

Learning Improved Entertainment Trading Strategies for the TAC Travel Game

L. Julian Schwartzman and Michael P. Wellman

University of Michigan
Computer Science & Engineering
2260 Hayward St
Ann Arbor, MI 48109-2121 USA

Abstract

For almost five years we have continually operated a simulation testbed exploring a variety of strategies for the TAC Travel game. Building on techniques developed in our recent study of continuous double auctions, we performed an equilibrium analysis of our testbed data, and employed reinforcement learning in the equilibrium environment to derive a new entertainment strategy for this domain. A second iteration of this process led to further improvements. We thus demonstrate that interleaving empirical game-theoretic analysis with reinforcement learning in an effective method for generating stronger trading strategies in this domain.

Introduction

Because it encompasses a variety of trading mechanisms and reasoning tasks, the TAC Travel game presents a plethora of strategy problems for trading agents [Wellman et al., 2007]. Agents buy flights at stochastically varying fixed prices, buy hotels at prices set by multiunit ascending auction, and buy as well as sell entertainment tickets through continuous double auctions (CDAs). Based on our recent successful effort to derive stronger strategies for trading in generic CDAs [Schwartzman and Wellman, 2009a], we sought to employ this method to trading entertainment in TAC Travel.

The TAC entertainment trading problem differs from that typically studied in the generic CDA trading literature in several essential respects. First, there are actually 12 simultaneous CDAs operating in TAC Travel, and interdependent values among the goods. Second, agents in the generic setup are typically specialized to buyer or seller roles, whereas in TAC Travel an agent may both buy and sell in the same auction. Third, valuations are not fixed and generated up front, but rather are determined indirectly by the surrounding market context (e.g., flight and hotel prices and holdings), which vary dynamically as the game progresses.

For these reasons, it is not straightforward to translate existing CDA strategies to entertainment trading in TAC, and indeed, to our knowledge, no previous entrants have reported attempting to perform simple adaptations from the CDA literature. Nevertheless, the methods by which we learned improved strategies for generic CDAs are quite applicable to this TAC problem, as we describe herein.

Background: EGTA/RL

One of the main difficulties of analyzing TAC strategies experimentally is that the performance of an agent's strategy can be greatly affected by the decisions of other agents. As a result, studies on strategy performance need to properly define the context of other-agent strategies under which the evaluation takes place. In the *empirical game-theoretic analysis* (EGTA) approach [Wellman, 2006], games are estimated by employing computer simulations. EGTA methods systematically compile information about strategy performance, and iteratively add strategies to a set of candidates to build a comprehensive model covering a broad strategy space, amenable to game-theoretic analysis.

A key issue that EGTA studies need to address is the *strategy exploration problem* [Schwartzman and Wellman, 2009b], which focuses on the basic step of adding a new strategy to a set of candidates, enlarging the profile space. This decision is significant because the typical problems that we study with EGTA have very large or infinite strategy spaces, and computational constraints restrict our analysis to finite sets. Since adding strategies expands the size of the problem exponentially, the decision to introduce a new strategy needs to be considered carefully.

In our CDA study [Schwartzman and Wellman, 2009a], we approached the strategy exploration problem by interleaving EGTA with reinforcement learning (RL) [Sutton and Barto, 1998]. The central idea of this interleaved approach is to focus learning effort on the contexts supported by equilibrium reasoning over the data collected thus far. Given the large amount of training data required for effective RL, it would not be practical to learn a best response to any but a tiny fraction of other-agent strategy profiles. By focusing on finding a deviation from a current equilibrium, we concentrate the training on the most promising regions of profile space. By definition, a new strategy that succeeds at deviating will qualitatively change the empirical game analysis, effectively producing a new equilibrium. By introduction of relevant strategies at successive EGTA/RL iterations, we increase our confidence in the ultimate results of analyzing the cumulative empirical game.

Other researchers have used a similar process to search for equilibrium profiles, starting from a few basic strategies and iteratively finding (exact) best-response profiles until reaching a fixed point. The process, when it converges, results in

a strategy profile that is a best response to itself, and consequently a Nash equilibrium (NE). Our EGTA/RL approach is similar in spirit to iterated best response, except that RL will typically provide *approximate* best responses for the scenarios that we are interested in.

The EGTA/RL approach can be summarized by the following broad stages:

1. Implement game simulator.
2. Select set of candidate strategies \hat{S} .
3. Estimate the empirical game.
4. Find a Nash equilibrium s^* .
5. Derive a new bidding strategy L using RL, applied in a context where other agents play s^* and the learning agent attempts to deviate.
6. Evaluate the learned policy. If L provides a positive deviation from s^* , add L to \hat{S} , and extend the empirical game by continuing with stage 3. Otherwise, if learning has converged and the RL model cannot be improved further, the process ends.

Stages 1 through 4 are part of the standard EGTA process. Our EGTA/RL approach adds stages 5 and 6, and repeats the cycle until convergence. The entire process is outlined in Figure 1.

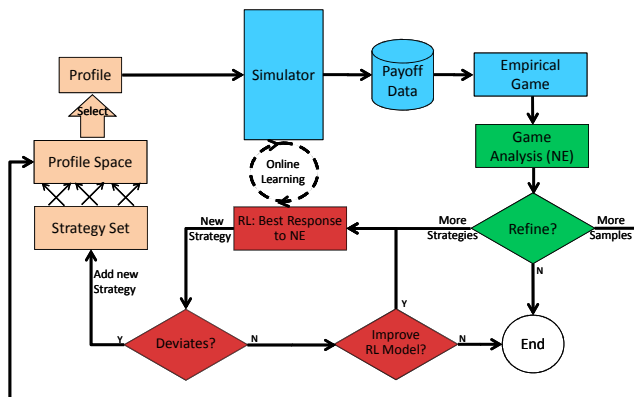


Figure 1: Interleaving empirical game-theoretic analysis with reinforcement learning.

One of the most basic trading scenarios is an abstract market based on the *continuous double auction* (CDA) mechanism [Friedman, 1993]. The CDA is a simple and well-studied auction institution, employed commonly in commodity and financial markets. Numerous strategies for trading in generic CDA environments have been proposed over the years, and analyzed experimentally in various combinations [Cliff, 1998, Rust et al., 1994, Tesaro and Bredin, 2002, Vytelingum et al., 2008]. In a recent work [Schwartzman and Wellman, 2009a], we analyzed a strategy set including representatives of the major strategies from this literature. We exhaustively sampled profiles over this set, and iteratively derived new strategies using our combined EGTA/RL process. When this converged, the equilibria of

the final empirical game were supported exclusively with learned strategies. This study demonstrated the effectiveness of RL interleaved with EGTA for deriving stronger CDA trading strategies, and inspired our present work.

TAC Travel Overview

TAC Travel is a game in the domain of travel shopping, in which eight autonomous travel agents assemble trip packages of hotel rooms, flights, and entertainment tickets on behalf of their clients. During a 9-minute game, agents attempt to maximize total client satisfaction (utility summed over eight clients) minus expenditures, by trading goods in three different markets. Flights are sold at fixed prices that vary according to a stochastic process. Hotel rooms are sold in multiunit ascending auctions, that close periodically in a random order. Entertainment ticket trading, our focus in this work, is mediated by CDAs.

More specifically, there are three different entertainment events across four days, and clients specify a *fun bonus* $(f_1, f_2, f_3) \sim U[0, 200]$ for attending each of them. There are 8 tickets available for each event type and day (96 tickets in total), and all agents receive an initial random endowment of 12 tickets. Game rules allow clients to attend at most one event per night of the trip, and do not provide additional utility for having a client attend the same type of event more than once. Tickets are traded throughout the entire 9-minute game via 12 standard CDAs (one per day and event type).

The game comprises 28 simultaneous auctions in total, with interdependencies dictated by market rules, client preferences, and trip feasibility constraints. Clients accrue utility 1000 for a feasible trip, minus a penalty for deviating from their preferred day, plus bonuses for staying in the premium hotel or consuming entertainment. At the end of a game instance, the TAC game server calculates the optimal allocation of goods to clients for each agent, and computes agent score as total client utility minus net expenditures. Agents holding negative balances of entertainment tickets are assessed a penalty of 200 per ticket owed.

Walverine’s Architecture

The architecture of *Walverine*, the TAC Travel entry from the University of Michigan, is depicted in Figure 2. Trading logic is divided into two main modules, one to purchase hotel rooms and flights, and another to trade entertainment tickets. A centralized optimizer computes optimal packages and marginal valuations, answering queries requested by the trading modules. All optimizations are based on transactions and prices (both actual and predicted) reflecting the state of both traders, turning the optimizer into an implicit link that allows trader coordination. A proxy mediates communications between the trading components and the game server, routing all bids and queries.

The optimizer provides an interface that allows trading components to set parameters (e.g., good holdings and prices) and issue queries, communicating with them through sockets. The optimization problem is modeled as an integer linear program [Wellman et al., 2007, Appendix B] written in AMPL [Fourer et al., 1993], and computed using the

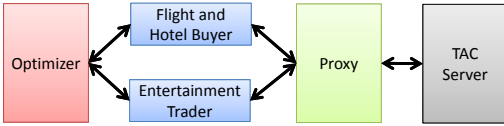


Figure 2: Walverine’s architecture.

CPLEX solver. It answers two basic queries.

Optimal package. Optimal bundle of goods, given good holdings and predicted prices.¹ Our implementation actually considers only opportunities to purchase goods, and neglects the possibility of selling entertainment tickets. This is an instance of the *acquisition problem*, a special case of the *completion problem* [Boyan and Greenwald, 2001] comprising a core subtask for bidding in simultaneous markets.

Marginal values. Incremental value of each additional unit of available goods. Let $v^*(g, x)$ denote the value of the optimal package, assuming current holdings and price predictions, except that the agent holds x additional units of good g and that no further units of g can be purchased. The marginal value of the k th unit of g is

$$MV_k = v^*(g, k) - v^*(g, k - 1). \quad (1)$$

The optimizer provides marginal values for open hotel auctions and $1 \leq k \leq 8$. For entertainment, the optimizer provides the marginal value of buying (MV_1) and cost of selling (MV_0) a single unit of each event type and day.

Empirical Game

Our research team at the University of Michigan has been conducting an ongoing EGTA study of TAC Travel based on parametric variations of Walverine since 2004, with over 200,000 game instances in its data set as of April, 2009. The experiment runs on a testbed comprising a total of five dedicated workstations (one running the TAC game server, one running all eight agents, and three running the optimizers), plus a shared workstation that controls experiment generation and data gathering.

All experiments and results described in this paper address the reduced four-player version of the game, denoted $TAC_{\downarrow 4}$, where each player controls the strategy of two TAC agents [Wellman et al., 2005]. This reduction, coupled with symmetry, shrunk the profile space from more than 10^{44} to 270725 distinct profiles, given 49 strategies introduced to date. Table 1 shows our data set divided among the 1-, 2-, and 4-player versions of the game. The unreduced 8-player game, due to its size, remains mostly unevaluated.

We reduce sample variance by adjusting scores, using the method of control variates [L’Ecuyer, 1994]. The control

¹Walverine predicts hotel prices by calculating the competitive equilibrium of the TAC economy [Wellman et al., 2004]. Entertainment and flight prices are given by the price quote.

p	Profiles			Samples/Profile	
	Total	Evaluated	%	Min	Mean
1	49	49	100.0	35	92.9
2	1225	998	81.5	22	41.1
4	270725	7254	2.7	19	27.8

Table 1: Evaluated profiles in TAC Travel for each reduction $TAC_{\downarrow p}$, and sampled games per profile.

variables combine premiums for good hotels and entertainment, initial flight quotes, and demand based on preferred arrival and departure dates. The specifics of this adjustment are provided by Wellman et al. [2007].

Existing Strategies

The strategies included in our EGTA study consist of different versions of Walverine, generated through fourteen parameters that control its behavior. One such parameter, for example, controls *bid shading*, the amount that hotel bids get reduced below Walverine’s value estimate. Thirteen parameters control behavior for flights and hotels bidding, with a single parameter selecting the strategy used to trade entertainment tickets. The full parametrization produces a large space of 864,000 possible strategies, given the individual parameter values tested to date. Of course, we never intended to test all these combinations. Over the course of the EGTA analysis, new strategies have been introduced one-by-one, each manually selected after carefully considering intermediate results from empirical equilibrium analyses.

We provide a list of strategies explored to date in Table 2.² For the purposes of this study we ignore the details about flights and hotels, and focus on the entertainment strategy only. Consequently, we assign uninterpreted labels to flight/hotel parameters, except for setting “H3” which is the base of our new (learned) entertainment strategies.

The entertainment strategies that we tested are:

- E10/E14: derived using RL for TAC-02
- E11/E15: derived using RL for TAC-03
- E12/E13: based on livingagents
- E16/E17: based on WhiteBear
- E19/E20: derived using our EGTA/RL methodology

As discussed below, E14 is the same as E10 except for a bug fix that updates the optimizer’s view of entertainment holdings properly. Similarly, E15 fixes E11, E13 fixes E12, and E17 fixes E16. In actuality, for any given case the “fixed” version is not necessarily better than the “bug” version, as other elements of the strategy may compensate for the unintended failure to update holdings. Therefore, we treat these as simply different strategies, and evaluate them empirically in our experimental testbed.

²Strategy IDs shown here do not necessarily correspond with labels employed in previous studies based on our testbed.

Strategy ID	Samples	Parameters		
		Ent.	Hotel/Flight	
1	7,698	E11	Various combinations	
2	7,092	E12		
3-7	237,776	E11		
8	13,478	E12		
9	33,840	E11		
10	25,164	E12		
11-18	277,350	E11		
19	18,376	E15		
20	46,780	E17		
21	5,264	E14		
22	55,412	E11		
23	19,018	E15		
24	52,140	E11		
25	30,062	E10		
26	50,734	E11		
27	21,280	E12		
28	15,318	E13		
29-30	36,626	E15		
31	85,512	E17		
32-35	39,994	E11		
36	4,990	E15		
37-39	60,762	E11		
40	24,570	E12		
41	5,354	E16		
42	126,830	E17		
43	31,172	E11		
44	57,250	E17		
45	24,838	E12		H3
46	9,766	E13		
47	106,686	E17		
49	61,954	E19		
50	34,762	E20		
Total	1,627,848			

Table 2: Strategies in TAC Travel empirical game. Samples are per game and agent, out of 203,481 games. Strategy 48 is not part of the empirical game, as it is only used temporarily for online learning purposes.

Walverine 2002 (E10/E14)

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 more limited task of entertainment trading.

The original entertainment trader used for TAC-02 employed Q-learning to derive a bidding policy. The Q function was encoded in two standard lookup tables, one for days $\{1, 4\}$ and the other for days $\{2, 3\}$. The agent considered each auction independently, and approximated their state using six parameters: BID , ASK , number of tickets held, game time, marginal value MV_1 of an additional unit, and marginal cost MV_0 of selling a unit (1). To keep the state space manageable, we discretized these dimensions into value sets of size 6, 6, 3, 3, 7, and 7, respectively.

Actions were defined as offsets from marginal value. We used sixteen discrete offset values, half for buying and half

for selling decisions. On each bidding iteration, the agent alternated between buy and sell decisions, treating them independently and sequentially to avoid considering all buy/sell combinations. The agent submitted only single-unit entertainment bids on each bidding iteration.

Rewards were given in both intermediate and terminal states. Intermediate rewards comprised cash flow originated from trading, resulting in positive rewards for sales and negative ones for purchases. Terminal rewards comprised fun bonus accrued to clients, based on the optimal allocation of goods at the end of a game.

We trained our agent by employing offline Q-learning on a training set consisting of batches of games against other TAC participants and instances of self-play, a total of more than 1800 games. *Walverine* employed a variety of entertainment trading policies while gathering experience, including a hard-coded strategy based on the one reportedly employed by livingagents [Fritschi and Dorer, 2002]. Once we had accumulated sufficient data, we ran some instances of *Walverine* based on preliminary learned policies, with various exploration-exploitation control methods.

During the actual finals, our learned strategy obtained an average reward nearly 400 over the no-trading strategy, but still below the livingagents baseline. Since we conducted this training while preparing our agent for the actual TAC tournament, our agent was subject to many changes during the learning process. These changes included substantial modifications of all main modules, including the optimizer and all trading components (entertainment, flights, and hotels), which undoubtedly confounds the results. In the present learning study we avoided these pitfalls, by following the more methodical approach described below.

One change in particular involved switching from one to two mirror optimizers for efficiency purposes, and having these handle queries of hotels and entertainment separately. The change required that our two trading components updated holdings and price information (for the goods they were in charge of) on both optimizers. An unfortunate bug, however, prevented the entertainment trader from updating entertainment holdings in the optimizer used for hotel queries. Consequently, our trading components were not in synchrony, and the flight/hotel buyer behaved as if no entertainment trading occurred throughout the games (including those in TAC finals). This bug was introduced while we were already in our training period, and was discovered in 2004 from experiments conducted in our EGTA studies. Strategies E10 and E14 both implement the policy derived from this learning experiment, but E14 includes a bug fix to update holdings in the optimizer properly. Note that E14 is not necessarily better than E10 because a large fraction of the bidding policy was learned from experience acquired while playing games with the bug.

Walverine 2003 (E11/E15)

In 2003 we tried Q-learning again, but this time using neural networks to represent the value function. This effort was short-lived and not particularly successful, but nonetheless we included the resulting policy in our testbed as part of

our EGTA study. Settings E11 and E15 implement the same policy, except that E15 updates optimizer holdings properly.

livingagents (E12/E13)

The top-scoring agent of the TAC competition held in 2001, *livingagents* [Fritschi and Dorer, 2002], chose all client itineraries at the beginning of a game, placing very high hotel bids (to secure the rooms) and purchasing appropriate flights immediately. The choice was made by calculating the optimal package of goods, assuming that flights could be purchased at their actual cost, hotels at their average (historic) cost, and all entertainment tickets at the average (historic) price of \$80. For the rest of the game, the agent traded entertainment tickets in order to match its optimal allocation. For the first seven minutes of a game, entertainment bidding was conditional on having an *ASK* price lower than \$80 or a *BID* price higher than \$80, for buy and sell offers, respectively. After the seventh minute, *livingagents* simply placed necessary bids to meet its optimal package, without any conditions. All bids, buy or sell, were priced at \$80 during the entire game.³

Our implementation of E12/E13 is limited to *livingagents*' entertainment strategy specified above, leaving the purchase of flights and hotels to our parameterized version of *Walverine*. Since *Walverine* does not initially commit to client itineraries, the optimal package of goods usually varies throughout the game. Consequently, our implementation attempts to buy (sell) tickets with a marginal value higher (lower) than 80, as indicated by the optimizer on every bidding opportunity. E12 and E13 implement the same policy, except that E13 updates optimizer holdings properly. Note that, in this case, we would expect E13 to be better than E12, as there is no learning involved.

WhiteBear (E16/E17)

WhiteBear [Vetsikas and Selman, 2003] was the top scoring agent in the Trading Agent Competitions held in 2002 and 2004, and obtained the third and second positions in 2003 and 2005, respectively. The creators of *WhiteBear* devoted most of their effort to dealing with hotels and flights. For entertainment, they used a simple but effective approach.

WhiteBear develops an overall plan for entertainment holdings, and makes offers to buy or sell accordingly at a price equal to the current price offset by a small amount. At the beginning of a game the agent also tries to buy low-priced tickets, either to allocate them to clients or potentially sell them for a profit later in the game. The agent limits the prices it bids in order to avoid very profitable deals for other agents, even if such deals were beneficial for *WhiteBear*. Price limits are relaxed somewhat during the last minute of the game.

Our implementation is based on the source code shared by the authors of *WhiteBear*, adapted to work within the framework used by *Walverine*. Specifically, the agent sub-

mits single-unit offers on each bidding iteration based on following rules:

- During the first minute of a game, the agent attempts to purchase tickets depending on current holdings. If it holds at most one unit of the specified ticket, *WhiteBear* attempts to buy for a price of 30. If it holds two or three tickets, it offers 20, and if three or more, the offer is 5.
- Between the second and eighth minutes, the agent queries the optimizer to determine the ticket quantities it needs to buy or sell in order to match the optimal package. The agent then bids $\min(ASK + 10, 66)$.
- During the last minute of a game, *WhiteBear* submits buy bids at price $\min(ASK + 10, 100)$.
- Sell offers needed to match the optimal package are set to $\max(BID - 2, 66)$ throughout the entire game.

Despite the simplicity of this strategy, *WhiteBear* obtained the highest entertainment score (as measured by the sum of fun bonus and cash flow obtained through trading) of all participants in the 2002 competition. This score was about 10% higher than that of the second highest scoring agent, and about 15% above the results obtained by *Walverine-02*. *Walverine* adopted E17 as its entertainment strategy for TAC-05, and this was likely the most important contributor to its improved performance [Wellman et al., 2006].

E16 and E17 implement the same policy, except that E17 updates optimizer holdings properly. As with *livingagents*, we would expect E17 to be generally better than E16.

Other Entertainment Strategies

Years of TAC Travel tournaments yielded various different entertainment strategies, many of which are not included in our testbed. *ATTac* [Stone et al., 2001, 2003], a top-scoring agent in TAC-00/01, calculated marginal values for each ticket and submitted linearly increasing (decreasing) buy (sell) bids as a function of game time, settling for smaller profits as the game progressed. *TeamHarmony* [Onodera et al., 2003], a participant in TAC-03/04, also submitted buy (sell) prices that increased (decreased) with time. *SouthamptonTAC* [He and Jennings, 2003], the second-highest scoring agent in TAC-02, calculated its optimal allocation of goods throughout the game, and employed fun bonus to value entertainment tickets (instead of marginal values). The agent defined a reservation price consisting of ticket value and a margin that decreased with time, and submitted buy (sell) bids whenever the *ASK(BID)* price approached the reservation price. *Thalis* [Fasli and Poursanidis, 2003], which achieved third and fourth places in TAC-02 and TAC-03, respectively, traded entertainment tickets seeking to meet its optimal allocation. This agent submitted sell bids after the first minute of the game, restricting prices to the range 81–125. Buy bids were restricted to the range 30–101, but were only submitted after the sixth game minute in order to exploit lower historical average trade prices. Another participant of TAC-02, 006 [Aurell et al., 2002], submitted bids at prices that approached the agent's estimated marginal value exponentially. *LearnAgents* [Sardinha et al., 2005], the agent achieving third place

³This description is adapted to the current 9-minute game. TAC Travel games ran for 15 minutes in 2000, 12 minutes from 2001 to 2003, and 9 minutes thereafter.

in TAC-04, submitted bids at prices indicated by its optimal allocation, offset by a fixed amount. Metarcor [Kehagias et al., 2006], the highest scoring agent in TAC-05, employed a set of bidding rules seeking a predefined average profit per auction per game. This agent determined profit by the difference between cash flow and ticket value, and its actual bids varied throughout the game based on time elapsed and other heuristic rules. RoxyBot, the highest scoring agent in TAC-06 [Lee et al., 2007], predicted future ticket prices based on trades from past games, deciding whether to submit bids at current prices or wait for future stages of the game.

Learning Framework

The learning model employed for entertainment strategies E19 and E20 is related to that of E14, and is also similar to the one used for our generic CDA study. We employ online Q-learning, and tile coding to represent the value function [Sutton and Barto, 1998]. The model is defined by a standard formulation of states, actions, and rewards.

Tile Coding

We partition state and action features into tiles, which combine to form a multidimensional tiling. Each tile maintains a weight representing the approximate Q-value of a (discretized) state-action pair. Given a training tuple, the method finds the containing tile t , calculates the standard Q-learning update, and adjusts the weight of t accordingly. We control generalization across each feature independently through a parameter b_i denoting *generalization breadth*, the farthest neighbor of t across feature i that gets updated. Neighbor tiles that are d_i ($0 < d_i \leq b_i$) tiles away from t across feature i get a fraction of such update equal to $\prod_{i \in F} (1 - \frac{d_i}{b_i+1})$, where F is the set of features encoded in the tiling.

State Space

The following observable features are used to describe a state and condition actions:

- Role: binary feature (buy or sell), encoded by two tiles (generalization not applicable).
- Day: binary feature to distinguish between events on days $\{1, 4\}$ or $\{2, 3\}$, encoded by two tiles.
- Value: marginal value provided by the optimizer (1), based on the role (MV_1 if buying, MV_0 if selling). This is encoded by 101 tiles and generalization breadth of five.
- Time: time elapsed in the game, encoded by 18 tiles and generalization breadth of two tiles.

As with entertainment strategy E14, here we also treat each auction independently, and alternate between purchase and sell decisions on each bidding iteration. Unlike E14, we ignore market quotes, ticket holdings, and one marginal value (depending on the role) — but use far greater fidelity to encode value and time. Unlike the generic CDA game, we do distinguish buying and selling roles. The (marginal) values for buying and selling goods are not symmetric in this case, and other strategies employ asymmetric bidding rules as well. We are also not considering features that encode price history explicitly, as we did for the CDA game.

Actions

An action A is a positive offset from the marginal value provided by the optimizer at the time the bid gets computed. Sell bids are submitted for $MV_0 + A$, and buy bids for $MV_1 - A$. These actions are encoded by 40 tiles, and a generalization breadth of two tiles. This configuration is similar to that of E14, with greater fidelity.

Rewards

Rewards are defined for both intermediate and terminal states. Intermediate rewards are assigned to states including a transaction, based on marginal value calculations (1). In contrast to the prediction-based marginal values provided dynamically during a game by the optimizer, however, we assign rewards based on marginal values calculated with respect to the agent’s actual holdings of flights and hotels at the end of a game, assuming that no additional goods can be traded. We traverse the sequence of transactions, set entertainment holdings in the optimizer to those that the agent had during the game (before each transaction occurred), and compute marginal values accordingly. For ticket sales, the reward is given by the cash amount obtained from the trade minus MV_0 . Similarly, for ticket purchases, the reward is given by MV_1 minus the amount paid.

Terminal reward is the difference between the final agent score and the sum of all intermediate rewards.

This scheme differs from that used for the CDA game, which did not need a final reward and had fixed (given) values for each unit being traded. It is also different from the scheme used for E14, as it attempts to better allocate rewards to actions by taking into account the marginal contribution of each trade with regards to the final allocation of goods.

Experiments

Our search for (approximate) equilibria in the empirical TAC Travel game focuses on two-strategy symmetric mixed profiles. Restricting attention to symmetric equilibria with small support simplifies the search, and enables us to confirm or reject equilibria with evaluations of only a relatively limited number of neighboring profiles. Given a symmetric four-player game, there are only five pure profiles over two strategies, and only four profiles required to evaluate deviation by a single player to a third candidate strategy. As noted in Table 1, despite years of continual simulation we have evaluations for only 2.7% of profiles of the 49-strategy game. This includes 359 pairs of strategies (out of $\binom{47}{2} = 1081$ possible pairs) for which we have all five profiles involving that pair. Each of these has been “challenged” to a different extent, as measured by the fraction of deviating profiles explored. We define the *regret bound* of a profile as the maximum gain from deviation by one player to another strategy, based on evaluated neighbor profiles. It is a lower bound on actual regret because potential deviations not yet evaluated could only increase the maximum gain.

Before we began our learning process, the two-strategy symmetric profile with lowest regret bound in TAC \downarrow_4 was

a mixture of strategies 31 and 47,⁴ played with probabilities .181 and .819, respectively. Such determination was based on an exhaustive search among the 359 available two-strategy combinations. Figure 3 presents a sensitivity analysis of these mixtures for 10,000 sampled payoff functions. Most curves in the figure are grayed and excluded from the legend, as they are essentially dominated.⁵ Note that mixture probabilities shown in the legend are slightly different from .181 (31) and .819 (47) because the figure is based on the latest data set at this writing, which includes additional samples taken since identifying this best pre-learning approximate equilibrium. Note also that there are other undominated mixtures with relatively low regret bounds. Because all potential deviations of the 31/47 mixture have been evaluated, .02 is the estimated actual regret of 31/47 (with respect to strategies 1–47 only), not just a lower bound.

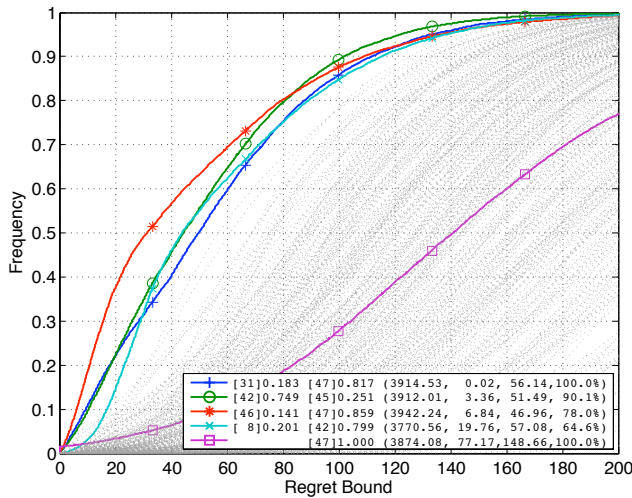


Figure 3: Distribution of regret bounds for 359 two-strategy mixtures of strategies 1–47. The legend shows mixture proportions, expected payoff and regret bound based on the maximum likelihood payoff matrix, mean regret bound over 10,000 sampled payoff matrices (also indicated by the markers on each curve), and probability-weighted percentage of potential deviations evaluated (count of evaluated neighbor profiles over all neighbor profiles, weighted by the probability of each profile of being played). Dominated mixtures are shown in gray.

In order to improve upon this 31/47 mixture and the entertainment strategies described earlier, we derived entertainment strategy E19 (strategy 49) by employing our EGTA/RL methodology. Since we are working on a four-player version

⁴Both 31 and 47 employ entertainment strategy E17 (based on WhiteBear), with different parameters for flights and hotels.

⁵Payoff function samples were generated based on mean and variance estimates of profile payoffs from the empirical game. Our criterion to test dominance is to compare regret bounds at discrete cumulative probabilities $\{.001, .002, \dots, .999\}$. A mixture that provides equal or higher regret bound than another mixture for every discrete point (with at least one strict inequality) is considered essentially dominated.

of the game, we set three players (six agents) to play the 31/47 mixture, while one player (two agents) attempted to deviate by learning a new policy. Our learning approach was similar to that used for the CDA study. Training was conducted online, repeatedly cycling over the experience collected during the last 200 games played (out of more than 900). Agents explored new actions with a linearly decreasing probability, using softmax action selection. The learning rate was fixed at .005, and the discount factor at .99. The payoff of a learned strategy was evaluated by playing all successive games with no further adaptation.

Figure 4 shows the learning curve of E19. At the time we conducted this training, mixture 31/47 provided a payoff of 3915.06 (now 3914.41, given additional samples). The evaluation of strategy 49, when other players adopt the 31/47 equilibrium, results in a payoff of 3976.24, or a deviation gain of 61.83.

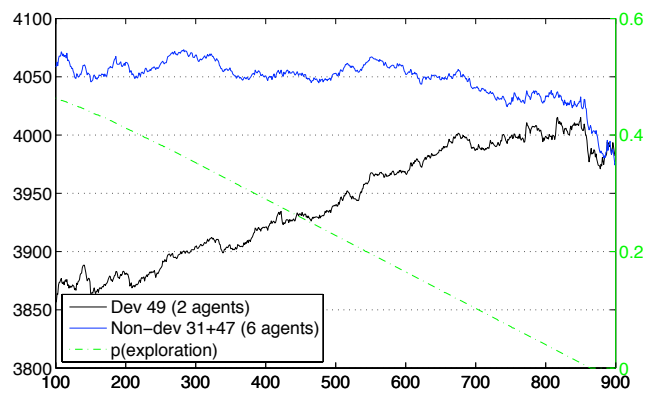


Figure 4: Learning curve of entertainment strategy E19 (strategy 49). Six agents (shown in blue) adopt the 31/47 mixture, while two agents (shown in black) seek to deviate. In order to deviate, 49 needs a payoff above 3914.41 (i.e., payoff obtained if all agents adopted the 31/47 mixture). The dashed diagonal line (right axis) shows the probability of exploring new actions.

Based on the deviation results of strategy 49, we incorporated it to the set of existing strategies and extended the empirical game. By evaluating some of the new profiles, we determined that strategy 49 was actually a pure-strategy NE.

Given this result, we set our learner to derive entertainment strategy E20 (strategy 50). We conducted training over 950 games, also decreasing the probability of exploring new actions linearly (but starting from a smaller initial value). The learning curve is shown in Figure 5.

Evaluating strategy 50 resulted in a deviation gain of 20.04 from the all-49 equilibrium. Consequently, we added strategy 50 to our data set, and extended the empirical game with further samples. This resulted in an approximate mixed-strategy NE (regret of .03) consisting of strategies 49/50, played with probabilities .295/.705. Figure 6 shows a sensitivity analysis of 385 two-strategy mixtures evaluated to date, considering our entire database up to strategy 50. Note that we have evaluated all potential deviations from the 49/50 equilibrium, and so far no other mixture provides

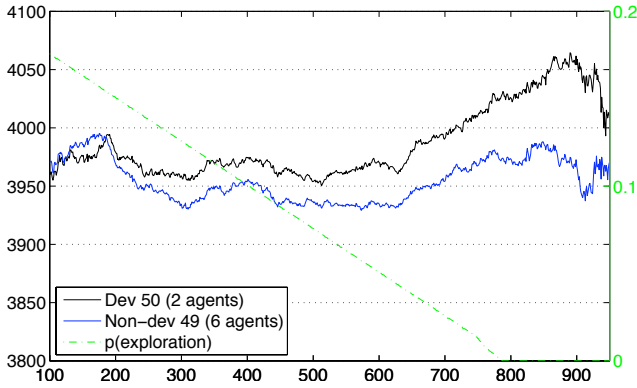


Figure 5: Learning curve of entertainment strategy E20 (strategy 50).

a comparable regret. Except for the 49/50 approximate equilibrium and the all-49 profile, all other 383 mixtures are essentially dominated.

The results of all these experiments are summarized in Table 3.

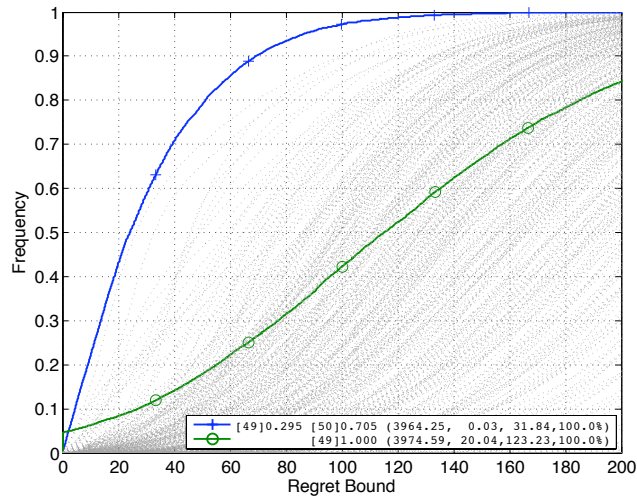


Figure 6: Distribution of regret bounds for 385 two-strategy mixtures of strategies 1–47 and 49–50. Bounds are based on 10,000 sampled payoff matrices.

Strategies	EGTA			Learning	
	Equil. Mix	Payoff	Num. Profiles	Strat.	Dev. Payoff
1-47	31 47	0.181 0.819	3914.41	230300	49 3981.54
49	49	1.000	3974.59	249900	50 3994.63
50	49 50	0.295 0.705	3964.25	270725	

Table 3: Results obtained by interleaving EGTA with RL.

We also evaluated a version of TAC₄ restricted to strate-

gies {45, 46, 47, 49, 50}, for which we have evaluations of all combinations of profiles (70 in total). These strategies use identical settings for flight and hotel parameters⁶ and only differ by their entertainment strategies—which makes them good candidates to evaluate our learned strategies E19 and E20. Using this restricted game, we performed a sensitivity analysis on strategy mixtures, sampling 100,000 payoff matrices and computing equilibria via replicator dynamics. The results indicate that strategies 49 and 50 are played most frequently in most of the samples, though strategy 47 (entertainment E17) appears in equilibrium occasionally.

As a final test, following the approach of Jordan et al. [2007], we elaborated a ranking of strategies by comparing deviation gains from equilibrium. We performed this analysis for the equilibrium prior to learning new entertainment strategies (31/47), and the one obtained afterwards (49/50). With respect to the pre-learning equilibrium, 49 (the strategy learned in exactly this context) is best with a gain of almost 62, statistically better than the next group ($p < .01$): 31 and 47, which have statistically indistinguishable gains close to zero. With respect to the current 49/50 equilibrium, no strategies are close to deviating (each worse at $p < .01$). Strategies 49 and 50 have statistically indistinguishable gains close to zero, statistically better than the next strategy, 42, which has a gain of -66 .

Discussion

Two iterations of our interleaved EGTA/RL process produced two new TAC entertainment strategies, which comprise an equilibrium in the cumulative empirical game. The deviation benefit decreased from the first iteration to the second, and we expect that little gain would have been produced by a third iteration.

It is difficult to make broad claims about the superiority of our new strategies for TAC entertainment strategy, for several reasons. First, although we did include a diverse sample of known approaches, this was not exhaustive. Second, it is impossible to completely separate entertainment strategy from the rest of the TAC Travel agent, as policies for flight and hotel trading significantly influence entertainment values. We controlled for this to some extent in our study, by coupling our new strategies with the best ranked flight and hotel strategy parameters, based on our previous empirical game analysis. Moreover, in a population of agents identical except for the entertainment strategies, the new learned strategies emerged as most prominent in equilibria under sensitivity analysis. Nevertheless, past success in entertainment trading by *WhiteBear* and the 2005 version of *Walverine* that uses *WhiteBear*'s entertainment strategy suggest that our baseline comparison is highly salient.

It may be surprising that the learned entertainment strategies do not condition on price quote or history information (except indirectly as the marginal value for one ticket may depend on price quotes on others). In fact, many of the previ-

⁶The specific setting is labeled “H3”, the best known combination of flight and hotel parameters. This is based on results from a linear regression that fitted parameter settings to score results against the best known equilibrium mixture up until strategy 47.

ous strategies from the TAC literature make little or no use of price quotes. Given the importance of such observations in generic CDA bidding, additional EGTA/RL iterations with a state space reformulated to include price features may be a promising approach to further improvements.

Acknowledgments

We thank Ioannis Vetsikas for sharing the WhiteBear source code with us. Kevin Lochner and Daniel Reeves assisted significantly in the operation of the TAC Travel testbed over the years. This work was supported in part by the US National Science Foundation.

References

- E. Aurell, M. Boman, M. Carlsson, J. Eriksson, N. Finne, S. Janson, P. Kreuger, and L. Rasmusson. 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, 2002.
- J. Boadway and D. Precup. Reinforcement learning applied to a multiagent system. Presentation at TAC Workshop, 2001.
- J. Boyan and A. Greenwald. Bid determination in simultaneous auctions: An agent architecture. In *Third ACM Conference on Electronic Commerce*, pages 210–212, Tampa, FL, 2001.
- D. Cliff. Evolving parameter sets for adaptive trading agents in continuous double-auction markets. In *Agents-98 Workshop on Artificial Societies and Computational Markets*, pages 38–47, Minneapolis, MN, May 1998.
- M. Fasli and N. Poursanidis. Thalix: A flexible trading agent. Technical Report CSM-388, University of Essex, Department of Computer Science, 2003.
- R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Boyd & Fraser, 1993.
- D. Friedman. The double auction market institution: A survey. In D. Friedman and J. Rust, editors, *The Double Auction Market: Institutions, Theories, and Evidence*, pages 3–25. Addison-Wesley, 1993.
- C. Fritschi and K. Dorer. Agent-oriented software engineering for successful TAC participation. In *First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Bologna, 2002.
- M. He and N. R. Jennings. SouthamptonTAC: An adaptive autonomous trading agent. *ACM Transactions on Internet Technology*, 3:218–235, 2003.
- P. R. Jordan, C. Kiekintveld, and M. P. Wellman. Empirical game-theoretic analysis of the TAC supply chain game. In *Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1188–1195, Honolulu, 2007.
- D. Kehagias, P. Toulis, and P. Mitkas. A long-term profit seeking strategy for continuous double auctions in a trading agent competition. In *Fourth Hellenic Conference on Artificial Intelligence*, Heraklion, Greece, 2006.
- P. L’Ecuyer. Efficiency improvement and variance reduction. In *Twenty-Sixth Winter Simulation Conference*, pages 122–132, Orlando, FL, 1994.
- S. J. Lee, A. Greenwald, and V. Naroditskiy. RoxyBot-06: An (SAA)² TAC travel agent. In *Twentieth International Joint Conference on Artificial Intelligence*, pages 1378–1383, Hyderabad, 2007.
- M. Onodera, H. Kawamura, M. Yamamoto, K. Kurumatani, and A. Ohuchi. Design of adaptive trading strategy for trading agent competition. In *International Technical Conference on Circuits/Systems, Computers and Communications*, pages 337–340, 2003.
- J. Rust, J. H. Miller, and R. Palmer. Characterizing effective trading strategies: Insights from a computerized double auction tournament. *Journal of Economic Dynamics and Control*, 18:61–96, 1994.
- J. A. R. P. Sardinha, R. L. Milidiú, P. M. Paranhos, P. M. Cunha, and C. J. P. Lucena. An agent based architecture for highly competitive electronic markets. In *Eighteenth International FLAIRS Conference*, pages 326–331, Clearwater Beach, FL, 2005.
- L. J. Schwartzman and M. P. Wellman. Stronger CDA strategies through empirical game-theoretic analysis and reinforcement learning. In *Eighth International Conference on Autonomous Agents and Multi-Agent Systems*, pages 249–256, Budapest, 2009a.
- L. J. Schwartzman and M. P. Wellman. Exploring large strategy spaces in empirical game modeling. In *AAMAS-09 Workshop on Agent-Mediated Electronic Commerce*, Budapest, 2009b.
- P. Stone, M. L. Littman, S. Singh, and M. Kearns. ATTac-2000: An adaptive autonomous bidding agent. *Journal of Artificial Intelligence Research*, 15:189–206, 2001.
- P. Stone, R. E. Schapire, M. L. Littman, J. A. Csirik, and D. McAllester. Decision-theoretic bidding based on learned density models in simultaneous, interacting auctions. *Journal of Artificial Intelligence Research*, 19:209–242, 2003.
- R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press, 1998.
- G. Tesauro and J. L. Bredin. Strategic sequential bidding in auctions using dynamic programming. In *First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 591–598, Bologna, 2002.
- I. A. Vetsikas and B. Selman. A principled study of the design tradeoffs for autonomous trading agents. In *Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 473–480, Melbourne, 2003.

- P. Vytelingum, D. Cliff, and N. R. Jennings. Strategic bidding in continuous double auctions. *Artificial Intelligence*, 172:1700–1729, 2008.
- M. P. Wellman. Methods for empirical game-theoretic analysis (extended abstract). In *Twenty-First National Conference on Artificial Intelligence*, pages 1552–1555, Boston, 2006.
- M. P. Wellman, D. M. Reeves, K. M. Lochner, and Y. Vorobeychik. Price prediction in a trading agent competition. *Journal of Artificial Intelligence Research*, 21:19–36, 2004.
- M. P. Wellman, D. M. Reeves, K. M. Lochner, S.-F. Cheng, and R. Suri. Approximate strategic reasoning through hierarchical reduction of large symmetric games. In *Twentieth National Conference on Artificial Intelligence*, pages 502–508, Pittsburgh, 2005.
- M. P. Wellman, D. M. Reeves, K. M. Lochner, and R. Suri. Searching for Walverine 2005. In *Agent-Mediated Electronic Commerce: Designing Trading Agents and Mechanisms*, number 3937 in Lecture Notes on Artificial Intelligence, pages 157–170. Springer, 2006.
- M. P. Wellman, A. Greenwald, and P. Stone. *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition*. MIT Press, 2007.