Searching for Walverine 2005

Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, and Rahul Suri

University of Michigan
Ann Arbor, MI 48109-2110 USA
{wellman,dreeves,klochner,rsuri}@umich.edu

Abstract

We systematically explore a range of variations of our TAC travel-shopping agent, Walverine. The space of strategies is defined by settings to behavioral parameter values. Our empirical game-theoretic analysis is facilitated by approximating games through hierarchical reduction methods.

1 Introduction

There are many ways to play the TAC travel-shopping game. Our agent, Walverine [Cheng et al., 2005], employs competitive analysis to predict hotel prices and formulate an optimal bidding problem. Other agents take different approaches to predicting hotel prices [Wellman et al., 2004], bidding under uncertainty [Greenwald and Boyan, 2004], and many other facets of the TAC game. Even within a particular approach to a particular subproblem, there is no end to possible variations one might consider, ranging from fine-tuning of policy parameters to qualitatively different strategies.

Like most TAC participants, we apply a mix of modeling and experimentation in developing our agent. Since our models of the TAC environment necessarily simplify the actual game, we rely on experimental offline trials to validate the ideas and set parameters. And since these offline experiments incorporate assumptions about other agents' behavior, we also depend on online experiments (e.g., during preliminary tournament rounds) to test our designs in the most realistic setting available. Also like most other participants (with the notable exception of Vetsikas and Selman [2003], discussed below), our combination of modeling and experimentation was essentially *ad hoc*, with only informal procedures for fixing a particular agent behavior based on the results.

For 2005 (following a preliminary effort for 2004), we decided to adopt a more systematic approach. The first element of our method is fairly standard: specify a parametrized version of Walverine, defining a space of strategies under consideration. We then explore the space through extensive simulation. A less conventional element of our method is that we use the simulation results to estimate an *empirical game*, and apply standard game-theoretic analysis to derive strategic equilibria. The particularly novel element we introduce in the current work is *hierarchical game reduction*, a general technique for approximating symmetric games by smaller games with fractional numbers of agents. In this instance, we show that 4-player and 2-player reductions of the TAC game are far more manageable than the full 8-player game, and argue that little fidelity is lost by the reduction proposed here.

In the next section we illustrate the parametrization of strategy space by describing some of Walverine's key parameters. Section 3 appeals to the TAC literature to demonstrate the importance of accounting for strategic interactions in evaluating agent designs. We describe the explosion of strategy profile space in Section 4, and introduce our hierarchical reduction operator. Results from our empirical game-theoretic analysis to date are summarized in Section 5.[1]

2 Walverine Parameters

TAC travel-shopping is an 8-player symmetric game, with a complex strategy space and pivotal agent interactions. Strategies include all policies for bidding on flights, hotels, and entertainment over time, as a function of prior observations. To focus our search, we restrict attention to variations on our basic Walverine strategy [Cheng et al., 2005], as originally developed for TAC-02 and refined incrementally for 2003 and 2004.

We illustrate some of the possible strategy variations by describing some of the parameters we have exposed to the calling interface. To invoke an instance of Walverine, the user specifies parameter values dictating which version of the agent's modules to run, and what arguments to provide to these modules.

2.1 Flight Purchase Timing

Flight prices follow a random walk with a bias that is determined by a hidden parameter that is chosen randomly at the start of the game. Specifically, at the start of each game, a hidden parameter x is chosen from the integers in $[-10, 30]$. Define $x(t) = 10 + (t/9{:}00)(x - 10)$. Every 10 seconds thereafter, given elapsed time t, flight prices are perturbed by

[1] Another paper presenting the hierarchical game-reduction idea and appealing to the TAC case study was presented at AAAI-05 [Wellman et al., 2005]; some of the material in Sections 4 and 5 also appears in that work.

a value chosen uniformly, with bounds $[lb, ub]$ determined by

$$[lb, ub] = \begin{cases} [x(t), 10] & \text{if } x(t) < 0, \\ [-10, 10] & \text{if } x(t) = 0, \\ [-10, x(t)] & \text{if } x(t) > 0. \end{cases} \quad (1)$$

Whereas flight price perturbations are designed to increase in expectation given no information about the hidden parameter, conditional on this parameter prices may be expected to increase, decrease, or stay constant.

Walverine maintains a distribution $\Pr(x)$ for each flight, initialized to be uniform on [-10,30], and updated using Bayes's rule given the observed perturbations Δ at each iteration: $\Pr(x|\Delta) = \alpha \Pr(x) \Pr(\Delta|x)$, where α is a normalization constant.

Given this distribution over the hidden x parameter, the expected perturbation $E[\Delta'|x]$ for the next iteration is simply $(lb + ub)/2$, with bounds given by (1). Averaging over the distribution for x, we have $E[\Delta'] = \sum_x \Pr(x) E[\Delta'|x]$.

Given a set of flights that Walverine has calculated to be in the optimal package, it decides which to purchase now as a function of the expected perturbations, current holdings, and marginal flight values. On a high level, the strategy is designed to defer purchase of flights that are not quickly increasing, allowing for flexibility in avoiding expensive hotels as hotel price information is revealed.

The flight purchase strategy can be described in the form of a decision tree as depicted in Figure 1. First, Walverine compares the expected perturbation ($E[\Delta']$) with a threshold $T1$, deferring purchase if the prices are not expected to increase by $T1$ or more. If $T1$ is exceeded, Walverine next compares the expected perturbation with a second higher threshold, $T2$, and if the prices are expected to increase by more than $T2$ Walverine purchases all units for that flight that are in the optimal package.

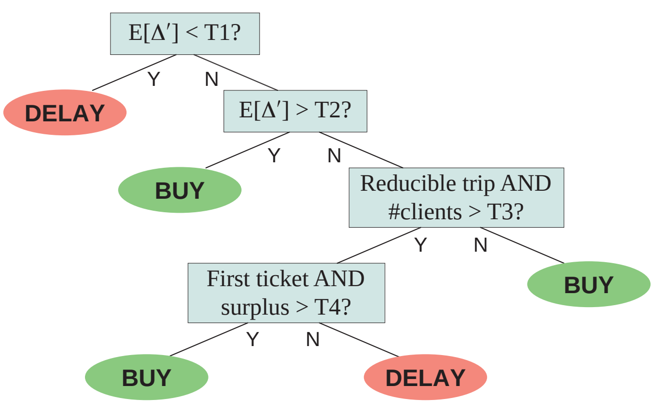

Figure 1: Decision tree for deciding whether to delay flight purchases.

If $T1 < E[\Delta'] < T2$, the Walverine flight delay strategy is designed to take into account the potential benefit of avoiding travel on high demand days. Walverine checks whether the flight constitutes one end of a *reducible* trip: one that spans more than a single day. If the trip is not reducible, Walverine buys all the flights. If reducible, Walverine considers its own demand (defined by the optimal package) for the day that would be avoided through shortening the trip, equivalent to the day of an inflight, and the day before an outflight. If our own demand for that day is $T3$ or fewer, Walverine purchases all the flights. Otherwise (reducible and demand greater than $T3$), Walverine delays the purchases, except possibly for one unit of the flight instance, which it will purchase if its marginal surplus exceeds another threshold, $T4$.

Though the strategy described above is based on sound calculations and tradeoff principles, it is difficult to justify particular settings of threshold parameters without making numerous assumptions and simplifications. Therefore we treat these as strategy parameters, to be explored empirically, along with the other Walverine parameters.

2.2 Bid Shading

The Walverine *optimal shading* algorithm [Cheng et al., 2005] identifies, for each hotel auction, the bid value maximizing expected utility based on a model of other agents' marginal value distributions. Because this optimization is based on numerous simplifications and approximations, we include several parameters to control its use.

Through a *shading mode* parameter, bid shading can be turned off, in which case Walverine bids its marginal value. Another parameter defines a *shade percentage*, specifying a fixed fraction to bid below marginal value. There are two modes corresponding to the optimal shading algorithm, differing in how they model the other agents' value distributions. In the first, the distributions are derived from a simplified competitive analysis. For this mode, another parameter, *shade model threshold* turns off shading in case the model appears too unlikely given the price quote. Specifically, we calculate the probability that the 16th highest bid is greater than or equal to the quote according to the modeled value distributions, and if too low we refrain from using the model for shading. For the second optimal shading mode, instead of the competitive model we employ empirically derived distributions keyed on the hotel closing order.

2.3 Other Parameters

Walverine predicts hotel prices based on competitive equilibrium analysis [Wellman et al., 2004]. The result, however, does not account for uncertainty in the predictions. We developed a simple method to hedge on our price estimates, by assigning an *outlier probability* to the event that a hotel price will be much greater than predicted. We can hedge to a greater or lesser degree by modifying this outlier parameter.

Given a price distribution, one could optimize bids with respect to the distribution itself, or with respect to the *expected* prices induced by the distribution. Although the former approach is more accurate in principle, necessary compromises in implementation render it ambiguous in practice which produces superior results [Greenwald and Boyan, 2004, Stone et al., 2003, Wellman et al., 2004]. Thus, we include a parameter controlling which method to apply in Walverine.

Several agent designers have reported employing *priceline* predictions, accounting for the impact of one's own demand quantity on price. We implemented a version of the *completion algorithm* [Boyan and Greenwald, 2001] that optimizes with respect to pricelines, and included it as a Walverine option. A further parameter selects how price predictions and optimizations account for outstanding hotel bids in determining current holdings. In one setting current bids for open hotel auctions are ignored, and in another the current hypothetical winnings are treated as actual holdings.

Finally, we choose among a discrete set of policies for trading entertainment. As a baseline, we implemented the strategy employed by livingagents in TAC-01 [Fritschi and Dorer, 2002]. We also applied reinforcement learning to derive policies from scratch, expressed as functions of marginal valuations and various additional state variables. The policy employed by Walverine in TAC-02 was derived by Q-learning over a discretized state space. For TAC-03 we learned an alternative policy, this time employing a neural network to represent the value function.

3 Strategic Interactions in TAC Travel

TAC agents interact in the markets for each kind of good, as competing buyers or potential trading partners. Based on published accounts, TAC participants design agents given specified game rules, and then test these designs in the actual tournaments as well as offline experiments. The testing process is crucial, given the lack of any compact analytical model of the domain. During testing, agent designers explore variations on their agent program, for example by tuning parameters or toggling specific agent features.

That strategic choices interact, and implications for design and evaluation, have been frequently noted in the TAC literature. In a report on the first TAC tournament, Greenwald and Stone [2001] observe that the strategy of bidding high prices for hotels performed reasonably in preliminary rounds, but poorly in the finals when more agents were high bidders (thus raising final prices to unprofitable levels). Stone et al. [2001] evaluate their agent ATTac-2000 in controlled post-tournament experiments, measuring relative scores in a range of contexts, varying the number of other agents playing high- and low-bidding strategies. A report on the 2001 competition [Wellman et al., 2003b] concludes that the top scorer, livingagents, would perform quite poorly against copies of itself. The designers of SouthamptonTAC [He and Jennings, 2002] observed the sensitivity of their agent's TAC-01 performance to the tendency of other agents to buy flights in advance, and redesigned their agent for TAC-02 to attempt to classify the competitive environment faced and adapt accordingly [He and Jennings, 2003]. ATTac-2001 explicitly took into account the identity of other agents in training its price-prediction module [Stone et al., 2003]. To evaluate alternative learning mechanisms through post-competition analysis, Stone et al. recognized the effect of the policies on the outcomes being learned, and thus adopted a carefully phased experimental design in order to account for such effects.

One issue considered by several TAC teams is how to bid for hotels based on predicted prices and marginal utility. Greenwald and Boyan [2004] have studied this in depth, performing pairwise comparisons of four strategies, in profiles with four copies of each agent.[2] Their results indicate that absolute performance of a strategy indeed depends on what the other agent plays. We examined the efficacy of bid shading in Walverine, varying the number of agents employing shading or not, and presented an equilibrium shading probability based on these results [Wellman et al., 2003a].

By far the most extensive experimental analysis reported for TAC travel-shopping to date is that performed by Vetsikas and Selman [2003]. In the process of designing Whitebear for TAC-02, they first identified candidate policies for separate elements of the agents' overall strategy. They then defined extreme (boundary) and intermediate values for these partial strategies, and performed experiments according to a systematic and deliberately considered methodology. Specifically, for each run, they fix a particular number of agents playing intermediate strategies, varying the mixture of boundary cases across the possible range. In all, the Whitebear experiments comprised 4500 profiles, with varying *even* numbers of candidate strategies (i.e., profiles of the 4-player game). Their design was further informed by 2000 games in the preliminary tournament rounds. This systematic exploration was apparently helpful, as Whitebear was the top scorer in the 2002 tournament. This agent's predecessor version placed third in TAC-01, following a less comprehensive and structured experimentation process. Its successor placed third again in 2003, and regained its first-place standing in 2004. Since the rules were adjusted for TAC-04, this most recent outcome required a new regimen of experiments.

4 Hierarchical Game Reduction

4.1 Motivation

Suppose that we manage to narrow down the candidate Walverine variants to a reasonable number of strategies (say 35). Because the performance of a strategy for one agent depends on the strategies of the other seven, we wish to undertake a game-theoretic analysis of the situation. Determining the payoff for a particular strategy profile is expensive, however, as each game instance takes nine minutes to run, plus another minute or two to calculate scores, compile results, and set up the next simulation. Moreover, since the environment is stochastic, numerous samples (say 12) are required to produce a reliable estimate for even one profile. At roughly two hours per profile, exhaustively exploring profile space will require $2 \cdot 35^8$ or 4.5 trillion hours simply to estimate the payoff function representing the game under analysis. If the game is symmetric, we can exploit that fact to reduce the number of distinct profiles to $\binom{42}{8}$, which will require 236 million hours. That is quite a bit less, but still much more time than we have.

The idea of hierarchical game reduction is that although a strategy's payoff does depend on the play of other agents (otherwise we are not in a game situation at all), it may be relatively insensitive to the exact numbers of other agents playing particular strategies. For example, let $(s, k; s')$ denote a profile where k other agents play strategy s, and the rest play s'. In many natural games, the payoff for playing any particular strategy against this profile will vary smoothly with k, differing only incrementally for contexts with $k \pm 1$. If such

[2] In our terminology introduced below, their trials focused on the 2-player reduced version of the game.

is the case, we sacrifice relatively little fidelity by restricting attention to subsets of profiles, for instance those with only even numbers of any particular strategy. To do so essentially transforms the N-player game to an $N/2$-player game over the same strategy set, where the payoffs to a profile in the reduced game are simply those from the original game where each strategy in the reduced profile is played twice.

The potential savings from reduced games are considerable, as they contain combinatorially fewer profiles. The 4-player approximation to the TAC game (with 35 strategies) comprises 73,815 distinct profiles, compared with 118 million for the original 8-player game. In case exhaustive consideration of the 4-player game is still infeasible, we can approximate further by a corresponding 2-player game, which has only 630 profiles. Approximating by a 1-player game is tantamount to ignoring strategic effects, considering only the 35 profiles where the strategies are played against themselves. In general, an N-player symmetric game with S strategies includes $\binom{N+S-1}{N}$ distinct profiles. Figure 2 shows the exponential growth in both N and S.

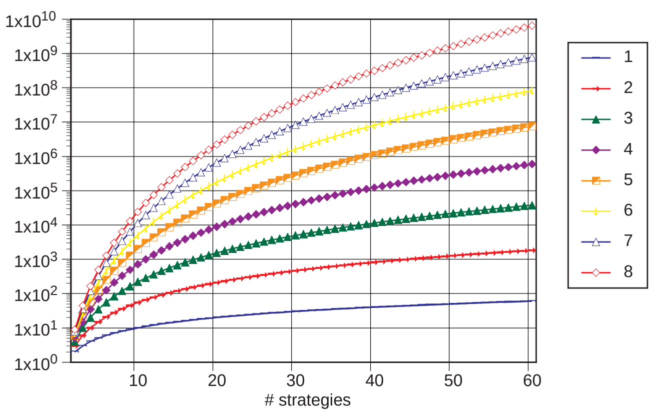

Figure 2: Number of distinct profiles (log scale) of a symmetric game, for various numbers of players and strategies.

4.2 Hierarchy of Reduced Games

We develop our hierarchical reduction concepts in the framework of *symmetric normal-form games*.

Definition 1 $\Gamma = \langle N, \{S_i\}, \{u_i()\}\rangle$ *is an* N-player normal-form game, *with* strategy set S_i *the available strategies for player* i*, and the* payoff function $u_i(s_1, \ldots, s_N)$ *giving the utility accruing to player* i *when players choose the* strategy profile $(s_1, \ldots, s_N)$.

Definition 2 *A normal-form game is* symmetric *if the players have identical strategy spaces* ($S_1 = \cdots = S_N = S$) *and* $u_i(s_i, s_{-i}) = u_j(s_j, s_{-j})$*, for* $s_i = s_j$ *and* $s_{-i} = s_{-j}$ *for all* $i, j \in \{1, \ldots, N\}$*. Thus we can write* $u(t, s)$ *for the payoff to any player playing strategy* t *when the remaining players play profile* s*. We denote a symmetric game by the tuple* $\langle N, S, u()\rangle$*.*

Our central concept is that of a *reduced* game.

Definition 3 *Let* $\Gamma = \langle N, S, u()\rangle$ *be an* N*-player symmetric game, with* $N = pq$ *for integers* p *and* q*. The* p-player reduced version of Γ, *written* $\Gamma\downarrow_p$*, is given by* $\langle p, S, \hat{u}()\rangle$*, where*

$$\hat{u}_i(s_1, \ldots, s_p) = u_{q \cdot i}(\underbrace{s_1, \ldots}_{q}, \underbrace{s_2, \ldots}_{q}, \ldots, \underbrace{s_p, \ldots}_{q}).$$

In other words, the payoff function in the reduced game is obtained by playing the specified profile in the original q *times.*

The idea of a reduced game is to coarsen the profile space by restricting the degrees of strategic freedom. Although the original set of strategies remains available, the number of agents playing any strategy must be a multiple of q. Every profile in the reduced game is one in the original game, of course, and any profile in the original game can be reached from a profile contained in the reduced game by changing at most $p(q-1)$ agent strategies.

The premise of our approach is that the reduced game will often serve as a good approximation of the full game it abstracts. We know that in the worst case it does not. In general, an equilibrium of the reduced game may be arbitrarily far from equilibrium with respect to the full game, and an equilibrium of the full game may not have any near neighbors in the reduced game that are close to equilibrium there. Elsewhere we provide evidence that the hierarchical reduction provides an effective approximation in several natural game classes [Wellman et al., 2005]. Intuition suggests that it should apply for TAC, and the basic agreement between TAC$\downarrow_2$ and TAC$\downarrow_4$ seen in our results tends to support that assessment.

5 TAC Experiments

To apply reduced-game analysis to the TAC domain, we identified a restricted set of strategies, defined by setting parameters for Walverine. We considered a total of 35 distinct strategies, covering variant policies for bidding on flights, hotels, and entertainment. We collected data for a large number of games: over 37,000 as of this writing, representing over ten months of (almost continuous) simulation.[3] Each game instance provides a sample payoff vector for a profile over our restricted strategy set.

Table 1 shows how our dataset is apportioned among the 1-, 2-, and 4-player reduced games. We are able to exhaustively cover the 1-player game, of course. We could also have exhausted the 2-player profiles, but chose to skip some of the less promising ones (around one-third) in favor of devoting more samples elsewhere. The available number of samples could not cover the 4-player games, but as we see below, even 2.3% is sufficient to draw conclusions about the possible equilibria of the game. Spread over the 8-player game, however, 37,000 instances would be insufficient to explore much, and so we refrain from any sampling of the unreduced game.

In the spirit of hierarchical exploration, we sample more instances per profile as the game is further reduced, obtaining

[3] Our simulation testbed comprises two dedicated workstations to run the agents, another RAM-laden four-CPU machine to run the agents' optimization processes, a share of a fourth machine to run the TAC game server, and background processes on other machines to control the experiment generation and data gathering.

p	Profiles			Samples/Profile	
	total	evaluated	%	min	mean
4	73,815	1775	2.4	10	20.8
2	630	467	74.1	15	31.1
1	35	35	100.0	20	91.5

Table 1: Profiles evaluated, reduced TAC games (TAC$\downarrow_p$).

more reliable statistical estimates of the coarse background relative to its refinement. On introducing a new profile we generate a minimum required number of samples, and subsequently devote further samples to particular profiles based on their potential for influencing our game-theoretic analysis. The sampling policy employed was semi-manual and somewhat *ad hoc*, driven in an informal way by analyses of the sort described below on intermediate versions of the dataset. Developing a fully automated and principled sampling policy is the subject of future research.

5.1 Control Variates

Since we estimate the payoffs (expected scores) by Monte Carlo simulation, there are several off-the-shelf variance reduction techniques that can be applied. One is the method of *control variates* [Ross, 2002], which improves the estimate of the mean of a random function by exploiting correlation with observable random variables. In our case the function is the entire game server plus eight agents playing a particular strategy profile, evaluating to a vector of eight scores. Random factors in the game include hotel closing order, flight prices, entertainment ticket endowment, and, most critically, client preferences. The idea is to replace sampled scores with scores that have been "adjusted for luck". For example, an agent whose clients had anomolously low hotel premiums would have its score adjusted upward as a handicap. Or in a game with very cheap flight prices, all the scores would be adjusted downward to compensate. Such adjustments reduce variance at the cost of potentially introducing bias. Fortunately, the bias goes to zero as the number of samples increases [L'Ecuyer, 1994].

For the analysis reported here, we adjust scores based on the following control variables (for a hypothetical agent A):

- ENT: Sum of A's clients' entertainment premiums ($8 \cdot 3 = 24$ values). $E[\mathrm{ENT}] = 2400$.
- FLT: Sum of initial flight quotes (8 values; same for all agents). $E[\mathrm{FLT}] = 2600$.
- WTD: Weighted total demand: Total demand vector (for each night, the number of the 64 clients who would be there that night if they got their preferred trips) dotted with the demand vector for A's clients. $E[\mathrm{WTD}] \approx 540.1$.
- HTL: Sum of A's clients' hotel premiums (8 values). $E[\mathrm{HTL}] = 800$.

The expectations are determined analytically except for $E[\mathrm{WTD}]$ which we estimated via 17,520 Monte Carlo samples from the client preference distribution.

Given the above, we adjust an agent's score by subtracting

$$\begin{aligned}\beta_{\mathrm{ENT}}(\mathrm{ENT} - E[\mathrm{ENT}]) \quad &+ \quad \beta_{\mathrm{FLT}}(\mathrm{FLT} - E[\mathrm{FLT}]) \\ &+ \quad \beta_{\mathrm{WTD}}(\mathrm{WTD} - E[\mathrm{WTD}]) \\ &+ \quad \beta_{\mathrm{HTL}}(\mathrm{HTL} - E[\mathrm{HTL}]),\end{aligned}$$

where the βs are determined by performing a multiple regression from $[\mathrm{ENT}, \mathrm{FLT}, \mathrm{WTD}, \mathrm{HTL}]$ to score using a data set consisting of 2190 games. Using adjusted scores in lieu of raw scores reduces overall variance by 22%.

5.2 Results

The detailed results and game-theoretic analysis of our experiments is presented elsewhere [Wellman et al., 2005]. Here we provide only a brief summary. A final account and discussion of how the experiments led to our choice of Walverine to run in the 2005 tournament will be presented at the workshop.

Analysis of the TAC$\downarrow_1$ "game" tells us which strategy performs best assuming it plays with copies of itself. We included a strategy (S34) designed to do well in this context: it shades all hotel bids by a fixed 50% rate. This indeed performs best, by about 250 points, since the result is very low hotel prices. However, the profile is quite unstable, as an agent who shades less can get much better hotel rooms, but still benefit from the low prices. Thus, this is not nearly an equilibrium in the less-reduced games.

With 70% of profiles evaluated, we have a reasonably complete description of the two-player game, TAC $\downarrow_2$, among our 35 strategies. We have identified five pure-strategy Nash equilibrium candidates, and 29 symmetric mixed-strategy profiles in equilibrium. Less than 1/3 of the considered strategies participate with probability exceeding 0.15 in some equilibrium found.

Results for TAC$\downarrow_4$ are still quite tentative. We have identified a few good candidate equilibrium mixtures over pairs of strategies. Further simulation in the next few months may confirm or refute these, or identify additional candidates. In general, strategies and combinations evaluated as stable in TAC$\downarrow_2$ tend to produce similar results in TAC$\downarrow_4$.

Analysis of the various reduced games does validate the importance of strategic interactions. As noted above, the best strategy in self-play, S34, is not nearly a best response in most other environments, though it does appear in a few mixed-strategy equilbria of TAC$\downarrow_2$. Strategy S34 achieves a payoff of 4302 in self-play. For comparison:

- The top scorer in the 2004 tournament, whitebear, averaged 4122.
- The best payoff we have found (so far) in TAC$\downarrow_2$ in a two-action mixed-strategy equilibrium candidate is 4220 (and this involves playing S34 with probability 0.4).
- The best corresponding equilibrium payoff we have found in TAC$\downarrow_4$ is 4031. No such equilibrium includes S34.

)

6 Conclusion

Given all this simulation and analysis, can we now identify the "best" strategy to play in TAC? We do have strong evidence for expecting that all but a fraction of the original 35 strategies will turn out to be unstable within this set. The supports of candidate equilibria tend to concentrate on a fraction of the strategies, suggesting we may limit consideration to this group.

The exact method that we used to determine our TAC-05 play based on this data will be reported at the workshop. However, we are committed to employing the current analysis to select candidate strategies for testing in preliminary rounds. Which will be played in the final tournament will depend on performance in games with the actual field of entrants.

Acknowledgments

Shih-Fen Cheng, Kevin O'Malley, and Julian Schvartzman have also contributed to the 2005 version of Walverine and our simulation process. We thank the anonymous reviewers for constructive comments. This work was supported in part by the National Science Foundation under grants IIS-0205435 and IIS-0414710.

References

Justin Boyan and Amy Greenwald. Bid determination in simultaneous auctions: An agent architecture. In *Third ACM Conference on Electronic Commerce*, pages 210–212, Tampa, FL, 2001.

Shih-Fen Cheng, Evan Leung, Kevin M. Lochner, Kevin O'Malley, Daniel M. Reeves, and Michael P. Wellman. Walverine: A Walrasian trading agent. *Decision Support Systems*, 39:169–184, 2005.

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

Amy Greenwald and Justin Boyan. Bidding under uncertainty: Theory and experiments. In *Twentieth Conference on Uncertainty in Artificial Intelligence*, pages 209–216, Banff, 2004.

Amy Greenwald and Peter Stone. The first international trading agent competition: Autonomous bidding agents. *IEEE Internet Computing*, 5(2):52–60, 2001.

Minghua He and Nicholas R. Jennings. SouthamptonTAC: Designing a successful trading agent. In *Fifteenth European Conference on Artificial Intelligence*, pages 8–12, Lyon, 2002.

Minghua He and Nicholas R. Jennings. SouthamptonTAC: An adaptive autonomous trading agent. *ACM Transactions on Internet Technology*, 3:218–235, 2003.

Pierre L'Ecuyer. Efficiency improvement and variance reduction. In *Twenty-Sixth Winter Simulation Conference*, pages 122–132, Orlando, FL, 1994.

Sheldon M. Ross. *Simulation*. Academic Press, third edition, 2002.

Peter Stone, Michael L. Littman, Satinder Singh, and Michael Kearns. ATTac-2000: An adaptive autonomous bidding agent. *Journal of Artificial Intelligence Research*, 15:189–206, 2001.

Peter Stone, Robert E. Schapire, Michael L. Littman, János A. Csirik, and David McAllester. Decision-theoretic bidding based on learned density models in simultaneous, interacting auctions. *Journal of Artificial Intelligence Research*, 19:209–242, 2003.

Ioannis A. Vetsikas and Bart 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.

Michael P. Wellman, Shih-Fen Cheng, Daniel M. Reeves, and Kevin M. Lochner. Trading agents competing: Performance, progress, and market effectiveness. *IEEE Intelligent Systems*, 18(6):48–53, 2003a.

Michael P. Wellman, Amy Greenwald, Peter Stone, and Peter R. Wurman. The 2001 trading agent competition. *Electronic Markets*, 13:4–12, 2003b.

Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, Shih-Fen Cheng, and Rahul Suri. Approximate strategic reasoning through hierarchical reduction of large symmetric games. In *Twentieth National Conference on Artificial Intelligence*, Pittsburgh, 2005.

Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, and Yevgeniy Vorobeychik. Price prediction in a trading agent competition. *Journal of Artificial Intelligence Research*, 21:19–36, 2004.