Multi-Agent Decision Support System for Supply Chain Management

Yevgeniya Kovalchuk
Department of Computing and Electronic Systems
University of Essex
+44(0)1206 87 3805

yvkova@essex.ac.uk

ABSTRACT

This paper presents an extended abstract of the author's doctoral research project on developing a multi-agent intelligent system for automatic managing supply chains. Supply chain management (SCM) is a very complex and dynamic environment. The doctoral work, which started in October 2005, is dedicated to finding better solutions for successful performance in the domain of real-time SCM.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

H.4.2 [**Information Systems Applications**]: Types of Systems – *decision support*

General Terms

Economics, Algorithms, Design, Experimentation

Keywords

Supply Chain Management, Trading Agents, Decision Support Systems, Multi-Agent Systems, Prediction, Learning, Neural Networks, Genetic Programming.

1. INTRODUCTION

While running their business, enterprises usually deal with a number of activities, such as: procurement, production, warehouse management, selling, marketing, and customer servicing among others. To help them to manage these activities, organisations try to automate their business processes. Usually, independent software and hardware solutions are used for each of the activities. However in practice, all the activities are highly connected and interdependent. To integrate some of them in a single process is the task of supply chain management (SCM). The SCM is concerned with negotiating with suppliers for raw materials, competing for customer orders, managing inventory, scheduling production, and delivering goods to customers. In addition to its complexity, the SCM is also a time-constrained and ever-changing process, especially nowadays, when enterprises move their business on-line. Taking into consideration market globalisation, companies often run distributed businesses, having suppliers and customers all over the world. To deal with their contractors, organisations use the Internet to participate in electronic commerce, where business occurs very fast. To be able to react to all changes quickly, companies are looking for applications that can support dynamic strategies and adapt to new conditions in the environment. The development of such an intelligent decision support system for SCM is the main objective of the author's PhD project.

Although the aim is to develop an integrated application for SCM, due to its complexity, it is difficult to address all the issues which can arise in the domain of SCM. To narrow the research scope, the project is mainly focused on the demand part of the supply chain. In particular, different methods for predicting customer offer prices that could result in customer orders (winning bidding prices) are explored and compared in the system. The motivation is that expected findings not only can improve a company's performance while running its supply chains, but could also be applied to financial markets and online auctions where the task of predicting winnings bidding prices is crucial. The TAC SCM game, where software agents developed by different research groups can compete against each other in the context of the SCM, is used as a test bed to evaluate the proposed algorithms. This simulated environment was implemented by Carnegie Mellon University and the Swedish Institute of Computer Science (SICS) in 2003 as part of the International Trading Agent Competition (http://www.sics.se/tac/). The game is now probably the best vehicle for testing SCM systems as it encapsulates many of the tradeoffs that could be found in real SCM environments: time-constraints, network latency, unpredictable opponents, etc.

The rest of this paper is organized as follows. The description of the TAC SCM scenario and overview of related work are provided first. Then, the research approach followed is presented. The results achieved so far along with the plans for future work are given next. The paper closes with the conclusions.

2. THE TAC SCM SCENARIO

According to the TAC SCM scenario [4], there are six agents competing in the game that act as product manufacturers (Figure 1). Their main tasks are to buy components from suppliers, produce computers and sell them to customers. The behaviour of both suppliers and customers are simulated by the TAC server.

10th Int. Conf. on Electronic Commerce (ICEC) '08 Innsbruck, Austria

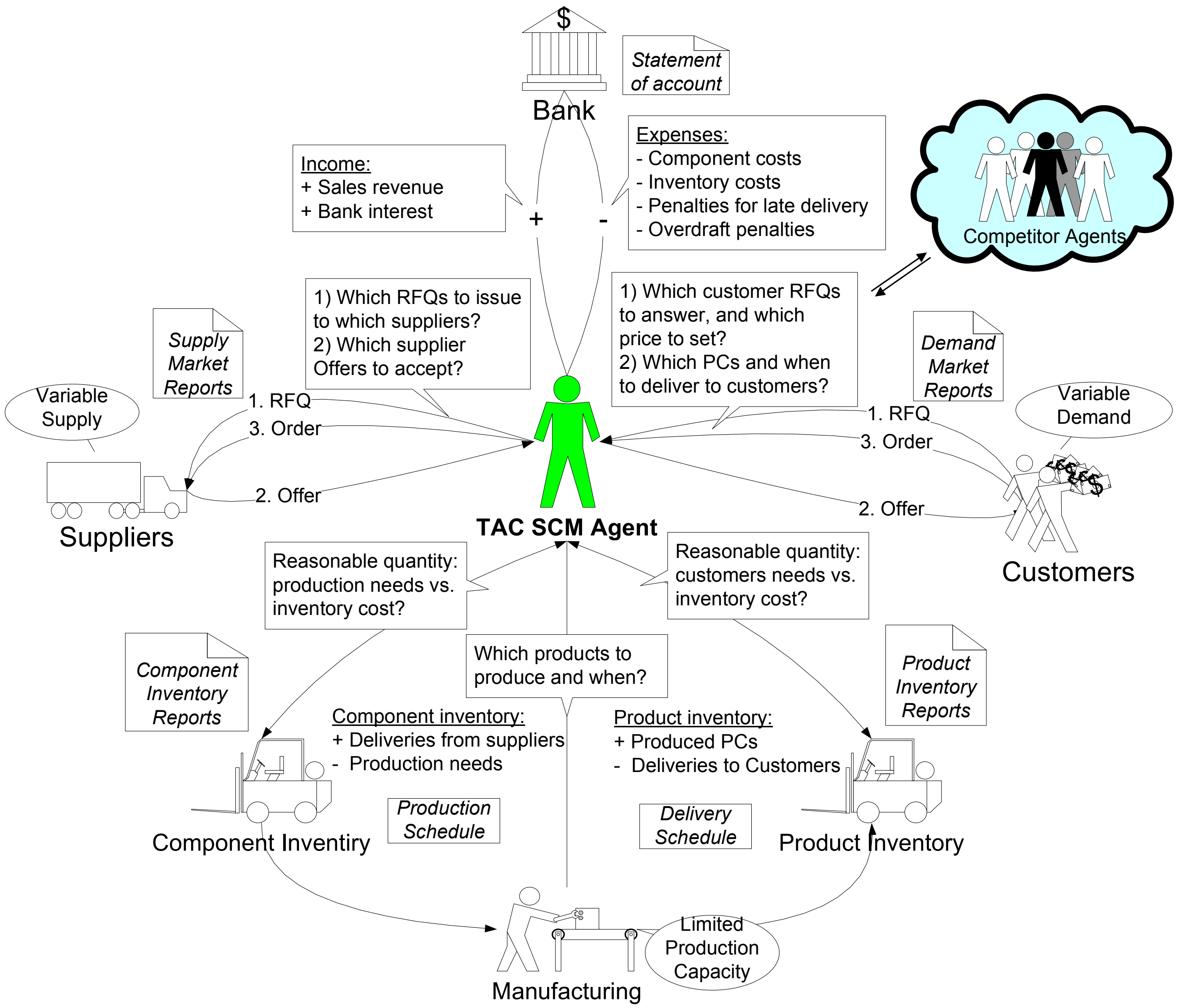

Figure 1. TAC SCM environment

The game lasts for 220 simulated days, 15 seconds of real time each. Each day an agent has to perform the following activities: (i) component procurement, (ii) product sales, (iii) production scheduling, and (iv) delivery scheduling. The aim of each participating manufacturer is to maximize their profit: the agent with the highest bank balance at the end of the game wins. The agent spends money on buying components, paying a storage cost for keeping an inventory of components and PCs, paying penalties for late deliveries of customer orders, and for bank overdrafts. The income of the agent consists of the revenue from PCs sales and interest on positive bank balance.

3. RELATED WORK

The TAC SCM community involves many research groups throughout the world. Each team investigates different issues within the SCM domain and develops various methods to tackle them. A number of works have been dedicated to the problem of finding optimal prices to offer customers in response to their requests. As this problem correlates with the objectives of the author's PhD thesis, the overview of these works is presented here.

The methods applied by different agents to solve the issue can be divided into two major categories. The first group of agents estimates the winning price for each RFQ and assumes that this price would result in an order [5, 7, 9]. The second group predicts for each possible bidding price the probability that it is going to be accepted [1, 10, 11, 12, 14].

An overview of the strategies applied to the problem of finding optimal offer prices up to 2004 is provided in [15]. The paper also presents the comparison of different learning algorithms for accomplishing the task in the context of the TAC SCM environment. Specifically, the following methods were analyzed: neural network with a single hidden layer and using back propagation, M5 regression trees, M5 regression trees boosted with additive regression, decision stumps (single-level decision trees) boosted with additive regression, J48 decision trees, J48 decision trees boosted with AdaBoost and BoosTexter [20, 21], support vector machines, naïve Bayes, and k-nearest neighbours. Their experimental results showed that M5 trees and BoosTexter give the minimum root mean squared error.

In their up-to-date versions in addition to the above mentioned methods, the TAC SCM agents also use other techniques. In particular, SouthamptonSCM [7] applies a fuzzy reasoning inference mechanism to determine offer prices according to the agent's inventory level, the market demand and the time in the game. TacTex uses additive regression with decision stumps [13]. In the earlier version of the agent, the developers used linear regression on six data points to generate a linear function which is modified then by the day factor [14]. The day factor measures the effect of the due date on offer acceptance. A similar approach is implemented in Botticelli [3] and CMieux [2]. The latter computes a linear least squares fit for the selling prices of each product over the past several game days. Additionally, the agent enforces lower and upper bounds on the predictions to ensure that the prediction remains relatively conservative. The agent maintains the probability distribution for each PC type mapping bidding prices to the likelihood of winning orders with these prices. The distributions are learned off-line using data from previously played games to build a regression tree. The developers of the agent showed that under certain assumptions this pricing problem can be reduced to the continuous knapsack problem [1]. Mertacor [12] selected the M5 data mining algorithm applied to historical data from past games in order to choose which attributes influence offer prices. It also uses two on-line modelling mechanisms in order to handle unexpected circumstances that may arise with regard to selling prices. The agent applies the k-Nearest Neighbours algorithm then to find the probability of offer acceptance for each bid placed. The probability of winning customer offers is also used in the bidding strategies implemented by MinneTAC [10] and DeepMaize [11]. RedAgent [9] uses an internal marketplace structure with competing bidders to set offer prices. The agent computes offer prices as a sum of 3 terms: a base price of the PC, an estimated discounted profit for the product (the difference between base price and order price, discounted according to the number of days left until the order expires), and a discounted penalty. PackaTAC [5] sets prices according to the market state taking into consideration the lowest and highest previous day prices and the current demand level.

According to [8] all the aforementioned methods do not take into consideration market conditions that are not directly observable. The authors propose a clustering based approach to identify the market regime and predict market changes. They use a Gaussian Mixture Model to represent the probabilities of market prices that allows the determination of the probability of receiving an order in different regimes for different prices. The authors assume the following factors which correlate with market regimes: the finished goods inventory of other agents; the ratio of offer to demand; and normalized price over time.

4. RESEARCH APPROACH

To deal with the complexity of the SCM domain, a multi-agent approach is applied to design the system. This allows to break the whole system down into separate building blocks, each concentrating on a particular part of the supply chain. By replacing one building block with another and by combining them in different ways, different versions of the system can be created in order to check how separate algorithms affect its overall performance. The system includes the following agents: Manager, Demand Agent, Supply Agent, Inventory Agent, Production Agent, and Delivery Agent. The Manager agent is responsible for the communication with the external contractors (suppliers, customers, bank, etc.), as well as managing all other agents. The Demand Agent decides which customer RFQs to answer and with what price. The remit of the Supply Agent is the procurement of low cost components on time from suppliers; the agent tracks the supplier market in order to choose the suppliers with lower prices and lower level of suspended deliveries. The Inventory Agent manages the component and PCs stocks in order to satisfy the needs of the Production and Delivery Agents while at the same time minimising holding costs. The Production Agent is responsible for scheduling current production and projecting production for the future. Finally, the Delivery Agent deals with delivering PCs to customers according to their orders and on time to prevent penalties.

To model the agents' behaviour, different techniques are used in the system, such as: constraint satisfaction, planning, logical rules, and online adjustments. The majority of the algorithms are based on simple heuristics. However, testing the system in the TAC SCM game showed that these algorithms do not perform well against stronger agents developed by other research teams. To improve the performance of the system, a predictive approach is required. According to this, a number of predictive algorithms are implemented in the Demand part of the SCM that deals with selling products to customers. The most crucial problem here is of predicting customer winning bidding prices. More specifically, a customer sends requests for quotes (RFQs) indicating which products, in what quantity and for when he wants them. The customer also indicates the reserve price – the highest price he is willing to pay for the product. Competing agents answer these customer RFQs with their offers specifying the bidding prices they are willing to offer to the customer. For each RFQ, the customer chooses the lowest price proposed by all manufacturers and places an order. So the problem here is to set optimal customer offer prices, which should be high enough to allow for profit and at the same time low enough to be accepted by customers.

So far, 3 different strategies have been developed to tackle the problem. According to the first strategy, the system predicts bidding prices for each customer RFQ which will more probably result in customer orders. The predictions are based on the current market situation and also on RFQs' details. 3 algorithms based on the Neural Network (NN) learning technique are implemented to perform the forecasting. In particular, for each algorithm a set of ensembles of 3-layered NNs for every product available on the market are constructed; each NN in the product ensemble predicts the probability that the winning bidding price will be in the price interval assigned to the ensemble. The algorithms differ in the number of inputs they consider and their methods for input normalization. The Back-propagation algorithm and sigmoid function as the activation function are used to train the NNs.

The second strategy for deciding on offer prices is to predict the lowest order prices for each product based on the time series of these prices. All TAC SCM competitors get daily market reports, where the lowest order prices proposed by all agents on the previous day for each product available on the market are specified. Using the previous values of these prices, their values for one and ten days in the future are predicted. The Neural Networks and Genetic Programming (GP) learning techniques are used to design 33 different models of predictors. Apart form the difference in the learning technique they use, the models also differ in their data transformation and normalization methods applied over inputs, and also the number of observables considered in the time series.

Finally, the third strategy implemented in the Demand Agent is to model the competitors' behaviour and to predict their bidding prices according to the models evolved. Having predicted prices of its competitors, the agent can bid just below them and thus win customer orders. Again, the NN and GP learning techniques are used and 4 different algorithms are developed to deal with the task. The algorithms differ in their approaches for selecting features to model competitors' behaviour.

To evaluate the proposed approaches and algorithms, a number of games were played in the TAC SCM simulated environment. Different combinations of participant agents were used. In some games, the competitors were different versions of own agent. For other games, highly competitive agents developed by other TAC SCM participants were run. Binary code of these agents is available from the TAC web-site repository. In order to decide on the most successful strategies to follow in each part of the supply chain, the game results were compared in terms of (a) overall scores of competing agents, (b) rates of customer offer prices proposed by them, and (c) order winning rates (the ratio between the number of offers send to the number of orders received). To evaluate different algorithms for predicting customer winning bidding prices implemented in the Demand Agent, the root mean square errors of their predictions were calculated to estimate the models' accuracies. In addition, the complexity of algorithm implementation and time of their execution were taken into consideration. The last parameter (execution time) is important as in the TAC SCM game all the decisions have to be made within 15 seconds.

5. RESULTS AND FUTURE WORK

The experiment results demonstrated that the agents that track the supplier market, plan their production in advance, and/or pick only profitable customer RFQs, perform better than those that do not support these strategies. The agents that use one of the proposed algorithms for predicting customer winning bidding prices outperform agents that do not make any predictions. The strategy of setting customer offer prices according to the algorithms which predict probabilities of the winning bidding prices to be in a particular price interval appeared to be less successful than using other predictive methods (predicting lowest order prices or competitors' prices). Although the algorithms for predicting lowest order prices and competitors' prices demonstrated different results across the games played, all of them showed high level of prediction accuracy. Both Neural Networks and Genetic Programming learning techniques appeared to be appropriate for predicting order price time series and competitors' bidding prices. At the same time, NN surpassed GP in terms of complexity of algorithm implementation and time of execution in the case of predicting competitors' prices (1 second for NN versus 90 seconds for GP). The disparity in the models' performance leads to another conclusion that different models might work better in different market conditions, which, in their turn, depend on the strategies applied by competitors. According to this, the task for future work is to develop a meta-model, which can consolidate the results obtained from individual models and find dynamically the best solution for the current market environment.

The experiments reveal that the prediction of the competitors' bidding prices themselves is not enough for making optimal decisions on offer prices: if the agent with the lowest predicted price does not bid for an RFQ, then the winning price will be the lowest among the ones set by the other agents who actually bid. Thus, in addition to the prediction of the agents' bidding prices for every RFQ, the classifiers, that will specify whether the agent will actually bid for the RFQ at such price level, have to be introduced. This will help to make decisions on which RFQs to bid for and what price to offer. Another task for future work is the problem of Feature Subset Selection. In particular, the experiments showed that the knowledge of the features that the competitors are using for making their decisions, could improve the predictive models of these competitors. The following claim has been proved empirically: if a player knows which features its competitor is using for making its bidding decisions, then, even without knowing the exact strategy of the competitor, it is possible to predict its bidding prices more accurately than in the case when these features are not known. Thus, there is a task of finding the method for predicting which parameters competitors are using.

With regard to the other agents implemented in the proposed SCM system, there is plenty of room for improvement of the performance of the Supply and Production Agents. Having the limited production capacity, the Production Agents tries to maximize its utility, i.e., the potential profit that might bring the scheduled production. At the moment, the agent schedules production for 12 days in the future using the following heuristics. For every day in the future, the agent leaves some capacity for future demand (the further production date, the more cycles are reserved), then schedules current and late orders, depending on their due date, profit and availability of components, and after this, it allocates current RFQs, again considering their due date, profit and availability of components. To schedule the production more accurately and to use the limited production capacity more efficiently, the agent needs to predict future customer demand, as well as reconsider its planning for the future dynamically, depending on the level of orders actually received from the customers. With respect to the Supply Agent, it places only short-term RFQs at the moment. On the one hand, this approach gives low holding costs. At the same time, the agent takes the risk not to get components on time or to get them at higher rates. Thus, there is the need to find the way to balance short-term and long-term requests for components.

6. CONCLUSIONS

The main objective of the author's PhD thesis is the development and implementation of an intelligent multi-agent decision support system for supply chain management (SCM). The SCM environment is very complex, highly dynamic, and with many constraints. It is unresolved issue at the moment on deciding which strategies to follow and which learning methods to use in order to perform more successfully in this domain. Within the scope of the presented work, the effort is made to contribute to finding better solutions by developing different algorithms and testing them in the TAC SCM simulated environment. In particular, a number of approaches for predicting customer offer prices that could result in customer orders are explored. To the best of author's knowledge, the proposed strategy of modelling competitors' selling behaviour is novel for the TAC community. With respect to the approaches of predicting winning price probabilities and the lowest order prices, which have been considered by other researchers, new methods to solve the problems are investigated. The results of the current research will be valuable for both academia and real industries. More specifically, the work is dedicated to applying machine learning techniques for forecasting and optimisation problems, which is an open issue within the research community. At the same time, the aim of the project is to build up an integrated solution to assist managing supply chains. Nowadays, enterprises are looking for implementing such systems to run their businesses. Moreover, various techniques for predicting bidding prices in the context of dynamic competitive environments are explored. Apart from the SCM, the solutions can be used in forecasting financial markets and participating in on-line auctions.

7. REFERENCES

[1] Benish M., Andrews, J., and Sadeh, N. 2006. Pricing for Customers with Probabilistic Valuations as a Continuous Knapsack Problem. In Proceedings of the Eighth International Conference on Electronic Commerce (Fredericton, Canada, 2006). ICEC-06, 38-46. DOI= http://doi.acm.org/10.1145/1151454.1151475

[2] Benish M., Andrews, J., Sardinha, A., and Sadeh, N. 2006. CMieux: Adaptive Strategies for Competitive Supply Chain Trading. ACM SIGecom Exchanges, 6, 1 (June, 2006). ACM Press, New York, NY, USA, 1-10. DOI= http://doi.acm.org/10.1145/1150735.1150737

[3] Benisch, M., Greenwald, A., Grypari, I., Lederman, R., Naroditskiy, V., and Tschantz, M. 2004. Botticelli: A supply chain management agent. In Proceedings of the Third International Conference on Autonomous Agents and Multi-Agent Systems (New York, NY, July 19-23, 2004). AAMAS-04, 1174-1181.

[4] Collins, J., Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., and Janson, S. 2006. The Supply Chain Management Game for the 2007 Trading Agent Competition. Technical Report CMU-ISRI-07-100. Carnegie Mellon University.

[5] Dahlgren, E. and Wurman, P. 2004. PackaTAC: A conservative trading agent. ACM SIGecom Exchanges, 4, 3. ACM Press, New York, NY, USA, 38-45. DOI= http://doi.acm.org/10.1145/1120701.1120707

[6] Dong, R., Tai, T., Yeung W., and Parkes D.C. 2004. HarTAC – the Harvard TAC SCM '03 Agent. In Proceedings of the Trading Agent Design and Analysis Workshop (New York, NY, 20 July, 2004). TADA-03, 1-8.

[7] He, M., Rogers, A., Luo, X., and Jennings, N.R. 2006. Designing a Successful Trading Agent for Supply Chain Management. In Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (Hakodate, Japan, 8-12 May, 2006). AAMAS-06, 1159-1166.

[8] Ketter, W., Collins, J., Gini, M., Gupta A., and Schrater P. 2005. Identifying and Forecasting Economic Regimes in TAC SCM. In Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (Edinburgh, Scotland, UK, 30 July – 5 August, 2005), 53-60.

[9] Keller, W., Dugay, F.-O., and Precup, D. 2004. RedAgent – winner of the TAC SCM 2003. ACM SIGecom Exchanges, 4, 3. ACM Press, New York, NY, USA, 1-8. DOI= http://doi.acm.org/10.1145/1120701.1120703

[10] Ketter, W., Kryznaya, E., Damer, S., McMillen, C., Agovic, A., Collins, J., and Gini, M. 2004. MinneTAC sales strategies for supply chain TAC. In Proceedings of the Third International Conference on Autonomous Agents and Multi-Agent Systems (New York, NY, July 19-23, 2004). AAMAS-04, 1372-1373.

[11] Kiekintveld, C., Miller, J., Jordan, P.R., and Wellman, M.P. 2006. Controlling a supply chain agent using value-based decomposition. In Proceedings of the Seventh ACM conference on Electronic commerce, Ann Arbor, Michigan, USA, 208-217. DOI= http://doi.acm.org/10.1145/1134707.1134730

[12] Kontogounis, I., Chatzidimitriou, K.C., Symeonidis, A.L., and Mitkas, P.A. 2006. A Robust Agent Design for Dynamic SCM Environments. In Proceedings of the Fourth Hellenic Joint Conference on Artificial Intelligence (Heraklion, Greece, May, 2006). SETN'06, 18-20.

[13] Pardoe, D. and Stone, P. 2007. Adapting in agent-based markets: A study from TAC SCM. In Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (Honolulu, HI, May, 2007). AAMAS-06, 677–679.

[14] Pardoe, D. and Stone, P. 2006. Predictive Planning for Supply Chain Management. In Proceedings of the Sixteenth International Conference on Automated Planning and Scheduling (Cumbria, UK, June, 2006). ICAPS-06, 21-30.

[15] Pardoe, D. and Stone, P. 2004. Bidding for Customer Orders in TAC SCM: A Learning Approach. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (New York, NY, July 19-23, 2004). AAMAS-04, 52-58.