up two new orders for a total of 15000 men's suits: one for 10000 and the other for 5000 men's suits. Martin, the director of Logistics, has already stated that it will take four months to make the suits. Together, they will gross over a million Euros, with a fine profit for the company. The problem is that Martin insists that the job will take four months and David's customer wants a two-month turnaround. Also, David claims that he can't afford to lose the customer. David and Martin are discussing and, so far, have accomplished little more than making each other angry. However, they can resolve their differences by negotiating a mutually beneficial agreement.

The remainder of this section illustrates how software agents equipped with the proposed model operate in the Sales-Logistics scenario. In particular, we demonstrate both how negotiation evolves and how software agents use different logrolling strategies (and their associated opening negotiation and concession tactics).

Table 1. Major issues, preferences, limits and targets (Sales agent)

Negotiation Issue	Weight	Limit	Target Point
quantity_1	0.350	9500	10000
date_1	0.300	1.25 (5 weeks)	1.00 (4 weeks)
quantity_2	0.175	4000	5000
date_2	0.175	1.50 (6 weeks)	1.00 (4 weeks)

For illustrative purposes, we consider the negotiation process from the viewpoint of David. There are four major issues of concern: `quantity_1`, `date_1`, `quantity_2` and `date_2`. The first two issues are the most important to David due to the inherent customer demands − he wants fast action on the 10000 suit order. Also, after a period of consultation with the customer, David concludes that he is overly firm about the 10000 suit order (and is willing to wait up to five weeks for 10000 suits, or ultimately 9500 suits), but he is moderately firm about the 5000 suit order (and is willing to wait up to six weeks for only 4000 suits). Table 1 shows the four issues, the (normalized) weights, the limits, and the target points of the Sales agent.

Figure 1 shows the joint utility space for David and Martin. The abscissa represents the utility to David, and the ordinate represents the utility to Martin. The solid line OCO' represents the Pareto optimal frontier (*i.e.*, the locus of achievable joint evaluations from which no joint gains are possible [16]). The small squares depict a few options for settling the issues at stake.

Now, we take up a few logrolling strategies, one at a time, and examine their nature and their impact on the negotiation outcome. As noted, it is of higher priority for Sales to get fast action on the 10000 suit order than the 5000 suit order. Suppose now that it is of higher priority for Logistics to handle the 5000 suit order (and to avoid the 10000 suit order). These two departments have the makings of a logrolling deal − each party can yield on issues that are of low priority to himself and high priority to the other party. Accordingly, David and Martin can reach the following solution: a 4-week schedule for 9750 suits and a 6-week schedule for 4500 suits. This agreement is represented by point A in Figure 1 and provides a (normalized) benefit of 0.562 to each agent.

Noticeably, logrolling strategies can permit negotiators to fully exploit the differences in the valuation of the issues to capitalize on optimal agreements. In this way, David and Martin can pursue specific logrolling strategies and agree on a four-week schedule for 10000 suits and a six-week schedule for 4000 suits. This agreement lies along the efficient frontier and is represented by point B in Figure 1 − it provides a (normalized) benefit of 0.65 to each party.

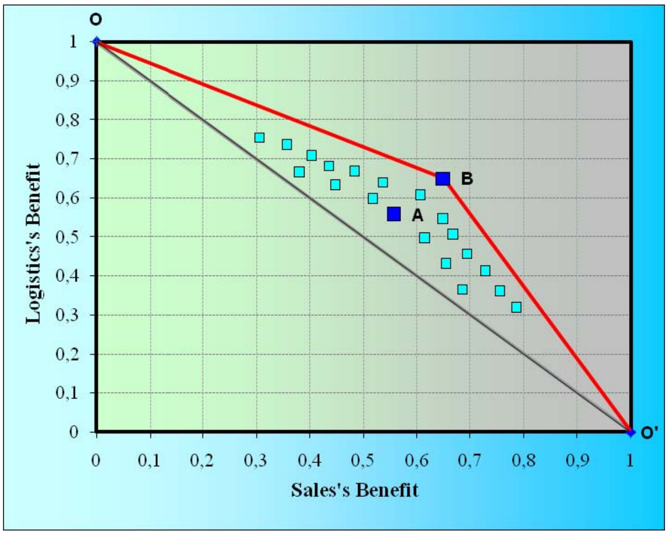

Fig. 1. Joint utility space for the Sales-Logistics negotiation situation

4 Related Work

Artificial intelligence researchers have paid a great deal of attention to automated negotiation over the past decade and a number of prominent models have been proposed in the literature (see, e.g., [4,9,15]. The majority of models primarily use either game-theoretic techniques or methods from the social sciences as a basis to develop autonomous negotiating agents. Furthermore, most models incorporate specific protocols (notably, the alternating offers protocol) and libraries of negotiation strategies (notably, concession and logrolling strategies). However, despite the power and elegance of existing models, we are aware of no similar efforts to define logrolling strategies as functions that specify the tactics to be used both at the outset and throughout negotiation. Tactics, in turn, are defined as functions that specify the short-term moves to be made at each point of negotiation.

5 Conclusion

This article has presented a simplified multi-agent supply chain system composed of a collection of software agents, each responsible for one or more supply chain activities, and each interacting with other agents in the execution of their responsibilities. Additionally, the article has presented the key features of a model for software agents that handles two-party and multi-issue negotiation. The model incorporates an alternating offers protocol, a set of logrolling strategies, and a set of negotiation tactics.

Autonomous agents equipped with the negotiation model are currently being developed. Our aim for the future is to perform a set of inter-related experiments to empirically evaluate the key components of the agents operating in the supply chain system. Each experiment will lay the foundation for subsequent experimental work.

References

1. Burgess, K., Singh, P., Koroglu, R.: Supply Chain Management: Structured Literature Review and Implications for Future Research. International Journal of Operations & Production Management 26, 703–729 (2006)
2. Fox, M., Barbuceanu, M., Teigen, R.: Agent-Oriented Supply Chain Management. International Journal Flexible Manufacturing Systems 12, 165–188 (2000)
3. Jennings, N., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems 1, 7–38 (1998)
4. Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Wooldridge, M., Sierra, C.: Automated Negotiation: Prospects, Methods and Challenges. Group Decision and Negotiation 10, 199–215 (2001)
5. Lewicki, R., Barry, B., Saunders, D., Minton, J.: Negotiation. McGraw Hill, New York (2003)
6. Lopes, F., Mamede, N., Novais, A.Q., Coelho, H.: A Negotiation Model for Autonomous Computational Agents: Formal Description and Empirical Evaluation. Journal of Intelligent & Fuzzy Systems 12, 195–212 (2002)
7. Lopes, F., Mamede, N., Novais, A.Q., Coelho, H.: Negotiation Strategies for Autonomous Computational Agents. In: 16th European Conference on Artificial Intelligence (ECAI-04), pp. 38–42. IOS Press, Amsterdam (2004)
8. Lopes, F., Mamede, N., Novais, A.Q., Coelho, H.: Negotiation Among Autonomous Agents: Experimental Evaluation of Integrative Strategies. In: 12th Portuguese Conference on Artificial Intelligence, pp. 280–288. IEEE Computer Society Press, Los Alamitos (2005)
9. Lopes, F., Wooldridge, M., Novais, A.Q.: Negotiation Among Autonomous Computational Agents: Principles, Analysis and Challenges. Artificial Intelligence Review 29, 1–44 (2008)
10. Osborne, M., Rubinstein, A.: Bargaining and Markets. Academic Press, San Diego (1990)
11. Pechoucek, M., Marik, V.: Industrial Deployment of Multi-agent Technologies: Review and Selected Case Studies. Autonomous Agents and Multi-Agent Systems 17, 397–431 (2008)
12. Pruitt, D.: Negotiation Behavior. Academic Press, New York (1981)
13. Pruitt, D., Carnevale, P.: Negotiation in Social Conflict. Open University Press, Philadelphia (1993)
14. Pruitt, D., Kim, S.: Social Conflict: Escalation, Stalemate, and Settlement. McGraw Hill, New York (2004)
15. Rahwan, I., Ramchurn, S., Jennings, N., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-based Negotiation. Knowledge Eng. Review 18, 343–375 (2004)
16. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press, Cambridge (1982)
17. Thompson, L.: The Mind and Heart of the Negotiator. Prentice-Hall, Englewood Cliffs (2005)