TacTex-03: A Supply Chain Management Agent

David Pardoe and Peter Stone
Department of Computer Sciences, The University of Texas at Austin
1 University Station C0500, Austin, Texas 78712-1188
{dpardoe,pstone}@cs.utexas.edu
http://www.cs.utexas.edu/˜{dpardoe,pstone}

Technical Report UT-AI-TR-04-308

***Abstract*— This paper introduces TacTex-03, an agent designed to participate in the Trading Agent Competition Supply Chain Management Scenario (TAC SCM). As specified by this scenario, TacTex-03 acts as a simulated computer manufacturer in charge of buying components such as chips and motherboards, manufacturing different types of computers, and selling them to customers. TacTex-03 was the top scorer in two of the preliminary rounds of the 2003 TAC SCM competition, and finished in 3rd place in the finals.**

I. Introduction

Supply chains are a current, challenging problem for e-commerce. In this paper, we consider the case of an individual agent managing just a single link in the chain. From this agent's perspective, it needs to do three major things: acquire supplies, sell products, and manage its local manufacturing process.

The Trading Agent Competition Supply Chain Management Scenario (TAC SCM) [1], allows us to study exactly this problem. It was introduced with the purpose of creating a standard testbed in which to compare concrete supply chain trading agent approaches. Roughly speaking, each TAC agent is a computer manufacturer in charge of buying components such as chips and motherboards, manufacturing different types of computers, and selling them to customers.

In this paper, we introduce TacTex-03, an agent in this TAC SCM scenario. At the high level, TacTex-03 attempts to acquire as many supplies as it can at cheap prices via a heuristic analysis of the scenario, and uses a greedy approach to the manufacturing process. The main innovation of TacTex-03 is an iterative search process to select a set of offers to make to customers.

TacTex-03 was the top scorer in two of the preliminary rounds of the 2003 TAC SCM competition, and finished in 3rd place in the finals. Following a brief summary of the TAC SCM scenario in Section II, the remainder of this paper details the TacTex-03 strategy (Sections III–V) and summarizes its performance in the competition (Sections VI). Section VII concludes.

II. TAC SCM

In this section, we provide a brief summary of the TAC SCM game and emphasize the parts that are most relevant to the strategy used by TacTex-03. Full details are available in the official game specification [2].

In a TAC SCM game, six agents act as computer manufacturers in a simulated economy that is managed by a game server. The length of a game is 220 simulated days, with each day lasting 15 seconds of real time. At the beginning of each day, agents receive messages from the game server with information concerning the state of the game, such as the customer requests for quotes (RFQs) for that day, and agents have until the end of the day to send messages to the server indicating their actions for that day, such as making offers to customers. The game can be divided into three parts: production and delivery, component supply, and computer demand.

A. Production and Delivery

Each day an agent must send a production schedule and a delivery schedule to the server indicating its actions for the next day. An agent has 2000 production cycles per day available for production, with a single computer requiring between 5 and 7 cycles to produce, depending on the components included. Delivery schedules indicate which customer orders will be filled from the available completed computers.

B. Component Supply

The computers sold are made from four components: CPUs, motherboards, memory, and hard drives. These components must be purchased from suppliers by the agents. There are two brands of CPU available, each requiring a specific type of motherboard. CPUs come in two speeds, and both hard drives and memories come in two sizes. Each brand of CPU is provided by a single supplier, while each of the other components is provided by two suppliers. There are a total of 10 different components, from which 16 different computer configurations can be made.

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. Suppliers respond with offers indicating the price at which they are willing to sell the components. Agents then accept or reject the offers. If a supplier does not expect that it will be able to provide the components by the date requested, it responds to an RFQ with two alternatives: an offer for as many components as it can provide on the requested due date, and an offer for the requested quantity on a later due date. An agent may only select one of these alternatives.

An agent may send 10 RFQs per supplier per day. These RFQs will be considered in the order they are sent. If a supplier receives RFQs from multiple agents, it repeatedly chooses an agent at random and processes the next RFQ in the list of RFQs sent by that agent, continuing until all RFQs have been considered.

Suppliers are able to produce only a limited quantity of components. At the beginning of the game, the number of each component that a supplier can produce is 500 per day, and this number varies according to a random walk. Suppliers base their offers to customers on their expected free capacity in the future. They assume a production capacity of 500 components per day and reserve the capacity needed to produce all current orders as late as possible. The prices suppliers offer are determined by the available capacity before the due date. The price offered on day d for an offer due on day $d+i$ is

$$Price(d+i) = P_{base}(1 - .5 * \frac{C_{available}(d+i)}{500 * i}) \qquad (1)$$

where P_{base} represents the fixed base price for the component requested and $C_{available}(d+i)$ represents the supplier's estimate of its total available capacity through day $d+i$.

C. Computer Demand

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, and a penalty that must be paid for each day the delivery is late. Orders are canceled on the fifth late day. Agents respond to the RFQs by making an offer to sell at a certain price, with the agent offering the lowest price on each RFQ winning the order. Agents are unable to see the prices offered by other agents or even the winning prices, but they do receive a report each day indicating the highest and lowest price at which each type of computer sold on the previous day.

The number of RFQs depends on the level of customer demand, represented by a parameter D. The actual number of RFQs each day is drawn from a Poisson distribution with D as its mean. Fluctuation in demand is modeled by multiplying D by an amount representing the current trend each day. This trend follows a random walk, and D is bounded between 80 and 320, with its initial value chosen randomly from this range.

III. Production and Delivery

Having summarized the game, we now turn to descriptions of TacTex-03's strategies for addressing the three main subtasks: production and delivery; supply acquisition; and sales.

Deciding what types of computers to produce and which orders to fill from the resulting computers is the most straightforward and self-contained problem faced by TacTex-03. The factors affecting these decisions are all known quantities: the components and computers in inventory, the available factory capacity, and the orders from customers. Our goal is clearly to deliver all customer orders on time if possible, and otherwise to find the feasible set of deliveries that maximizes our profit. We make the assumption that the strategy for obtaining components will take responsibility for obtaining all needed components on time and that the strategy for bidding on customer RFQs will take current customer orders into account, meaning that TacTex-03 can make decisions involving production and delivery without regard to the impact on these other modules. Rather than treating production and delivery as separate problems, TacTex-03 takes the approach of trying to decide which available resources should be allocated towards which customer orders, earmarking the resulting computers for delivery.

- Divide the current customer orders into two lists:
  - orders that are late or will be late if not produced immediately
  - all other orders
- Sort each list in order of decreasing value
- Append the second list to the first
- Go through the list attempting to fill each order:
  - Use any computers in inventory that are available
  - See if the remaining amount needed can be produced
  - If the order can be filled, earmark the computers for delivery

TABLE I

The greedy production scheduler used by TacTex-03.

Our initial approach to solving this problem was to formulate an optimal solution in the form of a linear program that plans for the next several days of production. The resulting linear program was similar to the one designed for a slightly simplified scenario by [3]. Unfortunately, with the computational resources available to us, the linear program failed to produce a result within the 15 seconds available per game day.

As a result, we devised an alternative heuristic solution that TacTex-03 uses in place of the linear program. This solution is a greedy approach and is detailed in Table I. Essentially, TacTex-03 attempts to fill orders in order of decreasing value, where the value of an order is taken to be its price plus the remaining penalty that can be avoided by filling the order minus the cost of the components used. After considering the customer orders, if there are any production cycles remaining, TacTex-03 attempts use them to build up an equal inventory of all computer types to be used to satisfy future orders.

The costs used in determining the value of an order deserve further explanation. Because the prices paid for components in inventory are sunk costs, TacTex-03 does not consider these prices in determining the cost of producing an order. Instead, it estimates the replacement cost of each component by predicting future component prices. The replacement cost of a component also depends on the current inventory and predicted future use – if the current inventory is sufficient to last through the game, there will be no need to replace components used, resulting in an effective replacement cost of zero.

IV. Supply Acquisition

On the supply side, TacTex-03 sends several large RFQs for each component on the first day. It then accepts as many offers as it expects to be able to use, based on the combinations of offers received (indicating the day on which production will be able to start) and the customer demand on the second day (which provides an estimate of the demand level thereafter). After that, TacTex-03's strategy is to prevent its inventory from dropping below a certain threshold by predicting future inventory based on current usage. It continually probes suppliers with small RFQs to estimate their capacity, and when components are needed, it chooses the supplier and due dates for the RFQ that it expects will result in the lowest cost. To decide if it should accept an offer, TacTex-03 projects its future production and sales both with and without the added components, and places an order only if it increases predicted profit. The first-day and post-first-day components of the strategy are described in full detail below.

A. First Day

The price a supplier offers in response to an RFQ for components is based entirely on the supplier's available capacity for producing that component and is given by equation 1. Because all suppliers have their full capacity available at the beginning of the game, the prices they offer in response to RFQs sent on the first day will all be at half of the base price, the lowest price possible. As a result, one obvious strategy is for an agent to attempt to buy the bulk of the components it plans to use on the first day. Doing this ties up supplier capacity early in the game and significantly drives up component prices after the first day, forcing other agents to consider using the same strategy.[1] The end result is that many agents end up sending large, competing RFQs on the first day, and the random order in which suppliers consider these RFQs has a significant impact on how each game unfolds. Also, agents using this strategy must commit to buying large quantities of components without knowing how many computers they will be able to sell. In games in which the customer demand is very low, this can result in bidding wars in which computer prices drop to extremely low levels as agents try to make use of their excess components, and negative scores are common. Changes to the supplier pricing model have been proposed to resolve these issues in the future, but the model was left in place for the first competition.

Thus, TacTex-03's first day component approach follows the general strategy of sending large RFQs to suppliers on the first day while trying to mitigate the potential problems mentioned above.

The most important decision involved in such a strategy is to determine the number of each component that should be ordered. The number should be at least the minimum number we expect to be able to profitably use during the game, and should be at most the maximum number that could possibly be used during the game. If we knew in advance how many computers TacTex-03 would sell, we could simply instruct it to try to order the exact components needed. This amount is obviously unknown, but it can be predicted. The main factor influencing the number of computers that will be sold is the level of customer demand. This level varies throughout the game, but a prediction of the total customer demand in the game (measured in terms of the total number of customer RFQs) can be found by computing the expected total given the demand observed at the beginning of the game, i.e., the size of the initial set of customer RFQs. We formed this prediction by running many simulated games and finding the average total RFQs for each initial observed number of RFQs. Despite the fact that the demand can fluctuate, we found that games with a low initial demand have a significantly lower number of average total RFQs than games with a high initial demand.

TacTex-03 determines the number of each component it wishes to order from each supplier at the beginning of the game as a function of this predicted total demand. We considered having TacTex-03 learn this function from past game data by looking at the number of computers sold for a profit and comparing this to the actual total demand, but we felt that values learned from data from the seeding rounds might not be applicable to the more aggressive set of agents we expected to face in the finals, and we decided that attempting to learn during the competition was too risky due to the small number of games played. Instead, TacTex-03 uses hand-tuned values that we feel are reasonable.

Unfortunately, the first set of RFQs are received from customers on the second day, not on the first day when the RFQs must be sent to suppliers. Therefore, the only way TacTex-03 can use the predicted total demand to control the number of components it buys at the beginning of the game is to reject some offers received from suppliers on the second day, after the customer RFQs have been received. This means that TacTex-03 should send RFQs for the largest number of components it could possibly need, and then accept a set of offers that provides it with approximately the number it determines it wants.

That leaves us with the question of how to divide that number up between the ten RFQs that can be sent to each supplier on the first day. Each supplier produces two components, so we allow five RFQs per supplier for each component. TacTex-03 sets the due date of all RFQs to day 1, meaning that it will receive only offers for the full amount requested, due as soon as that amount can be produced. The most important RFQ for each component is the first one, since it will be the first to be considered by the supplier and to be delivered. The first RFQ should be for an amount large enough to last until following deliveries arrive, but it should be no bigger than the minimum amount TacTex-03 might want to order, because we always want to accept the first offer so that TacTex-03 can begin producing as soon as possible. We ultimately decided on RFQ sizes of 8800, 4400, 2200, 1100, and 550, in that order. The smallest number of components we would want TacTex-03 to order is about 8800 per supplier, and the largest is about 17050 per supplier, the sum. By sending RFQs that are multiples of powers of 2, we make it easy for TacTex-03 to accept a subset of offers that approximates the actual amount desired.

On the second day of the game, after receiving offers from

[1] As noticed by [4], more and more agents did indeed use this strategy as the competition proceeded.

- **On the first day:**
  - send RFQs: 8800, 4400, 2200, 1100, and 550 of each component
- **On the second day:**
  - predict number of components needed based on customer RFQs
  - plan production using offered components to find useable amount
  - accept a subset of the offers providing the desired amount

TABLE II

THE FIRST-DAY ORDERING STRATEGY USED BY TACTEX-03.

suppliers and RFQs from customers, TacTex-03 computes the amount it wants to order based on the predicted total demand. There is one other factor that needs to be considered, however, before orders are placed. We need for TacTex-03 to determine whether it will actually be able to use the combination of components it has been offered. It may be the case that a shortage of one component prevents the use of another component. For example, if TacTex-03 receives no motherboards of one type until late in the game, then that limits the number CPUs of the corresponding type that can be used. As another example, if TacTex-03 receives an offer for a delivery of 8800 of a component just a few days before the end of the game (something that we have actually observed), then obviously it will be unable to use much of that delivery before the game ends.

TacTex-03 handles this problem by projecting its production of computers over the whole game, assuming that all offers are accepted and that each day's production is divided equally among the types of computers that can be built with the components remaining on that day. If TacTex-03 is able to use fewer of any component than the amount it was planning to order, it reduces the planned orders accordingly. Once TacTex-03 has determined the number of each component it wishes to order, it goes through the offers for each component in order of due date and accepts any offer that will not put the total amount ordered over the desired amount. The process is summarized in table II. [2]

B. After the First Day

After the first day, we need a different strategy for obtaining components if it becomes necessary to do so. Our goal is for TacTex-03 to maintain a certain minimum inventory while obtaining the necessary components as cheaply as possible. The threshold inventory is set at 750 for CPUs and 1500 for other components for most of the game, and gradually decreases to 0 towards the end of the game. Our strategy can be broken into three parts: deciding what components should be ordered and when they will be needed; determining what RFQs should be sent; and accepting or rejecting offers from suppliers.

TacTex-03 determines what components will be needed by projecting future inventory levels over the next fifty days. We set a limit of fifty days to prevent TacTex-03 from buying components so far in advance that its predictions may be unreasonable. The number of each component in inventory on a given day in the future can be computed as the current inventory plus expected deliveries minus estimated use. To find an estimate of the number of components it will use, TacTex-03 assumes that each component will be used at a constant rate in the future. The rate predicted for each component is the average number used per day over the past ten days due to deliveries of computers to customers. A cap is placed on the rate to prevent a short period of heavy use from causing an overestimate of future need. We base the rate on components in delivered computers rather than components used in production so that any production that simply builds up an inventory of completed computers does not create a false sense of demand for components.

Once TacTex-03 has predicted its inventory over the next fifty days, it steps through each day and determines whether there is a shortage of any component. If so, it makes a note of the amount needed and adds this amount to the inventory of each following day to simulate that the amount was delivered. The result is a list of the smallest and latest possible deliveries that are needed to prevent any shortages. To allow these deliveries to fit into the available RFQs, they are combined to form a set of at most five desired purchases per component, each needing to be delivered by a certain date.

For each desired purchase, we want TacTex-03 to create an RFQ by choosing a supplier and due date that will allow it to obtain the components on time at the lowest possible price. The price a supplier quotes for a fixed size request is a function of the supplier's available capacity as shown in equation 1, and is therefore a function of the request's due date. A supplier calculates its free capacity by planning to produce all existing orders as late as possible, so it may be possible to save money by requesting a delivery before it is needed in order to take advantage of a higher average available capacity per day.

By looking at an offer at price p from a supplier due on day $d+i$, where d is the current date, we can compute the supplier's available capacity between now and day $d+i$ as

$$C_{available}(d+i) = (1 - \frac{p}{P_{base}(component)}) * 1000$$

If we have two offers with different due dates, we can find the supplier's available capacity between the two due dates by subtracting the capacity remaining before the first date from the capacity remaining before the second. With enough offers, we can build a fairly accurate model of the supplier's remaining capacity before any date. TacTex-03 takes advantage of this fact by using all of its ten available RFQs per supplier per day to probe suppliers and estimate their available capacity. Any RFQs that are left after requesting needed components are used to send RFQs for a single component, spaced regularly over the next fifty days. All offers received are then used to determine the available capacity of a supplier over each

[2] The decision not to order components that cannot be used based on the set of offers received means that we are assuming that TacTex-03 will be unable to obtain the needed complementary components. This leaves it vulnerable to a preemptive strategy in which one agent consumes all of the supply RFQs on day 0 as described by [4]. A possible response is to repeat the first day strategy on subsequent days, but in the tournament, we simply fell back on the post-first-day acquisition strategy described below.

Sending RFQs:

- Predict component inventory over the next 50 days
- Determine the minimum deliveries needed to stay above the threshold
- Create RFQs to satisfy this need
- Choose suppliers and due dates for RFQs based on expected prices
- Use free RFQs to probe suppliers to determine their capacity

Handling offers:

- Accept offers if the marginal value of the components exceeds the cost
- Update estimates of supplier capacity by analyzing prices of offers

TABLE III

THE STRATEGY USED TO ORDER COMPONENTS AFTER THE FIRST DAY

interval between two RFQ due dates. TacTex-03 maintains an estimate of the capacity available to each supplier on each of the next fifty days, and updates its estimates by dividing the difference between the actual observed capacity and the estimated capacity in an interval equally among the days in the interval.

For each RFQ to be sent for a desired purchase, TacTex-03 considers all possible suppliers and all due dates between the current date and the desired delivery date, and chooses the supplier and due date that it believes will result in the lowest offered price for an on-time delivery. These price estimates will not be completely accurate due to the fact that additional orders may have been placed since the time the estimates of supplier capacity were formed, and the fact that capacities can only be determined for intervals, not for specific dates. Still, this strategy seems fairly effective at finding good prices on components.

The final element of the supply strategy used by TacTex-03 is deciding whether to accept offers from suppliers. We make the simplifying assumptions that due to the other parts of our strategy, TacTex-03 will have only requested components it can actually use, and that it will have obtained the best possible prices. The only question left is whether accepting an offer will be profitable. TacTex-03 makes this decision by calculating the marginal value of the components in each offer independently.

The marginal value of a component is equal to the difference in the expected profits with and without the added amount. TacTex-03 calculates these expected profits by projecting the future production needed to completely use up the type of component being considered, which may involve purchasing additional components of other types. TacTex-03 predicts the prices computers will be sold at and the prices of additional components based on recent history, and it predicts the ratios of computer types produced based on current inventory. If accepting the offered components increases the profit from such production by a larger amount than the price of the offered components, then TacTex-03 accepts the offer.

V. SALES

On the demand side, TacTex-03 uses a heuristic search process to try to find the set of offers that maximizes future expected profits. The profitability of a set of offers is estimated by first predicting the set of orders that will result and then planning for production based on these expected orders (and thus estimating revenue minus penalties). For each possible set of offers, the expected resulting orders are predicted by examining each offer individually and predicting the probability of the customer accepting it as a function of the offered price. These predictions are based on the daily price reports which indicate recent prices paid by customers.

Before describing the complete search process, we begin by laying out our method of predicting the probability that a given order will be accepted by a customer.

A. Predicting the Probability of an Order

In order to have some idea of what orders will result from offers to customers, we would like to have a way of predicting the probability of an offer being accepted as a function of the price offered. TacTex-03 makes this prediction for an RFQ by looking at the reported highest and lowest prices at which the requested type of computer sold over the past ten days. From these prices it finds the following five values:

- the lowest price the computer sold at
- the average low price
- the midpoint between the average low and the average high price
- the average high price
- the highest price

From an analysis of past game data, we obtain the following approximate estimates of the probability of an offer at each of these prices being accepted, in order: .95, .7, .45, .15, .05. To predict the probability of an order for an offered price, TacTex-03 finds which of the two above prices its offered price is between and linearly interpolates between the corresponding probabilities. Prices below the lowest price or above the highest price result in predictions of 1 and 0, respectively. Prices above the reserve price obviously generate predictions of 0.

These predicted probabilities depend only on the type of computer requested and the reserve price, ignoring the other properties of the RFQ. From our observations, this is reasonable for all properties except for the due date. For an agent operating on a make-to-order basis, the due date of an RFQ may be a very important factor, and later due dates may be preferable. This preference shows up in game data in the form of higher average prices for computers that are due sooner. The effect varies from game to game, most likely due to the differing agents' behavior.

TacTex-03 handles this issue by using values we call day factors, which are multipliers for the generated probabilities. The due dates for RFQs range from 3 to 12 days in the future, and a separate day factor is learned for each day in this range. TacTex-03 learns day factors by comparing actual orders received with expected orders. When an offer is made on an RFQ, TacTex-03 computes the probability of an order by multiplying its initial prediction by the corresponding day factor. It then records the expected number of orders for each due date. When TacTex-03 receives orders the next day, it divides the actual number of orders on each due date by the expected number of orders to find the ideal day factors that would have resulted in a correct prediction. The day factors are then updated by adjusting them slightly in the direction of the ideal day factors. Day factors are set to 1 at the beginning of the game. Thus, the day factors serve both as a means of gauging the impact of due dates on computer prices and as a mechanism for ensuring that the number of orders received is roughly the number expected. An analysis of game data shows that without the day factors, TacTex-03 tends to receive more orders with earlier due dates, but that the distribution evens out when day factors are used.

B. The Search Process

Our goal in bidding on RFQs is to find the set of offers that maximizes TacTex-03's expected profit. For a single RFQ, the optimal offer price is simply the value

$$argmax_{price}(price - cost) * P(order|price) \qquad (2)$$

If TacTex-03 were to offer this single-case optimal price on all RFQs, however, it might receive more orders than it was able to fill. There can be no advantage to offering any price lower than the single-case optimum, and we can only reduce the number of orders by offering higher prices, so the optimal set of offers will be one in which all prices are at or above the single-case optimum. So to find the optimal set of offers, TacTex-03 uses a heuristic search process that starts with the single-case optimum prices for each RFQ and iteratively raises prices on offers until doing so no longer increases the expected profit.

The expected profit for a set of offers can be determined by simulating production of the possible resulting orders. TacTex-03 plans production using a version of its greedy production scheduler that looks several days into the future and tries to maintain flexibility by producing each order as late as possible. First, TacTex-03 plans for the production of all current orders, to ensure that it reserves enough resources for these orders. Then, with the remaining resources, TacTex-03 can plan for the production of the orders resulting from the offers it is considering. The orders that will result can be determined by using the estimates of order probability described above. We first considered producing samples from these probabilities and finding the average profit, but we felt that it would not be possible to consider enough samples to produce an accurate result. Instead, TacTex-03 plans production of the expected orders, meaning that it considers the partial orders generated by multiplying the probability of an order by the quantity ordered.

If TacTex-03 considered all of its resources in planning for the production of orders resulting from the offers made on the current day, it might end up dedicating all of its resources for the near future to those orders and have nothing left to use in the production of future orders. We could solve this problem by having TacTex-03 allocate only a portion of its resources to be used in the production of the orders, but instead we take the approach of predicting the RFQs TacTex-03 will receive over the next several days and coming up with offers for these RFQs at the same time as the actual RFQs from the current day. The result is that TacTex-03 effectively reserves some resources for future RFQs that may be more attractive than some of the current RFQs. TacTex-03 randomly generates a predicted set of the future RFQs that would have due dates between 3 and 12 days in the future, because these are the RFQs that could contend for resources with the actual RFQs received on the current day.

The one piece of information TacTex-03 needs to generate the future RFQs is the number of RFQs that will be issued each day. We made some attempts to estimate the RFQ trend through line fitting, but we found that TacTex-03 obtained better results by simply setting the number of RFQs generated each day equal to the average number received over the past four days. A more sophisticated approach to estimating the trend is described in [5]

TacTex-03 begins its search process by planning for the production of its current orders, generating the predicted future RFQs and combining them with the current day's RFQs, and setting the initial offer price for each RFQ to the value obtained from formula 2. Each step of the search process then proceeds as follows. TacTex-03 generates the expected orders for the current set of offers by multiplying the predicted probability of winning each order by the quantity of computers requested. The greedy production scheduler is then applied to the expected orders, but this time the orders are sorted by their value divided by the production cycles required, because production cycles tend to be the main constraint during this process. Once this is done, TacTex-03 needs to determine which of the offer prices should be raised. We made the decision to have TacTex-03 raise the prices of RFQs resulting in orders that could not be produced by a set fraction of the base price. The reasoning behind this decision is that we want TacTex-03 to reduce the number of orders with values so low that it chooses to not produce them. While it is possible to construct situations in which this is not the correct decision, in practice the results obtained by our method appear fairly good.

After the prices have been raised, TacTex-03 recalculates the expected orders and repeats production. This process continues until all of the expected orders can be produced. This should also be the point at which the expected profit is highest due to the fact that the values of the computers produced increase at each step while the number of computers that cannot be delivered decreases. Once the final set of prices is determined, TacTex-03 sends offers to customers for those prices corresponding to actual RFQs. This iterative search process is summarized in Table IV.

VI. Competition Results

The 2003 TAC SCM competition consisted of a qualifying round, two seeding rounds, and the actual competition. Results are presented in Table VI. In the qualifying and seeding rounds, a large number of games (at least 60 per agent per round) were played between the 20 participating agents. TacTex-03 won the qualifying round with an average score more than twice that of any other competitor, partly due to

	Qualifying Round		*Seeding Round 1*		*Seeding Round 2*		*Quarterfinal Group 1*		*Semifinal Group 2*		*Finals*	
Rank	*Agent*	*Score*	*Agent*	*Score*	*Agent*	*Score*	*Agent*	*Score*	*Agent*	*Score*	*Agent*	*Score*
1.	**TacTex**	**33.65**	jackaroo	35.55	**TacTex**	**32.97**	PackaTAC	18.31	Botticelli	-4.83	RedAgent	11.61
2.	RedAgent	15.09	**TacTex**	**32.66**	RedAgent	29.52	PSUTAC	17.81	whitebear	-9.58	deepmaize	9.47
3.	jackaroo	14.89	UMBCTAC	30.16	Botticelli	28.03	RedAgent	12.75	**TacTex**	**-15.54**	**TacTex**	**5.02**
4.	Botticelli	13.88	RedAgent	24.57	jackaroo	19.23	Botticelli	5.69	Sirish	-20.21	Botticelli	3.33
5.	HarTAC	12.41	Botticelli	17.29	whitebear	16.50	whitebear	5.26	MinnieTAC	-24.98	PackaTAC	-1.68
6.	MinnieTAC	10.88	PSUTAC	15.52	PSUTAC	15.25	**TacTex**	**1.85**	UMBCTAC	-29.91	whitebear	-3.45

TABLE V

COMPETITION RESULTS FOR THE TOP SIX AGENTS IN EACH ROUND (SCORES REPRESENT AVERAGE SCORES IN MILLIONS OF DOLLARS)

- Reserve resources needed to fill current orders
- Predict future RFQs that will contend with current RFQs for resources
- Find the optimal bid for each RFQ by itself
- Repeat until all expected orders can be produced:
    - find the expected orders resulting from the current bids
    - sort orders by value per production cycle
    - plan production using the greedy scheduler
    - raise prices on all RFQs resulting in unfilled orders

TABLE IV

THE HEURISTIC SEARCH PROCESS FOR FINDING OPTIMAL BIDS

the fact that it was one of the first agents to use a first-day ordering strategy. As other agents improved, the performance gap shrank, but TacTex-03 still managed to place second in the first seeding round and win the second one.

During the qualifying and seeding rounds, it was interesting to watch the results as more and more agents began using the first-day ordering strategy. The level of customer demand often stays close to the upper or lower boundary for much of the game, and so most games can be described as high-demand or low-demand. Agents using a first-day ordering strategy tended to get very high scores in high-demand games and negative scores in low-demand games, with average scores much higher than those of agents not using such a strategy. Once we implemented the ability to base the number of components ordered at the beginning on the demand observed on the second day, TacTex-03 tended to achieve the highest scores in high-demand games while suffering more modest losses in low-demand games. Its scores were low in games where the demand began high and then moved significantly lower, however.

We expected to face much stronger agents during the actual competition, and that proved to be the case. During the qualifying and seeding rounds, most games included some agents that were not competitive against agents using the first-day ordering strategy, and sometimes agents failed to play in their scheduled games. This reduced the competition for customer orders and gave an artificial boost to those agents using the first-day ordering strategy. With six strong agents playing in a game, we knew that there would be a much higher risk of being unable to use all of the components ordered at the beginning and having a negative score.

The competition consisted of three rounds played over three days. Agents were divided into two brackets during the first two rounds. On the first day, each agent played in only six of the nine games per bracket, making comparisons between agents difficult. We finished sixth in our group, partly because we were the only agent to play in all four of the low demand games.

During the semifinal and final rounds, the same six agents always played against each other. In our semifinal bracket, all six agents used some form of a first-day ordering strategy, causing severe contention for components. The outcomes of games were heavily dependent on the order in which the suppliers considered the RFQs on the first day. Agents rarely had enough components for production until around day 50, and often an agent would still be unable to produce on day 100. In some cases, TacTex-03 received its first large delivery from a supplier near the end of the game, too late to be of much use. As a result, TacTex-03 would often have a shortage of one type of component for much of the game but end up with too much left at the end. Fortunately for us, all of the agents seemed to suffer from these problems. In fact, the average scores of all agents in our semifinal bracket were negative. TacTex-03 finished third out of the six agents, enough to move on to the finals.

As a result of the problems in the semifinal round, we modified TacTex-03's strategy for accepting offers on the second day to order fewer components and be more aggressive in rejecting offers that would be of no use based on the due dates of all offers, as described in section IV-A. Unfortunately, one of the agents in the final round, DeepMaize, surprised us by using a preemptive strategy of sending RFQs on the first day for more components than it could possibly use, in an attempt to block others from ordering the majority of their components at the beginning of the game [4]. This meant that not only did TacTex-03 not receive the normal number of offers for components, but it also rejected many of the offers it did receive, believing it would be unable to obtain the complementary components because these had been won by other agents. In fact, these components could probably have been bought following the first day, although not at the same low prices. Our strategy for obtaining components after the first day did not kick in until a few days had passed, and as it was a short-term strategy that only looked 50 days ahead, it was often unable to place any orders during the early part of the game because the supplier capacity had already been used up for that period. As a result, TacTex-03 ended up purchasing fewer components than all but one other agent during the finals. Still, we managed to come in third, behind RedAgent and DeepMaize.

The strategy used by TacTex-03 for bidding on customer RFQs worked well, keeping a steady stream of orders coming in without ever overburdening the production scheduler. An analysis of the finals provided by RedAgent[3] showed that TacTex-03 bid on fewer RFQs than most other agents but managed to sell computers at an average price that was about the same as, or better than, the prices of other competitors, with the exception of RedAgent. RedAgent bid on more RFQs than any other agent, seemed to make higher bids, and appeared willing to hold on to inventory rather than sell at low prices. The result was that RedAgent ended up with significantly higher average computer prices than any other agent. One potential flaw in TacTex-03's bidding process is that it tries to maximize its profits over the next few days, meaning that it will never plan to hang on to completed computers in inventory to sell in the future when prices may be higher. Also, TacTex-03 did not generally appear to bid prices that were significantly higher than the average selling prices, suggesting that our heuristic for estimating the probability of an order either underestimated the probability of winning an order at higher prices or otherwise discouraged high bids.

The greedy production scheduler appeared to work just fine. In general, TacTex-03 tended to have few penalties. As long as it is clearly possible to deliver all computers on time, any reasonable approach to production should perform optimally. It is only when some orders cannot be satisfied that differences between solutions will become apparent. For TacTex-03, penalties tended to occur in bunches when a delivery of components did not arrive when expected, preventing production of certain types of computers. In this situation where production is impossible for a sustained period, the choice of production strategy again makes little difference.

The situation where differences between production strategies would be most apparent is if an agent regularly received more orders from customers than it could produce at full production capacity, which is not a situation TacTex-03 ever seemed to face. This suggests that a good strategy for bidding on customer orders can reduce the impact of the production strategy.

VII. Conclusion and Future Work

In this paper we have introduced TacTex-03, a successful agent in the first TAC SCM competition. Its competition performance suggests that TacTex-03 can capably handle several of the duties required of an agent in a supply chain, and we hope to further analyze and tune its performance through more controlled experiments in the future.

Much of the prior discussion of the competition results centered around component supply in general, and the first-day ordering strategy in particular. While there is clearly room for improvement in making TacTex-03 more robust in adapting to the supply strategies of other agents, we anticipate significant changes to the game specifications in the area of component procurement that will lessen or change many of the problems we faced, and so have we set aside consideration of supply issues for the time being.

As described above, the only significant problem with the strategy used by TacTex-03 for production and delivery occurs when components are not delivered when expected. One possible solution, apparently used by some other agents, is to only consider current inventory when making offers to customers, but this would restrict the number of RFQs we could bid on. A better solution might be to improve our predictions of when future component deliveries will arrive through better modeling of supplier capacity.

The improvement of such predictions the main focus of our on-going research agenda related to TAC. Most of the decisions made by TacTex-03 are based on predictions of future circumstances. For example, determining the marginal value of an offer from a supplier requires predictions of future production, computer prices, and component prices. TacTex-03 makes many of its predictions by assuming that the current state of those factors will remain the same in the future. Finding ways to make more accurate predictions, such as by applying learning techniques, is a high priority.

VIII. Acknowledgments

We thank Ronggang Yu and Ali Niaz for participating in the design and development of TacTex-03. This research was supported in part by NSF CAREER award IIS-0237699.

References

[1] N. Sadeh, R. Arunachalam, J. Eriksson, N. Finne, and S. Janson, "TAC-03 a supply-chain trading competition," *AI Magazine*, Spring 2003.

[2] R. Arunachalam, J. Eriksson, N. Finne, S. Janson, and N. Sadeh, "The TAC supply chain management game," Swedish Institute of Computer Science, Tech. Rep., 2003, draft version 0.62.

[3] M. Benisch, A. Greenwald, V. Naroditsky, and M. Tschantz, "A stochastic programming approach to TAC SCM," in *Proceedings of the Fourth ACM Conference on E-commerce*, May 2004.

[4] J. Estelle, Y. Vorobeychik, M. Wellman, S. Singh, C. Kiekintveld, and V. Soni, "Strategic interactions in a supply chain game," University of Michigan, Tech. Rep., 2003.

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

[3] http://rl.cs.mcgill.ca/Projects/RedAgent/RedAgent.html