Presentation on Value-Based Decomposition strategies in TAC SCM

Given by Paul Ahern at 14:00-14:20 on 16 November 2006 at 4C in Cork.

Ladies and Gentlemen, welcome to this, the first presentation from members of the 2006-2007 CS5204 course at UCC.

My name is Paul Ahern and I would like to dedicate this talk to one of my favourite organizations the Plain English Campaign. There will be time for questions at the end. However, if at any point in the talk I say something which you do not understand then please stop me and I shall try to make my meaning more clear.

Unaccustomed as I am to public speaking, I may not project my voice enough to be heard by all. If I start to mumble, or you cannot hear what I am saying for some other reason, then please let me know immediately.

Today I am here to talk about the use of Value-Based Decomposition in the design of the University of Michigan entry in the Trading Agent Competition Supply Chain Management scenario for 2005. Henceforth referred to as TAC SCM. Their agent is called Deep Maize.

(Aside: Maize is another name for corn, a cereal is native to the Americas. I kind of knew that, but I still had to look it up to be sure.)

'Value-Based Decomposition' is a terrible piece of jargon, but sadly unavoidable since it is in the title of the paper I've been asked to speak about. Basically, it is a way of breaking a complex problem down into more soluble pieces. The solution of which should provide the answer to the original problem.

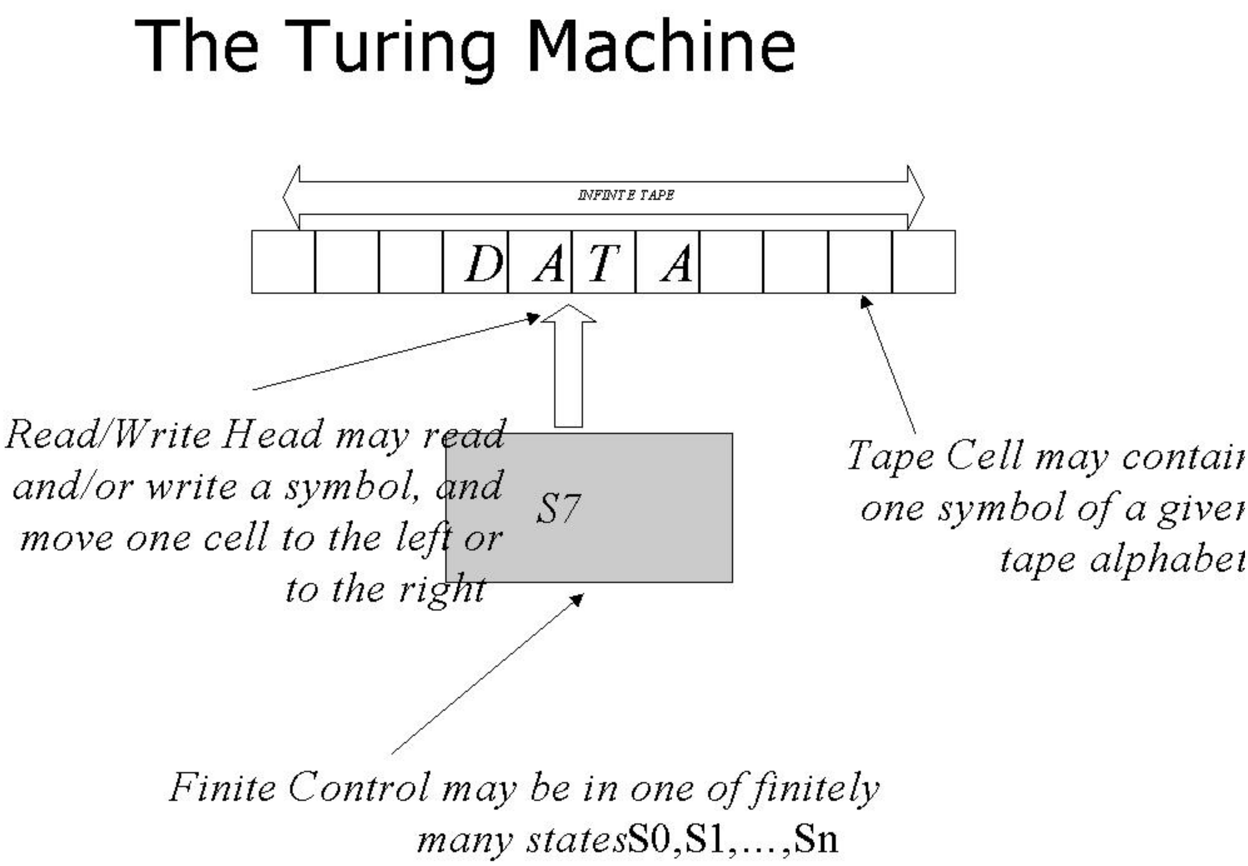
The term 'Value-Based Decomposition' is technical and precise, but its use in the title of the paper seems to owe more to a need to impress rather than to inform. It may also intimidate readers from outside the field and so could be considered a form of terminological terrorism.

The TAC SCM Scenario is based around a supply chain for the manufacture of PCs.

[Suppliers] <=> [Agents] <=> [Customers]

The TAC SCM game is a simulation of the supply chain by a set of computer programs.

Every executing computer program passes through a series of states. The current state can be represented by the values in all of the program variables, the program code and a pointer to the next statement in the code to be executed.



A Turing Machine – Infinite Tape + Finite set of states

The Turing Machine is a simple model of such a system, which nevertheless captures its essence.

State is an overloaded term in the English language. For example: The state of the world is different from the state of Illinois. Or the state of the Irish state.

The precise state of the world in the sense of the position and forces acting upon every single thing, animal, person, particle, atom and sub-atomic particle cannot be known. However, if it were known then it might be possible to predict the next state (and so on).

This is impossible. Donald H Rumsfeld a few years ago didn't even know the state of Osama Bin Laden: “We know for certain knowledge that he is in Afghanistan. Or in some other country. Or Dead.”

Over the course of a TAC SCM game the model, the set of programs making up the current simulation, moves though a series of states. These include the various and varying prices of Components and PCs.

The overall state of the simulation is not known by the agents. So it is called the Hidden State. If this Hidden State (and the future states) were known then the an agent could act in an idealised manner to maximise its profits.

Of course the agent is also part of the simulation and its actions contribute to the future states. So changing its actions would change the future states also, potentially leading to new optimum actions and so on ad infinitum.

The design of Deep Maize seeks to predict the hidden and future states of the game and then at least take approximately ideal actions.

Deep Maize – Controlling a Supply Chain Agent Using Value-Based Decomposition

Deep Maize is the University of Michigan entry in the Trading Agent Competition Supply Chain Management scenario. The central idea governing its design is to estimate marginal values for each finished product and component input as accurately as possible, given predictions about market conditions and constraints on production.

In other words the agent forecasts conditions in the customer and supplier markets and calculates the amount the agent expects to get for each additional PC sold and the price it expects to pay for each component bought.

These values provide a way to decompose the agent's decisions into manageable sub-problems, retaining many of the advantages of global optimization, while being more computationally manageable. Detailed decisions such as which suppliers to buy from and at what price are made by separate modules and algorithms which are informed by the high-level forecasts.

The high-level tasks are:

Raw Demand Prediction

- Given observations of market demand, predict the future demand distribution.
    - Daily observed demand is Poisson[1] distributed about some mean parameter. That mean parameter is a deterministic function of the prior day’s mean and trend parameters.

1 The probability of a number of events occurring in a fixed period of time if these events occur with a known average rate, and are independent of the time since the last event.

Effective Demand Curves

- Given a feature set, predict the demand curves for each future day.
    - Predict effective demand curve relative using k-nearest neighbours and affine updating[2].
      Using:
        - Reserve Price [Current day only]
        - Simulation Day
        - Minimum of the lowest selling price over past 5 days
        - Average of the lowest selling price over past 5 days
        - Average of the midpoint between highest and lowest price over the last 5 days
        - Average of the highest selling price over the past 5 days
        - Maximum of the highest selling price over the past 5 days
        - Price spread on previous day
        - Number of days without a sale
        - Average number of PCs actually requested by customers over the previous 5 days
        - Expectation of the mean demand variable for market segment
        - Expectation of the trend variable for the market segment
        - Estimated component supply (x4)
        - Estimated supplier capacity (x4)

Supplier Market Predictions

- Predict supplier capacity
    - Estimate current supplier capacity using 20 day market reports
- Predict supplier prices
    - Weighted average of prior price curves

The agent starts by using any new information it receives to update its beliefs about the hidden state of the supply chain. It then makes comprehensive predictions about the conditions it expects to face in the markets for components and finished PCs, accounting for the behaviour of the other agents.

The next stage makes high-level decisions about long-term production scheduling. The projected production schedule approximately optimizes the agent's expected profit margin with respect to the market predictions, subject to constraints on factory capacity and component arrival.

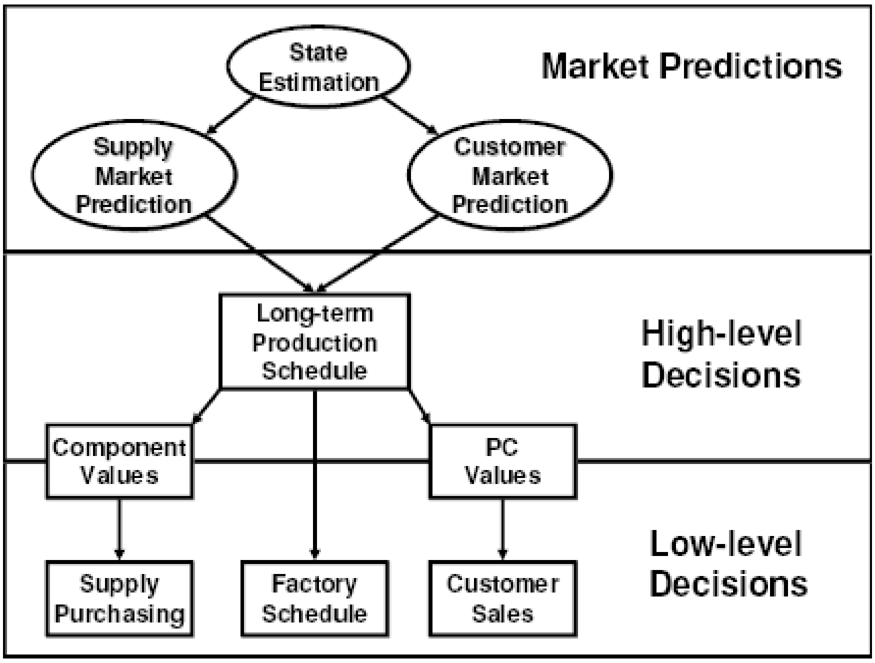
The final stage is a set of optimizations that make specific low-level decisions about factory scheduling, customer sales and component purchases.

---

2 $x \rightarrow Ax + b$

The key question is how to coordinate the low-level decisions with the overall production plan. Deep Maize accomplishes this by deriving values for PCs and components from the production plan and incorporating them into the objective functions for the low-level decisions.

Instead of optimizing the overall profit margin, the sales decision optimizes the margin between expected revenue and the value of the PCs sold. Similarly, the purchasing decision optimizes the margin between the value of the components purchased and the total cost.



Organization of Deep Maize's decision process on each TAC SCM day.

Reference summarized:

Controlling a supply chain agent using value-based decomposition., Christopher Kiekintveld, Jason Miller, Patrick Jordan, and Michael P. Wellman, Seventh ACM Conference on Electronic Commerce, Ann Arbor, MI, 2006.

The Deep Maize team have claimed a superior performance for their agent. It did finish fourth overall in the 2005 competition. Arguably it would have come second but for network connectivity problems.

I am not entirely convinced by their data and arguments. Some of their measures of 'success' seem a bit forced to me.

I would also argue that there is a flaw in the current TAC SCM scenario in that agents (or their parameters) can be changed between the rounds of the competition. It seems to me that a better test of agent intelligence would be to allow it to adapt autonomously to the different conditions in the earlier and later rounds.

Allowing human designers to intervene is more a test of their intelligence than that of the AI.

And now for the more interactive part of the presentation: Are there any questions?

Paul Ahern 15/11/2006