The 2007 Procurement Challenge: A Competition to Evaluate Mixed Procurement Strategies

A. Sardinha
School of Computer Science, Carnegie Mellon University

M. Benisch
School of Computer Science, Carnegie Mellon University

N. Sadeh
School of Computer Science, Carnegie Mellon University

R. Ravichandran
School of Computer Science, Carnegie Mellon University

V. Podobnik
Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

M. Stan
Department of Computer Science, University "Politehnica" of Bucharest, Romania

Abstract

This article discusses the design of the 2007 "Supply Chain Management - Procurement Challenge" (SCM-PC), a competition designed by the first three authors to evaluate the performance of mixed procurement strategies. Specifically, the SCM-PC Challenge revolves around a PC assembly scenario, where trading agents developed by different teams compete for components required to assemble different types of PCs. The Challenge requires the agents to manage supply chain risk through combinations of long-term, quantity-flexible procurement contracts and one-off procurement contracts for different components. Collectively the authors represent the top three entries in the 2007 Procurement Challenge. They present the strategies their teams developed for the competition, compare their performances, and discuss lessons learned from the competition.

Introduction

Supply chain management involves planning, implementing and controlling the buying and selling of raw materials, work-in-process inventory and finished goods. Traditionally, this process has been static, depending primarily on long-term relationships between existing trading partners. The increasing adoption of more flexible and dynamic relations has the potential to make markets more efficient by establishing better matches between suppliers and customers. However, as relationships become more flexible the decisions involved in supply chain management will become more complex due to both the shear number of factors that have to be taken into account while making these decisions, and the uncertainties in the markets. The role of technology to aid in supply chain management decision making has thus become inevitable. In the short term, technology can be used passively to provide insights to supply chain managers. In the longer term, technology can adopt a more active role by making decisions in an autonomous manner.

The TAC SCM Competition

The TAC SCM competition was established to simulate many of the challenges imposed by dynamic markets, while keeping the rules simple enough to entice researchers from various fields to submit entries. The rules of the game mimic many of the real world market forces, in order to be able to transfer the results of the game into practical managerial insights. In this competition, the participants make a software agent that plays the role of a PC manufacturer that needs to make decisions regarding procurement of raw materials, production of computers, and sale of these finished goods to consumers.

The Procurement Challenge

Over the years, the annual TAC SCM competition has become more and more competitive with contributions from top universities around the world. In order to push the science to the extreme, it is necessary to continuously change the rules of the game. While redesigning a game, one has to strike a very delicate balance between making a game too boring for incumbents, and too difficult for new entrants. In TAC SCM, we achieve this balance by introducing new *challenge games* that mimic smaller but more complex goals.

We discuss one such challenge in this paper - *the procurement challenge* where agents compete in securing profitable contracts for procuring raw materials from the suppliers. By limiting the agents' control to just the procurement of raw materials we are able to better analyze best practices in a more controlled setting. In (Andrews et al. 2007), the authors show that in the "baseline" TAC SCM game, the top agents make purchases with longer lead times. This is similar to the real world managerial insight that supply chain risks can usually be mitigated by adopting long-term procurement options versus one-off contract purchases. In order to further understand the tradeoffs between the uncertainty in demand in the long-term and uncertainty in supply in the short term, we introduced the option of negotiating long-term, quantity-flexible procurement contracts in addition to the one-off procurement contracts that are present in the "baseline" TAC SCM game.

We present relevant background work in the next section, and then present an extended description of the procurement challenge in following section. We then describe the approaches of the top three agents in the 2007 competition, and then present a detailed analysis of the results of the actual competition. We finally conclude by enlisting some insights we have gathered from this work that may be useful in real world supply chains.

Related Work

The work presented in this paper is closely related to three lines of research: i.) the development of other trading agent competitions, ii.) reports about supply chain trading agent design and analysis, and iii.) work on optimizing procurement decisions through analytical methods and simulations.

Related Work on Trading Agent Competitions

The original Trading Agent Competition (TAC) was first conceived in 1999 (Wellman and Wurman 1999) as a way to encourage research on automated trading in a competitive academic environment. The first tournament was held in July 2000, and centered around a travel scenario (Wellman et al. 2001). The travel game is now called TAC Classic and involves agents bidding against one another on the various components of travel packages to satisfy the demands of their simulated clients. The annual competitions led to the growth of a community centered around the topic of automated trading and significant developments in the understanding of the phenomena surrounding this topic (many of these developments are summarized in (Wellman, Greenwald, and Stone 2007)).

Building on the success of the original Trading Agent Competition, a new game scenario was introduced in 2003 that focuses on the challenges of automating a supply chain entity (Arunachalam and Sadeh 2005). The supply chain game, called TAC Supply Chain Management (TAC SCM), involves agents playing the role of PC manufacturers who buy components and sell assembled PCs to simulated customers. TAC SCM has cultivated a research community from over 60 different institutions and 20 different countries, and continues to produce research that is relevant to real-world supply chain entities (e.g. (Benisch, Andrews, and Sadeh 2006; Kiekintveld et al. 2007)).

In 2007 the TAC community introduced three new competitions. One was a completely new scenario called the TAC Market Design game (Cliff et al. 2007). This game requires entrants to develop automated market rules for matching simulated buyers and sellers. The other new competitions were challenges based on the same scenario as the TAC Supply Chain Management game. The TAC SCM challenges isolate specific components of the problem faced by agents competing in the full game. The TAC SCM Prediction Challenge (Pardoe 2007) focuses on the task of predicting and forecasting the stochastic supply and demand faced by a supply chain trading agent. The TAC SCM Procurement Challenge is the topic of this paper and was designed to isolate the procurement decisions faced by agents in the full TAC SCM game.

Related Work on Trading Agent Design and Analysis

The agent descriptions in this paper follow a long line of work describing successful agents for the TAC SCM scenario. For example, Benisch et. al. 2006 (Benisch et al. 2006) provides an in depth description of the different modules composing the CMieux agent. In Kiekintveld et. al. 2004 (Kiekintveld et al. 2004) the DeepMaize team describes how their agent dynamically coordinates sales, procurement and production strategies in an attempt to stay profitable. In (He et al. 2006) the SouthamptonSCM team presents their agent's strategy based on fuzzy reasoning. In (Pardoe and Stone 2004) the TacTex team describes machine learning techniques that were used to predict bid prices of other agents and offers considerable insight into the overall strategy behind their first-place agent in (Pardoe and Stone 2006) and (Pardoe and Stone 2007). Podobnik, Petric and Jezic describe the CrocodileAgent agent in (Podobnik, Petric, and Jezic 2006), (Petric, Podobnik, and Jezic 2007a) and (Petric, Podobnik, and Jezic 2007b), and PhantAgent is described by Stan et. al. in (Stan, Stan, and Florea 2006). The Botticelli team (Benisch et al. 2004) shows how the problems faced by TAC SCM agents can be modeled as mathematical programming problems, and offers heuristic algorithms for bidding on RFQs and scheduling orders. The RedAgent team (Philipp W. Keller 2004) presents an internal market architecture with simple heuristic-based agents that individually handle different aspects of the supply chain process.

Related Work on Supply Chain Optimization

The third line of related work involves optimizing supply chain decisions in a non-competitive setting by analyzing abstracted models of purchasing decisions or running simulations. Analytical techniques have come largely from the management science and operations research communities. A good overview of the work in this space is given by Lariviere (Lariviere 1999). This work typically attempts to characterize optimal replenishment policies (e.g. (Anupindi and Bassok 1999)) under various stochastic assumptions about supply and demand. The most closely related paper to our work is by Martinez-de-Albeniz and Simchi-Levi (de Albeniz and Simchi-Levi 2005). They address the problem of optimizing order quantities from a portfolio of flexible long-term contracts and spot market procurement opportunities when supply conditions are deterministic (e.g. suppliers do not refuse business and never default).

There have been a number of other simulation tools developed to analyze different aspects of supply chain performance. These simulations include software to evaluate different ways of re-engineering the supply chain (Swaminathan, Smith, and Sadeh 1998), determine the impact of different information exchange protocols on supply chain performance (Swaminathan, Sadeh, and Smith 1995), and understand the "bullwhip effect," (Lee, Padmanabhan, and Whang 1997) (i.e. the amplification of demand fluctuations as they travel through a supply chain).

TAC SCM Procurement Challenge

The TAC-SCM Procurement Challenge was introduced in 2007 to provide a competition platform designed to isolate the procurement decisions faced by manufacturer agents in the baseline TAC-SCM game (Arunachalam and Sadeh

2005; Collins et al. 2006). This challenge game also extends the space of procurement options available to the manufacturer agents by allowing them to enter long-term contracts with suppliers agents.

The TAC SCM Procurement Challenge (or "SCM-PC") was designed to promote the development of supply chain trading agents that are capable of effectively coordinating their procurement decisions. The game revolves around a personal computer (PC) assembly supply chain consisting of competing PC manufacturer agents, their component supplier agents and their customer agents. This challenge requires agents to manage supply chain risk by negotiating long-term, quantity-flexible procurement contracts and supplementing these contracts with one-off procurement contracts.

The SCM-PC game features three agents competing for supply contracts from ten different suppliers. Each game has one hundred simulated days, and each day lasts ten seconds of real time. A server simulates the customers and suppliers, and provides banking, production, and warehousing services to the individual agents. The agents receive messages from the server on a daily basis informing the state of the game, such as the current inventory of components, and must send responses to the same server until the end of the day indicating their actions, such as component orders to the suppliers. At the end of the game, the agent with the highest sum of money is declared the winner.

The long-term contracts are negotiated when the game starts, and each contract stipulates a minimum and maximum weekly quantity the agent commits to purchasing. Each day, the agents may also decide to procure additional components by negotiating one-off contracts. Each agent has an identical factory, where it can produce any type of computer. The factory is simulated by the game server, and also includes a warehouse for storing components and finished computers. A daily production and delivery schedule are also generated for the agent, and orders are only produced and delivered if the required components are available.

Customer Demand, Production and Delivery

On each day, the agents receive the exact same set of orders from customers representing one third of the total demand. Each order consists of a product type, a quantity, a due date and a price per unit. The customer demand is generated according to the same Poisson distribution as the baseline TAC-SCM game, with an average that is updated on a daily basis by a random walk. The server attempts to produce and deliver orders in a greedy fashion by giving priority to orders with higher revenue. When an order reaches the top of the queue the server checks whether or not each agent has enough components to produce it. Those agents with enough components exchange them for the revenue associated with the order.

Long-term Contracts

The long-term contracts are used to distribute risk between suppliers and agents. Each contract consists of a minimum (Q^{lts}_{min}) and maximum (Q^{lts}_{max}) weekly quantity the agent commits to purchasing, an execution price (p_{exec}) that the agent has to pay for each unit it actually purchases from the supplier, and a unit reservation price (p_{res}) that the agent has to pay independently of how much it actually orders. To ensure that each game presents a mix of long-term and one-off contract options, we assume that each component is available from a supplier that only offers long-term contracts and another one that only offers one-off contracts.

When the game starts, the agents have the option of negotiating the long-term contracts for each component, and these contracts are awarded based on second price auctions. The server first announces a reserve price (ρ) for each auction, and then waits ten seconds for the agents to submit their bids. Each bid consist of a requested maximum weekly quantity and an execution price that the agent is willing to pay for each component. The minimum weekly quantity and the reservation price are not specified by the agents in the bids, but are calculated by the suppliers as follows:

- $Q^{lts}_{min} = Q^{lts}_{max}/(1+\alpha)$, where α changes from one game to another and is announced at the start of each game.
- $p_{res}/(p_{res} + p_{exec}) = \beta$, where β also changes from one game to another and is announced at the start of each game.

The long-term contract supplier allocates 100% of its weekly nominal capacity (C^{nom}_{week}) to the bidding agents. Quantities are allocated based on the requested maximum weekly quantities, starting with the bid that has the highest execution price. Each agent's long-term contract has an execution price that is computed as the next highest price below its own bid ("second highest price" rule). The allocation proceeds until there are no bids left or until the long-term supplier has run out of capacity (based on its weekly nominal capacity). In the latter situation, the last manufacturer agent to receive a contract may end up with a maximum weekly quantity that is less than what it had requested.

At the beginning of any given week, each agent decides how much to actually order under its long-term contracts. If the total quantity requested by the agents exceeds the supplier's capacity, the supplier computes the ratio of demand it can satisfy based on its actual capacity. Each agent then receives a quantity that is proportional to this ratio, so that all agents with long-term contracts are treated equally and receive the same fraction of their actual demand that week.

One-off Contracts

Every day, manufacturer agents can send requests for quotes (RFQs) to suppliers with a given reserve price, quantity, type and delivery date. A supplier receives all RFQs on a given day, and processes them together at the end of the day to find a combination of offers that approximately maximizes its revenue. On the following day, the suppliers send back to each agent an offer for each RFQ with a price, a possibly adjusted quantity, and a due date. A detailed description of the one-off contract negotiation is presented in (Collins et al. 2006).

The suppliers have a limited capacity for producing a component, and this limit varies throughout the game according to a mean reverting random walk. Moreover, suppliers also limit their long-term commitments by reserving some capacity for future business. The pricing of components is based on the ratio of demand to supply, and higher ratios result in higher prices. Each day the suppliers estimate their free capacity by scheduling production of components ordered in the past and components requested that day as late as possible. The price offered in response to an RFQ is equal to the requested components base price discounted by a function proportionate to the supplier's free capacity before the RFQ due date. The manufacturer agents normally face an important trade-off in the procurement process: pre-order components for the future where customer demand is difficult to predict, or wait to purchase components and risk being unsuccessful due to high prices or availability.

TAC SCM-PC Agents

This section describes the approaches of the top three SCM-PC agents: PhantAgent (University "Politehnica" of Bucharest), CMieux (Carnegie Mellon University) and CrocodileAgent (University of Zagreb). Each of these agents use a different combination of long-term and one-off contract strategies. One of the primary differences between the agents is the way that future demand is predicted and how this prediction is used to create long-term procurement strategies. Another primary difference is how one-off contracts are handled, with PhantAgent and CrocodileAgent using repeated queries with fixed lead times and CMieux varying its lead times between queries.

PhantAgent

PhantAgent divides its decision making process into three different sub-problems: calculating needed components, handling long-term contract procurement, and generating one-off contract orders. Each of these problems is solved using relatively simple heuristics which we will now describe in detail. Many of the heuristics used in PhantAgent rely on external parameters which can be optimized by analyzing historical performance. However, due to limited availability of historical data prior to the 2007 SCM-PC competition these parameters were set largely by hand.

Calculating Needed Components At the beginning of each day, PhantAgent estimates the number of components it will need from the current day to the end of the simulation. In order to determine this number, PhantAgent first estimates the number of components it expects to have in inventory on each of the remaining days. The expected inventory levels are estimated by iterating through each day, adding component arrivals that are due and subtracting the estimated component usage. The main difficulty in this process involves determining a good estimate of each day's component usage. We will refer to the usage of component j on the current day d as $Q_{(d,j)}$. To estimate $Q_{(d,j)}$ PhantAgent combines two heuristic values:

- The first heuristic value, $E[Q_j]$, assumes a fixed usage each day based on the expected demand as described in the simulation parameters.
- The second, $\bar{Q}_{(d,j)}$, is a moving average of component usage from the past 10 days.

Both of these heuristic values have certain weaknesses. The problem with using $E[Q_j]$ is that it is not flexible to demand variations and fails to account for fluctuations in demand throughout the game. By ignoring such fluctuations the agent will often either run out of components or be left with excess inventory when the simulation nears its end. The problem with using $\bar{Q}$ is that demand at the beginning of the simulation can be significantly different than the demand at the end. Thus, the long lead time orders placed at the beginning of the game based on the demand at that time may not match the demand when the components arrive. To avoid these problems PhantAgent uses a weighted average of both heuristic values, where $E[Q_j]$ is weighted more heavily during the beginning of the game. The formula is given in Equation 1.

$$\widehat{Q}_{(d,j)} = \frac{D-d}{D} E[Q_j] + \frac{d}{D} \bar{Q}_{(d,j)} \tag{1}$$

where:
D - the total number of days in the simulation.

The result is then slightly scaled down to avoid excess inventory towards the end of the game due to changes in demand when the long lead time requests were made.

Handling Long-term Contract Procurement PhantAgent prioritizes availability over price in the long-term contracts. In order to capitalize on high selling prices during time of low availability, it was empirically determined that bidding the average one-off contract prices from the past several games enabled the agent to reliably procure the quantity it desired.

Throughout the game PhantAgent exercises the option to increase weekly order quantities if there is a need for components and the one-off contract suppliers are charging more than the long-term contract prices.

Generating One-off Contract Orders For one-off contract requests, PhantAgent uses all 5 RFQs each day to request components with fixed lead times (for SCM-PC in 2007, values used were $\{2, 3, 10, 25, 45\}$ days). The agent adjusts its requested quantities according to current market conditions, so on some occasion it might not use some of the RFQs. This fixed lead time strategy usually allows the agent to procure components consistently throughout the simulation while paying a price that is close to the average paid by any agent.

Requests with very short lead times (such as lead times of 2 and 3 days) are treated independently of the other RFQs and are used primarily to maintain a steady stock of components. These requests have been observed to vary significantly in price from one day to another (during low and high demand periods, the prices of RFQs with very short lead times are usually very low and very high respectively). Handling these daily variations is the main concern here and for this the reserve price mechanism is used to only accept offers with good prices.

Generating Long Lead Time Requests with One-off Contracts Since PhantAgent uses fixed lead times for all one-off contract requests, the main decision regarding long lead time requests is choosing appropriate order quantities. The general principle governing this decision is to make long lead time orders only if they are expected to be better or equal to orders with short lead times. In most cases the order with longer lead times have the lowest prices. However, orders with shorter lead times towards the end of the game are important due to the scaling down applied to the estimated component usage. Order quantities are chosen simply based on the difference between the expected usage and the components already expected from prior procurement.

CMieux

The strategy used by CMieux for the long-term contract negotiation and procurement is described in the next sub-section. The following sub-section then presents CMieux's negotiating strategy for the one-off contracts, which is essentially the same used in (Benisch et al. 2006).

Long-term Contract Negotiation and Procurement The negotiation of the long-term contracts is conducted on the first day of each game, and each agent receives a reserve price p_{res}, an α and a β before it computes a bid for each component. The main challenge the agent faces when computing a bid, composed of a desired maximum weekly quantity and a bid price, is the high uncertainty it has about the customer demand and the prices of components in the one-off contract markets. Thus, the strategy used by CMieux relies on average one-off contract prices ($\bar{P}_{one-off}$) from previous games to compute a bid price for p_{exec}. The formula is given in Equation 2.

$$p = max(\bar{P}_{one-off}, \rho + increment) \quad (2)$$

The first value of the max function in Equation 2 computes a bid price (p) for low reserve prices (smaller than a threshold, $\bar{P}_{one-off}$). Thus, when reserve prices are low the agent is willing to buy components from the long-term contract suppliers for a price equal to the average one-off contract price. The second value of the max function in Equation 2 computes a bid price for high reserve prices (when $\rho \geq \bar{P}_{one-off}$), and sets p to be very close to ρ.

The requested maximum weekly quantity in each bid is considered a parameter in the model that must be adjusted to reflect the amount of risk the agent is willing to take with the long-term contracts. Empirically, it was determined that on average the loss from unsold inventory in low customer demand games outweighed the profits from sales in higher demand games. This problem occurred due to the very low flexibility (approximately 15%) between the minimum and maximum weekly quantities, which was not large enough to cover all the different customer demand scenarios. Thus, a more conservative strategy was adopted and the requested maximum weekly quantity was adjusted to suit low demand games.

One-off Contract Procurement The module is designed to rapidly adapt to changing market conditions and exploit gaps in the one-off contract market to ensure that its procurement prices tend to fall below its competitors. Each day, the procurement module performs two tasks: i.) it attempts to identify a particularly promising subset of current supplier offers, and ii.) it constructs a combination of RFQs to be sent to suppliers that balances the agent's component needs with identified gaps in current supplier contracts.

Accepting Supplier Offers The module accepts supplier offers using a rule-based decision process. The agent begins by selecting offers that are satisfactory based on price, quantity and due date using historical data. In an effort to keep the agent's reputation as high as possible, the agent first accepts offers that satisfy the quantity and due date requirements of the corresponding RFQ ("full offers"). Next, if still needed, satisfactory offers with relatively large quantities ("partial offers"), or early due dates ("earliest complete offers") are also accepted.

Sending Supplier Requests In order to determine the amount of components needed, the procurement module computes the difference between the inventory required to maintain production levels specified by the customer orders, and the projected inventory from the long-term and one-off contract suppliers for the remainder of the game. However, CMieux does not *need* to procure this entire difference each day. The components are not needed immediately, thus it can divide the purchasing of components across several days. This will not only enable the agent to aggressively procure components within a specific scheduling window, but also allow the agent to buy some of the components it needs well in advance, when they are likely to be cheapest.

The process of computing what specific requests to send to suppliers is then decomposed by component type. For each component type, the procurement module generates several sets of lead times and searches for the set with the highest utility. This utility is computed by approximating the sum of the utility of the components they request and subtracting their forecast prices. The sets of lead times with the greatest utility for each component are sent as RFQs to the appropriate suppliers. The reserve price of each RFQ is set to be the average utility of the components it includes.

CrocodileAgent

CrocodileAgent's architecture is based on incorporating a generic intelligent software agent model (Podobnik, Trzec, and Jezic 2007) into the IKB framework (Vytelingum et al. 2005), a three layered agent-based framework for designing strategies in electronic trading markets. The following sections describe the approach for handling long-term contract negotiation and one-off contract procurement.

Negotiating Long-term Contracts At the start of the game the CrocodileAgent negotiates long-term contracts for each component. After the suppliers announce reserve prices for each component CrocodileAgent calculates the maximum weekly quantity it will request. The requested quantity, $\widehat{Q}_j$, for each component j is a linear function of its reserve price (ρ, where ρ is represented as a fraction of the component's base price). The reserve price may assume any value in the interval $[\rho_{min}, \rho_{max}]$, so $\widehat{Q}_j$ is calculated as shown in Equation 3.

$$\widehat{Q}_j = (\frac{\rho_{max} - \rho}{\rho_{max} - \rho_{min}})Q_j^{max} + (1 - \frac{\rho_{max}-\rho}{\rho_{max}-\rho_{min}})Q_j^{min} \quad (3)$$

where:
Q_j^{min}, Q_j^{max} - the minimum and maximum quantity requested for component j.

After the maximum weekly quantities have been determined, CrocodileAgent submits a long-term contract bid for each component with price equal to the reserve price ($p = \rho$).

Each week CrocodileAgent chooses an actual quantity to order (Q_j^{order}) for each long-term contract based on the amount of components it has in inventory (N_j). This quantity increases linearly from a fixed constant σ% to 100% of the maximum quantity in the long-term contract (Q_{max}^{lts}) as the number of components in inventory decreases. The formula is given in Equation 4.

$$Q_j^{order} = \min(Q_{max}^{lts} \times \frac{N_j^{+} - \sigma\% \times N_j^{-} + (\sigma\% - 100\%) \times N_j}{N_j^{+} - N_j^{-}}, Q_{max}^{lts}) \quad (4)$$

where:
N_j^{-}, N_j^{+} - minimum and maximum inventory level for component j,

Negotiating One-off Contracts CrocodileAgent breaks up the procurement problem with one-off contracts into two different sub-problems: strategy on the first day (*day 0*) and replenishment of components during the rest of the game. A close examination of the baseline TAC SCM Game rules (Collins et al. 2006) (which also defines the SCM-PC one-off contract supplier model) suggests that procurement of components at the very beginning of the game (*day 0* procurement) can provide components with low prices throughout the game (because there is no prior component demand). Although the concept of *day 0* procurement strategy has some similarities with long-term contract negotiation, these two procurement strategies are totally independent because each component is available from two different suppliers: one that only offers long-term contracts and one that only sells components with one-off contracts.

***Day 0* Strategy with One-off Contracts** The goal of using the *day 0* strategy is to acquire components for the beginning of the game and (if possible) buy cheap components with longer lead times. CrocodileAgent sends a fixed set of five RFQs on *day 0*. The parameters for *day 0* procurement were determined by conducting a series of experiments.

Component Purchase During the Game with One-off Contracts Throughout the game CrocodileAgent uses the one-off contracts to fill in gaps in its inventory not covered by the long-term contracts or *day 0* strategy. At the beginning of each day, the agent calculates the quantity of each component that has been previously ordered but not yet delivered ($Q_j^{outstanding}$). This quantity is multiplied by a fixed distance factor, so that orders with longer lead times have a smaller weight than orders with shorter lead times. If the quantity in inventory plus the outstanding quantity for a component is less than a fixed threshold, a more aggressive strategy is used where five RFQs are sent to the supplier with short lead times and relaxed reserve prices. Otherwise, RFQs are sent to suppliers with fixed lead times so that it can replenish its inventory without exceeding a maximum amount.

The one-off contract suppliers sometimes offer bargains on very short lead times (e.g. lead times of 2 or 3 days). In order to capitalize on these bargains, CrocodileAgent sends RFQs with short lead times and low reserve prices during the first few days of the game and as long as its outstanding components are not more than a certain percentage above an acceptable maximum.

2007 TAC SCM-PC Results and Analysis

This section presents the results of the final rounds of the 2007 TAC-SCM Procurement Challenge. The final rounds were held at the Twenty-Second Conference on Artificial Intelligence (AAAI-07). They featured twelve games, three games for each combination of three agents out of the four finalists. The final standings are presented in Table 1, the value in the third column is the average profit accumulated by each agent over the course of the nine games it played in.

Table 1: Final Standings for the 2007 SCM-PC

Agent	Games Played (out of 12)	Number of Games Won	Average Score
PhantAgent	9	4	8,731,513
CMieux	9	6	7,405,743
CrocodileAgent	9	2	6,399,115
Warrior	9	0	4,200,440

In addition to the overall competition results we also performed a finer pairwise comparison of the top three agents to account for the fact that they did not all participate in the same games. Table 2 presents the performance of pairs of agents in all of the games involving them both. For the top three agents there are three distinct pairs and each pair participated in six common games. As can be seen, the pairwise results provide a different ranking with CMieux ahead of the other two agents and PhantAgent ahead of CrocodileAgent. It is also worth noting that these results are consistent with the number of games won by each agent throughout the finals, with CMieux winning 6 out 9 games, PhantAgent 4 out of 9 games and CrocodileAgent 2 out of 9 games.

The discrepancy between the overall rankings and pairwise rankings can be explained by the varying demand conditions faced by agents in different games. PhantAgent achieved a higher overall score than CMieux because it participated in one game with a high customer demand, and was able to successfully take advantage of this opportunity.

Table 2: Final "Pairwise" Standings for the 2007 SCM-PC

Agent	Games Played	Number of Games Won	Average Score
CMieux	6	4	7,149,838
PhantAgent	6	2	6,788,197

Agent	Games Played	Number of Games Won	Average Score
CMieux	6	4	8,286,761
CrocodileAgent	6	1	4,385,217

Agent	Games Played	Number of Games Won	Average Score
PhantAgent	6	3	10,027,071
CrocodileAgent	6	1	7,096,601

We will now present analysis of several important aspects of the game as well as graphs that illustrate the effect of the strategies adopted by the top three agents. The following sub-sections describe the sales volume of each agent, the one-off and long-term contract mixes, and the average procurement costs.

Customer Orders and Deliveries

To measure the sales volume of the top three SCM-PC agents, we calculated their *realized demand percentage*, or the fraction of the total possible demand that they were able to satisfy. Figure 1 presents a pairwise comparison of the average realized demand (with 95% confidence intervals) of the top three agents. As in Table 2, the values shown for each pair in Figure 1 are calculated using only the games that pair participated in. CrocodileAgent had the highest average realized demand amongst all agents. However, the overlapping confidence intervals show that there was no statistically significant difference between any of the agents.

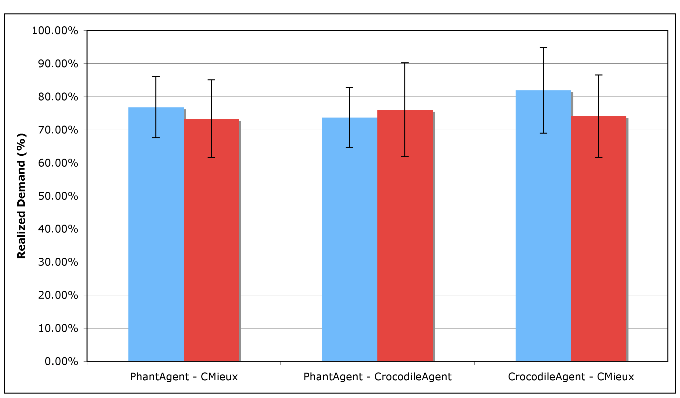

Figure 1: Average Realized Demand

Quantity Ordered from the Suppliers

The average number of components ordered by each of the top three agents from the one-off and the long-term contract suppliers with 95% confidence intervals is presented in Figures 2 and 3. These graphs show that all three of the top agents procure a substantial amount of components from the more stable long-term market, but tend to buy significantly more from the one-off contract market. While long-term contracts provide some amount of flexibility in the weekly orders, they are negotiated when the agents have no information about the customer demand. It is not surprising that these agents chose to rely more on one-off contracts since they can be negotiated on a daily basis giving them more flexibility to adapt to varying market conditions.

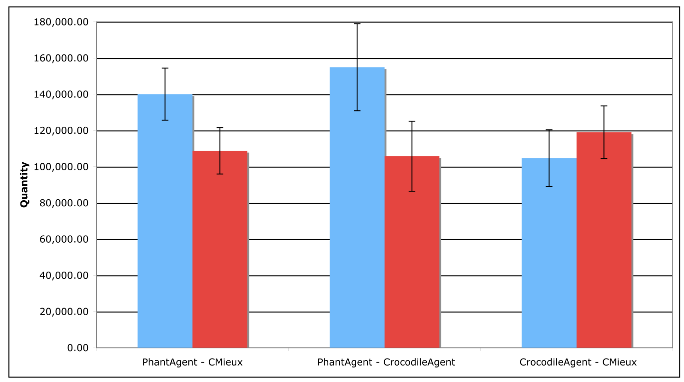

Figure 2: Average Number of Components Ordered from the One-off Contract Suppliers

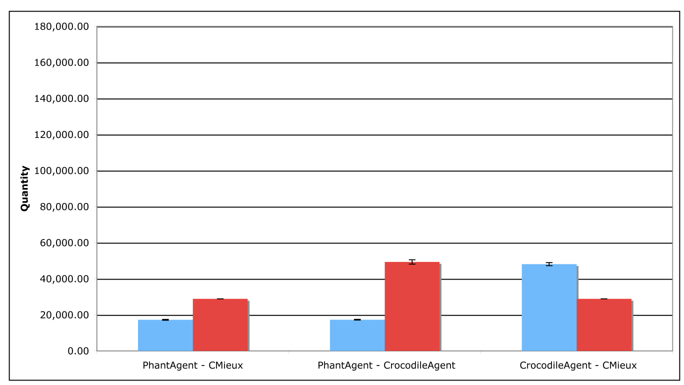

Figure 3: Average Number of Components Ordered from the Long-term Contract Suppliers

Component Prices

Figure 4 presents the average weighted prices of components purchased by each agent in both the long-term and one-off contract markets combined (with 95% confidence intervals). The graph shows that CMieux's procurement prices were significantly better than the other two agents when compared across both markets.

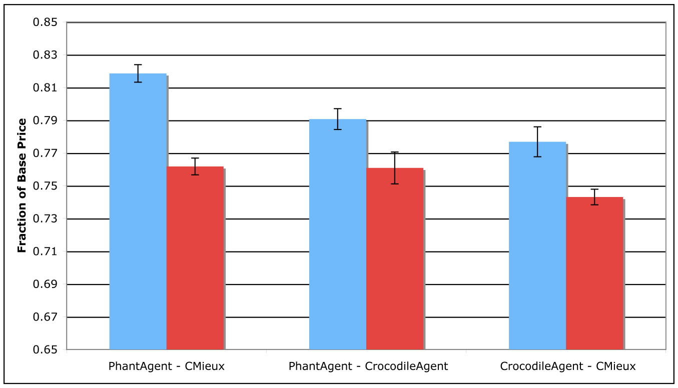

Figure 4: Average Weighted Prices of Components from Long-term and One-off Contracts

When we examine the one-off and long-term contract markets separately we see that CrocodileAgent was able to get significantly lower prices for long-term contracts than the other two agents. CMieux's long-term prices were a close second to CrocodileAgent with an average difference of 1.69%. In the one-off contract market CMieux has a significant edge over the other two agents.

CrocodileAgent's low long-term procurement costs can be explained by the fact that it bids the lowest possible price for all of the long-term contracts (as described in the previous section). However, the fact that CMieux had the lowest overall procurement costs suggests that CrocodileAgent was not able to procure enough from the long-term market to overcome CMieux's better prices in the one-off contract market. CMieux's dynamic one-off contract strategy for optimizing RFQs each day was more effective than the fixed procurement strategies of the other agents. The additional flexibility provided an advantage leading to the lowest overall average procurement costs.

Conclusions and Future Work

This paper began with a description of the Supply Chain Trading Agent Competition Procurement Challenge (SCM-PC). In addition to isolating the procurement decisions faced by agents in the "baseline" Supply Chain Trading Competition, the SCM-PC rules extended the purchasing options to include quantity flexible long-term contracts that are negotiated once at the start of the game.

We then described the approaches of the top three SCM-PC agents from the 2007 competition: PhantAgent (University "Politehnica" of Bucharest), CMieux (Carnegie Mellon University) and CrocodileAgent (University of Zagreb). These agents were shown to differ primarily in the ways they predicted future demand and the flexibility they employed in their one-off contract procurement. In particular, PhantAgent and CrocodileAgent used repeated queries with fixed lead times, while CMieux varied its lead times between queries.

Finally, we presented a detailed analysis of the results from the actual competition. The results showed that the agents used long-term procurement contracts to procure a baseline inventory, but purchased the bulk of their components with one-off contracts. This suggests that the existence of a flexible one-off contract market enabled the agents to mitigate the risk typically associated with long-term commitments.

One potential short-coming of the results from the 2007 SCM-PC competition is that the agents and agent designers had very little historical experience to learn from. This lack of historical data may have made the game less dynamic and slightly inefficient. In the future we plan to provide tools allowing agents to analyze information from previous game logs.

Another future change will involve improving the competition structure itself. Our analysis of the results suggested that CMieux was purchasing components significantly cheaper than the other two agents, while maintaining similar service levels. However, due to an imbalance in game conditions CMieux placed behind PhantAgent in overall average profit. A closer look at the reasons for this discrepancy revealed a flaw in the competition structure. Due to the high variance in customer demand between games the agents should not have been compared across games in which they did not all compete. Our attempts to control for this after the fact (by comparing performance between agents only in games where they were both present) appeared to correct the discrepancy.

We originally proposed the TAC SCM Procurement Challenge in order to better analyze best practices in procurement in a more controlled setting. Largely, we believe we have made significant progress towards this goal and have gained important insights about automated supply chain procurement markets. In 2008, we have decided to allow each supplier to offer both one-off and long-term contracts, rather than restricting each supplier to offer only one type of contract. This is consistent with practices found in many actual supply chains and also somewhat more challenging.

Acknowledgments

The platform for the procurement challenge is based on the baseline Supply Chain Trading Agent Competition game originally designed by Carnegie Mellon's e-Supply Chain Management Laboratory in collaboration with the Swedish Institute of Computer Science (SICS). The first three authors, who are with the e-Supply Chain Management Laboratory, would like to thank SICS for sharing its code and allowing them to customize it for the Procurement Challenge described in this paper. This work has been supported by the National Science Foundation under ITR Grant 0205435 as well as by a grant from SAP Research.

References

Andrews, J.; Benisch, M.; Sardinha, A.; and Sadeh, N. 2007. What differentiates a winning agent: An information gain based analysis of TAC-SCM. In *Proceedings of Trading Agent Design and Analysis Workshop*.

Anupindi, R., and Bassok, Y. 1999. Supply contracts with quantity commitments and stochastic demand. In Tayur, S.;

Ganeshan, R.; and Magazine, M., eds., *Quantitative Models for Supply Chain Management*. Kluwer Academic Publisher.

Arunachalam, R., and Sadeh, N. 2005. The supply chain trading agent competition. *Electronic Commerce Research and Applications* 4(1):66–84.

Benisch, M.; Andrews, J.; and Sadeh, N. 2006. Pricing for customers with probabilistic valuations as a continuous knapsack problem. In *Proceedings of ICEC'06*.

Benisch, M.; Greenwald, A.; Grypari, I.; Lederman, R.; Naroditsky, V.; and Tschantz, M. 2004. Botticelli: A supply chain management agent. In *Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*.

Benisch, M.; Andrews, J.; Sardinha, A.; and Sadeh, N. 2006. CMieux: Adaptive strategies for competitive supply chain trading. In *Proceedings of ICEC'06*.

Cliff, D.; Gerding, E.; Niu, J.; McBurney, P.; Moyaux, T.; and Miller, T. 2007. TAC Market Design Game (TAC CAT). Website. http://www.marketbasedcontrol.com/blog/index.php?page_id=5.

Collins, J.; Arunachalam, R.; Sadeh, N.; Eriksson, J.; Finne, N.; and Janson, S. 2006. The supply chain management game for the 2007 trading agent competition. CMU-ISRI-07-100, Carnegie Mellon University.

de Albeniz, V. M., and Simchi-Levi, D. 2005. A portfolio approach to procurement contracts. *Production and Operations Management* 14(1):90–114.

He, M.; Rogers, A.; Luo, X.; and Jennings, N. R. 2006. Designing a successful trading agent for supply chain management. In *Proceedings of AAMAS'06*.

Kiekintveld, C.; Wellman, M. P.; Singh, S.; Estelle, J.; Vorobeychik, Y.; Soni, V.; and Rudary, M. 2004. Distributed feedback control for decision making on supply chains. In *Fourteenth International Conference on Automated Planning and Scheduling*.

Kiekintveld, C.; Miller, J.; Jordan, P.; and Wellman, M. P. 2007. Forecasting market prices in a supply chain game. In *International Joint Conference on Autonomous Agents and Multiagent Systems*.

Lariviere, M. 1999. Supply chain contracting and coordination with stochastic demand. In Tayur, S.; Ganeshan, R.; and Magazine, M., eds., *Quantitative Models for Supply Chain Management*. Kluwer Academic Publisher.

Lee, H. L.; Padmanabhan, V.; and Whang, S. 1997. The bullwhip effect in supply chains. *Sloan Management Review* 93–102.

Pardoe, D., and Stone, P. 2004. Bidding for customer orders in TAC SCM. In *AAMAS-04 Workshop on Agent-Mediated Electronic Commerce*.

Pardoe, D., and Stone, P. 2006. Predictive planning for supply chain management. In *Proceedings of Automated Planning and Scheduling'06*.

Pardoe, D., and Stone, P. 2007. An autonomous agent for supply chain management. In Adomavicius, G., and Gupta, A., eds., *Handbooks in Information Systems Series: Business Computing*. Elsevier.

Pardoe, D. 2007. TAC SCM Prediction Challenge. Digitaly Distributed. http://www.cs.utexas.edu/ TacTex/PredictionChallenege/predictionChallenge.pdf.

Petric, A.; Podobnik, V.; and Jezic, G. 2007a. The CrocodileAgent 2005: An overview of TAC SCM agent. *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science* 4452:219–233.

Petric, A.; Podobnik, V.; and Jezic, G. 2007b. The CrocodileAgent: Designing a robust trading agent for volatile e-market conditions. *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science* 4496:597–606.

Philipp W. Keller, Felix-Olivier Duguay, D. P. 2004. RedAgent-2003: An autonomous, market-based supply-chain management agent. In *Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*.

Podobnik, V.; Petric, A.; and Jezic, G. 2006. The CrocodileAgent: Research for efficient agent-based cross-enterprise processes. *Lecture Notes in Computer Science* 4277:752–762.

Podobnik, V.; Trzec, K.; and Jezic, G. 2007. Context-aware service provisioning in next-generation networks: An agent approach. *International Journal of Information Technology and Web Engineering* 2(4):41–62.

Stan, M.; Stan, B.; and Florea, A. M. 2006. A dynamic strategy agent for supply chain management. In *SYNASC*.

Swaminathan, J. M.; Sadeh, N. M.; and Smith, S. F. 1995. Informaiton exchange in the supply chain. CMU-RI-95-36, Carnegie Mellon University.

Swaminathan, J. M.; Smith, S.; and Sadeh, N. M. 1998. Modeling supply chain dynamics: A multiagent approach. *Decision Sciences* 29:607–632.

Vytelingum, P.; Dash, R. K.; He, M.; and Jennings, N. R. 2005. A framework for designing strategies for trading agents. In *Proceedings of the IJCAI Workshop on Trading Agent Design and Analysis (TADA'05)*, 7–13.

Wellman, M. P., and Wurman, P. R. 1999. A trading agent comeptition for the research community. In *IJCAI-99 Workshop on Agent-Mediated Electronic Trading*.

Wellman, M. P.; Wurman, P. R.; O'Malley, K.; Bangera, R.; de Lin, S.; Reeves, D.; and Walsh, W. E. 2001. Designing the market game for a trading agent competition. *IEEE Internet Computing* 5(2):43–51.

Wellman, M. P.; Greenwald, A.; and Stone, P. 2007. *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition*. MIT Press.