

Evolution of a supply chain management game for the Trading Agent Competition

Joakim Eriksson, Niclas Finne and Sverker Janson *

Swedish Institute of Computer Science, Box 1263, SE-164 29 Kista, Sweden
E-mail: {joakime,nfi,sverker}@sics.se

Abstract. TAC SCM is a supply chain management game for the Trading Agent Competition (TAC). The purpose of TAC is to spur high quality research into realistic trading agent problems. We discuss TAC and TAC SCM: game and competition design, scientific impact, and lessons learnt.

With a conscious effort to these ends, a competition may help focus research on well-chosen problems in high impact domains, facilitate comparison of results, and create a community eager to analyze and build on the results of peers and in working jointly towards the perfection of solutions.

Potential pitfalls include emphasizing winning to the exclusion of testing scientific hypotheses, and carrying the overhead of designing, implementing, and operating a competition.

Keywords: AI competition, software agents, electronic markets, trading agents, supply chain management

1. Introduction

Write software to automatically control a manufacturing company and maximize profits while competing with adversarial manufacturers for limited and variable product demand and component supply. This is the challenge of TAC SCM, a supply chain management game for the Trading Agent Competition. Supporting the competition is a distributed simulation system developed by the authors: a server running the scenario and agent APIs for the participants.

In a game, six software agents connect to the TAC SCM server, and play six competing manufacturers for 220 simulated days. A competition is divided into hundreds of games, each lasting one hour. Game parameters are varied between and within games, and statistics are collected to calculate metrics and to benchmark the agents. The winner is the agent that makes the most profit.

TAC SCM was created in the context of the Trading Agent Competition (TAC), an international scientific forum promoting high quality research into challenging e-business problems. TAC was initiated by Wellman and associates at the University of Michigan in 2000 [33]. Since 2002, Swedish Institute of Computer Science (SICS) operates the annual international TAC

tournament and provides the supporting software infrastructure: game servers, agent development APIs, and visualization tools [27].

TAC SCM tournaments have been held in conjunction to IJCAI-03 [11], ICEC-03 [10], and AAMAS-04 [1], the latter attracting 30 participant research groups. A wide range of approaches have been explored by participants, a selection of which will be discussed here. Courses using TAC SCM as an experimental platform have been held at several universities, e.g., Brown University, Carnegie Mellon University, Melbourne University, University of Michigan, North Carolina State University, and University of Texas at Austin.

TAC SCM thus forms an established platform for research and training on real-time decision making in electronic markets, in use by leading research and educational institutions all over the world. The purpose of this paper is to discuss the design of the competition and lessons learnt, and in passing introduce TAC SCM to a wider AI audience.

2. The Trading Agent Competition

TAC was inspired by RoboCup and other successful AI competitions [15,32,33]. The goal was to create a competition format for software agents acting on, and

*Corresponding author: Sverker Janson, E-mail: sverker@sics.se.

interacting through, markets, competing to best satisfy their objectives by buying and selling (trading) goods. Several market game scenarios were considered before settling on the travel shopping game now known as TAC Classic.

2.1. TAC Classic: travel shopping

The goal of TAC Classic agents is to assemble travel packages for clients at the lowest possible cost. Performance is measured as the client utility of travel packages minus the cost of goods acquired. A full description is available on the TAC Web site [27]. We offer here a brief overview.

Travel packages consist of in- and out-flights, hotel rooms for the stay-over nights, and optional tickets for events. The trip takes place in a five-day work week. Clients will have preferences for specific arrival and departure days, one of the two possible hotels, and certain events. Preferences are parameters to the client utility function.

In a game instance, eight agents compete during 9 minutes for flights, room, and tickets on 28 simultaneous markets of three types (Fig. 1). Each agent has eight clients with random preferences, making a total of 64 clients.

Flights are of unlimited availability but their price will tend to increase over time, giving a cost to postponing buying decisions.

Hotel rooms are sold on auctions, one per hotel and night. Availability of rooms is limited; each of the two hotels has 16 rooms per nights. Mid-week hotel auctions will usually be highly competitive. Hotel auctions will close randomly, one per minute.

Event tickets are traded in double auctions. Each agent receives a random initial endowment of tickets that can be traded for profit and to satisfy client preferences.

2.2. Elements of a competition

The TAC Classic competition is played in three rounds over 8–10 weeks in June to August: qualifying, seeding, and finals. Games are scheduled around the clock during weekdays, matching different combinations of participants, with the constraint that all play the same number of games. The exact format depends on the number of participants but follows the same basic pattern. We present the format of TAC Classic 2002.

The qualification round ranks agents by average performance (TAC-02: 26 agents, 120 games/agent),

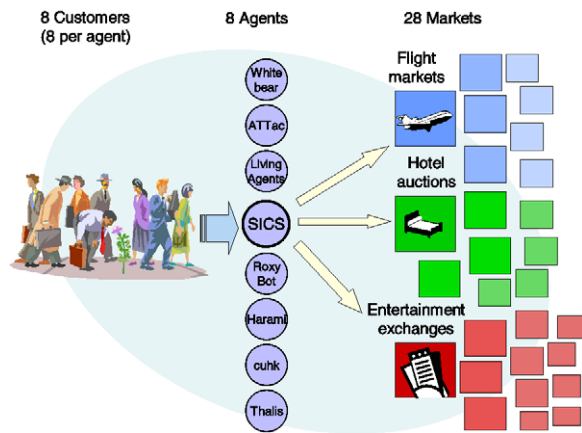


Fig. 1. Elements of the TAC Classic game scenario.

moving the (>16) acceptably performing agents on to seeding. The seeding rounds rank the qualified agents by average performance (TAC-02: 19 agents, 440 games/agent), of which sixteen move on to semi-finals. The semi-finals are two separate groups of eight agents, the four highest and lowest scoring in one group and the middle eight in one (TAC-02: 14 games/agent), of which the best four in each group move on to the finals.

The finals are played as repeated games with the eight finalist agents (TAC-02: 32 games/agent). The competition winner is the agent with the highest average score in this final round.

Note that fewer games are played in the two final rounds, decreasing the statistical significance of competition results. This is a conscious choice to allow the final rounds to be played during a conference event.

2.3. Lessons learnt from TAC Classic

A basic lesson from TAC Classic is that a TAC game is a trade-off to be established between several general concerns, including: science, realism, simplicity, and feasibility.

Science. TAC focuses on research into trading agents; a game should help participants focus on open research issues. In particular, TAC emphasizes complex market situations, with a multiplicity of interacting issues, requiring a combination of solution methods.

Realism. TAC aims not only to pose scientific problems to the AI community but to help bridge the gap between AI research and the real world. Games should encourage the development of more complete solutions, which are close to being useful in real applications.

Simplicity. TAC games should be simple enough to present a low barrier of entry for participants and be quickly understood by outside parties.

Feasibility. TAC is subject to resource constraints on arrangers and participants. The time spent on designing games, developing software, running competitions, etc, should be motivated by increased scientific progress (quantity, quality, and relevance).

Many specific issues discussed for TAC Classic apply to trading agent games, competitions, and operations in general [32], e.g.:

Roles. Some games, e.g., the MIT Beer Game, require participants to play different roles. For simplicity, the games considered for TAC are symmetric; all participants play the same role.

Scalability. The competition structure should ideally allow participation to grow with increasing interest in the game. In practice, only up to some fairly small number of participants is feasible using current competition formats (perhaps up to 50), but this limit has not yet been reached in TAC.

APIs and Agentware. TAC games require game and competition software platforms with standard APIs for participants. For maximum freedom to participants, the platforms are servers to which agents connect over the Internet. Agents may employ any hardware and software technologies as long as they adhere to the server protocol.

Stone draws on deep personal involvement in TAC Classic and RoboCup in an analysis of pitfalls and benefits of both games [26]. Both are multiagent games, with agents acting in dynamic real-time environments, with hidden state and unpredictable adversaries. Pitfalls listed include: obsession with winning, domain-dependent solutions, barrier to entry, and invalid evaluation conclusions. Benefits include: research inspiration, common platform for exchanging ideas, continually improving solutions, and generate realistic economies. Most of these carry over to any TAC-style game.

TAC Classic has successfully inspired scientific progress in the field, and several papers have surveyed, analyzed, and compared published approaches [6,9,29,31].

2.4. The next step

Following three years of running TAC Classic, 2000-2002, the game was perceived as well understood

and the major research ideas explored. The TAC community was ready for a new challenge. New scenarios discussed included FCC bandwidth auctions and stock markets, both of which strong candidates for new TAC games (cf. [12,21]).

Working against the FCC bandwidth auction scenario was the circumstance that European bandwidth auctions had suffered badly from winner's curse and caused political turmoil. Stock markets are noncontroversial, but concerns were raised that TAC-type situations, with clear strategic interactions between participants and interdependencies between markets, are not the main case in stock markets and could be perceived as artificial if designed into a TAC game.

Sadeh and Arunachalam of CMU then proposed the idea of a Supply Chain Management game for TAC, TAC SCM, which was immediately well received. New aspects and issues compared to TAC Classic include:

- a novel and important real-world domain, with a large body of independent industrial and academic method development, adding new connections between AI and operations research,
- a larger number of decision stages and explicit dynamic variation over time, changing the problem character from decision to control,
- a more complex state and action space, with three separate subproblems that need to be strongly coordinated, requiring new approaches to learning and decision making.

A design process began in collaboration with the authors with the goal to introduce the new TAC SCM game in TAC 2003.

3. TAC Supply Chain Management

We first briefly introduce some supply chain terminology and issues, then state the game idea and discuss game design issues. The full specification is available elsewhere [2].

3.1. Supply Chain Management

Supply chain management is a field of great practical importance and widespread academic interest. See, e.g., the book by Simchi-Levi et al. [24] for an overview of the main issues, and the books edited by Stadtler and Kilger [25] and Geunes et al. [7] for more technical material.

A *supply chain* is a network of production and distribution facilities (nodes), typically involving multiple organizations, that performs the function of transforming input resources (supplies) into finished products and services delivered to consumers.

Supply chain management (SCM) deals with planning and operating issues of a supply chain. This can be loosely divided into *strategic* issues such as optimizing the network (e.g., choosing sites for facilities), optimizing products for distribution (e.g., space-efficient packaging and delayed localization), and collaboration with network partners (e.g., information sharing and risk sharing); and *tactical and operational* issues such as demand forecasting, order promising, materials sourcing, production and distribution planning and scheduling, inventory control, etc.

Relations to suppliers and customers (or distributors) are often referred to as *vertical*, cf. vertical integration. Analogously, relations to nodes on the same level, competitors and complementary suppliers, are referred to as *horizontal*, cf. horizontal integration.

The famous *bullwhip effect* [17] is a problem of vertical coordination, where demand signals are magnified upstream a supply chain in the absence of information sharing. This is nicely captured by the MIT Beer Game [18], a management game where participants play the roles of supplier, manufacturer, distributor, and customer in a simple four-tier supply chain.

3.2. The game idea

Imagine being in charge of a PC manufacturing company (Fig. 2). PCs are built from components: CPUs, motherboards, hard disks, and memory. Other components are assumed non-critical and are not part of your decision making. The critical components come in varieties, which when combined form a range of possible PC configurations.

At your disposal you have a factory with a fixed production capacity. You have inventories for components and finished products, for practical purposes unlimited, but items carry a storage cost. You have a bank account with credit, on which you pay or receive interest on the balance.

For your component supplies, you have a range of available suppliers. You may send them RFQs (Requests For Quotes), detailing volumes and delivery dates, and receive in return offers, valid for a fixed time. Offers may specify smaller volumes and later delivery dates, depending on the available capacity of the

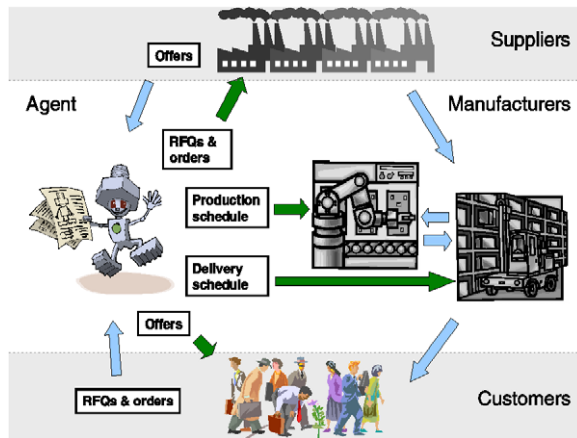


Fig. 2. Elements of the TAC SCM game scenario.

suppliers. Eventual deliveries may be delayed. Prices offered will reflect overall demand.

Your customers will send you RFQs for their PC purchases, detailing the specific configuration, the number of PCs, the due date, the maximum price acceptable, and a daily penalty for delays beyond the due date. You respond with matching offers.

Life would be easy if you didn't also have competitors, other PC manufacturers competing with you for the limited and varying capacity of the suppliers and demand from customers. Your customers will request quotes from all manufacturers and will strictly prefer offers with the lowest prices.

Your task is to devise a strategy for acquiring supplies, scheduling production, and selling the finished products. A good strategy will make high average profit in the expected range of situations and for a wide range of strategies of adversaries.

3.3. Designing the TAC SCM game

Further details are needed to make a playable game. We will discuss a choice of basic design issues and potential pitfalls of the TAC SCM design process.

Start-Up and Ending. The role played by start-up and ending in the game should be a function of the real-world domain modeled. Feasibility constraints on games also influence the choice.

One view is that we wish to measure "steady state" performance of strategies (a common objective of simulation experiments [4]), and reduce start-up and ending effects to an absolute minimum. Another view is that starting and ending production lines is a fact of life, and that performance should reflect product life cycles.

In the first case, there should be an initial phase of “warming up” to reach a steady state, followed by a period of performance measurement. Neither start-up nor ending should need to be reflected in the decision making of agents. This choice requires scoring (see below) that fairly, independently of agent strategies, reflects performance in a random interval, either integrating over states and actions all days or measuring a difference in the value of the initial and final states. Any scoring bias will make agents designers interested in guessing and planning for the start and end of a game.

It is possible to make the probability of ending the same for all days, but at the price of variation in game lengths, causing problems for competition scheduling and raising issues of comparing performance between games. In practice, an upper limit is needed and once known, it can just as well be a fixed game length, which agents may plan for.

It is also hard to see how to avoid the need for start-up strategies. We may attempt to define initial conditions that are close to the typical “steady state” with respect to the parameters of the game instance, but different agents will aim for different states depending on their strategies, and to get to them quickly will necessarily be an advantage. Agents compete for orders and supplies from the start of a game, and early decisions will affect future performance.

Having a fixed game length with a known, simple, starting condition appears to be feasible, and is the design choice made for TAC SCM. This can be explained as modeling a product life cycle.

An unrealistic aspect is that a number of competitors all start and end on the same days. In particular, this singles out the first day as a strategic singularity, for which special “Day Zero” tactics in the supply markets have an unproportional outcome on the game, and hence need to be used by all participants [30]. Arguably, fierce competition for supplies between competitors in a new market is to be expected, but the tactics available can be artifacts of game design.

We should on one hand seek to capture the essence of real world challenges, as they occur, rather than design the best fit to favorite research issues, solution methods, and general aesthetics. On the other hand, the resulting game has to be playable and support our scientific objectives.

Scoring. The score should appropriately reflect the goal of supply chain control. SCM practitioners has developed detailed performance indicators for supply chains known as SCOR metrics [23], with a very large number of metrics for all aspects of supply chains. Not

all are applicable to the proposed scenario, and using a large number would only be confusing to non SCM experts. Useful and easy to understand metrics include: cash-to-cash cycle time, delivery performance, inventory days of supply, and economic profit.

Profit is a simple and intuitive metric, and a default performance indicator for economic activity. Other metrics serve as heuristics, which are believed to correlate with profit, but are easier to use as goals for the optimization of business process. The profit made is the difference in value between the first and last day of a game. The assets are component and finished product stocks and bank account balance, initially empty/zero. The value to assign depends on the notion of ending. With an “unknown” ending, a fair value has to be assigned to that particular configuration with respect to the game state. If the state of customer and supply orders is considered, the problem becomes more complex. With a fixed, planned for, ending, the valuation is a matter of scrap value. For electronics in phased out production lines, this is close to zero.

The simplest choice is to only measure the bank account balance at the end of a game, and this is the design choice made for TAC SCM.

Structure and Timing. A first consideration is whether the game should be event-driven or time-driven. In computer gaming terminology, this corresponds to “real time” and “turn based” (with a deadline). It is highly desirable in TAC that a difference in connection speed (bandwidth and ping time) will play little or no role for agent performance.¹ The only possible design choice for TAC SCM is a turn-based game.

Thus, the game is naturally divided into a number of decision stages, called *days*, in which agents are given time to receive messages, make decisions, and reply with messages. The order in which agents act makes no difference for the game, only during which day.

The length of a day in real time is a compromise between three main desires:

- To run games with a large number of decision periods, giving time for natural patters of variation to be played out, and for start-up and ending effects not to dominate.
- To run a large number of games to increase the statistical significance of competition results. This is limited by the time available for a competition.

¹The first TAC Classic design, in 2000, exhibited a flaw giving an advantage to bidding as close to the end as possible, hence to faster connections. This was corrected the following year.

- To make each day long enough that differences and variation in message transmission speeds are a small fraction of a day. Agents have to act before the deadline and slow or variable connections will leave less time for decisions, but any amount of computing hardware may be used to compensate, if necessary. In practice, the choice of methods dominates the need for hardware resources.

The compromise chosen for TAC SCM is to make each day 15 seconds long and have 220 days per game, allowing games to be run under one hour, which can be conveniently scheduled on whole hours.

Competition Format and Rules. The competition format of TAC Classic is well tested and feasible for arrangers and participants. Reusing it is straightforward. Since the new games are longer, either fewer games can be played or more computing hardware is needed. To run games in parallel, arrangers have to run more servers and participants more agents. Experience shows that running two parallel games is feasible.

The number of participants in a game is a compromise between the scientific objective to explore strategic interactions between trading agents and the feasibility constraints to let each agent play a reasonable number of games in the time available. Having only two agents in games would emphasize modeling, identifying, and battling the strategies of single opponents. In TAC Classic, the number was eight. Keeping with the TAC tradition of modeling competitive oligopolic markets, the number for TAC SCM was set (arbitrarily) to six.

TAC Classic allows agents to be arbitrarily modified up to the finals. This is also adopted for TAC SCM. Not only does this meet the pragmatic need of participants for time to complete their agents, but, more importantly, encourages a continual exchange of ideas between teams, promoting more rapid progress.

As individual agents improve and weaker agents are knocked out in preliminary rounds, there is a tendency towards more competitive games. One effect is creating a moving target for the use of statistical/machine learning methods. We are free to choose to regard this as an artifact of running a competition in a particular way, obstructing real issues, or as a manifestation of a real world challenge, that markets will tend towards more competitiveness, and something that developers and agents have to deal with.

Suppliers and Customers. The agents interact through markets. While we could conceivably explore a range of market designs that support the needs of supply chains, a design decision was made early for TAC SCM to base both supplier-manufacturer and manufacturer-customer interaction on a uniform protocol: RFQ, offer, order.

RFQing works well, and is reasonably realistic, in the manufacturer-customer market. Customers have the simple choice to select the lowest bid. The agents will do the pricing, to best reflect supply and demand, and their individual objectives.

In the supplier-manufacturer market, agents RFQ suppliers, which select RFQs and price offers. The mechanism for selection and pricing are part of the game design. Agents can increase their relative chances of getting supplies in two ways: (1) by maintaining a good reputation (the fraction of the supplier's offers accepted in the past) and (2) through clever uses of the RFQ protocol and the given supplier mechanisms [30], changing the competition into who best outsmarts the game design. The latter is clearly tangential to the scientific objectives of TAC.

There are two possible remedies: To keep the RFQ protocol, the supplier has to be smarter, optimize well for its own self-interest, thus avoiding simple traps. Obviously, suppliers will never be perfect and an incentive to outsmart them remains. A smarter supplier also adds computation overhead to the game server.

Another option is to allow agents to bid for supplies in the supplier-manufacturer market. The challenge is then to design an unbiased bid clearing mechanism, which encourages competitive bidding true to the scientific aims of TAC.

Driving demand are probability distributions and random walks designed to offer realistic variation of demand between and during games. Similarly, supplier capacity varies in a random walk to model production and delivery problems. The demand distributions designed for the 2003 competition turned out to be strongly bi-modal. Once the random walk became high or low, it tended to stay there. For the 2004 competition, demand was redesigned and split into a sum of three overlapping segments, with individual random walks.

4. Server and agent software

A TAC SCM competition is run during several weeks, from mid-June to early August, with finals dur-

ing a major conference (IJCAI or AAMAS), but training games are run during most of the spring, and work towards the competition starts already in the preceding fall.

Agents need to be developed or improved, and the game and its software platform updated to reflect new insights. Developing a new agent typically requires a few person-months of work and has to be started in time (early spring). Initial design of the game and development of the software platform required in the order of 6-12 person-months, and annual updates another couple of person-months.

4.1. The game server

The TAC SCM server has two main tasks: running individual games and supporting competition management. Management is done through a Web interface: user registration, game scheduling, access to game viewers, result pages, game logs, etc.

To run games, the server implements a simulator of the supply chain scenario. For flexibility, not all of which needed for the present version of the game, all functionality is implemented as communicating agents, with the external agents interacting through local proxies.

The simulator logs all game events and messages for later analysis and debugging. All log files are stored and made available through the game result pages. A log toolkit is available for manually inspecting and visualizing games, and can also be used as an API for developing automatic analysis tools.

Running games can be viewed and inspected in real-time using the game viewer (Fig. 3), which includes a chat for real-time communication between competition participants, working from all around the world.

The server is open source software, which can be downloaded by participants to run games and competitions locally. Servers are also always in operation for training games, the addresses of which are published on the TAC Web site [27].

4.2. Developing agents

Competition participants implement agents that will connect to the server and play the roles of manufacturers in the scenario. Each simulated day, 15 seconds long, the agent collects information from the server, decides what to do, and responds to the server. Among its tasks are to

- bid on customer RFQs for PC orders

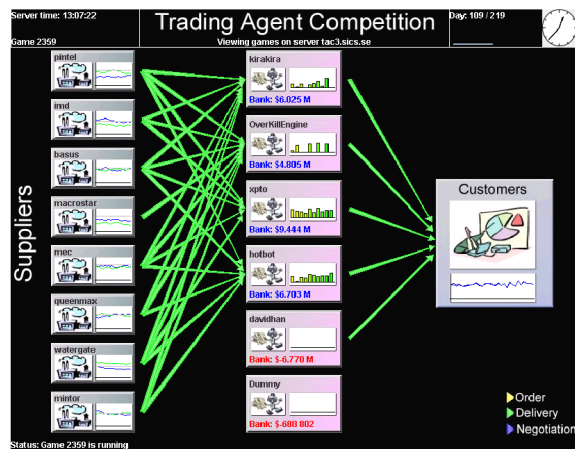


Fig. 3. The TAC SCM game viewer.

- procure components for production
- decide when to assemble PCs
- decide when to deliver PCs

To play games, a participant creates an account in the server to uniquely identify the agent. During training, games can be scheduled through the web interface and the agents can connect as needed. An unscheduled agent will automatically be assigned to the next game with free slots. During competitions, agents are connected continuously, waiting to be automatically scheduled for games. There is a small break between games to enable maintenance, e.g., restarting an agent.

It is possible in principle to implement agents in any technology on any platform supporting Internet connections. But requiring participants to reimplement the server protocol is an unnecessary barrier of entry and is error prone. For TAC Classic, software APIs implementing the protocol were provided. An XML-like protocol syntax was available, but was not fully specified and suffered from idiosyncrasies. For TAC SCM, all participants base their agents on the software API provided in Java, known as the AgentWare. The actual (compact binary) Internet protocol remains unspecified.

The AgentWare provides not only an implementation of the communication protocol, but also a base agent framework, SCMAgent, simplifying access to game information and assisting in bookkeeping of RFQs, orders, and inventory. Included with the AgentWare is also a simple example manufacturing agent. By using the AgentWare, a participant can avoid entire classes of software errors that would ruin game play, and focus on designing and implementing strategies.

5. State of the art

TAC SCM has been played for two years. Arunachalam and Sadeh [3] offer an analysis of the 2003 competition. The game and approaches are still very much evolving. We will briefly review five agents from the 2003 and 2004 finals, including the respective winners. For each agent, we summarize the overall approach and the elements of supply, production, and sales strategies. (Any errors are the responsibility of the present authors.)

The reviewed agents are characterized by different core approaches: heuristic decision rules, market-based resource management, distributed feedback control, stochastic programming, and machine learning for price prediction. To some extent, these correspond to different aspects of agent design, and could be combined in future agents. In contrast, the TAC Classic solutions involve solutions to two basic problems: prediction and completion, i.e., learning a model of the problem and making decisions with respect to this model. The approaches to TAC SCM have not yet been consolidated in this fashion. We present approaches in terms of separate strategies for supply, production, and sales, but it is conceivable that an analysis in terms of overall learning and decision problems will eventually be feasible and more rewarding.

A largely orthogonal aspect of TAC SCM agent design is the possibility to devise strategies that exploit specific properties of the domain or of the game design, and the corresponding need to counter strategies. Strategic interactions played a large role in the 2003 and 2004 games, and were essential for competitive game play. This is analyzed elsewhere and not further discussed here [30]. At this stage in the evolution of agent design, such strategies are ad hoc and hand-coded. We expect future agent designers to employ more principled methods also for this central aspect of competitive multi-agent interaction.

5.1. FreeAgent

FreeAgent of Tobiasson won the 2004 competition with an agent based on heuristics [28].

Supply. The highest acceptable component price depends on predicted future stock levels. Higher relative prices are accepted for cheaper components.

Production. Produce in delivery date order. If available capacity, produce safety stock of 200 units per product.

Sales. Sort RFQs by expected profit per cycle. Only bid on orders where components and production cycles are available and delivery is guaranteed. The bid price is

$$\min(p_t, \max(p_m, p_r)),$$

where p_t is the target price, p_m the minimum sell price, and p_r the customer reserve price. The target price p_t is last day's highest closing price with random variations. One bid is priced +5%. If factory utilization is low, the target can be lowered by up to 5%. The minimum sell price p_m is

$$\max(c + 10\%, pan),$$

where c is the sum of average component prices, a is the current average profit per cycle over all products, n is the number of cycles needed for the product, and p is a factor depending on the production queue, between 0.7 and 1 when all cycles are used, and 0 if there are free cycles. When 30 days remain, the minimum sell price is set to the target price.

5.2. RedAgent

RedAgent of Precup et al. [13] won the 2003 competition. The agent is based on a multi-agent design. Simple heuristic sub-agents manage tasks and resources. An internal market is used as a decision mechanism to coordinate the sub-agents, and to provide price estimates for goods to be sold and purchased.

Supply. One sub-agent per component type orders supply for a buffer (safety stock). The buffer size depends on the number of days left and the average demand. The components in the buffer are sold via an auction to assembler agents for production.

Production. One sub-agent sells production cycles and one sub-agent per product type bids for production cycles and components in to produce for their buffer. The size of buffers depends on the current day and the average demand for the product. Produced PCs are sold at the product auction for a price depending on the ratio of the current and target size of buffer (a low buffer level giving a higher price).

Sales. One sub-agent bids for customer RFQs and bases the offer price on price information from the internal product auctions. The price from the auction is incremented with an absolute margin per production cycle. The margin is increased if too many offers are accepted, and decreased otherwise. Each order received gets its own sub-agent that acquires the products from the product auction and delivers the product to the customer.

5.3. Deep Maize

Deep Maize of Wellman et al. [14] played in the 2003 and 2004 finals (2nd and 5th place). Through idealized equilibrium analysis, strategic aspects of the environment are identified and a profitable zone of operation defined. Distributed feedback control is employed to coordinate functional modules and to operate within the profitable zone.

Supply. Generate RFQs based on customer orders, the current inventory (to keep a buffer), and the expected future component utilization. To determine which offers to accept, maximize the total value of components ordered and in inventory.

Production. Sort the orders first by delivery date, secondly by PC value or, for late orders, by penalty. Produce and ship orders in that order. If available capacity, produce safety stock of 100 units per product.

Sales. Maximize the expected value of the offers for customer RFQs where the expected value is based on probability of winning the bid, probability of delivering, expected penalties, and the expected value the products could have been sold for in the future.

5.4. Botticelli

Botticelli of Greenwald et al. [5] played in the 2003 and 2004 finals (both 4th). Stochastic programming approaches are used in attempts to solve bidding and production scheduling optimally.

Supply. Order components early to fill component inventory with 70% of production capacity. Additional components are ordered by a procurement module that runs every 10 days.

Production. The production schedule is calculated during bidding optimization. Both actual customer orders and possible customer orders are considered. Scheduling is made for multiple days of production, but some production cycles for future days are saved for future RFQs (the amount to save is given by heuristics).

Sales. The bidding policy (pricing for current RFQs) and production scheduling problem is modeled as a stochastic program. The best bidding policy and production schedule is found by solving the stochastic program via an integer program formulation.

5.5. TacTex03

TacTex03 of Stone et al. [20] played in the 2003 finals (3rd) and in the 2004 semi-finals. The problem emphasized is bidding on customer RFQs. Several machine learning approaches have been explored for predicting the probability of a bid being accepted. Heuristic methods are used to search for the optimal set of bids.

Supply. Order a majority of components early to fill inventory. During the rest of the game probe suppliers and estimate their capacity and existing orders. Estimates of the customer demand trend (latest RFQs) and opponents inventories (market reports) are also made to ensure that the right components are ordered at the right time.

Production. The production schedule is calculated in the bidding procedure. The scheduler is greedy and takes both actual orders and RFQs into account. The scheduling horizon is 12 days.

Sales. The goal of the bidder is to optimize profit over the next 12 days. Current and expected future RFQs are considered. The starting bids are set to the reserve prices. Each bid is incrementally lowered to find the highest expected profit per PC produced. A learned approximation function for probability of winning and expected profit is used. A production scheduler is used to avoid overselling production capacity.

6. Discussion and future work

6.1. On the role of TAC

In any field, a lack of common problems and benchmarks risks leading to fragmented efforts and weak incentives for producing the strong generic solutions expected from scientific research.

Software agent research often addresses complex real-world domains with a multiplicity of issues. Solutions may require combinations of methods and considerable development effort. The diversity of domains and complexity of solutions adds to the challenge.

A competition helps focus research on standard problems, facilitates comparison of results, and, more importantly, creates a community with a vested interest in thoroughly analyzing and building on the results of peers and in jointly working towards the perfection of solutions.

Research on agents in electronic markets has to a large extent focused on idealized computational economies, or on the implementation of agents and market infrastructure without an accompanying analysis of their real-life properties. The agents of such systems have no true self-interest, are not competitive, do not speculate, but play the market game using a given strategy, working towards a common goal.

Through a trading agent competition, the self-interest of competing agents (and researchers) helps bring about a shift of interest from well-behaved, easy-to-analyze computational economies, to open full-fledged electronic markets in which the participants are free to act in their own best interest.

6.2. Pitfalls of competitions

It is worth keeping in mind that a competition is not in itself a means for scientific progress. Many efforts and achievements pertaining to competitions are not scientific results:

Winning a Competition. Appealing to human competitiveness as a driving force is a double-edged sword. Spending hours honing and perfecting a solution to make justice to an approach of general interest is one thing, tweaking by hand the tricks and parameters of an ad hoc solution is another. Winning to the exclusion of testing a scientific hypothesis is not in itself a scientific achievement. Adding to the problem of interpreting competition results is that initial conditions of agents (TAC Classic) and randomness in games (both) add an element of chance to winning [16].

Designing a Competition. A TAC game could potentially be developed as an experiment in mechanism design, using the competition for evaluation. But even the current ad hoc designs serve as scientific points of reference and have been published [22,32,33]. Methods of simulation experiment design and analysis could potentially be used to increase the scientific value of competition results [4]. Working against this is the need for agent development during running competitions.

Operations and Support. The competition software platform is publishable if it is of general scientific interest, e.g., the Michigan AuctionBot that served as the first platform for TAC Classic [34] and the RoboCup Soccer Server [19]. It remains an unavoidable fact that operations and support is a considerable overhead on the arrangers.

6.3. Actively promoting scientific progress

The main objective of TAC is to promote scientific progress. To this end, the TAC community makes a conscious effort to emphasize the competition as a research event:

- TAC is co-located with major scientific conferences (IJCAI and AAMAS).
- There is an associated reviewed workshop: Trading Agent Design and Analysis.
- Papers and team descriptions appear as edited volumes [8].

To coordinate these efforts, the community has formed a formal body, the Association for Trading Agent Research, with the aim to ensure the continued scientific impact of TAC.

6.4. The future of TAC

TAC SCM is a more complex game than its predecessor TAC Classic. For the latter, the solutions presented in the second year have continued to dominate the game. The performance gap between leading agents soon became small, indicating a convergence of approaches. A similar development is likely also for TAC SCM, but possibly over a longer period of time.

As performance converges, the issue is raised whether to modify the game to add or emphasize new research issues or to leave it unchanged for the purpose of future reference.

The TAC SCM design is being updated for TAC SCM 2005, to reduce the role of the “Day Zero” singularity and to make other improvements.² We expect the game to continue to evolve with increasing insights into this rich domain.

A game such as TAC SCM can potentially serve as an experimental platform and benchmark for multiple scientific disciplines. It is not an AI competition per se. Other methods, from automatic control, operations

²This is ongoing work at the time of writing and a topic for future papers.

research, and management science do apply, and are already being exploited by AI researchers. Using the competition as a link between fields is an exciting opportunity.

As long as all participants keep in mind the role of the competition for their own scientific objectives: to focus research on common problems, serve as a benchmark for experimental evaluation, and have a little fun along the way, all will benefit from a wider inflow of ideas.

The long term aim for TAC is to offer additional market games that help AI researchers focus on core challenges of high-impact real-world domains. The path to such a future is of an evolutionary nature, and depends on the continued involvement and commitment of the TAC community.

Acknowledgements

The research reported in this paper has been supported by SITI and Stiftelsen SISU.

TAC SCM is the product of a collaboration between the authors and Norman Sadeh and Raghu Arunachalam of Carnegie Mellon University's e-Supply Chain Management Laboratory.

The TAC SCM design team is indebted to the TAC community for crucial feedback on the design, enthusiastic participation in TAC, and first-rate scientific contributions to the field: including Michael Wellman, Peter Stone, Amy Greenwald, Peter Wurman, and many others.

The authors wish to thank Anders Hånström of Ericsson Telecom for discussions on TAC SCM design and real world supply chains.

References

- [1] AAMAS, the International Conference on Autonomous Agents and Multi-Agent Systems. <http://www.aamas-conference.org/>.
- [2] R. Arunachalam, J. Eriksson, N. Finne, S. Janson and N. Sadeh, The TAC supply chain management game, Computer Science Technical Report CMU-CS-03-184, Carnegie Mellon University, August 2003.
- [3] R. Arunachalam and N. Sadeh, The supply chain trading agent competition, *Electronic Commerce Research and Applications* 3(4) (2004), 389–404.
- [4] J. Banks, J.S. Carson, B.L. Nelson and D.M. Nicol, *Discrete Event Simulation*, 3rd edn, Prentice Hall, 2000.
- [5] M. Benisch, A. Greenwald, I. Grypari, R. Lederman, V. Naroditskiy and M. Tschantz, Botticelli: A supply chain management agent, in: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04)*, 2004, pp. 1174–1181.
- [6] A. Greenwald, ed., The 2002 trading agent competition: An overview of agent strategies, *AI Magazine* 24(1) (2003), 83–91.
- [7] J. Geunes, P.M. Pardalos and H.E. Romeijn, eds, *Supply Chain Management: Models, Applications, and Research Directions*, Kluwer Academic Publishers, 2002.
- [8] A. Greenwald, Special issue on TAC SCM, *SIGecom Exchanges* 4(3) (2003).
- [9] A. Greenwald and P. Stone, Autonomous bidding agents in the trading agent competition, *IEEE Internet Computing* 5(2) (2001), 52–60.
- [10] ICEC, the International Conference on Electronic Commerce. <http://icec.net/>.
- [11] IJCAI, the International Joint Conference on Artificial Intelligence. <http://www.ijcai.org/>.
- [12] M. Kearns and L. Ortiz, The Penn-Lehman automated trading project, *IEEE Intelligent Systems* 18(6) (2003), 22–31.
- [13] P.W. Keller, F.-O. Duguay and D. Precup, RedAgent-2003: An autonomous, market-based supply-chain management agent, in: *Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, New York, 2004.
- [14] C. Kiekintveld, M.P. Wellman, S. Singh, J. Estelle, Y. Vorobeychik, V. Soni and M. Rudary, Distributed feedback control for decision making on supply chains, in: *Proceedings of Fourteenth International Conference on Automated Planning and Scheduling*, 2004.
- [15] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda and M. Asada, The robocup synthetic agent challenge, 97, in: *International Joint Conference on Artificial Intelligence (IJCAI97)*, 1997.
- [16] P.L. Lanzi and A. Strada, A statistical analysis of the trading agent competition 2001, *SIGecom Exchanges* 3(2) (2002), 1–8.
- [17] H. Lee, V. Padmanabhan and S. Whang, The bullwhip effect in supply chains, *Sloan Management Review* 38(3) (1997), 93–102.
- [18] The MIT Beer Game, <http://beergame.mit.edu/>. A web version of a popular management game developed at MIT in the 60's.
- [19] I. Noda, H. Matsubara, K. Hiraki and I. Frank, Soccer server: a tool for research on multi-agent systems, in: *Applied Artificial Intelligence*, 1997.
- [20] D. Pardoe and P. Stone, Agent-based supply chain management: Bidding for customer orders, in: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04)*, 2004, pp. 1442–1443.
- [21] P.S.A. Reitsma, P. Stone, J.A. Csirik and M.L. Littman, Self-enforcing strategic demand reduction, in: *Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, Lecture Notes in Artificial Intelligence, Springer, 2002, pp. 289–306.
- [22] N. Sadeh, R. Arunachalam, J. Eriksson, N. Finne and S. Janson, TAC-03: A supply-chain trading competition, *AI Magazine* 24(1) (2003), 92–94.
- [23] The Supply-Chain Council, <http://www.supply-chain.org/>. The organization behind the SCOR metrics.

- [24] D. Simchi-Levi, P. Kaminsky and E. Simchi-Levi, *Designing and Managing the Supply Chain: Concepts, Strategies and Case Studies*, McGraw-Hill, Boston, 2000.
- [25] H. Stadler and C. Kilger, eds, *Supply Chain Management and Advanced Planning: Concepts, Models, Software and Case Studies*, 2nd edn, Springer, 2002.
- [26] P. Stone, Multiagent competitions and research: Lessons from RoboCup and TAC, in: G.A. Kaminka, P.U. Lima and R. Rojas, eds, *RoboCup-2002: Robot Soccer World Cup VI*, Springer, 2003, pp. 224–237.
- [27] The Trading Agent Competition (TAC) Website at SICS. <http://www.sics.se/tac/>.
- [28] M. Tobiasson, FreeAgent. Personal communication, November 2004.
- [29] M.P. Wellman, S.-F. Cheng, D.M. Reeves and K.M. Lochner, Trading agents competing: Performance, progress and market effectiveness, *IEEE Intelligent Systems* **18**(6) (2003), 48–53.
- [30] M.P. Wellman, J. Estelle, Y. Vorobeychik, S. Singh, C. Kiekintveld and V. Soni, Strategic interactions in a supply chain game, *Computational Intelligence* **21**(1) (2005), 1–26.
- [31] M.P. Wellman, A. Greenwald, P. Stone and P.R. Wurman, The 2001 trading agent competition, *Electronic Markets* **13**(1) (2003).
- [32] M.P. Wellman and P.R. Wurman, A trading agent competition for the research community, in: *IJCAI-99 Workshop on Agent-Mediated Electronic Commerce*, Stockholm, 1999.
- [33] M.P. Wellman, P.R. Wurman, K. O'Malley, R. Bangera, S. de Lin, D. Reeves and W.E. Walsh, Designing the market game for a trading agent competition, *IEEE Internet Computing* **5**(2) (2001), 43–51.
- [34] P.R. Wurman, M.P. Wellman and W.E. Walsh, The Michigan Internet AuctionBot: A configurable auction server for human and software agents, in: *Proceedings of the 2nd International Conference on Autonomous Agents (Agents '98)*, 1998.