In *SIGecom Exchanges*,
4(3):19-28, 2004.

TacTex-03: A Supply Chain Management Agent

DAVID PARDOE and PETER STONE
Department of Computer Sciences, The University of Texas at Austin

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.

1. 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) [Sadeh et al. 2003], 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. The strategy used by TacTex-03 is detailed in Sections 2–4, and Section 5 summarizes its performance in the competition.

2. PRODUCTION AND DELIVERY

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. 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

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.

program was similar to the one designed for a slightly simplified scenario by [Benisch et al. 2004]. 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.

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.

3. 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 more detail below. First, we introduce the supplier pricing model, which is key to our strategy.

Suppliers are able to produce only a limited number of components per day, and this number varies according to a random walk. The prices suppliers offer are determined by the expected available production capacity before the due date. The price offered on day d for an offer due on day $d+i$ is

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

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 $d+i$.

3.1 First Day

As indicated by Equation 1, the price a supplier offers in response to an RFQ for components is based entirely on the supplier's available production capacity. 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 ca-

—Divide the current 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.

pacity 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. As significant changes to the supplier pricing model have been proposed for future competitions, we present here only a summary of our first day strategy. Further details are available from [Pardoe and Stone 2003].

TacTex-03's first day component approach follows the general strategy of sending large RFQs to suppliers on the first day. The most important decision involved in such a strategy is to determine the number of each component that should be ordered. The main factor influencing the number of computers that will be sold is the level of customer demand. This level varies throughout the game according to a random walk, 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. 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.

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. TacTex-03 solves this problem by sending a set of RFQs to each supplier requesting the largest number of components it could possibly need, and then accepting the subset of the resulting offers that provides it with approximately the number it determines it needs based on predicted total demand.

Before accepting offers, however, TacTex-03 checks to make sure that 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. TacTex-03 handles this problem by projecting its production of computers over the whole game, based on the available components. If TacTex-03 is able to use fewer of any component than the

[1] As noticed by [Estelle et al. 2003], more and more agents did indeed use this strategy as the competition proceeded.

amount it was planning to order, it reduces the planned orders accordingly. [2] The entire first day strategy is summarized in Table II.

3.2 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 lost 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.

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 is given by 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 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 $d+i$ by solving for $C_{available}(d+i)$ in Equation 1. 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

[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 [Estelle et al. 2003]. A possible response is to repeat the first day strategy on subsequent days, but we simply fall back on the post-first-day acquisition strategy described below.

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

Table II. The first-day ordering strategy used by TacTex-03.

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

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 maintain an estimate of the capacity available to each supplier on each of the next fifty days.

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 low 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.

4. 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.

4.1 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, and 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. 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.

4.2 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

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

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.

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.

5. COMPETITION RESULTS

The 2003 TAC SCM competition consisted of a qualifying round, two seeding rounds, and the actual competition. Results are presented in Table 5. 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 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.

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 a 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

—Reserve the resources needed to fill current orders
—Predict the 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

Rank	*Qualifying Round*		*Seeding Round 1*		*Seeding Round 2*		*Quarter-final Group 1*		*Semifinal Group 2*		*Finals*	
	Agent	*Score*	*Agent*	*Score*	*Agent*	*Score*	*Agent*	*Score*	*Agent*	*Score*	*Agent*	*Score*
1.	**TacTex**	**33.7**	jackaroo	35.6	**TacTex**	**33.0**	PackaTAC	18.3	Botticelli	-4.8	RedAgent	11.6
2.	RedAgent	15.1	**TacTex**	**32.7**	RedAgent	29.5	PSUTAC	17.8	whitebear	-9.6	deepmaize	9.5
3.	jackaroo	14.9	UMBCTAC	30.2	Botticelli	28.0	RedAgent	12.8	**TacTex**	**-15.5**	**TacTex**	**5.0**
4.	Botticelli	13.9	RedAgent	24.6	jackaroo	19.2	Botticelli	5.7	Sirish	-20.2	Botticelli	3.2
5.	HarTAC	12.4	Botticelli	17.3	whitebear	16.5	whitebear	5.3	MinnieTAC	-25.0	PackaTAC	-1.7
6.	MinnieTAC	10.9	PSUTAC	15.5	PSUTAC	15.3	**TacTex**	**1.9**	UMBCTAC	-29.9	whitebear	-3.5

Table V. Competition results for the top six agents in each round (scores represent average scores in millions of dollars)

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 3.1. 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 [Estelle et al. 2003]. 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 may have underestimated the probability of winning an order at higher prices.

The greedy production scheduler appeared to work well. 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. This suggests that a good strategy for bidding on customer orders can reduce the impact of the production strategy.

6. 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.

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 the main focus of our on-going research agenda related to TAC.

REFERENCES

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

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

Pardoe, D. and Stone, P. 2003. TacTex-03: A supply chain management agent. Tech. Rep. UT-AI-TR-03-308, The University of Texas at Austin, Department of Computer Sciences, AI Laboratory. At `http://www.cs.utexas.edu/home/department/pubsforms.shtml`.

Sadeh, N., Arunachalam, R., Eriksson, J., Finne, N., and Janson, S. 2003. TAC-03 a supply-chain trading competition. *AI Magazine 24,* 1 (Spring), 92–94.

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