The CrocodileAgent: Analysis and comparison with other TAC SCM 2005 agents

Ana Petric, Vedran Podobnik, Gordan Jezic

Faculty of Electrical Engineering and Computing / Department of Telecommunications
10000 Zagreb, Croatia
{ana.petric, vedran.podobnik, gordan.jezic}@fer.hr
http://www.tel.fer.hr

Abstract. The Trading Agent Competition (TAC) is an international forum which promotes high quality research regarding the trading agent problem. One of the TAC competitive scenarios is Supply Chain Management (SCM) where six agents compete by buying components, assembling PCs from these components and selling the assembled PCs to customers. In this paper, we briefly describe the strategies implemented in the CrocodileAgent, our entry in 2005 TAC SCM. The agent's performance in a series of controlled experiments is discussed.

1 The CrocodileAgent

The CrocodileAgent is an intelligent agent developed at the Department of Telecommunications, Faculty of Electrical Engineering and Computing in Zagreb, Croatia. Designing the CrocodileAgent was an extension of a project that started in 2004 when the KrokodilAgent [1] was developed. The CrocodileAgent participated in the TAC SCM 2005 [2] and achieved 16th place in the qualifications, 12th place in the seedings and 4th place in its quarterfinal group.

The central agent component is the ZTELAgent which coordinates collaboration between all the other agent components. The CustomerImpl is responsible for the agent's interaction with customers. The InventoryImpl keeps track of component prices and provides enough components for continuous PC production. The SupplierImpl sends orders to suppliers and takes care of delivered components.

A simplified description of the algorithm used for component purchase follows. The CrocodileAgent implemented two supply tactics: day-0 procurement and ordering components during the game. The agent ordered a large number of components on day-0 by sending RFQs with the following delivery dates: 3, 9, 17, 27 and 69. The reserve prices the agent was willing to pay for the components were 102%, 107%, 92%, 82% and 77% of the nominal price on the respective delivery dates. If the need for components occurred during the game, additional components were ordered. The agent always accepted partial offers. Each day the agent calculated the component quantity ordered, but not delivered up to that moment, and multiplied it with a distance factor. The agent checked the number of components currently stored in the warehouse and the number of components ordered to see if it was necessary to order

Petric, Ana; Podobnik, Vedran; Jezic, Gordan.
The CrocodileAgent: Analysis and Comparison with other TAC SCM 2005 Agents.
Proceedings Trading Agent Design and Analysis and Agent Mediated Electronic Commerce (TADA AMEC) Joint International workshop AAMAS 2006
Fasli, Maria; Shehory, Onn, editor(s).
Hakodate, Japan: AAMAS, 2006. 202-205.

components that day. The agent sent long-term RFQs with larger quantities to ensure long-term occupancy of the warehouse. The quantities were modified depending on customer PC demand and component shortage on the market.

Next we describe the PC selling algorithm used. The CrocodileAgent graded RFQs from 1 to 10, where a higher grade marked more profitable RFQs; RFQs with the same grade were sorted in chronological order of their delivery dates. The grades were determined by calculating the difference between the customer's reserve price and the agent's cost of producing a certain PC. In case there were not enough components to produce the requested PCs, the agent checked to see if they could be delivered from the reserve PCs stored in the warehouse. The basic PC price was calculated by summing the average prices of each component incorporated in the PC. The agent's desired profit was a percentage of the basic PC price and it varied depending on customer PC demand, factory utilization, previous prices and other factors.

Both algorithms can be applied to auctions which include supply chain management subject to minor modifications regarding game parameters. Since the algorithms are not tightly connected, they can be used separately in First-price Sale-bid auctions.

2. Controlled Experiments

To evaluate the agent's performance, we held a competition with some of the best agents from the TAC SCM 2005. The chosen opponents were: TacTex [3, 4], Mertacor [5], DeepMaize [6], MinneTAC and the PhantAgent. The final ranking is shown in Table 1. We conducted a detailed analysis of the games played mostly with the CMieux Analysis and Instrumentation Toolkit for TAC SCM [7].

The average prices paid for each component type are shown in Figure 1. The CPU price accounts for more than 50 % of the PC price so it is very important to purchase cheap CPUs. TacTex bought some of the cheapest CPUs while DeepMaize paid the highest prices for CPUs. The CrocodileAgent bought the third cheapest CPUs. This is interesting since the CrocodileAgent does not have a sophisticated algorithm with supplier capacity estimations and customer demand predictions to determine component prices and required quantities. We can see that Mertacor bought the cheapest motherboards and memory, while the CrocodileAgent bought the cheapest hard disks. MinneTAC bought the most expensive motherboards, memory and hard disks.

The number of sold PCs are shown in Figure 2. All the agents mostly sold Mid Range PCs. MinneTAC sold more High Range PCs than Low Range PCs. The CrocodileAgent sold an equal number of Low and High Range PCs, while the remaining agents sold significantly larger quantities of Low Range PCs than High Range PCs.

Table 1. Competition results at server `mobility3.labs.tel.fer.hr` .

Position	Agent	Score	Position	Agent	Score
1	TacTex	5.64M	4	PhantAgent	0.98M
2	DeepMaize	4.58M	5	MinneTAC	-0.43M
3	Mertacor	1.36M	6	CrocodileAgent	-1.24M

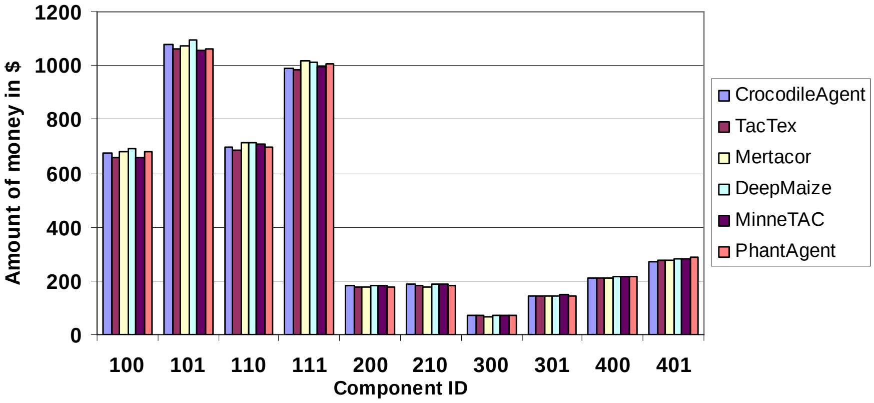

Fig.1 Average component prices

If we look at the average PC selling prices in Figure 2, we can see the reason for the CrocodileAgent placing last in this competition. Namely, it sold the cheapest PCs. The reason for this lies in the PC selling algorithm and the way the agent determined his desired profit. DeepMaize achieved the highest average PC selling prices for Low and Mid Range PCs and the second highest average PC selling price for High Range PCs. MinneTAC also has a successful selling algorithm which achieved the highest average PC selling price for High Range PCs.

To conclude, we can say that TacTex won the competition due to its highly efficient procurement algorithm and a good PC selling algorithm. The large number of components bought and PCs sold only contributed to the victory. DeepMaize lost first place due to its very high component purchase prices, especially CPUs. Mertacor performed best between the three agents which sell smaller amounts of components. It bought rather expensive CPUs but some of the cheapest motherboards, memory and hard disks. Its PC selling prices were average and that was enough for it to place third in the competition.

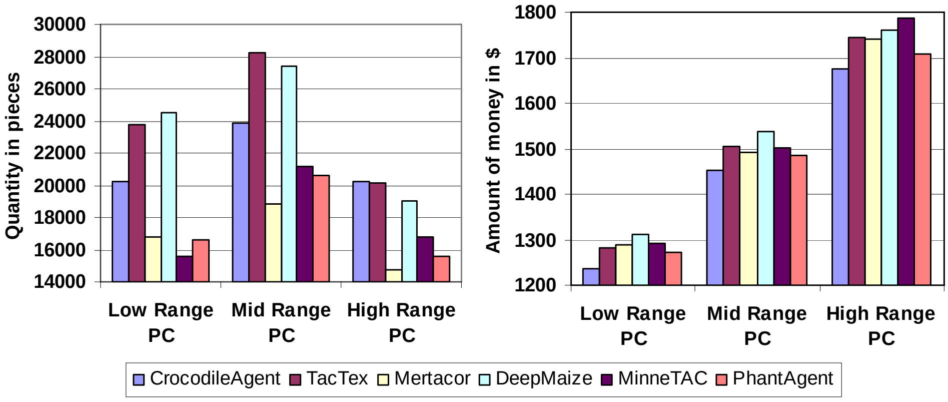

Fig. 2 Total quantity and the price of sold PCs.

The PhantAgent bought rather cheap components but also sold some of the cheapest PCs. When combined with the small amount of PCs sold, the final score of the PhantAgent put it in fourth place. The MinneTAC agent performed best in selling High Range PCs at very high prices. MinneTAC finished in fifth place. Some of the reasons include very high component purchase prices and a small total number of PCs sold.

3. Conclusions and Future Work

We briefly listed the important basic agent tasks and explained how they were implemented in the CrocodileAgent. In order to improve the functionalities of the agent, we held a competition with some of the best agents in TAC SCM 2005. We figured that this was a good way to determine the agent's soft spots. The results were a little discouraging since the CrocodileAgent placed last, but a lot was learned. The main reason for the CrocodileAgent's results lies in its reactive PC selling algorithm. This algorithm does not predict the fluctuation of prices on the PC market. Instead, it only reacts to the current state of the PC market and regulates its desired profit per PC. The component purchase algorithm functions quite well, but there is always room for improvement. The changes of the component purchase rules introduced in TAC SCM 2005 required a lot of work on the component purchase algorithm. Since there are no major changes in TAC SCM 2006, the CrocodileAgent represents a good basis for further development. Special attention needs to be dedicated to the PC selling algorithm with an emphasis on customer demand prediction and the prediction of winning PC prices. At this time, we are conducting some experiments which introduce fuzzy logic into the agent to help improve some of its functionalities.

References

1. Petric, A., Jurasovic, K.: KrokodilAgent: A Supply Chain Management Agent, ConTEL, Zagreb, Croatia, 2005, pp. 297-302.
2. Collins, J., Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., Janson, S.: The Supply Chain Management Game for the 2005 Trading Agent Competition, http://www.sics.se/tac/tac05scmspec_v157.pdf. Date accessed: Apr 13, 2005.
3. Pardoe, D., Stone, P.: Bidding for Customer Orders in TAC SCM: A Learning Approach, AAMAS-04 Workshop on Trading Agent Design and Analysis, New York, USA, 2004.
4. Pardoe, D., and Stone, P.: TacTex http://tac.eecs.umich.edu/research_reports/TacTex04poster.pdf
5. Kontogounis, I., Chatzidimitriou, K. C., Symeonidis, A. L., Mitkas, P. A.: TAC SCM 2005: Agent MerTACor, Aristotle University Thessaloniki. http://tac.eecs.umich.edu/research_reports/MertacorSCMPoster.pdf
6. Jordan, P., Kiekintveld, C., Miller, J., Wellman, M. P.: Deep Maize 2005 http://tac.eecs.umich.edu/research_reports/DM05-poster.pdf
7. Benisch, M., et. al.., "CMieux Analysis and Instrumentation Toolkit for TAC SCM", Carnegie Mellon University School of Computer Science Tech. Report CMU-ISRI-05-127, 2005.