

Walverine: A Walrasian Trading Agent

Shih-Fen Cheng, Evan Leung, Kevin M. Lochner, Kevin O'Malley,
Daniel M. Reeves, L. Julian Schwartzman, and Michael P. Wellman*

University of Michigan
Artificial Intelligence Laboratory
Ann Arbor, MI 48109-2110 USA
tac.support@umich.edu

February 11, 2003

Abstract

TAC-02 was the third in a series of Trading Agent Competition events fostering research in automating trading strategies by showcasing alternate approaches in an open-invitation market game. TAC presents a challenging travel-shopping scenario where agents must satisfy client preferences for complementary and substitutable goods by interacting through a variety of market types. Michigan's entry, Walverine, bases its decisions on a competitive (Walrasian) analysis of the TAC travel economy. Using this Walrasian model, we construct a decision-theoretic formulation of the optimal bidding problem, which Walverine solves in each round of bidding for each good. Walverine's optimal bidding approach, as well as several other features of its overall strategy, are potentially applicable in a broad class of trading environments.

*Revised and extended version of a paper to appear in the *Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-03)*.

1 Introduction

The annual Trading Agent Competition (TAC) provides a periodic forum for exploring the interaction of strategies for a challenging market game. The original motivation for TAC was to encourage agent researchers interested in trading to focus on a common problem, involving multiple interrelated goods traded simultaneously in a strategically complex setting (Wellman *et al.*, 2001). The TAC series has succeeded in attracting significant participation (approximately 20 entries per year, a majority of which clearly reflect a substantial effort), and most importantly, producing a sizable body of ideas and results, as reflected in over a dozen published reports to date.¹

After operating the competition for its first two years, the University of Michigan entered TAC for the first time in 2002. TAC-02, organized by the Swedish Institute of Computer Science (SICS), was held in Edmonton, Canada, in July. The field of 19 entrants included many strong contenders from the previous year (Wellman *et al.*, 2003)—several of which were significantly improved (Greenwald, 2003a)—as well as some newcomers.

Our agent, “Walverine”, gets its name from the University of Michigan team mascot (the wolverine—a variety of weasel), and Léon Walras, the 19th-century economist who first envisioned the concept of interacting markets in price equilibrium (Walras, 1954). Walverine’s overall approach can be characterized as “competitive analysis”—forming expectations based on a model where agents behave competitively (Arrow and Hahn, 1971). From such assumptions about the other agents’ behavior, Walverine formulates a decision-theoretic model of its bidding problem, and issues its offers accordingly.

Embodying the competitive analysis approach in a software trading agent has led us to develop several novel techniques. Although worked out in detail specifically for the TAC environment, we expect that the underlying ideas will prove applicable to a broad range of trading contexts.

2 Trading Agent Competition

2.1 TAC Rules

The TAC game presents a travel-shopping task, where traders assemble flights, hotels, and entertainment into trips for a set of eight probabilistically generated clients. Clients are described by their preferred arrival and departure days (pa and pd), the premium (hp) they are willing to pay to stay at the “Towers” (T) hotel rather than “Shanties” (S), and their respective values (e_1, e_2, e_3) for three different types of entertainment events. The agents’ objective is to maximize the value of trips for their clients, net of expenditures in the markets for travel goods. The three categories of goods are exchanged through distinct market mechanisms.

Flights. A feasible trip includes air transportation both ways, comprising an inflight day i and outflight day j , $1 \leq i < j \leq 5$. Flights in and out each day are sold independently, at prices determined by a stochastic process. The initial price for each

¹<http://auction2.eecs.umich.edu/researchreport.html>

flight is $\sim U[250, 400]$, and follows a random walk thereafter with an increasingly upward bias.

Hotels. Feasible trips must also include a room in one of the two hotels for each night of the client’s stay. There are 16 rooms available in each hotel each night, and these are sold through ascending 16th-price auctions. Agents submit bids for various quantities, specifying the price offered for each additional unit. When the auction closes, the units are allocated to the 16 highest offers, with all bidders paying the price of the lowest winning offer. Each minute, the hotel auctions issue *quotes*, indicating the 16th- (*ASK*) and 17th-highest (*BID*) prices among the currently active unit offers (Wurman *et al.*, 1998). Starting at minute four, one of the hotel auctions is selected at random to close, with the others remaining active and open for bids.

Hotel bidders are also subject to a “beat-the-quote” rule (Wurman *et al.*, 2001), requiring that any new bid offer to purchase at least one unit at a price of $ASK + 1$, and at least as many units at $ASK + 1$ as the agent was previously winning at *ASK*.

Entertainment. Agents receive an initial random allocation of entertainment tickets (indexed by type and day), which they may allocate to their own clients or sell to other agents through continuous double auctions (Friedman and Rust, 1993). The entertainment auctions issue *BID* and *ASK* quotes representing the highest outstanding buy and lowest sell offer, respectively, and remain open for buying and selling throughout the 12-minute game duration. A client may sell tickets that it does not own, but must pay a penalty of 200 per ticket for any “short sales” not covered by the end of the game.

A feasible client trip r is defined by an inflight day in_r , outflight day out_r , hotel type (H_r , which is 1 if T and 0 if S), and entertainment types (E_r , a subset of $\{1, 2, 3\}$). The value of this trip is given by

$$v(r) = 1000 - 100(|pa - in_r| + |pd - out_r|) + hp \cdot H_r + \sum_{i \in E_r} e_i. \quad (1)$$

At the end of a game instance, the TAC server calculates the optimal allocation of trips to clients for each agent, given final holdings of flights, hotels, and entertainment. The agent’s game score is its total client trip utility, minus net expenditures in the TAC auctions.

2.2 Lessons from Previous TAC Events

In designing Walverine, we had the benefit of learning from two years of observing the efforts of other TAC agent designers (Stone and Greenwald, 2003; Wellman *et al.*, 2001). We outline some of the lessons that particularly influenced our thinking about the competition.

First, agents are generally quite competent. Our initial game design embedded several key issues we thought relevant for agent strategy, and despite the lack of prior discussion, most of the entrants recognized these, and moreover discovered others we did not anticipate. The second year’s entrants explicitly built on methods disclosed after the first competition, and disclosed these advances as well. Thus, there was good reason to expect the agents to get better, and that the level of competition would be especially high in the final stage of the tournament.

Second, agents tend to improve dramatically during the course of the tournament. At the time of the preliminary rounds (qualifying and seeding), entrants are still debugging their implementations and refining their designs. Therefore, patterns observed in early games may not be strong evidence for behavior in the finals.

Third, calculating optimal allocations and marginal valuations is feasible, and important (Greenwald and Boyan, 2001; Stone *et al.*, 2001).

Finally, the hotel market is sufficiently competitive that depending on prices to be reasonable without even monitoring them is a viable strategy. This fact was evidenced most dramatically by the success of livingagents in TAC-01 (Fritschi and Dorer, 2002; Wellman *et al.*, 2001).

3 Walverine Framework

Based on these observations, we decided early in our design process to commit to the hypothesis that the TAC domain resembles a competitive economy. That is, we take as a basic presumption that the aggregate behavior of eight trading agents could be successfully approximated by a model in which each behaves according to the dictates of perfect competition. This does not mean we literally believe that the agents act as perfect competitors (they patently do not), nor that they should (indeed, Walverine tries to behave strategically). It is merely a modeling assumption chosen to balance accuracy and tractability.

One consequence of adopting a model of this sort is that we did not depend substantially on empirical data as input to our trading strategy. This is an advantage in light of the observation above that the preliminary rounds tend to differ qualitatively from the finals. Although Walverine does have some free parameters that could have benefited by tuning for performance, we deliberately resisted this approach in favor of maintaining a commitment to our analytical models.

3.1 Architecture

Walverine’s functional architecture is depicted in Figure 1. Dividing the agent into modular components facilitated the development of Walverine’s strategy, as well as its software realization, especially given the number of programmers involved (all of the coauthors). We partitioned the bidding decisions into one strategy for flight and hotel acquisition, and another for entertainment trading. Assuming the availability of only one direct API connection to the TAC server, we routed all bid messages and query results through a local proxy standing between SICS and our trading components. An optimization server answers queries about optimal packages and marginal values to both strategy components, given information about transactions, and actual and predicted prices. Although there is no direct communication between the flight/hotel and entertainment modules, information supplied by each is reflected implicitly in the results of optimization queries provided to the other.

The discussion below focuses on flight and hotel bidding, which dominates the game and exemplifies our competitive analysis approach. Walverine’s entertainment strategy takes a completely different tack, discussed in Section 7.

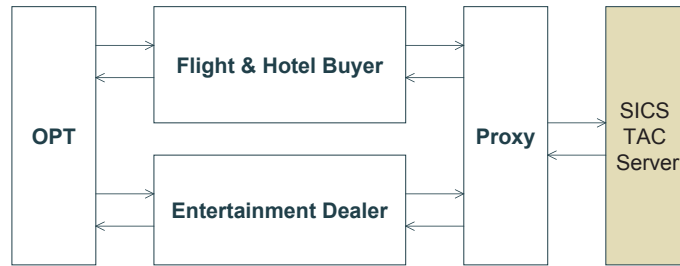


Figure 1: Walverine architecture.

3.2 Skeletal Trading Strategy

Analysis early in the design process suggested that it was not worthwhile to delay flight purchases, as the expected price increase exceeded the likely benefit of improved hotel information by the time hotels begin to close and reveal meaningful prices. Therefore, Walverine commits to flights as soon as possible, based on an assessment of expected optimal trips.

Specifically, upon game start, Walverine retrieves client information and initial flight prices. It then generates an initial prediction of hotel prices (details in Section 4), and calculates the optimal trip at these prices. It then issues bids immediately to purchase the flights for these trips. In the TAC-02 finals, Walverine purchased all 16 flights within the first four seconds of the game, on average.

After the initial flight purchases, decision making is effectively divided into discrete rounds, delimited by the release of hotel price quotes each minute, with one random hotel auction closing each minute starting at minute four. Therefore, at 3:00 and each minute mark thereafter, Walverine executes its flight/hotel bidding-round routine, which comprises the following sequence of steps:

1. Update price quotes and holdings for flights and hotels.
2. Recalculate hotel price predictions based on updated information.
3. Recalculate optimal package, and purchase any indicated flights beyond those currently held.
4. Calculate marginal values of hotel rooms.
5. Generate hotel bids based on these marginal values.

Price quotes and holding information (transactions) are retrieved directly from the TAC server. Walverine's methods for price prediction and bid generation are discussed in Sections 4 and 6, respectively. Calculating optimal packages and marginal values is the domain of our optimization server, discussed in the next section.

3.3 Optimal Packages and Marginal Values

Walverine’s formulation of the trip optimization problem takes the general form:

$$\max_{\mathbf{r}} v(\mathbf{r}) - c(\mathbf{r}, \hat{\mathbf{p}}), \quad (2)$$

where $v(\mathbf{r}) = \sum_{i=1}^8 v(r^i)$, for r^i the trip assigned to client i , with $v(r^i)$ given by (1) specialized for this client. The expression $c(\mathbf{r}, \hat{\mathbf{p}})$ captures the cost of purchasing any travel goods (flights, hotels, entertainment) required for trips \mathbf{r} , beyond the agent’s holdings, at estimated or actual prices $\hat{\mathbf{p}}$. Unavailable items (i.e., closed hotels) are considered to have an effective price of ∞ .

This optimization problem can be expressed as an integer linear program (Stone *et al.*, 2001). We formulate the model in AMPL (Fourer *et al.*, 1993), and calculate results using the CPLEX solver.² Walverine’s optimization server wraps this optimization core with an interface for setting parameters and issuing queries, communicating with the strategy components through sockets.

Queries supported by the optimization server include:

Best Package Return the optimal package of goods, given current holdings and estimated or actual prices. This addresses the *completion problem* (Boyan and Greenwald, 2001), which has come to be recognized as a core problem in TAC bidding.³

Marginal Value Calculate the marginal (incremental) value of each additional unit of available goods. The server accommodates separate queries for hotel and entertainment goods.

We describe “hedged” variants on these queries in Section 5.

Let $v^*(g, x)$ denote the value of the best package, assuming we hold x additional units of good g , and taking $\hat{\mathbf{p}}_g = \infty$. The *marginal value*⁴ of the k th unit of g is simply $v^*(g, k) - v^*(g, k - 1)$. The standard marginal-value query for hotels performs this calculation for every open hotel, $1 \leq k \leq 8$. The marginal-value query for entertainment performs it for every entertainment good, $k = 0$ and $k = 1$.

4 Price Prediction

Walverine predicts hotel prices based on a literal application of its presumption that TAC markets are competitive. Specifically, it calculates the *Walrasian competitive equilibrium* of the TAC economy, defined as the set of prices at which all markets

²<http://www.ilog.com>

³Walverine’s best-package query actually solves a special case, the *acquisition problem* (Boyan and Greenwald, 2001), as it accounts for opportunities to buy goods at (estimated) prices but neglects the possibility of selling entertainment. Note that our version employs linear prices, in contrast with the more general *priceline* approach in which the agent faces varying estimated prices depending on the number of units demanded.

⁴Several previous TAC agents employed some concept of marginal value (Aurell *et al.*, 2002; Stone and Greenwald, 2003; Stone *et al.*, 2001), although the technical definitions applied have varied substantially.

would clear, assuming all agents behave as price takers (Hildenbrand and Kirman, 1976). Since flight prices are exogenous, it is only hotel prices that may adjust to balance supply and demand. Walverine attempts to find a set of hotel prices that would support such an equilibrium, and returns these values as its prediction for the hotels' final prices.

4.1 Calculating Competitive Equilibrium

Let \mathbf{p} be a vector of hotel prices, consisting of elements $p_{h,i}$ denoting the price of hotel type $h \in \{S, T\}$ on day $i \in \{1, 2, 3, 4\}$. Let $x_{h,i}^j(\mathbf{p})$ denote agent j 's demand for hotel h , day i at these prices, with its vector of such demands written as $\mathbf{x}^j(\mathbf{p})$. The *aggregate demand* is simply the sum of agent demands, $\mathbf{x}(\mathbf{p}) = \sum_j \mathbf{x}^j(\mathbf{p})$.

Demand for a given hotel is a function of *all* hotel prices, as changing the price of any hotel can affect the agent's choice of trips, and thus the demand for any other hotel. The interconnection of markets renders this a problem in *general equilibrium* (as opposed to partial equilibrium), and prevents us from analyzing each hotel in isolation.

Note that an agent's demand also depends on flight prices as well as its clients' preferences. We leave these factors implicit in our notation since both flight prices and preferences are considered constant with respect to the equilibrium calculation. We provide full detail on our demand calculations in Section 4.2 below.

Prices \mathbf{p} constitute a *competitive equilibrium* if aggregate demand equals aggregate supply for all hotels. Since there are 16 rooms available for each hotel on each day, we have in competitive equilibrium, $\mathbf{x}(\mathbf{p}) = \mathbf{16}$.

General equilibrium theory develops technical conditions on agent preferences under which such an equilibrium can be guaranteed to exist (Hildenbrand and Kirman, 1976; Mas-Colell *et al.*, 1995). However, these conditions do not hold in the TAC environment, and indeed the TAC economy may not possess a competitive equilibrium. Reasons include the fundamental discreteness and satiability of agents' demands for hotel rooms. Nevertheless, we may still expect to find approximate equilibria (i.e., prices inducing relatively small imbalances of supply and demand), and these may serve adequately for our prediction purpose.

The classic method for determining competitive prices is the *tatonnement* protocol, an iterative price adjustment procedure originally conceived by Walras (Arrow and Hahn, 1971). Tatonnement begins with an arbitrary price vector, and revises price elements respectively up or down as there is an excess of demand or supply. This procedure is guaranteed to converge on equilibrium prices when they exist, assuming in addition that demand obeys the *gross substitutes* property (Mas-Colell *et al.*, 1995). In the TAC domain, however, preferences for hotel rooms exhibit strong complementarities. This represents a patent violation of gross substitutes, as raising the price for one hotel can easily decrease demand for another, for example in the case of two hotels of the same type on adjacent days.

Notwithstanding these theoretical impediments, Walverine searches for a competitive equilibrium using tatonnement. Starting from an initial guess \mathbf{p}^0 , it iteratively computes a revised price vector according to:

$$\mathbf{p}^{t+1} = \mathbf{p}^t + \alpha^t (\mathbf{x}(\mathbf{p}^t) - \mathbf{16}). \quad (3)$$

We experimented with several schemes for varying the adjustment rate α , settling on an exponential decay. The process tended to converge quickly on an approximate equilibrium, with no detectable sensitivity to particular parameter choices. The version employed in the TAC-02 tournament happened to set $\alpha^t = 0.387 \cdot 0.95^t$. We ran tatonnement for 300 iterations, although the bulk of the adjustment generally occurred within the first 10% of that.

4.2 Calculating Expected Demand

A central part of the tatonnement update (3) is determination of demand as a function of prices. This is straightforward if client preferences are known, as it corresponds essentially to an instance of the best-package query described in Section 3.3.⁵ Whereas we do know the preferences of our own eight clients, we have no direct knowledge about the 56 clients assigned to the other seven agents.

Therefore, we partition the demand problem into a component from Walverine (\mathbf{x}_w), and one from the other agents:

$$\mathbf{x}(\mathbf{p}) = \mathbf{x}_w(\mathbf{p}) + \mathbf{x}_{\bar{w}}(\mathbf{p}).$$

We calculate $\mathbf{x}_w(\mathbf{p})$ using a simplified version of the best-package query (ignoring entertainment holdings). In place of $\mathbf{x}_{\bar{w}}(\mathbf{p})$, we attempt to estimate its expectation, exploiting our knowledge of the distribution from which client preferences are drawn. If agent demand is separable by client,

$$E[\mathbf{x}_{\bar{w}}(\mathbf{p})] = E\left[\sum_{i=1}^{56} \mathbf{x}_{client_i}(\mathbf{p})\right]. \quad (4)$$

Since client preferences are i.i.d.,

$$E[\mathbf{x}_{\bar{w}}(\mathbf{p})] = 56 \cdot E[\mathbf{x}_{client}(\mathbf{p})].$$

At the beginning of the game when there are no holdings of flights and hotels, the agent optimization problem is indeed separable by client, and so (4) is justified. At interim points when agents hold goods, the demand optimization problem is no longer separable. However, since we are ignorant about the holdings of other agents, we have no particular basis on which to determine *how* (4) is violated, and so we adopt it as an approximation.

It remains to derive a value for $E[\mathbf{x}_{client}(\mathbf{p})]$. Our solution follows directly from the distribution of clients. Preferred arrival and departure days (pa, pd) are drawn uniformly from the ten possible arrival/departure pairs:

$$E[\mathbf{x}_{client}(\mathbf{p})] = 0.1 \sum_{(pa, pd)} E[\mathbf{x}_{pa, pd}(\mathbf{p})]. \quad (5)$$

For a *given* (pa, pd), the only remaining uncertainty surrounds the hotel premium hp . We observe that the optimal choice of travel days is independent of hp , *conditional*

⁵Indeed, we originally validated our prediction concept by applying it to data from the TAC-01 finals, taking the client preferences as given.

on hotel choice. Let $r^*(pa, pd, h)$ denote the optimal trip for the specified day preferences, conditional on staying in hotel h (T or S). We can calculate this trip by taking into account the flight prices, prices for hotel h , day deviation penalties, and expected entertainment bonus (see Section 4.3). Note that the optimal trip for preferences (pa, pd) must be either $r^*(pa, pd, T)$ or $r^*(pa, pd, S)$. Let σ_h denote the net valuation of $r^*(pa, pd, h)$, based on the factors above but *not* accounting for hp .

Hotel premiums are also drawn uniformly, $hp \sim U[50, 150]$. Since T and S differ only in the bonus hp , we can determine the choice based on the relation of σ_T and σ_S :

$$h = \begin{cases} S & \text{if } \sigma_S - \sigma_T \geq 150 \\ T & \text{if } \sigma_S - \sigma_T \leq 50 \end{cases}$$

If instead $50 < \sigma_S - \sigma_T < 150$, then the choice of hotel depends on the actual hp . The uniform distribution of hp entails that the *probability* of S being the optimal choice is

$$\Pr(h = S) = \frac{\sigma_S - \sigma_T - 50}{100}.$$

Given the choice of trip days and hotel, the demand for this case is established. We aggregate these cases (weighting by probability of hotel choice if applicable) using (5), yielding the overall demand per client. Multiplying by 56 gives us $E[x_{\bar{w}}(\mathbf{p})]$, and combining with our own demand, finally, the overall expected demand estimate.

4.3 Expected Entertainment Surplus

The derivation above deferred detailed explication of our accounting for entertainment bonuses in evaluating alternative trips. We employ estimates of net entertainment contribution as a function of arrival and departure days. Our analysis is based on the distribution of client entertainment preferences, along with the empirical observation (reported by the *livingagents* team (Fritschi and Dorer, 2002)) that entertainment tickets tend to trade at a price near 80. We verified that this indeed obtained during the TAC-01 finals, and refined the estimate by distinguishing the entertainment tickets on congested days 2 and 3 (average price 85.49), from tickets on less congested days 1 and 4 (average price 76.35). Our analysis proceeds by assuming that agents can buy or sell any desired quantity at these prices.

Consider a client staying for n days, with given entertainment values. Its maximal entertainment surplus would be obtained by allocating its most valuable ticket to the cheapest day of its trip if profitable (that is, if the entertainment value exceeds the average price for that day), then if $n \geq 2$, its second most valuable to the next cheapest day, and finally, if $n \geq 3$, its least valuable to a remaining day.

Let x_i denote the cost of the i th least expensive day for the given trip $1 \leq i \leq \min(3, n)$. The expected entertainment surplus of the trip, then, is given by

$$\sum_{i=1}^{\min(3, n)} EV_i(x_i), \quad (6)$$

where $EV_i(x)$ denotes the expected value of allocating the i th most valuable ticket to a day costing x . Three ticket values are drawn independently from a uniform distribution.

Arrive:Depart	Expected Entertainment Surplus	
	TAC-01 Prices	TAC-02 Prices
1:2, 4:5	74.7	78.1
1:3, 3:5	101.5	112.2
1:4, 2:5	106.9	120.1
1:5	112.7	121.0
2:3, 3:4	66.2	76.8
2:4	93.0	110.9

Table 1: Expected contributions from entertainment, based on prices from TAC-01 and TAC-02 finals, respectively. Wolverine employs these summary values in its demand calculations.

Given m i.i.d. draws $\sim U[a, b]$, the i th greatest is less than z with probability

$$F_{i,m}^{[a,b]}(z) = \sum_{j=0}^{i-1} \binom{m}{j} \left(\frac{z-a}{b-a} \right)^{m-j} \left(\frac{b-z}{b-a} \right)^j.$$

The expectation of this i th order statistic $Z_{i,m}^{[a,b]}$ is given by

$$E \left[Z_{i,m}^{[a,b]} \right] = a + (b-a) \left(1 - \frac{i}{m+1} \right).$$

We need to determine the expected value of the i th ticket, net of its cost x . The expected surplus of the i th order statistic with respect to x , given that x is between the j th and $(j+1)$ st order statistic ($i \leq j$) is

$$E \left[Z_{i,m}^{[a,b]} - x \mid Z_{j+1,m}^{[a,b]} \leq x < Z_{j,m}^{[a,b]} \right] = E \left[Z_{i,j}^{[x,b]} - x \right]. \quad (7)$$

The probability of the condition in (7) is

$$\Pr(Z_{j+1,m}^{[a,b]} \leq x < Z_{j,m}^{[a,b]}) = F_{j+1,m}^{[a,b]}(x) - F_{j,m}^{[a,b]}(x).$$

Using these expressions, we can sum over the possible positions of x with respect to the order statistics (positions in which value minus cost is positive) to find the expected value of allocating the i th best ticket to a day costing x :

$$\begin{aligned} EV_i(x) &= E \left[\max \left(0, Z_{i,3}^{[0,200]} - x \right) \right] \\ &= \sum_{j=i}^3 \left(E \left[Z_{i,j}^{[x,200]} \right] - x \right) \left(F_{j+1,3}^{[0,200]}(x) - F_{j,3}^{[0,200]}(x) \right), \end{aligned}$$

providing the value we need to evaluate the expected entertainment surplus for a trip (6).

The results of these calculations for each of the ten possible trips are presented in Table 1. In the 2002 competition, Wolverine used the average prices from the TAC-01 finals. It turned out that the entertainment prices observed in the TAC-02 finals were somewhat lower (averaging 74.18 on days 2 and 3, 72.79 on days 1 and 4), thus supporting greater entertainment surplus.

4.4 Interim Price Prediction

The description above covers Walverine’s procedure for initial price prediction. Once the game is underway, there are several additional factors to consider.

- Agents already hold flight and hotel goods.
- Flight prices have changed.
- Hotel auctions have issued price quotes, providing a source of information about actual demand.
- Some hotel auctions are closed, precluding further acquisition of these rooms.

Walverine adopts a fairly minimal adjustment of its basic (initial) price prediction method to address these factors. It continues to employ initial flight prices in best-trip calculations, and ignores its own flight and hotel holdings in calculating own demand for open hotels. For closed hotels, it does fix its demand at actual holdings. Since we do not know the holdings of other agents, we make no attempt to account for this in estimating their demand. This applies even to closed hotels—in the absence of information about their allocation, Walverine’s tatonnement calculations attempt to balance supply and demand for these as well.

Given price quotes, we modify the price-adjustment process to employ *ASK* (or final price of closed auctions) as a lower-bound price for each hotel. This constraint is enforced within each iteration of the tatonnement update (3).

4.5 Prediction Quality

After the TAC-02 finals, we undertook a comprehensive comparative study of price-prediction methods employed by the participating agents. The results of this study are presented in a separate report (Wellman *et al.*, 2002). Among other conclusions, the investigation indicates that Walverine’s equilibrium method produced initial price predictions more accurate than those of any other TAC-02 agent.

On the other hand, we did not systematically evaluate the quality of interim price predictions, and suspect that Walverine has considerable room for improvement there. In principle, price quotes provide significant evidential value regarding uncertain demand, and Walverine fails to exploit this information directly in these terms.

5 Price Hedging

Walverine’s equilibrium analysis results in a point price prediction for each hotel auction. In reality, prices are inherently uncertain, and thus decisions about bidding and trip choice should take into account the potential deviations from any point estimate. Some agents, such as *ATTac* (Schapire *et al.*, 2002) and *RoxyBot* (Greenwald, 2003b), explicitly generate and use predictions in the form of distributions over prices. Others, including Walverine, generate point predictions but then make decisions with respect to distributions around those estimates.

The greatest source of risk stems from the possibility that a hotel’s price might greatly exceed the estimate, causing the agent to pay a painfully high price or fail to obtain its room(s). Thus, Walverine assigns a small *outlier probability*, π , to the event that a given hotel will reach an unanticipated high price. In the event the hotel is an outlier, we take its price to be $\max(2\hat{p}, 400)$, where \hat{p} is the estimated price of the hotel if it is not an outlier (i.e., according to the equilibrium price-prediction procedure, described above). Walverine’s overall price distribution is thus defined by a set of disjoint events with exactly one outlier, at probability π for each of the open hotel auctions, and the residual probability for the event of no outliers.

We apply this price distribution model in our initial calculation of optimal trips, on which we base our starting flight purchases. The resulting choice *hedges* for the potential that some price will deviate significantly from our baseline prediction. The typical effect of our hedging method is to reduce the duration of some trips, thus decreasing Walverine’s exposure to hotel price risk.

Walverine’s optimizer employs this same outlier model in computing responses to its *hedged marginal value* query. A hedged marginal value is simply a weighted average of marginal values, where v^* is calculated with respect to each outlier event (as well as the no-outlier event), with the results weighted according to the outlier probability π . Because it involves repeated optimization with respect to a variety of price and quantity combinations, hedged marginal value is the most computationally intensive operation performed by Walverine.

In the TAC-02 tournament, Walverine employed the setting $\pi = 0.06$. This is one of the few free parameters in its flight/hotel strategy, along with the outlier price expression itself. Although we made no systematic effort to tune this parameter, we did verify empirically that $\pi = 0.06$ significantly outperformed $\pi = 0$ (improvement was on the order of 300 points per game).

6 Optimal Bidding

An agent behaving competitively would bid in hotel auctions by offering to buy units at their marginal values. Again assuming separable clients, this means that each agent will submit an offer for a unit at marginal value for each hotel and each client. Under price uncertainty, the bidding decision problem is more complicated (Greenwald, 2003b), but a competitive agent would still not take into account its own effect on prices.

Walverine assumes that other agents bid competitively and itself bids strategically by calculating an optimal set of bids taking into account its own effect on hotel prices. This amounts to placing bids that maximize our expected surplus given a distribution from which other bids in the auction are drawn.

6.1 Generating Bid Distributions

As for our price-prediction algorithm, we model the seven other agents as 56 individual clients, again using the zero-holdings assumption to render the computation tractable. Our approach is to generate a distribution of marginal valuations assuming each of the

(pa, pd) pairs, and sum over the ten cases to generate an overall distribution val for the representative client.

For a given (pa, pd) pair, we estimate the value of a given room (h, i) as the difference in expected net valuation between the best trip assuming room (h, i) is free, and the best trip of the alternative hotel type h' . In other words, the value of a given room is estimated to be the price above which the client would prefer to switch to their best trip using the alternate hotel type.

Setting the price of (h, i) to zero and that of all other hotels to predicted prices, we calculate best packages $r^*(pa, pd, h)$, $r^*(pa, pd, h')$ and their associated net valuations σ_h and $\sigma_{h'}$ as in Section 4.2. If $(h, i) \notin r^*(pa, pd, h)$, we say that val_h is zero,⁶ otherwise it is the expected difference in net valuations:

$$\begin{aligned} val_S &= \max(0, \sigma_S - \sigma_T - hp), \\ val_T &= \max(0, \sigma_T - \sigma_S + hp). \end{aligned}$$

Since $hp \sim U[50, 150]$, these expressions represent uniform random variables:

$$\begin{aligned} \sigma_S - \sigma_T - hp &\sim U[\sigma_S - \sigma_T - 150, \sigma_S - \sigma_T - 50], \\ \sigma_T - \sigma_S + hp &\sim U[\sigma_T - \sigma_S + 50, \sigma_T - \sigma_S + 150]. \end{aligned} \quad (8)$$

For each (pa, pd) we can thus construct a cumulative distribution $Val_{pa, pd}$ representing the marginal valuation of a given hotel room. In general, $val_{pa, pd}$ will include a mass at zero, representing the case where the room is not used even if free. Thus, we have

$$Val_{pa, pd}(x) = \begin{cases} 0 & \text{if } x < \max(0, \alpha) \\ \frac{x - \alpha}{\beta - \alpha} & \text{if } \max(0, \alpha) \leq x \leq \beta \\ 1 & \text{if } x \geq \beta \end{cases}$$

where α and β are the lower and upper bounds, respectively, of the corresponding uniform distribution of (8).

The overall valuation distribution for a representative client is the sum over arrival/departure preferences,

$$Val(x) = \frac{1}{10} \sum_{(pa, pd)} Val_{pa, pd}(x).$$

Finally, it will also prove useful to define a valuation distribution conditional on exceeding a given value q . For $x \geq q$,

$$Val(x | q) = \frac{Val(x) - Val(q)}{1 - Val(q)}. \quad (9)$$

6.2 Computing Optimal Bids

After estimating a bid distribution, Walverine derives an optimal set of bids with respect to this distribution. Our calculation makes use of an order statistic, $Val_{k, n}(x)$, which

⁶As for σ , we omit the arguments for pa, pd , and day i where these are apparent from context.

represents the probability that a given value x would be k th highest if inserted into a set of n independent draws from Val .

$$Val_{k,n}(x) = [1 - Val(x)]^{k-1} Val(x)^{n-k+1} \binom{n}{k-1}$$

We can also define the conditional order statistic, $Val_{k,n}(x|q)$, by substituting the conditional valuation distribution (9) for Val in the definition above.

Once hotel auctions start issuing price quotes, we have additional information about the distribution of bids. If H is the hypothetical quantity won for Walverine at the time of the last issued quote, the current ASK tells us that there are $16 - H$ bids from other clients at or above ASK, and $56 - (16 - H) = 40 + H$ at or below (assuming a bid from every client, including zero bids). We therefore define another order statistic, B_k , corresponding to the k th highest bid, sampling $16 - H$ bids from $Val(\cdot | ASK)$ as defined by (9), and $40 + H$ bids from Val .

Note that our order statistics are defined in terms of other agents' bids, but we are generally interested in the k th highest value in an auction overall. Let n_b be the number of our bids in the auction greater than b . We define B_k so as to include our own bids, and employ the $(k - n_b)$ th order statistic on others, $Val_{k-n_b,n}(b)$, in calculating B_k .

Given our definitions, the probability that a bid b will be the k th highest is the following:

$$B_k(b) = \sum_{i=0}^{k-n_b-1} Val_{i,16-H}(b) \cdot Val_{k-n_b-i,40+H}(b | ASK). \quad (10)$$

We characterize the expected value of submitting a bid at price b as a combination of the following statistics, all defined in terms of B_k .

- $B_{16}(b)$: Probability that b will win and set the price.
- $B_{15}^+ \equiv \sum_{i=1}^{15} B_i(b)$: Probability that b will win but not set the price
- $M_{15} \equiv \{x \mid \sum_{i=1}^{15} P_i(x) = .5\}$: Median price if we submit an offer b .
- $M_{16} \equiv \{x \mid \sum_{i=1}^{16} P_i(x) = .5\}$: Median price if we do not bid.

Before proceeding, we assess the quality of our model, by computing the probability that the 16th bid would be above the quote given our distributions. If this probability is sufficiently low: $(B_{16}^+(ASK) < .02)$ then we deem our model of other agents' bidding to be invalid and we revert to our most conservative bid: our own marginal value.

If the conditional bid distribution passes our test, based on these statistics we can evaluate the expected utility EU of a candidate bid for a given unit, taking into consideration the marginal value MV of the unit to Walverine, and the number of units n_b of this good for which we have bids greater than b . Expected utility of a bid also reflects the expected price that will be paid for the unit, as well as the expected effect the bid will have on the price paid for all our higher bids in this auction. Lacking an

expression for expected prices conditional on bidding, we employ as an approximation the median price statistics, M_{15} and M_{16} , defined above.⁷

$$EU(b) = B_{16}(b) [(MV - b) - n_b(b - M_{16})] + B_{15}^+(b) [(MV - M_{15}) - n_b(M_{15} - M_{16})]$$

Walverine's proposed offer for this unit is the bid value maximizing expected utility,

$$b^* = \arg \max_b EU(b). \quad (11)$$

6.3 Beat-the-Quote Adjustments

Upon calculating desired offer prices for all units of a given hotel, Walverine assembles them into an overall bid vector for the auction, taking the beat-the-quote rule (BTQ) into consideration. BTQ dictates that if the hypothetical quantity won for an agent's current bid is H , any replacement bid for that auction must represent an offer to buy at least H units at a price at least $ASK + 1$. For example, suppose the current bid offers to pay (200, 150, 50) for three units, respectively, of a given hotel room. If $ASK = 100$, then the agent is winning its first two units (i.e., $H = 2$). To satisfy BTQ, the agent's new bid must be at least (101, 101).

Let $b = (b_1, \dots, b_8)$ be the agent's current bid for the eight potentially valuable units in this auction ($b_i = 0$ corresponds to no offer for that unit), and let b' be the proposed new bid, derived according to the optimization procedure above (11). To ensure satisfaction of BTQ, the agent could submit the modified bid

$$b'' = (\max(b'_1, ASK + 1), \dots, \max(b'_H, ASK + 1), b'_{H+1}, \dots, b_8).$$

However, this may not be a wise solution. Consider $b = (200, 150, 50, 0, \dots)$ as in the example above, but with $ASK = 150$ (equal to the agent's lowest winning bid), and desired new bid $b' = (500, 0, \dots)$. In this situation, the agent would like to revise upward its offer for the first unit, but would prefer that its offer of 150 for the second unit were outbid by another agent. Considering that other agents also follow BTQ, there will likely be several new bids at a price of $ASK + 1$ in the next round of bidding, meaning that an unrevised bid of 150 stands a much better chance of being outbid than does a revised bid of 151. In this case, the agent must balance the desirability of revising its bid for the first unit against its aversion to increasing its offer for the second.

Walverine decides whether to revise its bid based on a crude comparison of these factors. It assesses the value of bidding in terms of the magnitude of its desired price changes that are allowed by BTQ, and the cost of bidding in terms of the amount by which BTQ requires bidding above actual value. If this latter value exceeds the former, or a constant threshold, then Walverine refrains from submitting a revised bid. Otherwise it submits b'' .

⁷Offline analysis using Monte Carlo simulation verified that the approximation is reasonable.

6.4 Analysis

Using game data from the TAC-02 finals, we tested the utility of our bidding algorithm as well as the accuracy of the bid distributions.

Informal analysis reveals that our distributions systematically underestimate the actual values of the bids. It appears that the distributions are fairly accurate during the initial stages of the game, when our modeling assumptions hold (zero holdings, all auctions open). The deterioration in accuracy of our distributions is not a fatal problem, as our algorithm reverts to bidding marginal values when the observed price quote is judged too unlikely with respect to our estimates. Of course, more accurate distributions would enable more effective bid optimization.

Toward this end we devised an alternative bid estimation scheme, intended to correct for some of the invalid simplifying assumptions underlying our original method.⁸ It turned out that, for the TAC-02 finals at least, the alternative distributions more closely resemble the actual distribution of bids.

To test whether the more accurate estimates actually support improved bidding, we devised a measure based on past data that determines the effectiveness of a set of bids with respect to predicted prices and other agents' actual bids. For each open hotel in each bidding round, we calculated our winnings in that auction based on various bidding strategies and the actual bids placed by other agents. We then scored each hypothetical outcome under the assumption that rooms in other open hotels were available at predicted prices.

We used this method to score bids over 256 closings (32 games times eight bidding rounds) from the TAC-02 finals, generating 1152 data points (4.5 open hotels in average bidding round). Surprisingly, bidding based on our original, nominally less accurate distributions produced superior results, to both marginal-value bidding and bidding based on the more accurate new distribution. However, a mean-difference test did not reveal the differences to be statistically significant. Future work will further test and refine our model of other agents' bid distributions.

7 Entertainment Trading

Walverine's approach to entertainment trading can be considered a polar opposite of the competitive analysis approach it takes to flight and hotel buying. Equilibrium analysis has little to say about the dynamics of prices produced through continuous auctions, yet these transient behaviors seem particularly salient for effective entertainment trading. Thus, for this domain, we employ no model of the market, and no explicit calculations of the expected outcomes of alternative bid choices. Instead, Walverine adopts a *model-free*, empirical approach called *Q-learning*—a variety of *reinforcement learning* (Sutton and Barto, 1998).

⁸We forgo detailed specification, as this redesign effort is still in flux.

7.1 Learning Framework

The idea of applying Q-learning to TAC strategies was proposed by Boadway and Precup (2001), and employed in their TAC-01 entry. This agent attempted to learn a policy for the entire TAC game, but this proved too ambitious given the time available for development and training. Inspired by their example, we sought to pursue this approach for the much more limited TAC task of entertainment trading.

The aim of Q-learning is to estimate a function $Q : S \times A \rightarrow \mathfrak{R}$, representing the value of taking a given action in a given state. Value is typically measured by (discounted) cumulative future rewards, and the function can be represented in tabular or implicit form. From the Q function one can derive an optimal policy, namely that performing the maximally valued action in any given state. The recurrence (Bellman) equation relating values of adjacent states provides the basis for updating Q from experience of taking actions and observing state transitions and rewards.

Walverine’s entertainment component considers each auction independently. We approximate the state of an entertainment auction as the settings of six parameters: BID , ASK , number of tickets held, marginal value of first unit (MV_1), marginal value of zeroth unit (MV_0), and game time. To keep the state space manageable, we discretized these dimensions into value sets of size 6, 6, 3, 7, 7, and 3, respectively. Marginal values provided by the optimizer summarized client preferences and provided the necessary link to our flight/hotel module.

The reward from entertainment has two components: cash flow from trading and the entertainment bonus accrued to clients at the end of the game.

In each entertainment auction, Walverine maintains an offer to buy one unit, and an offer to sell one unit (at a higher price, of course). Rather than take the offer prices as actions, however, we define the action space in terms of *offsets from marginal value*. That is, the action $\text{buy}(x)$ means to revise its current unit buy offer to the price $MV_1 - x$. Similarly, $\text{sell}(x)$ corresponds to a sell offer at $MV_0 + x$. We defined eight discrete offset values. However, rather than consider all 64 buy/sell combinations, Walverine alternates between buy and sell decisions, considering only the eight available options for each case.

7.2 Learning Results

Our learning procedure encodes Q as a table. Walverine maintained two tables: one for entertainment events on days {1,4}, and the other for days {2,3}. Within each category (six auctions apiece), the learning agent shared its trading experience. Given the size of each table (6291 states and 16 actions⁹), Walverine required a great deal of training experience. We ran the Q-learning algorithm over data gathered from 14,839 games, including matches against other TAC participants during preliminary rounds, as well as many instances of self-play. Walverine employed a variety of entertainment trading policies while gathering experience, including a hard-coded strategy based on that reportedly employed by livingagents in TAC-01 (Fritschi and Dorer, 2002). Once

⁹There are 15,876 distinct combinations of state variables, but many of these do not represent legal states. In all of its training, Walverine actually experienced 2588 and 2602 states, respectively, in the two auction categories.

we had accumulated sufficient data, we ran some instances of Walverine based on preliminary learned policies, with various exploration-exploitation control methods.

Figure 2 displays a learning curve representing the evolution of Walverine’s entertainment performance during the training period. We took as a baseline the value of the null (no-trading) strategy, which we determined experimentally to provide an entertainment reward (through retaining endowed tickets) of 1019 on average. As a second baseline, we evaluated the performance of the aforementioned `livingagents` entertainment strategy, embedded in Walverine. The performance axis of Figure 2 measures Walverine’s learned entertainment strategy compared to this second baseline. In each interval of training games represented, we evaluate the policy learned based on games prior to that interval (thus the first interval represents the no-trading baseline). The evaluation consists of self-play games with half the agents following the learned entertainment policy and the other half following the `livingagents` entertainment strategy. By the time of the TAC-02 finals, we had reached within 50 points of the hand-coded strategy.

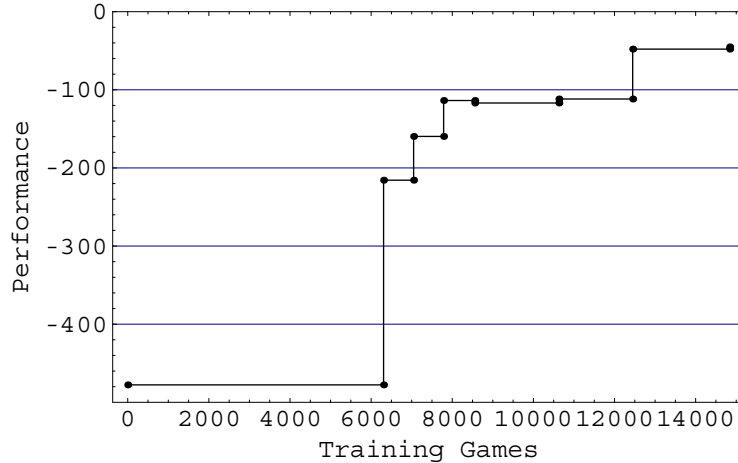


Figure 2: Entertainment learning curve.

It is important to note that Walverine itself underwent many changes during the learning process, which undoubtedly confounds the results. Moreover, the policies evaluated in Figure 2 retain an exploration element, except the last interval, which is pure exploitation.

In the TAC-02 finals, Walverine averaged an entertainment reward of 1409, nearly 400 over the non-bidding baseline. A summary of entertainment performance by agent is included in Table 2. Interestingly, `whitebear`, the high scorer in TAC-02, was extremely successful on entertainment, achieving an average reward of 1623. Although Vetsikas and Selman (2003) report employing a simple entertainment strategy, the high payoff achieved suggests that there may be room for improvement through further learning in this environment.

Agent	Final score	CP Adj	Ent
whitebear	3413	+66	1623
SouthamptonTAC	3385	-48	1464
Thalis	3246	-36	1393
UMBCTAC	3236	+55	1327
Walverine	3210	+67	1409
livingagents ¹⁰	3181	-20	1362
kavayaH	3099	-60	1460
cuhk	3069	-24	1452

Table 2: Scores during the finals. Each agent played 32 games. The second column represents our calculated adjustment due to client preference assignments. The third column presents the entertainment component of agent scores.

8 TAC-02 Agent Performance

8.1 Tournament Results

Average scores for the eight agents that played in the final round are posted in Table 2. See <http://www.sics.se/tac> for a list of participant affiliations and team leaders, as well as results from preliminary and semifinal rounds. Complete game logs are available, as for the previous TAC events.

Performance in the tournament is one relevant measure of agent quality, although we agree with those who have cautioned against focusing excessively on ranked results in the context of research competitions (Stone, 2002). One interesting question is how to measure progress over time. The two top-scoring agents in TAC-01, *livingagents* and *ATTac-2001*, participated with essentially unchanged agents in TAC-02. As noted above, *livingagents* did quite well, assuming we ignore the bug that caused it to skip two games. *ATTac* was top scorer in the TAC-02 seeding rounds, but then was eliminated in the semifinals. One possible explanation is that prices during the preliminary rounds in 2002 (which *ATTac* uses as training data) were not sufficiently representative of the final rounds. However, we believe it is also reflective of a general increase in competence of the other agents in the field.

The two top-scoring agents in TAC-02, *whitebear* and *SouthamptonTAC* (He and Jennings, 2002), also contended in TAC-01. These agents reportedly evolved from their 2001 designs, improved through adopting refined classifications of game environments (He and Jennings, 2003), and through extensive experimentation and parameter tuning (Vetsikas and Selman, 2003).

¹⁰The score of *livingagents* was adversely affected by missing two games. Discounting these would have led to an average score of 3393.

Round	Global	TAC Market	TAC (%)	Uniform H+E (%)	Endowed E (%)
Qualify	618	415	67.0	95.2	85.4
Seeding	618	470	75.7	95.2	85.4
Semi-Final	608	534	87.7	95.2	85.5
Final	609	542	89.1	94.6	85.0

Table 3: The efficiency of the TAC market compared to the global optimum. Global optimization with uniform allocation of hotel and entertainment to agents is reported as a benchmark, as is a second benchmark with uniform hotel allocation and no entertainment trading.

8.2 TAC Market Efficiency

Another gauge of agent effectiveness is how well they allocate travel goods, *in the aggregate*, through their market interactions. We can measure this by comparing actual TAC market allocations with ideal global allocations. Consider the total group of 64 clients, and the set of available resources: 16 hotel rooms of each type per day, plus 8 entertainment tickets of each type per day. The global optimizer calculates the allocation of resources maximizing total client utility, net of expenditures on flights assuming they are available at their initial prices. Note that this optimization neglects hotel and entertainment prices, as these are endogenous to the TAC market. The average achievable net utility, per client, in the various rounds of the TAC tournament as determined by global optimization is reported under the heading “Global” in Table 3. Average net utility achieved in the actual TAC games (also neglecting hotel and entertainment expenditures, but counting actual payments for flights) is reported under “TAC Market”.

As seen in the table, we found that the TAC market achieved 89% of the optimal value, on average, over the 32 games of the TAC-02 finals. There was a steady improvement from the qualifying round (67% optimal), seeding round (76%), and semifinals (88%). It is difficult to assess this effectiveness in absolute terms, so we provide a couple of benchmarks for comparison. In “Uniform H+E”, we distribute the hotel rooms and entertainment evenly across the eight agents, then optimize each agent’s allocation to clients. This approach yields 95% of the globally optimal value. The relative value drops to around 85% if we distribute only the hotels, leaving agents with their original endowment of entertainment. It is perhaps surprising that simply dividing the goods uniformly achieves such a high fraction of the available surplus—better than the market if entertainment is included in the distribution.

One reason that the uniform distribution is relatively so effective is that the agents are *ex ante* symmetric, with i.i.d. clients. Potential gains from trade are thus not so great for hotels. Second, a direct allocation avoids the significant obstacles posed to agents pursuing their allotments individually through the market. Agents face substantial risk (price uncertainty, exposure due to complementarities, unknown hotel closing patterns), and this necessarily entails some loss in expected allocation quality. For example, the set of available hotels is sufficient to obtain trips for all clients (albeit shortened from desired lengths), and given a definite allocation the agent can optimize for its clients accordingly. With uncertainty, the agents may plan for longer trips than

are jointly feasible, and thus wind up wasting flights, hoarding hotel rooms (to hedge), or resorting to suboptimal fallback trip options. In future work, we will investigate in greater depth the various sources of misallocation in TAC play.

9 Conclusions

The hallmark of Walverine’s approach is its basis in competitive analysis of the TAC travel economy. Walverine displays this characteristic most directly in its use of Walrasian competitive equilibrium to predict hotel prices, and its method for optimal bidding, which relies on the competitive property in its model of other agent’s bids. The agent’s hotel and flight bidding strategy is decision-analytic to the core, as every action is based on an explicit optimization with respect to its model assumptions.

As designers, we avoided empirical parameter tuning, except in the case of entertainment bidding, where we ceded *all* discretion to an automatic learning procedure. (Admittedly, we exercised subjective judgment in formulating the learning problem, inevitably introducing some bias.) Our aim is to enable sharper evaluation of our fundamental hypothesis regarding the utility of competitive analysis.

Although the results cannot be definitive, we regard Walverine’s TAC-02 experience as broad validation of its underlying approach. More focused studies, for example on initial price prediction (Wellman *et al.*, 2002), are need to evaluate specific components of the agent’s strategy. There is clearly room for improvement; in particular we have identified interim price prediction and modeling for optimal bidding as areas where Walverine fails to exploit available information. We intend to pursue such topics in preparing for future competitions in the TAC series.

Planning for TAC-03 is well underway. The 2003 tournament will include a division devoted to the travel-shopping game described here (now dubbed “TAC Classic”), and will also introduce a new game involving trading multi-faceted goods in a supply-chain context. This new game will introduce several interesting strategic issues not emphasized in TAC Classic. We hope that many of the agent researchers interested in trading domains will participate in one or both of these games, and find it—as we have—a stimulating and fertile environment for developing and evaluating novel trading-agent techniques.

10 Acknowledgments

The Michigan TAC team would like to congratulate and thank SICS (in particular Erik Aurell, Joakim Ericsson, and Niclas Finne) for operating a successful TAC-02 event. Our effort in reinforcement learning benefited from the advice of Satinder Singh Baveja. We are also grateful to TAC agents past and present, who provided many useful lessons for Walverine. This research was supported in part by NSF grant IIS-9988715.

References

- Arrow, K. J. and Hahn, F. H. (1971). *General Competitive Analysis*. Holden-Day, San Francisco.
- 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*, Aix-en-Provence.
- Boadway, J. and Precup, D. (2001). Reinforcement learning applied to a multiagent system. Presentation at TAC Workshop.
- Boyan, J. and Greenwald, A. (2001). Bid determination in simultaneous auctions: An agent architecture. In *Third ACM Conference on Electronic Commerce*, pages 210–212, Tampa, FL.
- Fourer, R., Gay, D. M., and Kernighan, B. W. (1993). *AMPL: A Modeling Language for Mathematical Programming*. Boyd & Fraser.
- Friedman, D. and Rust, J., editors (1993). *The Double Auction Market*. Addison-Wesley.
- 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*, Bologna.
- Greenwald, A. (2003a). The 2002 trading agent competition: An overview of agent strategies. *AI Magazine*.
- Greenwald, A. (2003b). Bidding under uncertainty in simultaneous auctions. Technical report, Brown University.
- Greenwald, A. and Boyan, J. (2001). Bidding algorithms for simultaneous auctions: A case study. In *Third ACM Conference on Electronic Commerce*, pages 115–124, Tampa, FL.
- He, M. and Jennings, N. R. (2002). SouthamptonTAC: Designing a successful trading agent. In *Fifteenth European Conference on Artificial Intelligence*, pages 8–12, Lyon.
- He, M. and Jennings, N. R. (2003). SouthamptonTAC: An adaptive autonomous trading agent. *ACM Transactions on Internet Technology*.
- Hildenbrand, W. and Kirman, A. P. (1976). *Introduction to Equilibrium Analysis: Variations on Themes by Edgeworth and Walras*. North-Holland Publishing Company, Amsterdam.
- Mas-Colell, A., Whinston, M. D., and Green, J. R. (1995). *Microeconomic Theory*. Oxford University Press, New York.

- 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. (2002). Multiagent competitions and research: Lessons from RoboCup and TAC. In *Sixth RoboCup International Symposium*, Fukuoka, Japan.
- Stone, P. and Greenwald, A. (2003). 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.
- Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning*. MIT Press.
- Vetsikas, I. A. and Selman, B. (2003). A principled study of the design tradeoffs for autonomous trading agents. In *Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne.
- Walras, L. (1954). *Elements of Pure Economics*. Allen and Unwin. English translation by William Jaffé, originally published in 1874.
- 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.
- Wellman, M. P., Reeves, D. M., Lochner, K. M., and Vorobeychik, Y. (2002). Price prediction in a trading agent competition. Technical report, University of Michigan.
- Wellman, M. P., Greenwald, A., Stone, P., and Wurman, P. R. (2003). The 2001 trading agent competition. *Electronic Markets*.
- Wurman, P. R., Walsh, W. E., and Wellman, M. P. (1998). Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, **24**, 17–27.
- Wurman, P. R., Wellman, M. P., and Walsh, W. E. (2001). A parametrization of the auction design space. *Games and Economic Behavior*, **35**, 304–338.