# The 2001 Trading Agent Competition

**Michael P. Wellman***
University of Michigan
wellman@umich.edu

**Amy Greenwald**
Brown University
amygreen@cs.brown.edu

**Peter Stone**
AT&T Labs — Research
pstone@research.att.com

**Peter R. Wurman**
North Carolina State University
wurman@csc.ncsu.edu

**Abstract**

The 2001 Trading Agent Competition was the second in a series of events aiming to shed light on research issues in automating trading strategies. Based on a challenging market scenario in the domain of travel shopping, the competition presents agents with difficult issues in bidding strategy, market prediction, and resource allocation. Entrants in 2001 demonstrated substantial progress over the prior year, with the overall level of competence exhibited suggesting that trading in online markets is a viable domain for highly autonomous agents.

## Introduction

Automated trading in online markets is increasingly recognized as a promising domain for agent technology. Programmed trading is nothing new, but the emergence of electronic marketplaces has dramatically increased the opportunities for such trading. The trading task may be particularly well-suited for automation, as the interfaces are relatively simple. For example, messages from agents are typically limited to offers to exchange quantities of standardized goods for standardized currency using standardized exchange protocols. Although decisions about desirable trades may be based on a multitude of factors, specifying reasonable strategies seems often quite feasible with normal levels of effort.

As researchers we would like to make that last statement more precise, and develop an understanding of just how effective agent strategies can be, and how automated traders might affect the conduct of electronic markets. Understanding behaviors of other agents is clearly an advantage in designing one's own, as well as in designing the market itself.

Unfortunately, data about real-world trading agents is difficult to obtain. Designers of successful trading agents are naturally reluctant to compromise their proprietary advantages by revealing their strategies. Designers of unsuccessful agents seem equally reluctant to discuss their experience, for perhaps different reasons. This has led interested researchers to study their own designs, leading in some cases to useful observations, but all conclusions from such investigations must be qualified by questions of realism. Since the effectiveness of one agent's strategy depends on the strategies of others, having one designer choose all strategies introduces a great source of fragility to the research exercise.

One natural approach is for researchers to cooperate, by addressing their design energy to a common problem. The Trading Agent Competition (TAC) is an attempt to induce this cooperation, by organizing an event providing infrastructure, and promising external attention to the results. The first TAC (Wellman *et al.* 2001) was held in July 2000 in Boston, in conjunction with the International Conference on Multiagent Systems (ICMAS-00). TAC-00 attracted 18 participants from six countries, several of whom based on this experience contributed to the research literature on trading agents (Fornara & Gambardella 2001; Greenwald & Boyan 2001; Stone & Greenwald to appear; Stone *et al.* 2001). The competition can also claim spinoff contributions to research on visualizing real-time market data (Healey, St. Amant, & Chang 2001). Based on the success of that event, we held a sequel in October 2001 in Tampa, as part of the ACM Conference on Electronic Commerce (EC-01).

One positive result of the second TAC was the possibility of measuring actual progress, through performance and through the transfer of ideas from one competition to the next. The TAC-01 experience provides a wealth of lessons for trading agent designers, and designers of competitions.

## TAC Market Game

To enter the competition, TAC participants developed software agents to play a challenging market game. Entries in the game play the role of travel agents, striving to arrange itineraries for a group of clients who wish to travel from TACTown to TAC's host city and back again during a five-day period. Travel goods are traded at simultaneous on-line auctions that run for twelve minutes (reduced from 15 minutes in the 2000 competition). An agent's objective is to secure goods serving the particular desires of its clients, but to do so as inexpensively as possible. An agent's score is the difference between the utility it earns for its clients and its net expenditure. In this section, we summarize the TAC game, noting differences between 2000 and 2001 rules; for further details, visit `http://tac.eecs.umich.edu`.

*Contact author. University of Michigan AI Laboratory, 1101 Beal Av, Ann Arbor, MI 48109-2110, USA.

## Trading Travel Goods

Agents trade three types of travel goods: (1) flights to and from the host city, (2) room reservations at two available hotels–one considered higher quality than the other, and (3) entertainment tickets for three kinds of events. Each type is traded according to distinct market rules, with separate auctions corresponding to every combination of good type and day, yielding 28 auctions in total: eight flight auctions (there are no inbound flights on the fifth day, and there are no outbound flights on the first day), eight hotel auctions (two hotel types and four nights), and twelve entertainment ticket auctions (three entertainment event types and four nights). All 28 auctions operate simultaneously and asynchronously on the Michigan Internet AuctionBot server (Wurman, Wellman, & Walsh 1998). We describe the auctions rules for each good type in turn.

**Flights.** An effectively infinite supply of flights is offered by the "TAC seller" at continuously clearing auctions. No resale or exchange of flights is permitted. The seller's offers follow a random walk, setting prices initially between $250 and $400, and perturbing them every 30-40 seconds by a random value uniformly selected in the range $[-10, x(t)]$, where $t$ is the number of seconds from game start. All prices were confined within bounds: $150 to $600 in 2000, and $150 to $800 in 2001. In 2000, price changes followed an unbiased random walk; that is, $x(t) = 10$ for all times $t$. Most of the entrants therefore waited until near the end of the game to purchase flights. Waiting avoided a commitment pending revelation of other relevant information (e.g., hotel prices), at zero expected cost (modulo the boundary effects). To present agents with a meaningful tradeoff, we changed the policy in 2001 so that prices were biased to drift upwards: for each flight auction $a$, a number $x_a$ was uniformly drawn in the range $[10, 90]$, and $x(t) = 10 + (x_a - 10)(t/720)$.

**Hotels.** The TAC seller also makes available 16 rooms per hotel per night, which are sold in ascending, multi-unit, sixteenth-price auctions. In other words, the winning bidders are those who place the top sixteen bids, and these bidders uniformly pay the sixteenth-highest price. No bid withdrawal or resale in hotel auctions is permitted. There is a tendency in such auctions (observed, for example, in the Santa Fe Double Auction Tournament (Rust, Miller, & Palmer 1994)) to wait until the end to bid, both to avoid undue commitment and to withold strategic information from competing bidders. In an attempt to encourage agents to bid early, in 2000 we subjected hotel auctions to early closing after random periods of inactivity; otherwise, the auctions closed simultaneously at the end of the game. However, this countermeasure proved ineffective, as agents merely entered minimal increments at measured intervals in order to ensure the auctions stayed alive. Much of the meaningful price movements occurred as hotel prices skyrocketed near the end. In 2001, we induced early bidding through more drastic means: closing hotel auctions randomly at one-minute intervals. Specifically, one randomly selected hotel would close after four minutes, a second after five minutes, and so on, until the last auction closed after eleven minutes. From the agents' point of view, the order of auction closings was unknown and unpredictable.

**Entertainment.** TAC agents buy and sell entertainment tickets in continuous double auctions. Each agent receives an initial endowment of tickets. In 2000, for each event on each night, each agent received the following number of tickets: zero with probability 1/4, one with probability 1/2, and two with probability 1/4. Trading entertainment was not a major factor in 2000, as the symmetric distribution meant that agents could usually do reasonably well by keeping its initial endowment. In 2001, to promote trade, each agent received exactly 12 tickets, partitioned as follows: for day 1 or day 4, four tickets of one type and two tickets of a second type; and, for day 2 or day 3, four tickets of one type and two tickets of a second type.

## Trip Utility

Eight trading agents compete for travel goods in a TAC game instance, with each agent representing eight clients. The market demand is thus determined by the sixty-four clients' preferences, which are randomly generated from specified probability distributions. A client's preference is characterized by (1) ideal arrival and departure dates (IAD and IDD, respectively, which range over days 1 through 4, and days 2 through 5, respectively), (2) value for staying in the premium quality hotel (HV, which takes integer values between 50 and 150), and (3) values for each of the three types of entertainment events (integers between 0 and 200). The three available entertainment events in 2001 were Amusement Park (AP), Alligator Wrestling (AW), and Museum (MU).

A TAC travel *package* is characterized by arrival and departure dates (AD and DD, respectively), a hotel type (H, which takes on value G for good or F for fair), and entertainment tickets ($I_X$ is an indicator variable that represents whether or not the package includes a ticket for event $X$). In order to obtain positive utility for a client, a TAC agent must construct a *feasible* package for that client; otherwise, the client's utility is zero. A feasible package is one in which (1) the arrival date is strictly less than the departure date, (2) the same hotel is reserved during all intermediate nights, (3) at most one entertainment event per night is included, and (4) at most one of each type of entertainment ticket is included.

A client's utility for a feasible package is given by:

$$\text{utility} = 1000 - \text{travelPenalty} + \text{hotelBonus} + \text{funBonus}$$

where

$$\begin{aligned} \text{travelPenalty} &= 100(|\text{IAD} - \text{AD}| + |\text{IDD} - \text{DD}|) \\ \text{hotelBonus} &= \begin{cases} \text{HV} & \text{if H} = \text{G} \\ 0 & \text{otherwise} \end{cases} \\ \text{funBonus} &= I_{\text{AP}}\text{AP} + I_{\text{AW}}\text{AW} + I_{\text{MU}}\text{MU}. \end{aligned}$$

### Allocating Goods to Clients

In 2000, TAC agents were responsible for assigning goods to clients, reporting their allocations at the end of the game. By 2001, this problem was considered well-understood; thus, we had the TAC server compute and report each agent's optimal allocation.

## Themes in Agent Design

Although each TAC entry exhibited its own special features, it is possible to identify some common structures and general themes. Points of comparison can be organized into the common decisions the agents face, which we collect into the canonical agent "bidding cycle" of Table 1. Different agents may naturally frame the questions somewhat differently, or in an implicit way. Nevertheless, this skeletal structure provides a convenient organization for a discussion of characteristic strategic features.

| |
|---|
| REPEAT |
| 1. For each good, do I want to bid now or later? |
| 2. For each good that I want to bid on now, what quantity do I want to buy or sell? |
| 3. For each exchange quantity, what price should I offer? |
| UNTIL game over |

Table 1: Trading agent bidding cycle: a skeletal view.

### When to Bid

The three TAC auction types present agents with distinct timing concerns. Flights are offered continuously on a posted-price take-it-or-leave-it basis, so the agent's decision is simply whether to commit at a given time. Since flight prices are expected to increase over time, agents face the tradeoff of buying early for less, or paying more with benefit of gaining information about other goods (e.g., hotel prices and winnings). Different agents approached this problem in different ways. For example, Caisersose and livingagents always acquired all their flights immediately, on average about one minute into the game. Urlaub01 was even faster on average (46 seconds), even though it occasionally picked up some extra flights late into the game. ATTac makes its flight bidding decisions based on a cost-benefit analysis: if the cost of postponing a bid on a particular flight exceeds the benefit of winning that flight under multiple scenarios, then ATTac bids. This led to some immediate purchases and others spread later in the game, with an overall mean time of about two minutes. The remaining agents in the finals deliberated further, with Tacsman getting its flights on average over four minutes after game start.

Hotels, in contrast, are exchanged through ascending auctions with periodic revelation of price quotes and one-time clearing. Once per minute, each hotel auction would release a price quote and one was randomly selected to clear and close. Since no information was revealed during these one-minute intervals, bidding was effectively organized into discrete rounds.[1] Agents typically spent the bulk of each round calculating their bidding decisions, placing the bids at round end. Exactly what time constituted the "end", though, depended on an agent's assessment of network latency and the risk of placing a late bid. Note that all agents were compelled to maintain active bids for all open hotels, since the next hotel to close was unknown and unpredictable.

Like flights, entertainment is exchanged in continuous auctions, giving agents the opportunity to time their offers based on strategic considerations. Some agents (e.g., livingagents, ATTac, 006) explicitly maintained separate control threads for entertainment bidding decisions. Most agents in TAC-01 bid on all goods only once per minute (after the first three minutes), since this timing strategy was appropriate for the sequentially closing hotel auctions.

### What to Bid On

One of the key problems that a TAC agent faces is the so-called *completion* problem (Boyan & Greenwald 2001):

> Given my current holdings, and given (expected) market prices, what goods would I choose to buy or sell at these prices?

We observed two general approaches in TAC-01:

- Agents such as whitebear solved the completion problem using global optimization techniques employed by TAC-00 participants, including integer linear programming (Stone *et al.* 2001) and heuristic search (Greenwald & Boyan 2001).
- TacsMan constructed travel packages by optimizing utility client-by-client, rather than globally.

Most agents used completion to choose a limited set of goods to bid on. However, ATTac always bid on all open hotel rooms, based on independent assessments of each room's predicted marginal utility.

### How Much to Bid

The decision about what price to offer for a given good was typically decomposed into the problems of first establishing a reservation value, and then determining a strategic bidding policy based on that value. It is not straightforward to assign reservation values to individual goods, however, due to the interdependences among them. Perfectly complementary goods (e.g., an inflight and outflight for a particular client) are worthless in isolation, and perfectly substitutable goods (e.g., rooms in different hotels for the same client and day) provide added value only in isolation. Nonetheless, most agents employed marginal utility—ignoring interdependencies–at least as a baseline reservation value. Note that taken literally, each good necessary for trip feasibility (e.g., a hotel room on a particular night, once flights have been committed and no alternative rooms are available) has a marginal utility equal to the value of the

[1]This is in contrast with TAC-00, where all price information was revealed continually, and hotel auctions cleared at the end. As a result, most TAC-00 agents placed their serious hotel bids at or near the end, and prices often rose dramatically at that point.

whole trip. In TAC-00 several agents entered bids on this basis, causing hotel prices to escalate wildly.

### Price Prediction

At several points in its bidding cycle (Table 1), an agent must solve a version of the completion problem. Determining what to bid on directly poses the question, and marginal utility calculations employed in pricing require that it be calculated twice—once each with and without the good included. The completion problem in turn requires a set of market prices as input. But before closing, actual prices are unknown. TAC agents employed a variety of statistical estimation techniques to predict market clearing prices.

For flights, TAC participants employed maximum likelihood estimation, least squares regression, and other simple prediction methods. For entertainment tickets, historical data suggested that most traded at or near 80. One agent (livingagents) therefore placed all its offers at this value, and another (Urlaub01), used the prediction to set upper and lower bounds on its entertainment bids.

Predicting hotel clearing prices was a central element of trading strategy in TAC-01. (In TAC-00, little information relevant to clearing prices was revealed during the course of a game.) There are many possible approaches to this hotel price estimation problem. Among those we observed in TAC-01 are the following, associated in some cases with agents that seemed to exemplify that approach.

1. Just use the current price quote, $p_t$.
2. Adjust based on historic data. For example, if $\Delta_t$ is the average historical difference between clearing price and price at time $t$, then the predicted clearing price is $p_t + \Delta_t$.
3. Predict by fitting a curve to the price points seen in the current game (polimi_bot).
4. Predict based on closing price data for that hotel in past games (livingagents). 006 combined this approach with extrapolation from current prices.
5. Same as above, but condition on hotel closing time, recognizing that the closing sequence will influence relative prices (Retsina, which also conditioned on current prices).
6. Same as above, but condition on full ordering of hotel closings (Tacsman), or which hotels are open or closed at a particular point (RoxyBot, Urlaub01).
7. Learn a mapping from features of the current game (including current prices) to closing prices based on historic data (ATTac).
8. Hand-construct rules based on observations about associations between abstract features (SouthamptonTAC).

Some agents, rather than using a point estimate of prices, took into account *distributions* of prices, solving the completion problem repeatedly for various prices sampled from this distribution. This kind of analysis reveals the sensitivity to the estimates of the conclusions drawn from them, and also permits some accounting for correlation of prices across goods. The challenge to agents that employed such sampling techniques was to combine the results of their sampling. Simple averaging is not necessarily correct, as the appropriate action when prices are equally likely to be $p$ or $p'$ may be entirely different from the appropriate action when the price is certainly $(p + p')/2$.

## TAC 2001 Tournament

TAC-01 was organized as a series of four competition phases, culminating with the semifinals and finals at the EC-01 conference. First, the qualifying round served to select the 16 agents that would participate in the semifinals. Second, the seeding round was used to divide these agents into two groups of eight. After the semifinals, on the morning of the 14th, four teams from each group were selected to compete in the finals, which took place that same afternoon.

### Preliminary Rounds

The qualifying round ran from 10-17 September and included 28 agents, each of which were randomly selected to play in about 270 games. The main purpose of the qualifying round was to encourage competitors to create functional agents well in advance of the finals, thus ensuring a competitive field by the main event. Later scores were weighted more heavily, thus encouraging teams to experiment early on but create a stable agent by the end of the round.

Several groups entered more than one agent in the qualifying round. However only one agent per group was allowed to proceed to the seeding round. The top twelve agents automatically qualified, and all others with positive scores were invited to participate in the finals contingent on attendance at the workshop.

For the resulting field of 16 teams, a seeding round was held from 24 September until 5 October to determine how the semifinal groups would be formed. The top four and bottom four teams from the seeding round formed one group, with the rest of the teams (places 5-12) forming the other. The extensive seeding round offered a consequential testing scenario for agents during this period of intensive agent development. As a side effect, the seeding round provided a source of realistic game data for designers taking a statistical approach. Again, the scores were weighted such that those later in the round counted more.

In addition to the 16 qualifying teams, two additional agents were included in the seeding rounds for calibration purposes. First, ATTac-2000 (Stone *et al.* 2001) is a copy of the highest-scoring agent from the TAC-00 finals. To account for the rule changes between TAC-00 and TAC-01, ATTac-2000 was modified with a one-line change that caused it to place all of its bids before the first hotel closed as opposed to during the last minute of the game.

Second, dummy_buyer is included in this round's pool as a benchmark from the qualifying round. dummy_buyer is the agent provided by the TAC team to play in test games that do not have a full slate of agents. Whereas most of the other agents' behaviors were modified between (and during) the qualifying and seeding round, the dummy was left unchanged. Indeed, we observed substantial deterioration in

| Agent | Affiliation | Score |
|---|---|---|
| SouthamptonTAC | U Southampton | 3164 |
| whitebear | Cornell U | 3120 |
| Urlaub01 | Penn State U | 3076 |
| livingagents | Living Systems AG | 3012 |
| TacsMan | Stanford U | 2984 |
| CaiserSose | U Essex | 2870 |
| polimi_bot | Politecnico di Milano | 2858 |
| umbctac | U Maryland Baltimore Cty | 2765 |
| RoxyBot | Brown U | 2732 |
| ATTac | AT&T Research | 2686 |
| Retsina | Carnegie Mellon U | 2675 |
| PainInNEC | NEC Research | 2575 |
| ATTac-2000 | | 2412 |
| harami | Bogazici U | 2156 |
| dummy_buyer | | 1673 |
| jboadw | McGill U | 1307 |
| bang | NCST Bangalore | 1306 |
| 006 | Swedish Inst Comp Sci | 1115 |
| arc-2k | Chinese U Hong Kong | -36 |

Table 2: Scores during the seeding round.

| Heat 1 | | Heat 2 | |
|---|---|---|---|
| Agent | Score | Agent | Score |
| livingagents | 3660 | Retsina | 3294 |
| SouthamptonTAC | 3615 | ATTac | 3249 |
| Urlaub01 | 3485 | CaiserSose | 3038 |
| whitebear | 3470 | TacsMan | 2966 |
| 006 | 3241 | PainInNEC | 2906 |
| arc-2k | 1746 | polimi_bot | 2835 |
| jboadw | 1717 | umbctac | 2773 |
| harami | 94 | RoxyBot | 2112 |

Table 3: Scores for the two semifinal heats. Each agent played 11 games.

the dummy's standing as the preliminary rounds progressed. Results of the seeding round are displayed in Table 2.

## The Main Event

The semifinals and finals were held together on a single day, 14 October. This format severely limited the number of games that could be played. On the other hand, the single-day format allowed the culmination of the tournament to take place in a workshop environment with most of the participants present. It also ensured that agents would remain more or less unchanged during these rounds.

Each of the semifinal heats consisted of eleven games among identical agents. The top four teams from each heat advanced to the finals. The results of the semifinals are shown in Table 3.

The finals consisted of 24 games among the same eight agents. Right from the beginning, it became clear that livingagents was the team to beat in the finals. They jumped to an early lead in the first two games, and by eight games into the round, they were more than 135 points per game ahead of the next team (SouthamptonTAC). After another eight games, they were more than 250 points ahead of their two closest competitors (ATTac and whitebear).

| Agent | Final score | Client pref adjust |
|---|---|---|
| livingagents | 3670 | -66 |
| ATTac | 3622 | 42 |
| whitebear | 3513 | -72 |
| Urlaub01 | 3421 | -2 |
| Retsina | 3352 | -30 |
| SouthamptonTAC | 3254* | -64 |
| CaiserSose | 3074 | 202 |
| TacsMan | 2859 | -11 |

Table 4: Scores during the finals. Each agent played 24 games. *SouthamptonTAC's score was adversely affected by a crash in one game. Discounting that game would have led to an average score of 3531.

At that point, ATTac began making a comeback. With one game to be played, ATTac was only an average of 22 points per game behind. It thus needed to beat livingagents by 514 points in the final game to overtake it, well within the margins observed in individual game instances.

As the game completed, ATTac's score of 3979 was one of the first to be posted by the server. The other agents' scores were reported one by one, until only the livingagents score was left. After agonizing seconds, the TAC server posted a final game score of 4626, enough for livingagents to retain the lead. Final scores are posted in Table 4.

## Influence of Client Preferences

Because they determine the scoring function, randomly generated client requests for a particular game can have a significant bearing on scores. Whereas this effect can be expected to wash out over a large number of games, it may not in a smaller set (e.g., the TAC finals).

To try to assess the affect of this factor on TAC results, we identified a small number of statistics on client parameters that we would expect to be correlated with performance. We tested these for significance over the seeding round games, employing the variables in a linear regression along with indicator 0-1 variables for each of the agent identities. After a very small amount of trial-and-error, we came up with the following significant measures:

1. total client preferred travel days
2. total entertainment values
3. ratio of "easy" days (1 and 4) to hard (2 and 3) in preferred trip intervals

Applying the resulting regression model to the finals data yields an "adjustment factor" that accounts for the chance effect of client preference parameters. These values (normalized) are displayed in the final column of Table 4.

If the scores were adjusted based on these factors, there would be two changes in the rankings. First, livingagents had somewhat more favorable client data than did ATTac, and so in the adjusted rankings ATTac would come out in front. Caisersose had by far the least favorable inputs, and so it too would rise by one spot. Adjustment would result in several ranking changes within the semifinals, but no change in the top four selection for either heat.

## Strategy: Livingagents vs. ATTac

A sharper contrast in agent strategy can be drawn by examining more specifically the approaches of the two particular agents that finished at the top of the standings in the finals.

ATTac uses a predictive, data-driven approach to bid based on expected marginal values of all available goods. A price-predictor based on boosting techniques (Schapire *et al.* 2002) is at the heart of the algorithm. This price-predictor generates distributions over expected hotel closing prices. ATTac then samples from these distributions in an effort to compute the expected marginal utility of each good. It then bids exactly these expected marginal utilities. As the game proceeds, the price distributions change in response to the observed price trajectories, thus causing the agent to continually revise its bids. Note that by using this strategy, provided that the price is right, ATTac automatically buys contingency goods to guard against the possibility of the most desired goods becoming too expensive.

In terms of the skeletal bidding cycle of Table 1, ATTac focuses mainly on step 3. On every cycle it determines prices for every available hotel room and entertainment ticket. For flights, like most other agents, it does determine a single coherent set of candidate flights before performing its expected marginal utility calculations to determine whether it is worth it to buy now or to wait for additional price information to reduce uncertainty.

The strategy of livingagents (Fritschi & Dorer 2002) is strikingly different. livingagents takes the initial flight prices and calculates optimal client trips, assuming hotel prices will be at historical averages.[2] It then purchases flights immediately, and bids for the required hotels at prices high enough to ensure successful acquisition. These choices are not reconsidered, and indeed the flight and hotel auctions are not monitored at all. livingagents similarly makes a fixed decision about which entertainment to attempt to buy or sell, assuming they will be priced at their historical average of $80. It does monitor the entertainment auctions, taking acceptable offers opportunistically until putting in final reservation prices at the seven-minute mark.

At first blush, it is quite surprising that an effectively open-loop strategy such as that employed by livingagents could be so successful. In general, the optimal configuration of trips will depend on hotel prices, yet the open-loop strategy ignores all the predictive information about them that is revealed as the game progresses. Moreover, the behavior is quite risky. If the initial hotel buy offers are not high enough, the agent will fail to complete some trips and thus lose substantial value. But if they are placed sufficiently high to ensure purchase, there is a danger that the agent will have to pay a price such that the trip is unprofitable (or less profitable than an alternative).

In particular, it is quite clear that if all agents followed the strategy of livingagents, the result would have been disastrous. With all eight agents placing very high bids for the hotels, the prices will skyrocket and most of the trips will be unprofitable. Indeed, experiments with analogous behaviors for a version of the ATTac-2000 agent bear out this result (Stone *et al.* 2001).

But of course, livingagents was *not* competing with copies of itself. Most of the other agents, like ATTac, employed closed-loop, adaptive strategies that condition their behaviors on the evolution of prices. By steering away from goods that are becoming expensive (or predicted to become so), they also attenuate the forces raising those prices. Thus, these agents effectively "stabilize" the system, keeping the prices lower, and less variable, than they would be without such tight monitoring. This benefits all agents, whether or not they are monitoring.

The open-loop strategy has several advantages. It is simple, and avoids the expected tangible costs of waiting (e.g., letting flight prices rise) and hedging (e.g., buying contingency goods that may not be used). Whether it is worthwhile to monitor the markets and adapt bidding behavior depends pivotally on predictability of closing prices.

- If the prices are perfectly predictable from the start of the game, then there is no benefit to an adaptive strategy. (Indeed, the optimal closed-loop strategy would degenerate to an open-loop behavior.)
- With large price variances, a closed-loop strategy should do better. Typically, it will place mid-range bids in all the auctions and end up buying the cheapest goods. In the end, it may have to pay some high prices to complete itineraries, but it should largely avoid this necessity. The open-loop strategy picks its goods up front and ends up paying whatever price they end up at, which in some cases will be quite high.
- With small price variances, an *optimal* closed-loop strategy would in principle still be as good as any open-loop strategy. Nevertheless, the increase in complexity may be great for a small potential benefit, and even small miscalculations (e.g., underconfidence in predicted values, leading to excessive waiting and hedging) can prevent the agent from achieving this benefit. Thus, the relative simplicity of the open-loop approach may more than compensate for its suboptimality.

The foregoing argument suggests that there is some natural equilibrium between adaptive and open-loop behavior in the TAC game. Exactly what this equilibrium is, and whether the configuration of agents participating in TAC-01 achieved a close balance, are subjects for further analysis and empirical study.

## Discussion

All told, a tremendous amount of effort has gone into organizing and competing in TAC. Is this effort justified?

[2] For this estimate, livingagents used data from the preliminary rounds. As the designers note (Fritschi & Dorer 2002), hotel prices in the finals turned out to be significantly lower than during the preliminary rounds, presumably because the more successful agents in the finals were better at keeping these prices down.

Although not entirely realistic, the TAC game incorporates more realistic elements than most models previously studied in research on economically motivated agents. TAC has provided a platform on which to demonstrate new technologies (e.g., the livingagents development platform (Fritschi & Dorer 2002)), and to apply favorite techniques (e.g., fuzzy rules used by SouthamptonTAC, constraint programming by 006 (Aurell *et al.* 2002)). In addition, it serves as a benchmark problem against which comparisons can be drawn between competing techniques.

It seems clear that TAC has been instrumental in focusing a wide variety of researchers' attention on the trading agent problem domain. However, it is still too early to tell if the agent architectures developed by the competitors will influence the technologies that are eventually deployed in commercial settings. While the game is more complex than most other research models, it is far less so than real e-commerce settings, and the abstraction may well create incentives that are not aligned with real market conditions. The game design was influenced by the desire to make it interesting and challenging, which sometimes ran counter to the desire to keep it realistic. For example, the periodic closing of randomly selected hotel auctions is a reasonable measure to promote early bidding, but one not typically seen in real-world market mechanisms.

We are encouraged that the research will remain relevant by the fact that much of the focus of 2001 competition was on price prediction and timing of bid decisions, two topics that we think will be widely relevant in commercial applications. In addition, the TAC servers have been under almost constant use since the competition by several of the participants running ongoing experiments. Indeed the two servers which host TAC-01 logged more than 25 million bids in the last six months of 2001. About 1/4 of that load came after the official competition.

Whether the research efforts succeed or not, we have been amply rewarded by the success that instructors have had using TAC in education. The design of a TAC agent incorporates many central AI and e-commerce concepts, and provides a valuable framework around which to struture many related lessons. The game has been used as a class project in AI and e-commerce courses at several universities in the United States and Europe.

The Swedish Institute of Computer Science (SICS) will organize the next TAC event, to be held in Edmonton, Canada in July 2002. SICS has released an open-source version of the TAC server, developed using SICStus Prolog. Although TAC-02 will follow the same rules as TAC-01, we expect that future events will address domains—particularly business-to-business scenarios—that present other types of challenges to trading agents.

## Acknowledgments

The 2001 Trading Agent Competition was the product of dedicated effort by the Michigan TAC Team: Kevin O'Malley, Christopher Kiekintveld, Daniel Reeves, Sowmya Swaminathan, and William Walsh. The research value of the event is attributable to the energies devoted by the entrants. This account was informed by presentations from the entrants at the TAC-01 workshop, as well as their comments on earlier drafts of this paper (but please blame the authors for remaining inaccuracies). The research was supported in part by NSF grant IIS-9988715. TAC-01 also benefited from the generous sponsorship of Net Exchange, and the E-Commerce Learning Center at North Carolina State University.

## References

Aurell, E.; Boman, M.; Carlsson, M.; Eriksson, J.; Finne, N.; Janson, S.; Kreuger, P.; and Rasmusson, L. 2002. A trading agent built on constraint programming. In *Eighth International Conference of the Society for Computational Economics: Computing in Economics and Finance*.

Boyan, J., and Greenwald, A. 2001. Bid determination in simultaneous auctions: An agent architecture. In *Third ACM Conference on Electronic Commerce*, 210–212.

Fornara, N., and Gambardella, L. M. 2001. An autonomous bidding agent for simultaneous auctions. In *Fifth International Workshop on Cooperative Information Agents*, number 2182 in Lecture Notes on Artificial Intelligence, 130–141.

Fritschi, C., and Dorer, K. 2002. Agent-oriented software engineering for successful TAC participation. In *First International Joint Conference on Autonomous Agents and Multi-Agent Systems*.

Greenwald, A., and Boyan, J. 2001. Bidding algorithms for simultaneous auctions: A case study. In *Third ACM Conference on Electronic Commerce*, 115–124.

Healey, C. G.; St. Amant, R.; and Chang, J. 2001. Assisted visualization of e-commerce auction agents. In *Graphics Interface*, 201–208.

Rust, J.; Miller, J. H.; and Palmer, R. 1994. Characterizing effective trading strategies: Insights from a computerized double auction tournament. *Journal of Economic Dynamics and Control* 18:61–96.

Schapire, R. E.; Stone, P.; McAllester, D.; Littman, M. L.; and Csirik, J. A. 2002. Modeling auction price uncertainty using boosting-based conditional density estimation. In *Nineteenth International Conference on Machine Learning*.

Stone, P., and Greenwald, A. to appear. The first international trading agent competition: Autonomous bidding agents. *Journal of Electronic Commerce Research*.

Stone, P.; Littman, M. L.; Singh, S.; and Kearns, M. 2001. Attac-2000: An adaptive autonomous bidding agent. *Journal of Artificial Intelligence Research* 15:189–206.

Wellman, M. P.; Wurman, P. R.; O'Malley, K.; Bangera, R.; Lin, S.-d.; Reeves, D.; and Walsh, W. E. 2001. Designing the market game for a trading agent competition. *IEEE Internet Computing* 5(2):43–51.

Wurman, P. R.; Wellman, M. P.; and Walsh, W. E. 1998. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Second International Conference on Autonomous Agents*, 301–308.