TAC Supply Chain Management - Autonomous negotiation in a supply chain

Henrique Pires dos Santos da Fonseca
henrique.fonseca@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

May 2014

Abstract

Nowadays, supply chains are central entities in our global economy with a tendency to become bigger and more complex. Regarding the digital era, it is imperative to use Information Systems in order to support the supply chains business processes. This way the information flow will become more fluid resulting in an improved efficiency. The simulator TAC-SCM (Trading Agent Competition - Supply Chain Management) [4] was developed with the goal to explore and test Intelligent Agents with autonomous negotiation capabilities in a supply chain where computers are produced from computer components and then sold, in order to generate profit margin for the producer. This work documents an agent for an extension to TAC-SCM, the TAC-SCM-Procurement Challenge [1], which focuses it's activities on component procurement.

Keywords: TAC SCM, Supply Chain Management, Business Automation, Procurement Market

1. Introduction

In todays complex world, there is an incremental need of information systems [5] capable of supporting decisions in highly dynamic and unpredictable supply chains. A supply chain is composed of several interlinked organizations, each one focused on procurement, production and distribution. This work focus is the development of an intelligent agent for a supply chain simulator, the TAC-SCM-PC (Trading Agent Competition Supply Chain Management Procurement Challenge) [1] [8] which by design is an extension to the TAC-SCM [4]. This simulator hosts agents competitions since 2002 and is mainly a test bed for artificial intelligence techniques applied to the decision support systems of a supply chain. The TAC-SCM simplifies the scenario by breaking it down to only first level suppliers and first level clients, thus there are no distributors nor retailers. The intelligent agent is part of the manufacturer, which is the entity between distributors and retailers. The TAC-SCM-PC narrows the focus of the agent only to the procurement market, taking ownership of client and the supplier entities, and also the negotiation between the manufacturer and the clients. In the end, the implemented agent will be evaluated based in a group of key performance indicators (KPI), tested against agents developed by other teams: Warrior [6], Crocodile [6] and CMieux [2] [8].

2. Background

2.1. TAC-SCM

The TAC-SCM simulates a simplified and dynamic supply chain, represented in figure 1. In this supply chain, manufacturers produce computers with compute parts acquired from the suppliers. Then they are sold to the clients.

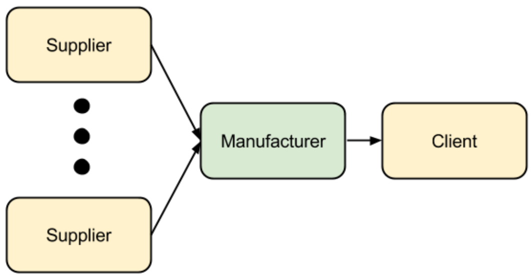

Figure 1: Supply Chain

The game consists of 6 producer agents, whom must compete in order to get the highest profit margin in the end of the game. This is accomplished by combining decisions on the suppliers market, the clients market and also the manufacturer factory schedule. In figure 2 one may observe the negotiation process both in the clients market and in the suppliers market. Starting at the suppliers market, the agent first makes a bid in the form of a Request for Quote (RFQ). The RFQ is composed by a reserve price, which is the maximum price that the agent is willing to pay for a unit, by a due date and by a quantity. In the next day the supplier may respond with an offer which can differ from the RFQ nevertheless respecting its price and quantity limits. The agent analyses the offer and may then answer with an order with the quantity and price dictated in the offer. Finally the supplier sends the order on the defined due date. The negotiation with the clients is similar except that in this case the agent assumes the role of the RFQ receiver. Client orders are generated by a *poisson* distribution and the supplier prices are determined by the level of procurement.

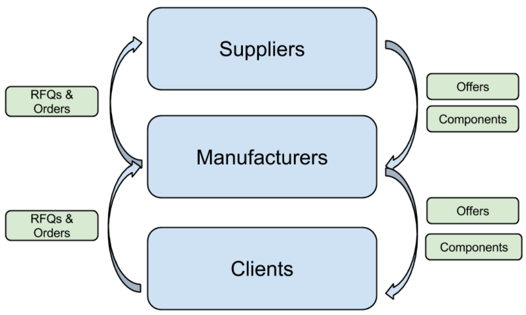

Figure 2: TAC-SCM - Supply Chain

2.2. TAC-SCM-PC

The main focus of this work is the development of a TAC-SCM-PC agent. The game differs from the base game as there are only 3 agents competing and the games duration is 100 days. In TAC-SCM-PC the negotiation between the manufacturer and the clients and also the factory scheduling are overtaken by the simulator. The clients send the exact group of orders to all the agents and then the agents factory (controlled by the simulator) will try to produce the requested computers from the most profitable to the least profitable. For each order the simulator will check if there are available components in order to produce the computer, if positive the order is shipped to the client. Also in this version there are two type of contracts, the Short-term contracts and the Long-term contracts. The short-term contracts are one-off contracts, they support only one transaction an they are negotiated exactly the same way as in the base game, using RFQs. On the other hand the Long-term contracts are negotiated before the game starts (day -1) and are eligible until the end of the game. The agent will then use a combination of both contracts in order to answer effectively to the clients demand.

2.2.1 Long-term contract negotiation

The Long-term contracts are negotiated before the beginning of the game with a full game duration. This type of contract determines a certain flexible quantity to be delivered weekly to the manufacturer agent, for a certain price per unit. For each Long-term supplier, the negotiation 3 occurs in the following way: first the supplier sends the minimum price p_{min} to the agent. The agent decides the maximum quantity Q_{max} per week and the execution price $p_{exec} \geq p_{min}$ and sends the pair to the supplier. The supplier analyses the pairs received from each supplier and orders them from the highest p_{exec} to the lowest, distributing its weekly capacity for the requested Q_{max}. Finally the supplier sends the final contract values to the agent, defining in this way p_{exec}, Q_{max} and Q_{min}. In the beginning of each week the agent decides how much quantity to order within the interval $[Q_{min}, Q_{max}]$

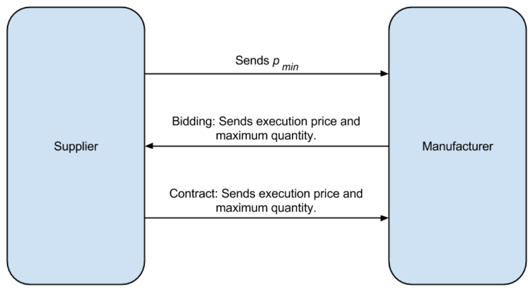

Figure 3: Long-term contract negotiation

3. Implementation - Bull Agent

The agent documented in this work is the Bull Agent, which is defined by a modular architecture 4. The forecast module observes the components market in order to store and analyse the information regarding the component prices for each day. The strategy module applies inventory quantity management while the forecast provides the agent with information about component prices. Both the previous modules feed the Short-term and Long-term negotiation modules enabling both of them to take coordinated action in the procurement market.

3.1. Strategy Module

The strategy module applies a maximum threshold limit in order to control component quantity in inventory. Also there is a cutoff mechanism that lowers the threshold limit in the ending phase of the game, enabling the agent to finish the game with lower quantities in the inventory.

3.2. Forecast Module

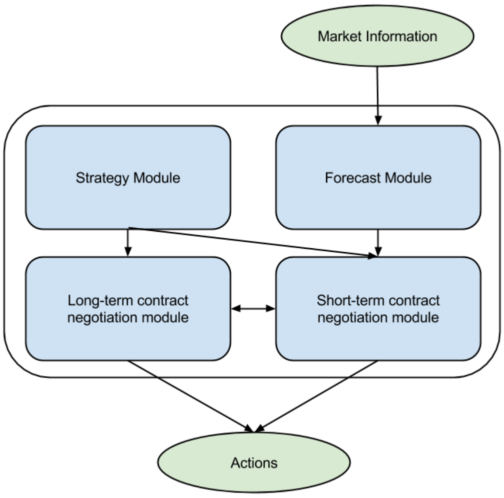

Figure 4: Bull Agent Architecture

The forecasting module provides the agent with information about component prices in a time window of 12 days. For a specific day when theres no information available, the base price of the component is used as the predicted price. During gameplay a regression k-nearest neighbors [7] algorithm is used to predict the prices for the days without information, using the nearest days with information as the source, in this case the algorithm will use $k = 3$, meaning with will perform the weighted mean with 3 neighbors, based on the distance from the day being processed. For the days where there where already orders or probes being sent, the agent uses a moving average [3] in order to obtain the most probable price based on the price trend for that specific component in that specific day.

3.3. Long-term contracts negotiation module

The long-term contracts negotiation occurs before the games start. First each Long-term supplier sends the minimum price p_{min} they are willing to accept. Then the bull agent must decide on the values of the execution price p_{exec} and also the maximum quantity Q_{max}. For the p_{exec}, the agent sets it to the minimum possible value, which is $p_{exec} = p_{min}$. This is a very conservative enabling the agent to pay the lowest price it can for the long-term components. For the Q_{max} the value chosen was 400 for non CPU type components and 200 for the CPU type components. This is because there are 4 brands of CPUs *Vs.* 2 brands of all the other component types, and it is assumed that computers demand is equal along the list of purchasable computers. The values where chosen based on a sensitivity analysis performed with different inputs. During the gameplay the agent must choose the quantity to be ordered, within the interval $[Q_{min}, Q_{max}]$. This decision is supported by a defined maximum inventory threshold applied to each component. In the moment of the decision, if a particularly component is above the designated threshold, the agent will order the Q_{min} quantity, otherwise it will order the Q_{max}.

3.4. Short-term contracts negotiation module

In contrast with the Long-term contracts, Short-term contracts are valid for only one transaction between the supplier and the manufacturer agent. The agent first sends a RFQ with the maximum price it is willing to pay, the due date and the required quantity. The agent only considers acquiring components with maximum due date $t + 11$ with t being the current day. This is because clients orders have a maximum due date of 12 days in the future. During gameplay the agent observers the clients orders, registering all the components needed to serve them in a vector segmented by day. Each day, for each component the vector is analysed and the required are ordered using the date with the lowest price possible for each day registered in the vector. This is done until 5 RFQs (maximum per supplier per day) are sent to a particular supplier.

4. Evaluation

Agents evaluation is based in a set of key performance indicators:

- Profit margin
- Factory Usage percentage
- Delivered orders percentage
- Inventory at the end of the game

Additional analysis is provided regarding the forecast mechanisms used. The Bull agent was tested against TAC-SCM-PC 2007 competitors, CMieux [2] [8] , CrocodileAgent [6] and Warrior [6]. The analysis was performed using an API that could interpret the TAC-SCM-PC simulator logs. For each competitor tested, it was retrieved 35 samples in order to approximate the sample set to a normal distribution (Central limit theorem [7]). This way it is possible to calculate 95% confidence intervals.

The forecast error (table 1) analyses the 2 forecast techniques mentioned in section 3.1, the k-nearest neighbors and the moving average. Also the base price prediction is analyzed.

The techniques were tested separately and against the real price paid for a specific day *versus* the price predicted with each mechanism. We can conclude that the forecast mechanisms provide

Base price	59,44%
Moving Average	6,91%
kNN	10,11%

Table 1: Forecast mechanisms error

the agent with valuable information since the error is inferior from the base price error, which is the standard mechanism for predicting prices.

4.1. *Versus* Warrior

First the Bull agent was analyzed against the Warrior agent.

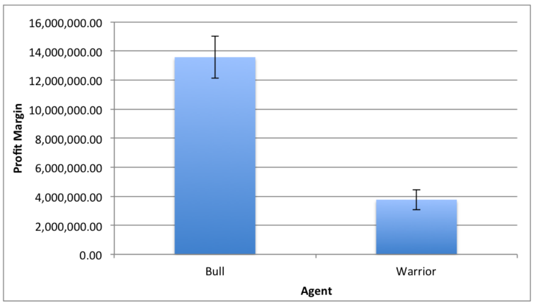

Figure 5: Profit Margin

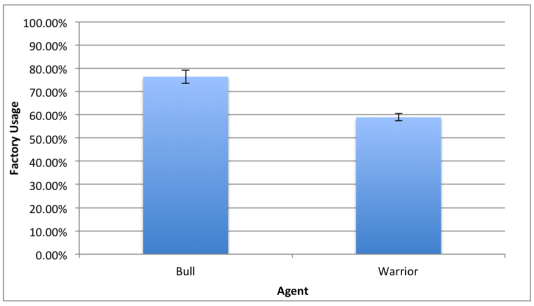

Figure 6: Factory Usage

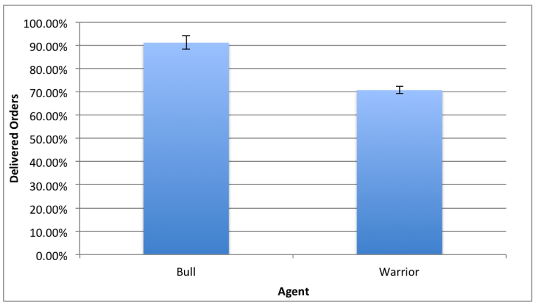

Figure 7: Delivered orders

The Bull as an higher profit margin then the warrior agent (fig. 5). The same is true for the factory

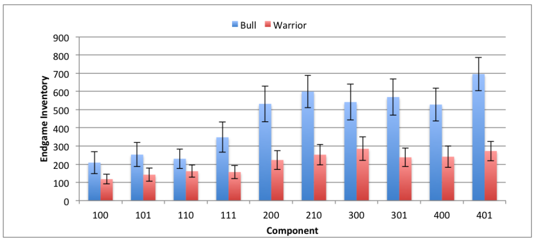

Figure 8: Endgame Inventory

usage (fig. 6) and order delivery (fig. 7) percentage. On the other hand the warrior agent has a lower inventory at the end of the game (fig. 8). Also the forecast mechanisms (section 3.1) provide Bull agent with an edge regarding the profit margin.

4.2. *Versus* Crocodile

Following, the agent was testes agains the CrocodileAgent.

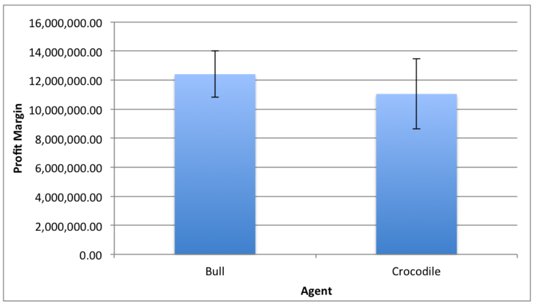

Figure 9: Profit Margin

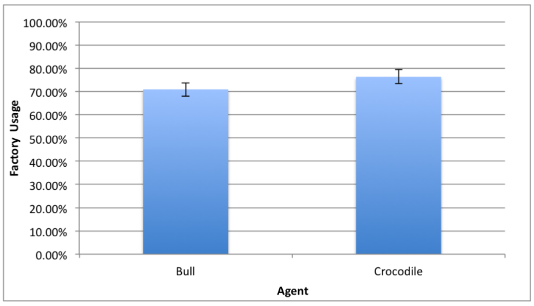

Figure 10: Factory Usage

The profit margin (fig. 9) of the Bull and Crocodile overlap, meaning it is not possible to tell the most efficient agent regarding this performance indicator. Meanwhile regarding factory utilization (fig. 10) and orders delivered (fig. 11) percentages, the Crocodile performs a little better. The difference is in the inventory left (fig. 12) in the end of the game which in the case of the Crocodile is higher than the Bull. This is due the cutoff strategy (section 3.2 implemented in the Bull agent.

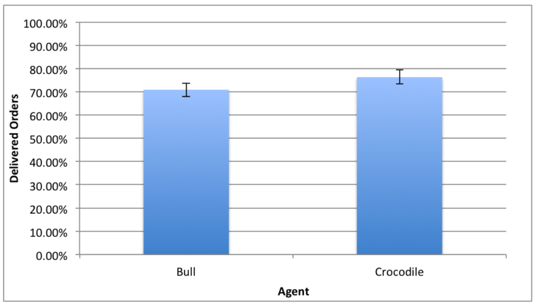

Figure 11: Delivered orders

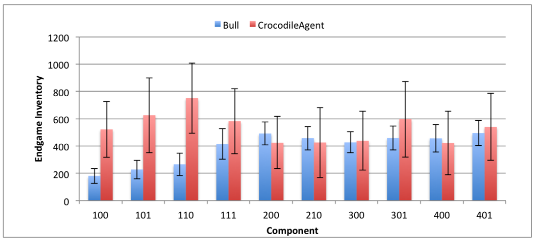

Figure 12: Endgame Inventory

4.3. *Versus* CMieux

The final agent to whom the Bull agent will be tested, is the CMieux.

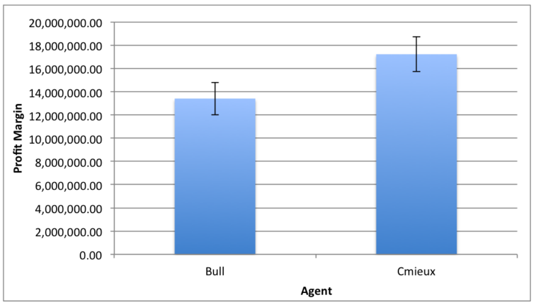

Figure 13: Profit Margin

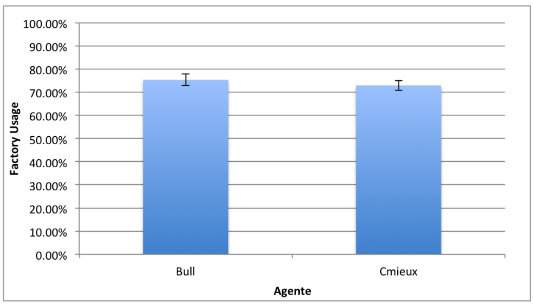

Figure 14: Factory Usage

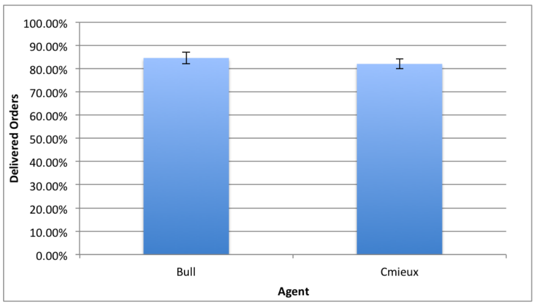

Figure 15: Delivered orders

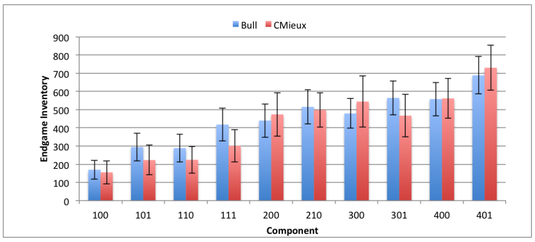

Figure 16: Endgame Inventory

Cmieux performs better than Bull regarding the profit margin (fig. 13). For the rest of the performance indicators (figures 14 15 16) , CMieux and Bull are close. In fact the CMieux doesnt get more edge to the Bull agent because of the strategies implemented in the Bull agent like the component maximum threshold and cutoff (section 3.2), the procurement strategies (sections 3.4 and 3.3) and finally the forecasting mechanisms (section 3.1).

5. Conclusions

The digital era turned information into the base of economy, becoming the most valuable commodity. Information systems have a central role supporting businesses all around the world. In the particular case of Supply chains, these systems can provide the manager with market trends and prince predictions in order to support decisions. The TAC-SCM competition [4] serves as a test bed for artificial intelligence techniques applied to supply chain management. In this work it was developed an agent for the TAC-SCM Procurement challenge [1]. The agent was tested against several 2007 TAC-SCM-PC competition participants and performed positively.

6. Future Work

As future work it would be interesting to extend the strategy module (section 3.2) in order to include a minimum threshold value. This way the agent can guarantee a minimum quantity for each component. Regarding the forecast module (section 3.1), it can be extended to include client demand forecast. This allows the agent to react more promptly to market fluctuations. Additionally it would be interesting to train the agent with data from past games in order to improve it's forecasting capabilities. As for the Long-term contracts module, further analysis is required regarding the initial negotiations optimal values for each component. The Short-term contracts module should become more selective about the prices offered by the suppliers, so it can improve even more the profit margin.

References

[1] J. A. N. S. Alberto Sardinha, Michael Benisch. The 2007 Supply Chain Trading Agent Competition - Procurement Challenge. *Supply Chain Management*, (June 2007), 2007.

[2] M. Benisch, A. Sardinha, J. Andrews, R. Ravichandran, and N. Sadeh. CMieux: Adaptive strategies for competitive supply chain trading. *Electronic Commerce Research and Applications*, 8(2):78–90, Mar. 2009.

[3] S. Chopra and P. Meindl. MSupply Chain Management - Strategy, Planning, and Operations. chapter 7. Second ed edition, 2004.

[4] J. Collins, R. Arunachalam, N. Sadeh, J. Eriksson, N. Finne, and S. Janson. The Supply Chain Management Game for the 2007 Trading Agent Competition. *Supply Chain Management*, (December 2006), 2007.

[5] K. C. Laudon and J. P. Laudon. Management Information Systems. chapter 9. Twelfth ed edition, 2012.

[6] T. Milivic, V. Podobnik, A. Petric, and G. Jezic. The CrocodileAgent: A Software Agent for SCM Procurement Gaming.

[7] T. M. Mitchell. *Machine Learning*. International edition, 1997.

[8] A. Sardinha, M. Benisch, N. Sadeh, R. Ravichandran, V. Podobnik, and M. Stan. The 2007 procurement challenge: A competition to evaluate mixed procurement strategies. *Electronic Commerce Research and Applications*, 2009.