Strategic Trading Agents via Market Modelling

Dongmo Zhang, Kanghua Zhao, Chia-Ming Liang, Gonelur Begum Huq, and Tze-Haw Huang
Intelligent Systems Laboratory
School of Computing and Information Technology
University of Western Sydney, Australia

This paper presents some key strategies applied in jackaroo agent. Most of the strategies are rooted in theoretical modelling and statistic analysis of TAC-03 SCM game. We model the product market with a variation of Cournot game and specify the component market by constant-supply model. We outline the basic theory and algorithms dealing with component procuring, product pricing, production scheduling and price forecasting.

Categories and Subject Descriptors: J.4 [**SOCIAL AND BEHAVIORAL SCIENCES**]: Economics; K.4 [**COMPUTERS AND SOCIETY**]: Electronic Commerce

Additional Key Words and Phrases: trading agent, supply chain management, Cournot model

1. INTRODUCTION

Trading Agent Competition (TAC) has been successfully run for four years since it was introduced by Wellman and Wurman [Wellman and Wurman 1999]. This annual activity offers "an international forum designed to promote and encourage high quality research into the trading agent problem". The first three games proceeded with a travel agency scenario (TAC Classic). In year 2003 a new game scenario of Supply Chain Management (SCM) was introduced by CMU and SICS [Arunachalam *et al.* 2003]. This scenario specifies a supply chain integration of Personal Computer(PC) marketplace. Participants are required to design a trading agent capable of sourcing of components, manufacturing of PC's and sales of products. The proposed game provides a competitive environment to stimulate solutions to the problems involved in supply chain integration and multiple market e-trading. About twenty teams from different universities and research institutes around the world were attracted to the challenge and competed each other in 2003. As one of the participants, jackaroo team, representing University of Western Sydney, contributed an agent to the tournament. The team received the third place in the qualifying round, the first in the seeding round 1 and the fourth in the seeding

This research has been supported in part by UWS Research Grant Scheme and Intelligent Systems Lab at University of Western Sydney.
Authors' address: School of Computing and Information Technology, University of Western Sydney, Locked Bag 1797, Penrith South DC, NSW 1797, Australia.

round 2. Unfortunately it was not able to proceed with the final round due to network problems at the conference venue. Nevertheless the agent has demonstrated its contributions of several leading strategies to the competition.

This paper describes the key strategies adopted in jackaroo agent. Most of the strategies are rooted in theoretical modelling and statistic analysis of the game. We model the product market with a variation of Cournot game and specify the component market by constant-supply model. We will present the basic theory and algorithms dealing with component procuring, product pricing, production scheduling and price forecasting.

2. TAC SCM SCENARIO MODELLING

In this section, we present an abstract model for TAC SCM game. The game scenario specifies a typical supply chain with three nodes: *component suppliers, PC manufacturers* and *end customers*, linked with two marketplaces: *component market* and *PC product market*[Arunachalam *et al.* 2003].

$$suppliers \overset{component\ market}{\Leftrightarrow} manufacturers \overset{product\ market}{\Leftrightarrow} customers$$

We will model the downstream market as a Cournot oligopoly and the upstream market with constant-supply model.

2.1 Product market model

The SCM product market is a typical oligopoly where the manufacturers (agents) choose quantities supplied to maximize their profit. Since each manufacturer's payoff structure is common knowledge to each agent, the market can be easily specified by Cournot model with a slight variation[Mendenhall *et al.* 1986]. To simplify the exposition, we assume that all PC products are homogeneous with the same market price and the cost of production for each manufacturer is the same. Obviously the model is also applicable to single product analysis.

In general, we assume that there are n manufactures competing the market. Let q_i denote the quantity of PC produced by manufacturer i and Q the aggregate quantity on the market, that is, $Q = \sum_{i=1}^{n} q_i$. Assume that the market demand (expressed by customer's RFQs) for all products is D_m. If the aggregate quantity on the market is no more than the market demand, we assume that the market-clearing price of the product is constant at p_0[1]. If the aggregate quantity is larger than the market demand, the market-clearing price is decreasing with over level of products on the market until the price becomes 0. Let $P(Q)$ denote the market-clearing price over aggregate quantity Q. Then

$$P(Q) = \begin{cases} p_0, & \text{if } Q \leq D_m; \\ \Gamma(Q - D_m), & \text{otherwise.} \end{cases}$$

where $\Gamma(Q)$ is a monotonous decreasing function, called *price descent function.* Specially, if the price decline is linear, the price function can be further simplified:

[1]If market demand is more than supply, agents can normally get customer orders with reserve price.

$$P(Q) = \begin{cases} p_0, & \text{if } Q \le D_m; \\ p_0 - \gamma(Q - D_m), & \text{if } D_m < Q \le \frac{p_0}{\gamma} + D_m; \\ 0, & \text{otherwise.} \end{cases}$$

where γ is called the *price descent coefficient* ($\gamma > 0$). Based on the analysis in Section 3.2, we will assume that the price decline is linear.

Let δ be the unit cost of the product and c_0 be the base cost of each manufacturer. The cost for each manufacturer to produce q products is then:

$C(q) = \delta q + c_0$, where δ and c_0 are non-negative.

For each manufacturer i, a strategy, q_i, is the quantity the manufacturer chooses to produce. A strategy profile, S, is a decision of production by all manufactories: $(q_1, \cdots, q_n)$, where $q_i \ge 0$ for any i.

The profit of each manufacturer i (payoff function) can then be written as:

$$\pi_i(S) = q_i P(Q) - C(q_i) = q_i P(Q) - \delta q_i - c_0.$$

If we assume that the price function is linear, the payoff function can be further specified as the following:

$$\pi_i(S) = \begin{cases} (p_0 - \delta)q_i - c_0, & \text{if } Q \le D_m; \\ (p_0 - \delta)q_i - \gamma q_i(Q - D_m) - c_0, & \text{if } D_m < Q \le \frac{p_0}{\gamma} + D_m; \\ -\delta q_i - c_0, & \text{otherwise.} \end{cases}$$

Therefore a strategy profile $(q_1^*, \cdots, q_n^*)$ is a Nash equilibrium if, for each player i, q_i^* solves the optimization problem:

$$\max_{0 \le q_i < \infty} \pi_i(q_1^*, \cdots, q_{i-1}^*, q_i, q_{i+1}^*, q_n^*)$$

THEOREM 2.1. *Assume that the product descent function Γ is linear. If $p_0 > \delta$, there exists a unique Nash equilibrium $(q_1^*, \cdots, q_n^*)$ to the problem where if $D_m \ge n\frac{p_0 - \delta}{\gamma}$, $q_i^* = \frac{1}{n} D_m$; if $D_m < n\frac{p_0-\delta}{\gamma}$, $q_i^* = \frac{1}{(n+1)}(\frac{p_0-\delta}{\gamma} + D_m)$ for each i.*

Due to space limitation, we omit the proof of the theorem[2]. This theorem shows that the SCM product market is not a standard Cournot oligopoly. Since the market price is capped with customer reserve price, the manufactures can not fully control the market price with market supply.

EXAMPLE 2.2. *Consider a PC marketplace with n suppliers. Assume that $p_0 = 2000$, $\gamma = 2.0$, $\delta = 1000$, $c_0 = 0$ and $D_m = 2000$. Then the market-clearing price function is*

$$P(Q) = \begin{cases} 2000 & \textit{if } Q \le 2000; \\ 6000 - 2Q & \textit{if } 2000 < Q \le 2500; \\ 0 & \textit{otherwise.} \end{cases}$$

For each manufacturer i, its profit is decided by the following function:

[2]A proof of the theorem will be presented in a sequent paper[Zhang 2004].

$$\pi_i(S) = \begin{cases} 1000q_i & \text{if } q_i \le 2000 - \sum_{k \ne i} q_k; \\ 5000q_i - 2q_i^2 - 2q_i \sum_{k \ne i} q_k & \text{if } 2000 - \sum_{k \ne i} q_k < q_i \le 2500 - \sum_{k \ne i} q_k; \\ -1000q_i & \text{otherwise.} \end{cases}$$

The following table lists the equilibrium production in case of no more than 6 manufacturers.

Number of Manufacturers	*Equilibrium Production*	*Aggregate Production*	*Market Price*
1	*2000*	*2000*	*2000*
2	*1000*	*2000*	*2000*
3	*666*	*1998*	*2000*
4	*500*	*2000*	*2000*
5	*417*	*2085*	*1830*
6	*357*	*2142*	*1716*

The last line shows a typical situation in TAC'03 SCM game. According to the game specification, the average base cost of each PC product is \$2000. If an agent orders components on day 0, the actual average cost of each PC is half of base price, so it is \$1000. The customers' reserve price for each PC product is between 75% to 125% of the base PC cost. Thus the capped market price of each PC product is averagely about \$2000. The above example shows that if the average market demand is no less than 2000 PCs per day, an agent should make full use of its production capacity (see more analysis in Section 3).

We remark that in the real TAC SCM game, both the base price p_0 and the price descent coefficient γ vary with market demands. According to the statistic analysis on the game data in TAC-03 SCM final, p_0 is a non-linear function of market demand and price descent coefficient can be approached by a linear function of market demand(see [Zhang 2004] for more details).

2.2 Component Market Model

Different from the product market, the market of components is a typical constant-supply market. Market supply is fixed with only a small reverting random walk. Each supplier of components has an allocated output quota for each component type it supplies, called *nominal capacity*, denoted by $C_{nominal}$. A supplier accepts orders assuming that it will have $C_{nominal}$ products available every day for each component. The price of the component is determined by its market demand capped with a base price, specified in the game specification.

Consider a single component. Let $C_{ordered}(d, d')$ denote the total ordered amount of the component up to day d which are requested for delivery on day d' (where $d' > d$). Specifically, $C_{ordered}(0, d') = 0$ for any d' since no any component order can be made on day 0 according to TAC SCM specification. Then the total available capacity from day d to day d' will be:

$$C_{available}(d, d') = \sum_{j=d}^{d'-1} (C_{nominal} - C_{ordered}(d, j))$$

Note that the nominal capacity on day d' is not included since it is assumed that products can only be shipped on the next day after they are produced.

With all the information, the component supplier decides its product price by using the following formula:

$$p(d,d') = p_{base}(1 - \delta_p \frac{C_{available}(d,d')}{(d'-d)C_{norminal}})$$

where $p(d,d')$ denotes the day d's price for the product requested for delivery on day d'; p_{base} is its base price; δ_p is a price discount factor, which is assumed to be a fixed value at 50% in TAC-03 SCM specification.

From the pricing function of components, we can easily see that the earlier an order for a component is placed, the more available capacity for the component is left, and the cheaper the component will be. Since it is not allowed $C_{avaliable}$ to be negative, the discount of component price is always between 0 to 50% on the base price. Notably, ordering components on day 0 can always receive the top discount of 50%, i.e., $p(0,d') = 0.5p_{base}$ for any $d' > 0$. This gives every agent a great bargain. Unfortunately this setting led the component market into an unintended situation in TAC-03 competition (see more analysis in Section 3.1).

3. STRATEGIZING AGENTS

We are now ready to present some key strategies that have been applied in the implementation of jackaroo agent.

3.1 Component Procuring

The component market model has suggested that ordering components earlier would reduce product cost up to 50%. One successful strategy adopted by jackaroo agent was that of ordering all components for a whole game at the very beginning of the game.

As we know, each manufacturer is capable of producing around 360 PCs per day, which requires 90 units of each type of CPU and 180 units of each type of the other components. Since the number of RFQs an agent can send to each supplier is limited (maximal ten RFQs per supplier per day), we package several day's component usage into one RFQ (about 270 to 540 for CPUs and 540 to 1080 per RFQ). With this approach, we can order all the components for whole game in the first few days. Figure 1 shows a result by using the strategy during the early state of TAC-03 game (game 418 on tac6).

In this game, we ordered 213 day's components in the first 8 days. The due date of each order was set a little bit earlier than it is actual needed in order to avoid possible delay with every 5 day's interval. By using this approach, we only paid 63.66% of full price for all components.

This strategy was only used in the qualifying round before we realized that ordering components on day 0 would received the top discount of 50%. From the seeding round 1, we changed the strategy to order all components on day 0. Sooner after most of the other competitors applied the similar strategy as well. Unfortunately this caused the component market out of functioning. Due to the limitation of supplier's production capacity, component delay became the most headache of every agent. The result of a game heavily depended on the random arrival of components. The game became less interesting.

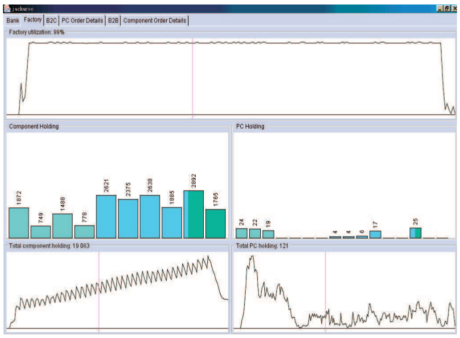

Fig. 1. An example of component procurement scheduling (data from TAC SCM game 418 on tac6).

There are numbers of remedies to the problem. One idea is to randomize the price discount on day 0 so that ordering components on day 0 does not necessarily gain an advantage. Another idea is to price components on RFQed quantity rather than on ordered quantity. Additionally, raising bank interest would be also a solution to the problem. We are now testing these possible solutions and hopefully can have some suggestions available for TAC-04.

3.2 Price coefficient learning

The decision of daily production is one of the biggest challenges faced by every SCM agent designer, especially if components are ordered at the very beginning. The product model provides a guideline for such a decision-making. Once we have a price function $P(Q)$, Theorem 1 would suggests a production quantity for each manufacturer. However, several problems need to be solved before it can be actually used:

—How many factors affect the market-clearing price?

—Whether the market-clearing price of each product is linear correlated to its redundant supply in the PC market?

—If it is linear, how to determine the price descent coefficient γ?

Besides market demand and aggregate quantity of products in the market, there are several other factors affecting the market product price: *agent's pricing strategy*, *stages of a game*, *component supply*, and *so on*. Nevertheless, market demand and market supply are still the most significant factors affecting market prices. Figure 2 shows a statistic result that produced by an off-line learning program:

The data picked up from game 427 on tac5. To reduce the influences of other factors, we ignored the first and last 50 day's data and average every five day's statistic results. Instead of a use of multiple linear regression, we simply consider the correlation between the following two variables:

—*price descent* (the difference between customer's reserve price p_0 and market-clearing price $P(Q)$),

—*redundant market supply* (the difference between product supply Q and market demand D_m).

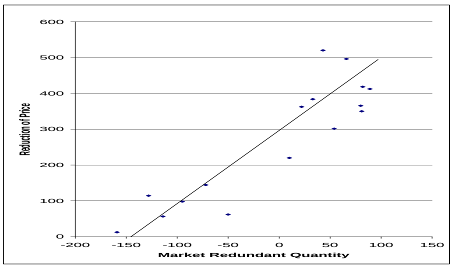

Fig. 2. Correlation between market price and market supply of PC products(data from TAC SCM game 427 on tac5

The statistic result shows a highly linear correlation between these two variables. The coefficient of correlation is up to 0.8357. The associated price decline coefficient γ is 2.0496. This encouraged us to use linear method to approximate price function. With the linear regression method, we can easily estimate the price decline coefficient γ, which could range from 0.1 to 10, depending on market demands, different competitors and stages of a game. We will present more statistic results and detail algorithms in a sequent paper.

Interestingly, the product market model implies that an agent should maximize its use of production capacity if market demand is above average level of 2000 PCs per day. Considering some unproductive days, this level can be even lower. According to our observation, if the average market demand is higher than 1500, maximizing production capacity is always profitable if components can be obtained at a good price.

We would like to remark that a linear approximation of price function does not imply the price descent coefficient to be constant during a whole game. In fact, the price descent coefficient γ is a function of time. Normally the abstract value of the coefficient would be much smaller than normal in the early stage of a game because most agents wait for components (less market supply) and much higher at the very end since most agents tend to dump their goods before the game ends.

3.3 Production Scheduling

There are many of tradeoffs faced by TAC SCM agent designers. One of them is to decide which types of products should be produced for inventory[3]. On one hand, one should produce more profitable products to keep a good inventory of these products. On the other hand, a balance of products in inventory should be maintained to maximize the ability of bidding customer orders. In the implementation of our agent, we adopted a dynamic weighting approach to production scheduling.

Let $v(k)$ denote the current inventory of product k $(0 \le k < 16)$ and v_{max} be their maximal value. We first calculate the inventory weight $w_1(k)$:

$$w_1(k) = (v_{max} - v(k))/(\sum_{i=0}^{16}(v_{max} - v(k)))$$

[3]This wasn't applicable for those agents who never produce inventory.

Next, we calculate the market profit weight. Let $p_m(k)$ and $c(k)$ be the current market price and component cost of product k, respectively. The profit weight $w_2(k)$ is defined as follows:

$$w_2(k) = \begin{cases} 0, & \text{if } p_m(k) - c(k) \leq 0; \\ (p_m(k) - c(k))/(\sum\limits_{p_m(i)-c(i)>0} (p_m(i) - c(i))), & \text{otherwise.} \end{cases}$$

Finally we combine these two weights with a balance coefficient λ, which was learnt from previous games:

$$w(k) = \lambda w_1(k) + (1 - \lambda) w_2(k), \quad k = 0, \cdots, 16$$

The actual quantity produced for inventory is then

$$q(k) = \frac{w(k) * r_{clycles}(k)}{u_{cycles}(k)}$$

where r_{cycles} represents the current remaining cycles after producing all the outstanding orders and $u_{cycles}(k)$ the required cycles per unit of product k.

3.4 Price forecasting

According to TAC-03 SCM specification, customer's reserve price(cap price) was setup as 75% to 125% of its base cost. This means that there is no much margin between baseline product cost and its market value. Manufacturers have to try their best to reduce their component cost. Since there is no cost for inventory and bank interest is relatively low, one can stock a large amount of components when their prices are low. Therefore finding an appropriate method to predict the market price of components is essential to agent design. We tested three different forecasting methods: *linear regression*, *moving-averaging* and *exponential smoothing*[Mendenhall *et al.* 1986]. Experiments show that the classical linear regression is the most inefficient way among the others, which could produce up to 30% of error by means of Mean Absolute Deviation(MAD), while the exponential smooth model can normally produce a high quality of predicting. Figure 3 shows a snapshot of price prediction window produced by our program when we test different forecasting models.

The green line (the third one at the beginning) represents the actual value of the predicted component price. The yellow line (the first one), blue line (the second one) and red line(the forth one) show the predicting values with linear regression, second-order smoothing and third-order smoothing, respectively. The maximal MAD of forecasting error is 28.3%, 12.5% and 11.0%, respectively.

We used the standard algorithm for calculating regression line, which can be found in any statistic book. The algorithms for high-order smoothing forecasting come from [Mendenhall *et al.* 1986]. Briefly, suppose that we have the observations of price of a component: $p_1, p_2, \cdots, p_d$ for a time series from day 1 to day d and we are going to predict the price on day $d + i$. The second-order smoothing model

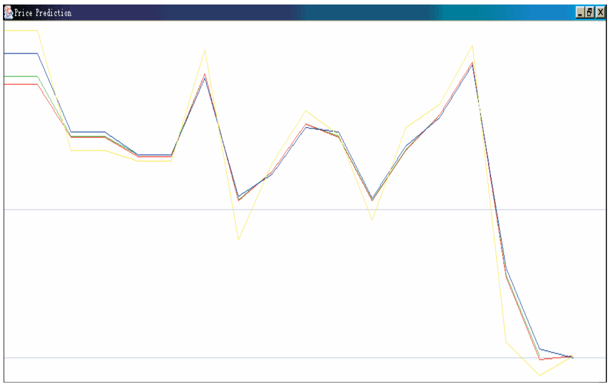

Fig. 3. Comparison on forecasting models for component price.

gives a predicting value of the price as:

$$\hat{p}_{d+i} = (2 + \frac{\alpha i}{1-\alpha})S_d - (1 + \frac{\alpha i}{1-\alpha})S_d(2)$$

where α is the smoothing constant; $S_d = \alpha p_d + (1-\alpha)S_{d-1}$; $S_d(2) = \alpha S_d + (1-\alpha)S_{d-1}(2)$.

The third-order smoothing forecasting model gives a value as:

$$\begin{aligned}\hat{p}_{d+i} = \; & (6(1-\alpha)^2 - (6-5\alpha)\alpha i + \alpha^2 i^2)\tfrac{S_d}{2(1-\alpha)^2} \\ & -(6(1-\alpha)^2 + 2(5-4\alpha)\alpha i + 2\alpha^2 i^2)\tfrac{S_d(2)}{2(1-\alpha)^2} \\ & +(2(1-\alpha)^2 + (4-3\alpha)\alpha i + \alpha^2 i^2)\tfrac{S_d(3)}{2(1-\alpha)^2}\end{aligned}$$

where $S_d(3) = \alpha S_d(2) + (1-\alpha)S_{d-1}(3)$.

Interestingly, the actual value of component price mostly goes between the values given by the second-order model and third-order model, respectively (see Figure 3). Therefore, in the implementation of our agent, we used the mean value of the second-order prediction and third-order prediction. We call the approach $2\frac{1}{2}$-order smoothing model. The smoothing constant we used is a value between 0.6 and 0.8. To get the component price update, we send RQFs to suppliers everyday, ordering only one unit. However we does not keep all past data for forecasting since it requires huge of memory and does not necessarily give us better results. We normally keep past twenty day's data for predicting purpose. Unfortunately this approach did not give us too much benefits during the competition since we only order a very small portion of components in the later days. We believe that component price prediction will play more important role once the problem of component pricing is fixed.

4. CONCLUDING REMARKS

The previous TAC games have proved its efficiency in promoting high quality of research on autonomous agent design and E-trading[Wellman and Wurman 1999] [Wellman *et al.* 2001] [Greenwald and Stone 2001] [Wellman *et al.* 2002]. The game platform not only provides a competitive environment to evaluate different trading strategies and different structure of agents but also a testbed for examining artificial market rules. An electronic market running with purely autonomous agents without sufficient testing would be in great dangerous. Researches pushed by the competition could be in two directions: efficient trading strategies which are capable of coping with any market situations and evolvable market-building mechanisms to avoid possible degeneration of market functions.

The authors would like to thank the anonymous referees for their valuable comments. We also wish to thank Yan Zhang, Ian Walsh, Chris Sheerman, Luke-karol Petelczyc for their supports during TAC-03 competition.

REFERENCES

Arunachalam, R., Eriksson, J., Finne, N., Janson, S., and Sadeh, N., The TAC supply chain management game, http://www.sics.se/tac/TAC03_spec.PDF, 2003.

Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., and Janson, S., Supply chain management game for the Trading Agent Competition, in: *Proceeding of IJCAI-03 Workshop on Trading Agent Design and Analysis*,pages 1-6,2003.

Chopra, S. and Meindl, P., *Supply Chain Management: Strategy, Planning and Opertiation*, Pearson Prentice Hall, 2000.

Greenwald, A. and Stone, P., Autonomous bidding agents for the trading agent competition, *IEEE Internet Computing*, 2(5), pages 52-60, 2001.

Kelly, A., *Decision Making using Game Theory: an Introduction for Managers*, Cambridge University Press, 2003.

Mendenhall, W., Reinmuth, J., Beaver, R. and Duhan, D., *Statistics for Management and Economics*, Fifth Edition, Duxbury Press, 1986.

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

Wellman, M. P., Wurman, P. R., O'Malley, K., Bangera, R., Lin, S., Reeves, D. and Walsh, W. E., Designing the market game for a Trading Agent Competition, *IEEE Internet Computing* March/April, pages 43-51 2001.

Wellman, M. P., Greenwald, A., Stone, P. and Wurman, P. R., The 2001 Trading Agent Competition, in: *IAAI-02*, pages 935-941,2002.

Zhang, D., Economic Model of TAC SCM Game, mineo, School of Computing and Information Technology, University of Western Sydney, Australia, 1-8, 2004.