Foreseer – A Constraint Based Agent for TAC SCM

David A. Burke, Kenneth N. Brown, Onur Koyuncu, Brahim Hnich and Armagan Tarim

Cork Constraint Computation Centre (4C), Dept. of Computer Science, UCC, Cork, Ireland

d.burke@4c.ucc.ie

The annual international Trading Agent Competition – Supply Chain Management competition is based around the manufacture and supply of PCs. There are multiple agents in the game, scheduling production, competing for orders from customers and components from suppliers.

Foreseer combines constraint-based optimisation, reasoning with probabilities, and learning of market conditions in an attempt to determine what customer requests to bid on and what prices to bid.

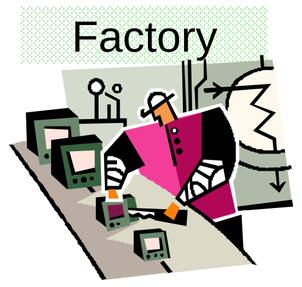
Agent

Customers send Request For Quotes that include product type, quantity, reserve price, due date and penalty.

Customers make orders based on quotes issued the previous day.

Suppliers send quotes in response to requests sent by Agent on previous day.

Current inventory will include all components received up to today, less components used in production.



Production and delivery schedule for next day.

Update Customer Model

- In order to reason about what offers to make to customers our agent maintains prices that correspond to different probabilities of success in winning contracts using an online learning approach.
- By keeping track of the ratio of offers accepted to those made, the prices can be updated iteratively to move closer to the target probability.
- This range of price/probability pairs is then used as input to a constraint model.

Learning Algorithm

$p_{target} = 0.5$	// Target probability
$w = w_{default}$	// Default weight
o	// Number of offers made
a	// Number of offers accepted
$p_{actual} = o/a$	// Current actual probability
$? = p_{actual} - p_{target}$	// Diff between actual & target
$w = w + ? * a$	// Update the weight by a
	// learning step size a
$price = base * w$	// Calculate new price

Update customer model.

Scheduling Production and Deliveries

- Produce confirmed orders.
- Produce in order of ascending due date (tie-break on largest profit) until all available production capacity and supplies are used up.
- Orders not produced in the current schedule are provisionally scheduled for being produced the day before they are due to be delivered.
- If there is spare capacity and supplies, build, in random blocks of 5, up to 100 of each product for storage in inventory.
- Orders are delivered when ready.

Choose and send offers to customers.

Update suppler model.

Decide Customer Offers

- OPL Studio is used to model the problem of choosing offers to make to customers.
- Each day the current days RFQs are input to the model, along with:
  - Fixed game parameters such as Bill of Materials and processing cycles used to produce products.
  - Current and expected inventory levels of each product and component.
  - Non-committed factory capacity over the planning horizon
  - Range of price/probability pairs from which to choose selling price.
  - Component costs based on average paid by agent over last 10 days.

Objective Function:

- Maximise the profit, where the profit is calculated by subtracting from the selling price the cost of components together with late delivery penalties.

Decision Variable:

- For each request, choose whether or not to bid, and select a price from the range of input prices, that are being learnt and updated throughout the game.

Constraints

- Ensure that we will be able to schedule any new orders we receive with existing orders such that the factory capacity for each day in the current horizon is not exceeded.
- We know the current amount of components available and by ordering components in advance we also know how much of each component will be arriving at each day. This allows us to add a constraint for availability of supplies.
- No offer made should exceed the reserve price of the request.

Update Supplier Model

- Monitor demand and order trends to determine future expected component needs.
- Examine latest market reports to determine reserve price to use for component requests.
- Calculate average component costs over the past 10 days to determine a minimum selling price.

Decide what requests to send to suppliers.

Supply Procurement

- Order components in advance.
- Try to maintain buffer level based on expected customer orders.
- Order majority of components with long due dates (t+10,20,30).
- Use orders with shorter due dates (t+5,7) to top up inventory.
- Reserve price set to 5% above the last reported market average.

Make orders to suppliers. Requests are only made for components that are needed and a reserve price is specified.
Our Agent makes orders on all offers received from the suppliers.

Implementation

- Foreseer was implemented using Java and OPL Studio 3.7.1.
- It runs on Windows XP using a 2 GHz Processor and 1 GB RAM.

Acknowledgements

- This work is funded by Science Foundation Ireland under Grant No. 03/CE3/I405 as part of the Centre for Telecommunications Value-Chain-Driven Research (CTVR)