In *Proceedings of the Twenty-First National Conference on Artificial Intelligence* (AAAI 06), Boston, MA, July 2006.

TacTex-05: A Champion Supply Chain Management Agent

David Pardoe and **Peter Stone**
Department of Computer Sciences
The University of Texas at Austin
dpardoe, pstone@cs.utexas.edu

Abstract

Supply chains are ubiquitous in the manufacturing of many complex products. Traditionally, supply chains have been created through the interactions of human representatives of the companies involved, but advances in autonomous agent technologies have sparked an interest in automating the process. The Trading Agent Competition Supply Chain Management (TAC SCM) scenario provides a unique testbed for studying supply chain management agents. This paper introduces TacTex-05 (the champion agent from the 2005 competition), describes its constituent intelligent components, and examines the success of the complete agent through analysis of competition results and controlled experiments.

Introduction

In today's industrial world, supply chains are ubiquitous in the manufacturing of many complex products. Traditionally, supply chains have been created through the interactions of human representatives of the various companies involved. However, recent advances in autonomous agent technologies have sparked an interest in automating the process.

Creating a fully autonomous agent for supply chain management is difficult due to the large number of tasks such an agent must perform. At a high level, the agent must procure resources for, manage the assembly of, and negotiate the sale of a completed product. To perform these tasks intelligently, the agent must be able to plan in the face of uncertainty, schedule the optimal use of its resources, and adapt to changing market conditions. The Trading Agent Competition Supply Chain Management (TAC SCM) scenario provides a unique testbed for studying and prototyping such agents by providing a competitive environment in which independently created agents can be tested against each other over the course of many simulations in an open academic setting.

In this paper, we describe TacTex-05, the winner of the 2005 TAC SCM competition. In particular, we describe the various intelligent components composing the agent and discuss how they are combined to result in an effective supply chain management agent. The remainder of this paper is organized as follows. We first summarize the TAC SCM scenario, and then give an overview of the design of TacTex-05. Next, we describe in detail the individual intelligent components: three predictive modules, two decision-making modules that attempt to identify optimal behavior with respect to the predictions, and two methods of adapting to opponent behavior based on past games. Finally, we examine the success of the complete agent, through both analysis of competition results and controlled experiments.

We would like to thank Jan Ulrich and Mark VanMiddlesworth for contributing to the development of TacTex-05. This research was supported in part by NSF CAREER award IIS-0237699.

The TAC Supply Chain Management Scenario

In this section, we provide a summary of the TAC SCM scenario. Full details are available in the official specification document.[1] In a TAC SCM game, six agents act as computer manufacturers in a simulated economy managed by a game server. The length of a game is 220 simulated days, with each day lasting 15 seconds of real time. The game can be divided into three parts: i) component procurement, ii) computer sales, and iii) production and delivery, as detailed in the remainder of this section.

Component Procurement

The computers are made from four components: CPUs, motherboards, memory, and hard drives, each of which come in multiple varieties. From these components, 16 different computer configurations can be made. Agents must purchase these components from a set of suppliers managed by the game server.

Agents wanting to purchase components send requests for quotes (RFQs) to suppliers indicating the *type* and *quantity* of components desired and the *date* on which they should be delivered. Agents may send at most 5 RFQs per component per supplier each day. Suppliers respond to RFQs by offering a price for the requested components if the request can be satisfied. Agents may then accept or reject the offers.

Suppliers have a *limited capacity* for producing components, and this capacity varies throughout the game according to a random walk. The price offered in response to an RFQ depends on the fraction of the supplier's capacity that is free before the requested due date.

Computer Sales

Customers wishing to buy computers send the agents RFQs consisting of the *type* and *quantity* of computer desired, the *due date*, a *reserve price* indicating the maximum amount the customer is willing to pay per computer, and a *penalty* that must be paid if the delivery is late. Agents respond to the RFQs by bidding in a first-price auction: the agent offering

[1] www.sics.se/tac/tac05scmspec_v157.pdf

the lowest price on each RFQ wins the order. The number of RFQs sent by customers each day depends on the level of customer demand, which fluctuates throughout the game.

Production and Delivery

Each agent manages a factory where computers are assembled. Factory operation is constrained by both the components in inventory and assembly cycles. Each day an agent must send a production schedule and a delivery schedule to the server indicating its actions for the next day. The production schedule specifies how many of each computer will be assembled by the factory, while the delivery schedule indicates which customer orders will be filled from the completed computers in inventory.

Overview of TacTex-05

We begin by presenting a high-level overview of TacTex-05. Details on specific agent components are contained in the sections that follow. Figure 1 illustrates the basic components of TacTex-05 and their interaction. There are five basic tasks a TAC SCM agent must perform: i) sending RFQs to suppliers to request components, ii) deciding which offers from suppliers to accept, iii) bidding on RFQs from customers requesting computers, iv) sending the daily production schedule to the factory, and v) delivering completed computers. We assign the first two tasks to a *Supply Manager* module, and the last three to a *Demand Manager* module. The Supply Manager handles all planning related to component inventories and purchases, and requires no information about computer production except for a projection of component use over a future period, which is provided by the Demand Manager. The Demand Manager, in turn, handles all planning related to computer sales and production. The only information about components required by the Demand Manager is a projection of the current inventory and future deliveries, along with an estimated replacement cost for each component used. This information is provided by the Supply Manager.

We view the tasks to be performed by these two managers as optimization tasks: the Supply Manager tries to minimize the cost of obtaining the components required by the Demand Manager, while the Demand Manager seeks to maximize the profits from computer sales subject to the information provided by the Supply Manager. In order to perform these tasks, the two managers need to be able to make predictions about the results of their actions and the future of the economy. TacTex-05 uses three predictive models to assist the managers with these predictions: a *Supplier Model*, a *Demand Model*, and an *Offer Acceptance Predictor*.

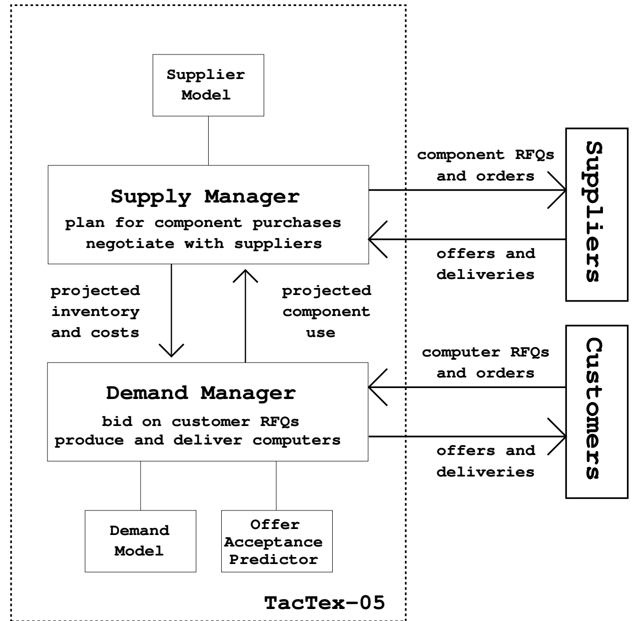

Figure 1: Agent overview

The Predictive Models

We first describe the functions of the three predictive models, and we then explain their use in the section that follows.

Demand Model

When planning for future computer production, the Demand Manager needs to be able to make predictions about future demand. The Demand Model is responsible for making these predictions, and does so using an approach introduced by the agent DeepMaize in 2003 (and fully described in (Kiekintveld *et al.* 2004)). Basically, this is a Bayesian approach that involves maintaining a probability distribution over the parameters used in the game server's algorithm for generating customer demand. Using this information, it is possible to project expected future demand.

Offer Acceptance Predictor

In order to bid on customer RFQs, the Demand Manager needs to be able to predict the orders that will result from the offers it makes. The Offer Acceptance Predictor makes these predictions possible. For each customer RFQ received, the Offer Acceptance Predictor generates a function mapping the possible offer prices to the probability of the customer accepting the offer. (The function can thus be viewed as a cumulative distribution function.) In (Pardoe & Stone 2004) we explored the possibility of learning to generate these functions based on past games. In TacTex-05, however, we use a simpler approach adapted from the method used by the agent Botticelli in 2003 (Benisch *et al.* 2004). Essentially, a linear function is generated for each computer type by performing regression on data points representing recent prices offered by TacTex-05 along with the resulting acceptance rate.

Supplier Model

The Supplier Model keeps track of all information sent to and received from suppliers. This information is used to model the state of each supplier, allowing the Supplier Model to predict the price that a supplier will offer in response to an RFQ with a given quantity and due date.

Recall that the price offered in response to an RFQ requesting delivery on a given day is determined entirely by the fraction of the supplier's capacity that is committed through that day. As a result, the Supplier Model can compute this fraction from the price offered. With enough offers, the Supplier Model can form a reasonable estimate of the fraction of capacity committed by a supplier on any single day. For each supplier and supply line, the Supply Manager maintains an estimate of free capacity, and updates this estimate daily based on offers received. Using this estimate, the Supplier Model is able to make predictions on the price a supplier will offer for a particular RFQ.

The Decision-Making Modules

We now describe the Demand and Supply Managers, which are responsible for making use of the predictions provided by the predictive models in deciding which actions TacTex-05 should take each day.

The Demand Manager

The Demand Manager is responsible for bidding on customer RFQs, producing computers, and delivering them to customers. All three tasks are performed using the same greedy production scheduling algorithm. (A greedy algorithm is used due to the 15 second time limit per game day.) As these tasks compete for the same resources (components, completed computers, and factory cycles), the Demand Manager begins by planning to satisfy existing orders, and then uses the remaining resources in planning for RFQs. Due to the range of due dates that customers may specify in RFQs, at most 10 days will be available in which to complete production of requested computers, so the Demand Manager always plans for the next 10 days of production.

The Demand Manager begins each day by initializing its production resources using the values provided by the Supply Manager. The production scheduler is then applied to existing orders, and orders that are due immediately and can be filled from inventory are scheduled for delivery.

Next, the Demand Manager tries to identify the set of bids in response to customer RFQs that will maximize the expected profit from using the remaining production resources for the next 10 days. This profit depends not only on the RFQs being bid on on the current day, but also on RFQs that will be received on later days for computers due during the period. The Demand Manager therefore generates (in a random manner equivalent to what the game server does) a predicted set of all RFQs that will be received for computers due during the period. The number of RFQs generated per future day is determined by the Demand Model's projection of future customer demand. Bids for these RFQs are chosen at the same time as those for the actual RFQs from the current day, effectively causing a portion of the remaining production resources to be reserved for the actual RFQs that will be received in the future.

Once the predicted RFQs are generated, the Offer Acceptance Predictor is used to generate an acceptance prediction function for every RFQ, both real and predicted. The Demand Manager then considers the production resources remaining, set of RFQs, and set of acceptance prediction functions and simultaneously generates a set of bids on RFQs and a production schedule that produces the expected resulting orders. This process involves the use of a variation of our greedy production scheduler in which expected order quantities (where the expected quantity ordered for an RFQ is the probability of acceptance times the actual quantity requested) are considered. Full details are available in (Pardoe & Stone 2004).

After applying the production scheduler to the current orders and RFQs, the Demand Manager is left with a 10-day production schedule and a set of bids for the actual and predicted RFQs. The bids on actual RFQs are sent to the customers, and the first day of the production schedule is sent to the factory specifying the instructions for the next day.

Finally, the Demand Manager projects component use for the period between 11 and 40 days in the future by predicting customer demand with the Demand Model and assuming that production of some fraction of this demand will be required.

Supply Manager

The Supply Manager's goal is to obtain, at the lowest possible cost, the components that the Demand Manager projects it will use. This process is divided into two steps: first the Supply Manager decides *what* components will need to be delivered, and then it decides *how* best to ensure the delivery of these components. These two steps are described below.

Deciding What to Order The Supply Manager seeks to keep the inventory of each component above a certain threshold. This threshold is 800, or 400 in the case of CPUs, and decreases linearly to zero between days 195 and 215. Each day, the Supply Manager determines the exact deliveries that would be needed to maintain the threshold on each day in the future given current inventory, expected deliveries, and projected component use. The result is a list of needed deliveries that we will call *intended deliveries*. When informing the Demand Manager of the expected future component deliveries, the Supply Manager will add these intended deliveries to the actual deliveries expected from previously placed component orders. The idea is that although the Supply Manager has not yet placed the orders guaranteeing these deliveries, it intends to, and is willing to make a commitment to the Demand Manager to have these components available.

Deciding How to Order Once the Supply Manager has determined the intended deliveries, it must decide how to ensure their delivery at the lowest possible cost. We simplify this task by requiring that for each component and day, that day's intended delivery will be supplied by a single order with that day as the due date. Thus, the only decisions left for the Supply Manager are when to send the RFQ and which supplier to send it to. For each individual intended delivery, the Supply Manager predicts whether sending the RFQ immediately will result in a lower offered price than waiting for some future day, and sends the RFQ if this is the case.

In order to make this prediction correctly, the Supply Manager would need to know the prices that would be offered by a supplier on any future day. Although this information is clearly not available, the Supplier Model does have the ability to predict the prices that would be offered by a supplier for any RFQ sent on the current day. To enable the Supply Manager to extend these predictions into the future, we make the simplifying assumption that the price pattern predicted on the current day will remain the same on all future days. This assumption is not entirely unrealistic due to the fact that agents tend to order components a certain number of days in advance, and this number generally changes slowly. Essentially, the Supply Manager follows a heuristic saying, "Given the current ordering pattern of other agents, prices are lowest when RFQs are sent x days in advance of the due date, so plan to send all RFQs x days in advance."

Adaptation over a Series of Games

The predictions made by the predictive modules as described above are based only on observations from the current game. Another source of information that could be useful in making predictions is the events of past games, made available in log files kept by the game server. During the final rounds of the TAC SCM competition, agents are divided into brackets of six and play a number of games (16 on the final day of competition) against the same set of opponents. When facing the same opponents repeatedly, it makes sense to consider adapting predictions in response to completed games. TacTex-05 makes use of information from these games in its decisions during two phases of the game: buying components at the beginning of the game, and selling computers at the end of the game. In both cases, only past games within a bracket are considered, and default strategies are used when no game logs are yet available.

Initial Component Orders

At the beginning of each game, many agents place relatively large component orders (when compared to the rest of the game) to ensure that they will be able to produce computers during the early part of the game. Prices for some components may also be lower on the first day than they will be afterwards, depending on the due date requested. Determining the optimal initial orders to place is difficult, because no information is made available on the first day of the game, and prices depend heavily on the orders of other agents.

TacTex-05 addresses this issue by analyzing component costs from past games and deciding what components need to be requested on the first two days in order to ensure a sufficient supply of components early in the game and to take advantage of low prices. The process is very similar to the normal one used by the Supply Manager, except that predictions of prices offered by suppliers are based on past games.

On the first day, the Supply Manager begins by deciding what components will be needed over the first 80 days. Because no demand information is available (customers begin sending RFQs on the second day), the Supply Manager simply projects an average level of customer demand and assumes it will win some fraction of the demand each day. This projected component use is converted into a list of intended deliveries as before.

Next, the Supply Manager must decide which intended deliveries will be cheapest if they are requested immediately and send the resulting RFQs. Because the Supplier Model will have no information from the current game to use in predicting prices, information from past games is used. By analyzing the log from a past game and modeling the state of each supplier, it is possible to determine the exact price that would have been offered in response to any possible RFQ. Predictions for the current game can thus be made by averaging the results from all past games.

Since information about customer demand is available on the second day of the game but not the first, it might be beneficial to wait until the second day to send RFQs if similar prices can be obtained. For this reason, the Supply Manager does not send a request on the first day for an intended delivery if the price expected on the second day is less than 3% higher, and the above process is always repeated on the second day.

Endgame Sales

Near the end of each game, some agents tend to run out of inventory and stop bidding on computers, while other agents tend to have surplus computers, possibly by design, that they attempt to sell up until the last possible day. As a result, computer prices on the last few days of the game are often either very high or very low. When end-game prices will be high, it can be beneficial to hold on to inventory so as to sell at a premium during the last days. When prices will be low, the agent should deplete its inventory earlier in the game. TacTex-05 adapts in response to the behavior of its competitors in past games by adjusting the predictions of the Offer Acceptance Predictor during the final days of each game.

TacTex-05's endgame strategy is essentially to reserve only as many computers for the final few days as it expects to be able to sell at high prices. In particular, from day 215 to 217, the Demand Manager will always respond to a customer RFQ (if it chooses to respond) by offering a price slightly below the customer's reserve price. For RFQs received on these days, the probability predicted by the Offer Acceptance Predictor is set to the fraction of computers that would have sold at the reserve price on that day in past games. When the Demand Manager plans for a period of production that includes one of these days, these acceptance probabilities should result in an appropriate number of computers being saved for these three days.

Competition Results

In this section we look at the results of the 2005 TAC SCM competition, which TacTex-05 won, paying special attention to the performance of TacTex-05 on the final day of the competition. Out of 32 teams that initially entered the competition, 24 advanced past a seeding round to participate in the finals, held over three days at IJCAI 2005. On each day of the finals, half of the teams were eliminated, until six remained for the final day. Game outcomes depended heavily on the six agents competing in each game, as illustrated by the progression of scores over the course of the competition, underscoring the potential value of adaptation. In the seeding round, TacTex-05 won with an average score of \$14.9 million, and several agents had scores above \$10 million. Making a profit was much more difficult on the final day of competition, however, and TacTex-05 won with an average score of only \$4.7 million, followed by SouthamptonSCM with \$1.6 million and Mertacor with \$.5 Million. The other three agents (each of which averaged at least \$6 million in the seeding round) lost money.[2]

[2] See `www.sics.se/tac/scmserver` for scores.

In order to visualize the game results from the final day of competition, we tracked four quantities over the course of each game, and plotted the average over all 16 games, shown in Figure 2. These quantities are component costs, time between component order and use, revenue, and profit. To determine daily component costs for each agent, a record of each component order was placed into a queue at the time the order was delivered. Whenever a computer order was delivered to a customer, components were removed from the queue, and the cost (including storage costs) recorded for that day (the day of delivery). Revenue from computer deliveries was similarly recorded on the day of the delivery. Penalties were not tracked. The precise meanings of the quantities graphed in Figure 2 are as follows. Cost represents the average cost, as a fraction of the base price, for each component used (delivered as part of a computer) on a given day. Order time represents the average length of time between the date a component is ordered and the date a component is used, regardless of when the component is delivered to the agent. Revenue represents the total sales prices for all orders delivered on a given day. Profit is equal to revenue minus all costs for the day. Thus both costs and order times are given as an *average* for all components *used* (not ordered) on a given day, while revenues and profits represent *totals* for the day. The data shown in Figure 2 has been smoothed to reduce day-to-day noise. For clarity, only the data for the top three agents is shown. Similar patterns can be observed in the data for the three remaining agents. Below, we discuss what can be learned from these graphs about TacTex-05's performance.

Computer Sales

From the revenue graph in Figure 2, we see that TacTex-05 had nearly the same daily revenue as SouthamptonSCM for most of the game, but significantly higher daily revenue than Mertacor. As there was little difference in the prices at which agents sold computers, differences in revenue were due primarily to differences in the volume of computers sold. In general, TacTex-05 was at or near the top in revenue in every game it played throughout the competition.

Endgame Computer Sales

The goal in adaptively predicting offer acceptance probabilities at the end of a game is to determine whether computers should be held back in hopes of high prices or sold early before prices drop. The first strategy turned out to be the correct one during the final day of the competition – on average, the fraction of customer RFQs receiving offers dropped from 57% on day 215 to 37% on day 217, and prices rose accordingly. From that standpoint, TacTex-05 behaved correctly, reducing computer sales near the end of the game and then increasing them during the last few days, as shown by Figure 2. However, TacTex-05 sold fewer computers overall during the last 20 days than it could have, and did not have particularly high profits in this period. The problem appears to be a lack of components caused by factors other than the adaptation, such as the reduction of the inventory threshold.

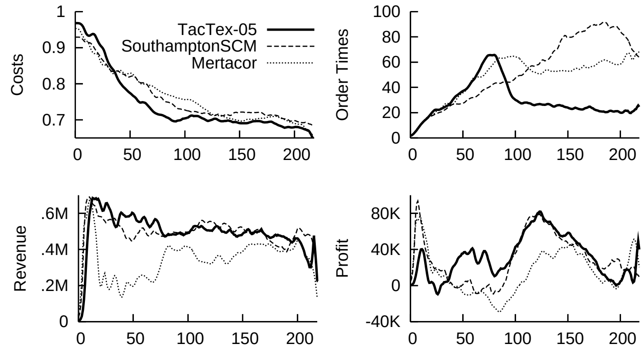

Figure 2: Competition results

Initial Component Orders

Recall that TacTex-05 will order components on the first or second day of a game if it predicts that prices will be lowest on these days, based on the results of past games. (For simplicity, we will say that components were ordered on a day when in fact the order was placed the following day in response to an offer.) First day orders jumped immediately after the first game (when fixed values were used because no past games were available), and continued to rise to about 95,000, while second day orders gradually dropped to nearly zero. By the later games, TacTex-05 was ordering nearly all of the components it expected to use over the first 80 game days on the very first day. For comparison, Southampton-SCM averaged 22,000 first day and 6,000 second day orders, while Mertacor averaged 18,000 first day and 10,000 second day orders.

The effects of these large first day orders on TacTex-05's performance can be seen in Figure 2. The most significant of these graphs is the daily profit. For roughly the first 40 days, TacTex-05's profit is below the profits of the other two agents, but between days 40 and 90, TacTex-05's profit is much higher. After day 90, TacTex-05 and Southampton-SCM have mostly similar profits.The differences in profits during these two periods can be explained as a result of the large first day orders.

During the period in which TacTex-05's profit is highest, it receives slightly more revenue than SouthamptonSCM, but not enough to explain the gap. The difference must therefore be in costs, and this is seen to be the case in the costs graph. This difference in costs comes as no surprise – TacTex-05 ordered most of its components on the first day precisely because it expected costs to increase on later days. What *is* a surprise is the fact that TacTex-05's costs are higher than those of other agents for the first 40 days. The explanation is related to the limited number of RFQs allowed per supplier per day – because TacTex-05 uses several of these RFQs for long term orders, it has less flexibility in generating short term RFQs. Essentially, higher short term costs are traded for lower long term costs.

Remaining Component Orders

After the components ordered at the beginning of the game have been used, the Supply Manager tends to favor relatively short term ordering. As shown in the order times graph of Figure 2, on average components are ordered about twenty days in advance of their use, much less than with the other agents. We can assume that the Supplier Model predicts lower prices from short term orders than long term ones, and in fact, the costs graph shows that TacTex-05 maintains slightly lower costs than the other agents.

Experimental Results

In addition to analyzing competition results, we have studied the performance of TacTex-05 through a number of controlled experiments against various sets of opponents taken from the TAC Agent Repository (`http://www.sics.se/tac/showagents.php`), a collection of agents submitted by TAC participants. Due to space limitations, we fully describe only one of these experiments, and summarize the results of others.

Experimenting with Initial Component Orders

From the analysis of the competition results, it appears that TacTex-05's early-game adaptation played a large role in its success. To see how this adaptation would perform under different circumstances, and to better measure the impact of this adaptation, we ran an experiment using two versions of the agent. A non-adaptive version used the price predictions resulting from the final day of the competition, leading to the large first day orders described above. An adaptive version began with these same predictions, but then adapted them as normal. 30 games were played between these agents and four agents that were not part of the final day of competition.

For this set of opponents, it turned out that prices were no longer consistently lowest on the first day. First day orders immediately dropped to about 20,000, while second day orders averaged about 22,000. The adaptive agent had lower component costs during the first part of the game, resulting in much higher daily profits. The average score of the adaptive agent was \$.81 million higher than the non-adaptive agent's score, and this difference is statistically significant with 95% confidence according to a paired t-test. The results of this experiment indicate that optimal first and second day orders may be very different for different sets of opponents, and that TacTex-05 is able to quickly adapt to take advantage of this fact.

Additional Results

Through similar experiments, we have obtained the following results:

- Of the three predictive models, it appears that performance is most dependent on the accuracy of the Supplier Model's predictions. Replacing the Offer Acceptance Predictor with a much simpler heuristic results in only a small decrease in profit, while doing so for the Demand Model results in very little change (Pardoe & Stone 2006).
- As with the early-game adaptation, the behavior resulting from the late-game adaptation depends heavily on the set of opponents faced. Overall, the use of the adaptive strategy tends to result in higher profits than the use of any simple fixed strategy (including the default behavior of the Offer Acceptance Predictor) (Pardoe, Stone, & VanMiddlesworth 2006).
- Changes to the Supply Manager designed to encourage short term orders, such as reducing the inventory threshold, may increase profit. It may be worth paying slightly more for components if by waiting to order, more informed purchases can be made (Pardoe & Stone 2006).

Related Work

A number of agent descriptions for TAC SCM have been published presenting various approaches to the tasks faced by an agent. (See `http://tac.eecs.umich.edu/researchreport.html` for a collection of papers.) Strategies used for bidding on customer RFQs range from game-theoretic analysis of the economy (Kiekintveld *et al.* 2004) to fuzzy reasoning (He *et al.* 2005). While attention has also been paid to the problem of component procurement, much of it has focused on an unintended feature of the game rules (eliminated in the 2005 competition) that caused component prices to *always* be cheapest at the very beginning of the game (Kiekintveld, Vorobeychik, & Wellman 2005). Methods of adaptation to a set of opponents over a series of games in TAC SCM have not been reported on to our knowledge. Such adaptation has been used in the TAC Travel competition, however (Stone *et al.* 2001).

Conclusion

In this paper we described TacTex-05, a fully implemented supply chain management agent consisting of predictive, optimizing, and adaptive components. In addition, we analyzed its championship performance during the 2005 TAC SCM competition and reported on the results of controlled experiments. Improving TacTex-05's predictive models through additional forms of adaptation remains an important area for future work.

References

Benisch, M.; Greenwald, A.; Grypari, I.; Lederman, R.; Naroditskiy, V.; and Tschantz, M. 2004. Botticelli: A supply chain management agent. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 3, 1174–1181.

He, M.; Rogers, A.; David, E.; and Jennings, N. R. 2005. Designing and evaluating an adaptive trading agent for supply chain management applications. In *IJCAI 2005 Workshop on Trading Agent Design and Analysis*.

Kiekintveld, C.; Wellman, M.; 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.; Vorobeychik, Y.; and Wellman, M. P. 2005. An analysis of the 2004 supply chain management trading agent competition. In *IJCAI 2005 Workshop on Trading Agent Design and Analysis*.

Pardoe, D., and Stone, P. 2004. Bidding for customer orders in TAC SCM. In *AAMAS 2004 Workshop on Agent Mediated Electronic Commerce VI: Theories for and Engineering of Distributed Mechanisms and Systems*.

Pardoe, D., and Stone, P. 2006. Predictive planning for supply chain management. In *Sixteenth International Conference on Automated Planning and Scheduling*.

Pardoe, D.; Stone, P.; and VanMiddlesworth, M. 2006. TacTex-05: An adaptive agent for TAC SCM. In *AAMAS 2006 Workshop on Trading Agent Design and Analysis / Agent Mediated Electronic Commerce*.

Stone, P.; Littman, M. L.; Singh, S.; and Kearns, M. 2001. ATTac-2000: An adaptive autonomous bidding agent. *Journal of Artificial Intelligence Research* 15:189–206.