

# TacTex-05: An Adaptive Agent for TAC SCM

David Pardoe, Peter Stone, and Mark VanMiddlesworth

Department of Computer Sciences  
The University of Texas at Austin, Austin TX 78712, USA  
{dpardoe,pstone,markvanm}@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, focusing on its ability to adapt to opponent behavior over a series of games. The impact of this adaptivity is examined through both analysis of competition results and controlled experiments.

## 1 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 intricate interactions of human representatives of the various companies involved. However, recent advances in autonomous agent technologies have sparked an interest, both in academia and in industry, in automating the process [1] [2] [3].

One barrier to supply chain management research is that it can be difficult to benchmark automated strategies in a live business environment, due to the proprietary nature of the systems and the high cost of errors. The Trading Agent Competition Supply Chain Management (TAC SCM) scenario provides a unique testbed for studying and prototyping supply chain management 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, and focus on its ability to adapt to opponent behavior over a series of games. In particular, we describe how decisions concerning purchases at the start of a game and sales at the end of a game are influenced by observations from past games. While the start- and end-game effects observed in a TAC SCM game may appear to be artifacts of the fixed game length, there are analogues in real-world supply chains: products are frequently introduced or phased out, and analyzing the behavior of competitors when such events have occurred in the past may provide clues for the future.

The remainder of this paper is organized as follows. We first summarize the TAC SCM scenario, and then describe the design of TacTex-05. Next, we describe the methods used by TacTex-05 to adapt to past opponent behavior. Finally, we examine the impact of this adaptivity, through both analysis of competition results and controlled experiments.

## 2 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.<sup>1</sup>

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 expanded on in the remainder of this section.

### 2.1 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, the *date* on which they should be delivered, and a *reserve price* stating the maximum amount the agent is willing to pay. Agents may send at most 5 RFQs per component per supplier each day. Suppliers respond to RFQs the next day 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.

### 2.2 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 for each day the delivery is late. Agents respond to the RFQs by bidding in a first-price auction: the agent offering 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.

---

<sup>1</sup> [http://www.sics.se/tac/tac05scmspec\\_v157.pdf](http://www.sics.se/tac/tac05scmspec_v157.pdf)

### 2.3 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. Agents are required to pay a small daily storage fee for all components in inventory at the factory.

## 3 Overview of TacTex-05

In this section we present a high-level overview of TacTex-05. Details on specific agent components are contained in the sections that follow.<sup>2</sup>

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

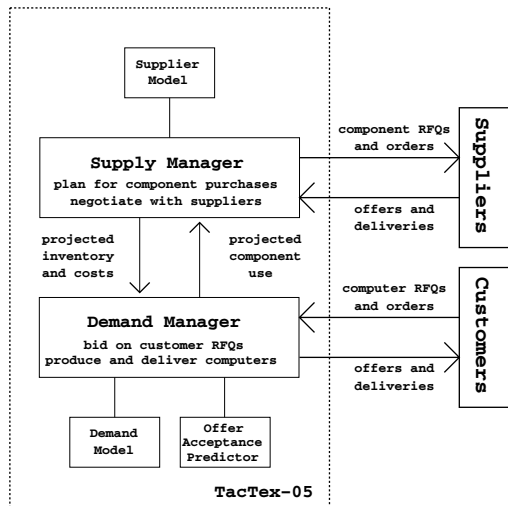


Fig. 1. Agent overview

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 component 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

<sup>2</sup> The information in sections 3-5 is condensed from [4] and is included to provide a fully self-contained agent description. Section 6 presents new information about the adaptive aspects of TacTex-05, the focus of this paper.

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 predictive *Supplier Model*, a predictive *Demand Model*, and an *Offer Acceptance Predictor*.

The Supplier Model keeps track of all information available about each supplier, such as TacTex-05's outstanding orders and the prices that have been offered in response to RFQs. Using this information, the Supplier Model can assist the Supply Manager by making predictions concerning future component prices.

The Demand Model tracks customer demand and tries to estimate the underlying demand parameters. With these estimates, it is possible to predict the number of RFQs that will be received on any future day. The Demand Manager can then use these predictions to plan for future production.

When deciding what bids to make in response to customer RFQs, the Demand Manager needs to be able to estimate the probability of a particular bid being accepted (which depends on the bidding behavior of the other agents). This prediction is handled by the Offer Acceptance Predictor. Based on past bidding results, the Offer Acceptance Predictor produces a function for each RFQ that maps bid prices to the predicted probability of winning the order.

## 4 The Demand Manager

The Demand Manager handles all computation related to computer sales and production. This section describes the Demand Manager, along with the Demand Model and the Offer Acceptance Predictor upon which it relies.

### 4.1 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 [5]). 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.

### 4.2 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 [6] 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 [7]. 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.

### 4.3 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. 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. The latest possible due date for an RFQ received on the current day is 12 days in the future, meaning the production schedule for the needed computers must be sent within the next 10 days. The Demand Manager thus 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 uses the future level of customer demand predicted by the Demand Model to generate a predicted set of all RFQs that will be received for computers due during the period. 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 [6].

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 from 11 to 40 days in the future by using the Demand Model to predict customer demand and assuming that some fraction of this demand will be produced.

## 5 The Supply Manager

The Supply Manager is responsible for purchasing components from suppliers based on the projection of future component use provided by the Demand Manager, and for informing the Demand Manager of expected component deliveries and replacement costs. In order to be effective, the Supply Manager must be able to predict future component availability and prices. The Supplier Model assists in these predictions.

### 5.1 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.

### 5.2 Supply Manager

The Supply Manager's goal is to obtain the components that the Demand Manager projects it will use at the lowest possible cost. 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 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."

The final step is to predict the replacement cost of each component. The Supply Manager assumes that any need for additional components that results from the decisions of the Demand Manager will be felt on the first day on which components are currently needed, i.e., the day with the first intended delivery. Therefore, for each component's replacement cost, the Supply Manager uses the predicted price of the first intended delivery of that component, even if no RFQ was sent.

## 6 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 (impacting mainly the behavior described in

Section 5.2), and selling computers at the end of the game (impacting the behavior in Section 4.2). We chose to focus on these areas for two reasons. Behavior during these two phases varies significantly from one agent to another, possibly due to the fact that these phases are difficult to reason about in general and may thus be handled using special-case heuristic strategies by many agents. At the same time, each agent's behavior remains somewhat consistent from game to game (e.g. many agents order the same components at the beginning of each game). This fact is critical to the success of an adaptive strategy – the limited number of games played means that it must be possible to learn an effective response from only a few past games.

## 6.1 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 one described in Section 5.2, except that predictions of prices offered by suppliers are based on past games. First, the components needed are identified, then the decision of which components should be requested is made, and finally the RFQs are generated.

The Supply Manager begins by deciding what components will be needed. On the first day, when no demand information is available (customers begin sending RFQs on the second day), the Supply Manager assumes that it will be producing an equal number of each type of computer, and projects the components needed to sustain full factory utilization for 80 days. On the second day, the Supply Manager projects future customer demand as before and assumes it will receive orders for some fraction of RFQs over each of the next 80 days. The projected component use is converted into a list of intended deliveries as before.

Next, the Supply Manager must decide which components should be requested on the current day (the first or second day of the game). As in Section 5.2, the Supply Manager must determine which intended deliveries will be cheapest if they are requested immediately. At the beginning of the game, the Supplier Model will have no information to use in predicting prices, and so 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 be made by averaging the results from all past games. When modeling the states of suppliers, RFQs and orders from TacTex-05 are omitted to prevent



the agent from trying to adapt to its own behavior. If the initial component purchasing strategies of opponents remain the same from game to game, these average values provide a reasonable means of estimating prices.

At the beginning of the game, the Supply Manager reads in a table from a file that gives the average price for each component for each pair of request date and due date. Using this table, the Supply Manager can determine which intended deliveries will cost less if requested on the current day than on any later day. Intended deliveries due within the first 20 days are always requested on the first day, however, to avoid the possibility that they will be unavailable later. If opponents request many components on the first day of the game but few on the second, the prices offered in response to RFQs sent on the second day will be about the same as if the RFQs had been sent on the first day. 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. For this reason, the Supply Manager will not send a request for an intended delivery if the price expected on the second day is less than 3% more than the price expected on the first.

Once the Supply Manager has decided which intended deliveries to request, it must decide how to combine these requests into the available number of RFQs (five, or ten if there are two suppliers). In Section 5.2, this problem did not arise, because there were typically few requests per day. On the first two days, it is possible for the number of intended deliveries requested to be much larger than the number of RFQs available. Intended deliveries will therefore need to be combined into groups, with delivery on the earliest group member's delivery date. The choice of grouping can have a large impact on the prices offered. When there is only one supplier, the Supply Manager begins by dividing the 80 day period into five intervals, defined by six interval endpoints, with a roughly equal number of intended deliveries in each interval. Each interval represents a group of intended deliveries that will have delivery requested on the first day of the interval. One at a time, each endpoint is adjusted to minimize the sum of expected prices plus storage costs for those components delivered early. When no more adjustments will reduce the cost, the Supply Manager sends the resulting RFQs. When there are two suppliers, ten intervals are used, and intervals alternate between suppliers.

## 6.2 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 it 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 (Section 4.2) during the last few 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 reserve. 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 will hopefully result in an appropriate number of computers being saved for these three days.

## 7 Competition Results and Additional Experiments

In this section, we look at the results of the final day of the 2005 TAC SCM competition to determine how TacTex-05's adaptivity contributed to its performance. We also present the results of controlled experiments designed to test this adaptivity under differing conditions.

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.<sup>3</sup>

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 4. 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 4, from top to bottom, 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

<sup>3</sup> Competition scores are available at <http://www.sics.se/tac/scmserver>

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 4 has been smoothed with a Gaussian filter with a standard deviation of 5 days 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 the performance of TacTex-05's adaptive methods.

### 7.1 Initial Component Orders During the Competition

We begin by examining the component orders placed by TacTex-05 at the beginning of games. 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.) Figure 2 shows the number of components ordered on the first and second day of each game. As two games were played at a time, there are eight points plotted per day, each representing the average of two simultaneous games. For the first pair of games, when no previous games against the same opponents were available to use in predicting prices, data from the previous day of the competition was used. After the first two games, price predictions were based only on games from the final day of competition. First day orders jumped immediately after the first pair of games, 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, the second largest orders came from the agent Deep Maize, which averaged 40,000 first day and 7,000 second day orders. SouthamptonSCM averaged 22,000 first day and 6,000 second day orders, Mertacor averaged 18,000 first day and 10,000 second day orders, and the other two agents had somewhat smaller orders. While the order sizes of the agents other than TacTex-05 did vary from game to game, they did so by relatively small amounts, and it is not clear if such changes were related to intentional adaptation. The fact that TacTex-05's first day orders continued to increase suggests the possibility of a self-reinforcing process: when more components are ordered on the first day, prices on later days may be driven up, making first day orders even more appealing.

The effects of these large first day orders on TacTex-05's performance can be seen in Figure 4. 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 SouthamptonSCM 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. These higher costs (along with what the revenue graph suggests is a slight lag in initial computer deliveries) explain the lower profit during this period, but are unexpected given TacTex-05’s attempts to place orders in a way that minimizes cost. The explanation is related to the limited number of RFQs allowed per supplier per day. After deciding which components need to be ordered on the first day, TacTex-05 must decide how to group these components into the allowed number of RFQs, as described previously. The cost per component for one group can only be reduced by increasing the costs of another group, and so the differences between the two periods can be seen as a tradeoff: lower prices during the second period at the expense of higher prices in the first. Because the other agents order fewer components on the first day, they are not as affected by this tradeoff, and can obtain lower prices during the early part of the game.

## 7.2 Experimenting with Initial Component Orders

To see how TacTex-05’s early-game adaptation would perform under different circumstances, and to better measure the impact of this adaptation, we ran an experiment using two versions of TacTex-05. A non-adaptive version used the price predictions resulting from the final day of the competition, leading to initial orders similar to those in the later games of Figure 2. 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: PhantAgent, Botticelli, RationalSCM, and CrocodileAgent.<sup>4</sup>

Figure 3 shows the changes in first and second day orders over the 30 games. For this set of opponents, it appears that prices are no longer consistently lowest on the first day. First day orders immediately drop to the lowest level allowed (since some components will always be ordered on the first day), and more components are usually ordered on the second day than the first. The variation in second day orders is mainly due to the fact that customer demand can be taken into account on the second day. Figure 5 shows that the adaptive agent had lower component costs during the first part of the game, and these lower costs resulted in much higher profits during the first 50 days. The reason for slightly higher profits and revenue for the non-adaptive agent shortly after day 50 is not immediately clear. 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.

---

<sup>4</sup> Taken from the TAC Agent Repository, <http://www.sics.se/tac/showagents.php>

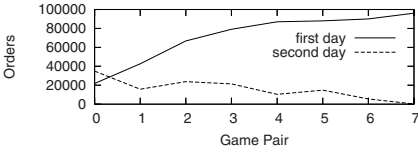


Fig. 2. Initial orders – competition

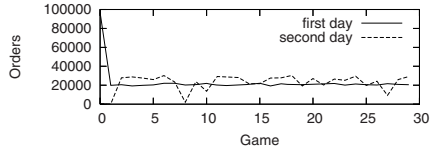


Fig. 3. Initial orders – experiment

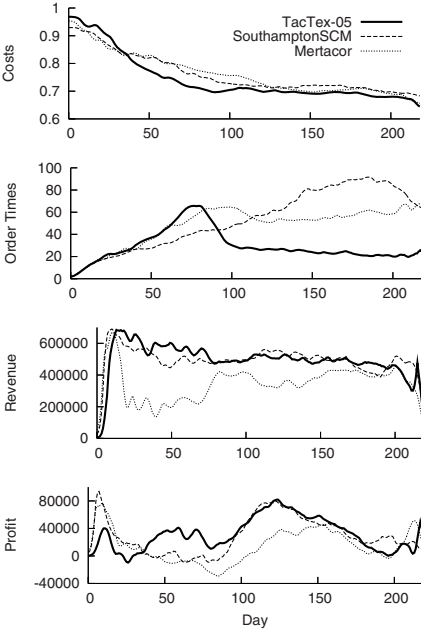


Fig. 4. Results – final day of competition

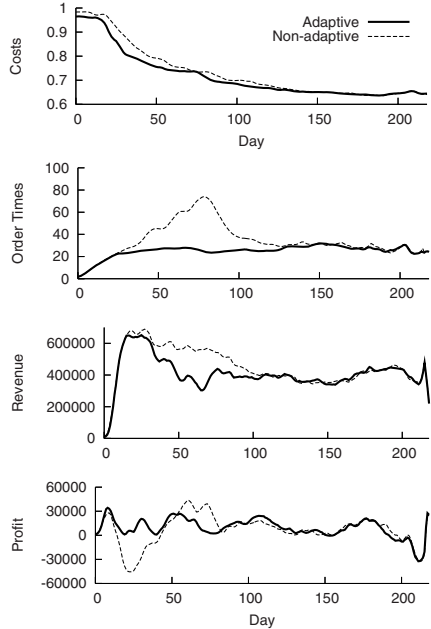


Fig. 5. Results – experiment

### 7.3 Endgame Sales During the Competition

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 4. However, TacTex-05 sold fewer computers overall during the last 20 days than it could have, and did not have particularly high profits during 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.

## 7.4 Experimenting with Endgame Sales

Since it is difficult to evaluate the success of the endgame adaptation that took place during the competition, we instead rely on experimental evaluation. We performed an experiment using four different versions of TacTex-05. One used endgame adaptation as normal, one assumed 100% offer acceptance during the final three days, one assumed 0% acceptance, and one (which we'll call "standard") used the same method of predicting offer acceptance as during the rest of the game. The other two agents were the "dummy" agents provided by the game server. The results are presented in the first column of Table 1. The adaptive agent outperformed the 0% agent and the standard agent (statistically significant with 95% confidence according to paired t-tests), but was slightly worse than the 100% agent (not significant). The 0% agent and the standard agent both had fairly large penalties during the final few days that accounted for much of the difference, caused by the fact that they underestimated offer acceptance probabilities.

Although the 100% agent worked well, under different circumstances this strategy might not work. To demonstrate, we ran another experiment with one adaptive agent and three 100% agents. The results are presented in the second column of Table 1. This time, the adaptive agent outscored the 100% agents (statistically significant). The 100% agents tended to have more unsold inventory than the adaptive agent, indicating that they overestimated their chances of selling computers during the final days. These two experiments demonstrate the need to handle sales differently during the end of a game than during the rest of the game, and the drawbacks of using a fixed strategy.

**Table 1.** Endgame adaptation experiments

<i>Agent</i>	<i>Exp. 1 Score</i>	<i>Exp. 2 Score</i>
adaptive	\$7.14M	\$6.85M
100%	\$7.20M	\$6.41M
0%	\$6.46M	–
standard	\$6.74M	–

## 8 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.) Although several agents make efforts to adapt to changing conditions *during* a single game (e.g. [8], [9]), 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, both during a round of competition [10], and in response to hundreds of previous games [11].) While attention has been paid to the problem of early-game component procurement, much of it has focused on an unintended feature of the game rules (eliminated in the 2005 competition) that caused prices to *always* be lowest on the first day of the game [12].

## 9 Conclusion

In this paper we described the TacTex-05 agent, focusing on its ability to adapt over a series of games. The adaptation of component purchases during the beginning of a game was shown to have a positive impact on performance during the 2005 TAC SCM competition, and both start-game and end-game adaptation were experimentally shown to allow the agent to respond to various types of opponent behavior. Improving TacTex-05's predictive modules through additional forms of adaptation remains an important area for future work.

## Acknowledgments

We would like to thank Jan Ulrich for contributing to the development of TacTex-05, the SICS team for developing the game server, and all teams that have contributed agents to the agent repository. This research was supported in part by NSF CAREER award IIS-0237699.

## References

1. Fox, M.S., Chionglo, J.F., Barbuceanu, M.: The integrated supply chain management system. Internal Report, Dept. Industrial Engineering, University of Toronto, available from <http://www.eil.utoronto.ca/public/iscm-intro.ps> (1993)
2. Sadeh, N., Hildum, D., Kjenstad, D., Tseng, A.: Mascot: an agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling. In Workshop on Agent-Based Decision Support in Managing the Internet-Enabled Supply-Chain, at Agents '99 (1999)
3. Chen, Y., Peng, Y., Finin, T., Labrou, Y., Cost, S.: A negotiation-based multi-agent system for supply chain management. In Workshop on Agent-Based Decision Support in Managing the Internet-Enabled Supply-Chain, at Agents '99 (1999)
4. Pardoe, D., Stone, P.: Predictive planning for supply chain management. In: Sixteenth International Conference on Automated Planning and Scheduling. (2006)
5. Kiekintveld, C., Wellman, M., Singh, S., Estelle, J., Vorobeychik, Y., Soni, V., Rudary, M.: Distributed feedback control for decision making on supply chains. In: Fourteenth International Conference on Automated Planning and Scheduling. (2004)
6. Pardoe, D., Stone, P.: 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. (2004)
7. Benisch, M., Greenwald, A., Grypari, I., Lederman, R., Naroditskiy, V., Tschantz, M.: Botticelli: A supply chain management agent. In: Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS). Volume 3. (2004) 1174–1181
8. Ketter, W., Collins, J., Gini, M., Gupta, A., Schrater, P.: Identifying and forecasting economic regimes in tac scm. In: IJCAI 2005 Workshop on Trading Agent Design and Analysis. (2005) 53–60

9. He, M., Rogers, A., David, E., Jennings, N.R.: Designing and evaluating an adaptive trading agent for supply chain management applications. In: IJCAI 2005 Workshop on Trading Agent Design and Analysis. (2005)
10. Stone, P., Littman, M.L., Singh, S., Kearns, M.: ATTac-2000: An adaptive autonomous bidding agent. *Journal of Artificial Intelligence Research* **15** (2001) 189–206
11. Stone, P., Schapire, R.E., Littman, M.L., Csirik, J.A., McAllester, D.: Decision-theoretic bidding based on learned density models in simultaneous, interacting auctions. *Journal of Artificial Intelligence Research* **19** (2003) 209–242
12. Kiekintveld, C., Vorobeychik, Y., Wellman, M.P.: An analysis of the 2004 supply chain management trading agent competition. In: IJCAI 2005 Workshop on Trading Agent Design and Analysis. (2005)