Robust Solutions in Stackelberg Games: Addressing Boundedly Rational Human Preference Models

Manish Jain, Fernando Ordóñez, James Pita, Christopher Portway, Milind Tambe, Craig Western
***Praveen Paruchuri,** and ****Sarit Kraus**
University of Southern California, Los Angeles, CA 90089
*Intelligent Automation, Inc., RockVille, MD 20855
**Bar-llan University, Ramat-Gan 52900, Israel
Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742

Abstract

Stackelberg games represent an important class of games in which one player, the leader, commits to a strategy and the remaining players, the followers, make their decision with knowledge of the leader's commitment. Existing algorithms for Bayesian Stackelberg games find optimal solutions while modeling uncertainty over follower types with an *a-priori* probability distribution. Unfortunately, in real-world applications, the leader may also face uncertainty over the follower's response which makes the optimality guarantees of these algorithms fail. Such uncertainty arises because the follower's specific preferences or the follower's observations of the leader's strategy may not align with the rational strategy, and it is not amenable to a-priori probability distributions. These conditions especially hold when dealing with human subjects. To address these uncertainties while providing quality guarantees, we propose three new robust algorithms based on mixed-integer linear programs (MILPs) for Bayesian Stackelberg games. A key result of this paper is a detailed experimental analysis that demonstrates that these new MILPs deal better with human responses: a conclusion based on 800 games with 57 human subjects as followers. We also provide run-time results on these MILPs.

Introduction

In Stackelberg games, one player, the leader, commits to a strategy publicly before the remaining players, the followers, make their decision (Fudenberg & Tirole 1991). Stackelberg games are important in many multiagent security domains such as attacker-defender scenarios and patrolling (Brown *et al.* 2006; Paruchuri *et al.* 2007). For example, security personnel patrolling infrastructure decide on a patrolling strategy first, before their adversaries act taking this commited strategy into account. Indeed, Stackelberg games are at the heart of the ARMOR system deployed at the Los Angeles International Airport to schedule security personnel (Murr 2007; Paruchuri *et al.* 2008). Moreover, these games have potential applications for network routing, pricing in transportation systems and many others (Korilis, Lazar, & Orda 1997; Cardinal *et al.* 2005).

Existing algorithms for Bayesian Stackelberg games find optimal solutions considering an *a-priori* probability distribution over possible follower types (Conitzer & Sandholm 2006; Paruchuri *et al.* 2007; 2008). Unfortunately, to guarantee optimality, these algorithms make strict assumptions on the underlying games, in particular that the players are perfectly rational and that the followers perfectly observe the leader strategy. However, these assumptions rarely hold in real-world domains. Of particular interest are the security domains mentioned earlier — even though an automated program may determine the strategy of the leader (security personnel), they face a human adversary. As is well known, such human adversaries may not be utility maximizers, computing optimal decisions. Instead, their preference models may be governed by their bounded rationality (Simon 1956). Followers may also have limited observability of the security personnel's strategy. Thus a follower may not provide the game theoretic rational choice, but rather may have another preference based on bounded rationality or uncertainty, and cause the leader to face uncertainty over the gamut of follower's actions. Therefore, in general, the leader in a Stackelberg game must commit to a strategy considering three different types of uncertainty, where no prior probability distribution is available for the first two types: (i) follower response uncertainty due to its bounded rationality, where the follower may not choose utility maximizing optimal strategy; (ii) follower response uncertainty due to its errors in observing the leader's strategy; (iii) follower reward uncertainty modeled as different reward matrices with a Bayesian *a-priori* distribution assumption, i.e. a Bayesian Stackelberg game. While existing algorithms handle the third type of uncertainty (Paruchuri *et al.* 2007; Conitzer & Sandholm 2006; Paruchuri *et al.* 2008), the optimality guarantees of these algorithms fail when faced with the first two types of uncertainty, and the leader reward may degrade unpredictably.

To overcome this limitation, we propose three new algorithms based on mixed-integer linear programs (MILPs) that provide *robust* solutions, i.e. they provide quality guarantees despite uncertainty over the follower's choice of actions due to its bounded rationality or observational uncertainty. Our new robust MILPs complement the prior algorithms for Bayesian Stackelberg games, handling all three types of uncertainty mentioned above. We provide run-time results and extensive experiments for 800 games with 57 human followers. The key result of this paper is to show that our new MILPs, while not optimal in a game-theoretic sense, perform better against human followers, who as is well known,

are not utility maximizers.

Background

Stackelberg Game: In a Stackelberg game, a leader commits to a strategy first, and then a follower optimizes its reward, *considering the action chosen by the leader.* To see the advantage of being the leader in a Stackelberg game, consider the game with the payoff as shown in Table 1. The leader is the row player and the follower is the column player. The only pure-strategy Nash equilibrium for this game is when the leader plays *a* and the follower plays *c* which gives the leader a payoff of 2. However, if the leader commits to a mixed strategy of playing *a* and *b* with equal (0.5) probability, then the follower will play *d*, leading to a higher expected payoff for the leader of 3.5.

	c	d
a	2,1	4,0
b	1,0	3,2

Table 1: Payoff table for example Stackelberg game.

Bayesian Stackelberg Game: In a Bayesian game of N agents, each agent n must be one of a given set of types. For the two player Stackelberg games, inspired by the security domain of interest in this paper we assume there is only one leader type (e.g. only one police force enforcing security), although there are multiple follower types (e.g. multiple types of adversaries), denoted by $l \in L$. However, the leader does not know the follower's type. For each agent (leader or follower) n, there is a set of strategies σ_n and a utility function $u_n : L \times \sigma_1 \times \sigma_2 \rightarrow \Re$. Our goal is *to find the optimal mixed strategy* for the leader given that the follower knows this strategy when choosing its own strategy.

DOBSS: While the problem of choosing an optimal strategy for the leader in a Stackelberg game is NP-hard for a Bayesian game with multiple follower types (Conitzer & Sandholm 2006), researchers have continued to provide practical improvements. DOBSS is currently the most efficient algorithm for such games (Paruchuri *et al.* 2008) and in use for security scheduling at the Los Angeles International Airport. It operates directly on the compact Bayesian representation, giving exponential speedups over (Conitzer & Sandholm 2006) which requires conversion of the Bayesian game into a normal-form game by the Harsanyi transformation (Harsanyi & Selten 1972). Furthermore, unlike the approximate approach of (Paruchuri *et al.* 2007), DOBSS provides an exact optimal solution.

We present DOBSS first in its more intuitive form as a mixed-integer quadratic program (MIQP) and then show its linearization into an MILP. DOBSS finds the optimal mixed strategy for the leader while considering an *optimal* follower response for this leader strategy. Note that we need to consider only the reward-maximizing pure strategies of the followers, since if a mixed strategy is optimal for the follower, then so are all the pure strategies in the support of that mixed strategy. We denote by x the leader's policy, which consists of a vector of the leader's pure strategies. The value x_i is the proportion of times in which pure strategy i is used in the policy. For a follower type $l \in L$, q^l denotes its vector of strategies, and R^l and C^l the payoff matrices for the leader and the follower respectively, given this follower type l. Furthermore, X and Q denote the index sets of the leader and follower's pure strategies, respectively. Let M be a large positive number. Given *a priori* probabilities p^l, with $l \in L$, of facing each follower type, the leader solves the following problem (Paruchuri *et al.* 2008):

$$\begin{array}{ll} \max_{x,q,a} & \sum_{i\in X}\sum_{l\in L}\sum_{j\in Q} p^l R^l_{ij} x_i q^l_j \\ \text{s.t.} & \sum_{i\in X} x_i = 1 \\ & \sum_{j\in Q} q^l_j = 1 \\ & 0 \le (a^l - \sum_{i\in X} C^l_{ij} x_i) \le (1 - q^l_j)M \\ & x_i \in [0 \dots 1] \\ & q^l_j \in \{0,1\} \\ & a \in \Re \end{array} \quad (1)$$

Where for a set of leader's actions x and actions q^l for each follower type, the objective represents the expected reward for the leader considering the *a-priori* distribution over different follower types p^l. Constraints 1 and 4 define the set of feasible solutions x as probability distributions over the set of actions X. Constraints 2 and 5 limit the vector q^l of actions of follower type l to be a pure distribution over the set Q (i.e., each q^l has exactly one coordinate equal to one and the rest equal to zero). The two inequalities in constraint 3 ensure that $q^l_j = 1$ only for a strategy j that is optimal for follower type l. In particular, the leftmost inequality ensures that for all $j \in Q$, $a^l \ge \sum_{i\in X} C^l_{ij} x_i$, which means that given the leader's vector x, a^l is an upper bound on follower type l's reward for any action. The rightmost inequality is inactive for every action where $q^l_j = 0$, since M is a large positive quantity. For the action that has $q^l_j = 1$ this inequality states that the follower's payoff for this action must be $\ge a^l$, which combined with the previous inequality shows that this action must be optimal for follower type l. We can linearize the quadratic programming problem 1 through the change of variables $z^l_{ij} = x_i q^l_j$, thus obtaining the following equivalent MILP (Paruchuri *et al.* 2008) :

$$\begin{array}{ll} \max_{q,z,a} & \sum_{i\in X}\sum_{l\in L}\sum_{j\in Q} p^l R^l_{ij} z^l_{ij} \\ \text{s.t.} & \sum_{i\in X}\sum_{j\in Q} z^l_{ij} = 1 \\ & \sum_{j\in Q} z^l_{ij} \le 1 \\ & q^l_j \le \sum_{i\in X} z^l_{ij} \le 1 \\ & \sum_{j\in Q} q^l_j = 1 \\ & 0 \le (a^l - \sum_{i\in X} C^l_{ij}(\sum_{h\in Q} z^l_{ih})) \le (1 - q^l_j)M \\ & \sum_{j\in Q} z^l_{ij} = \sum_{j\in Q} z^1_{ij} \\ & z^l_{ij} \in [0 \dots 1] \\ & q^l_j \in \{0,1\} \\ & a \in \Re \end{array} \quad (2)$$

Robust Algorithms

There are two fundamental assumptions underlying current algorithms for Stackelberg games, including DOBSS. First,

the follower is assumed to act with perfect utility maximizing rationality, choosing the absolute optimal among its strategies. Second if the follower faces a tie in its strategies' rewards, it will break it in favor of the leader, choosing the one that gives a higher reward to the leader. This standard assumption is also shown to follow from the follower's rationality and optimal response under some conditions (von Stengel & Zamir 2004). Unfortunately, in many real-world domains the follower does not respond optimally: this is due to the follower's preferences, which arise from bounded rationality, or its uncertainty. In essence, the leader faces uncertainty over follower responses — the follower may not choose the optimal but from a range of possible responses — potentially significantly degrading leader rewards. For example, in Table 1, despite the leader's committing to a strategy of playing *a* and *b* with equal probability, if the follower plays *c* instead of its optimal *d*, the leader's reward degrades from 3.5 to 1.5. Notice that no *a-priori* probability distributions are available or assumed for this follower response uncertainty.

To remedy this situation, we draw inspiration from robust optimization methodology, in which the decision maker optimizes against the worst outcome over the uncertainty (Aghassi & Bertsimas 2006; Nilim & Ghaoui 2004). In our Stackelberg problem the leader will make a robust decision considering that the boundedly rational follower could choose a strategy from its range of possible responses, or with imperfect observations of leader strategy, that degrades the leader rewards the most. We introduce three mixed-integer linear programs (MILPs) to that end. Our first MILP, BRASS (Bounded Rationality Assumption in Stackelberg Solver) addresses follower's bounded rationality. Our second robust MILP, BOSS (Bounded Observability in Stackelberg Solver), is a heuristic approach for robust leader strategy despite the follower's observational uncertainty. Our third MILP, MAXIMIN, provides a robust response no matter the uncertainty faced.

BRASS

When employing this MILP, we assume a boundedly rational follower who does not strictly maximize utility. As a result, the follower may select a ε-optimal response strategy, i.e. the follower may choose any of the responses within ε-reward of the optimal strategy. Given multiple ε-optimal responses, the robust approach is to assume that the follower could choose the one that provides the leader the worst reward — not necessarily because the follower attends to the leader's reward, but to robustly guard against the worst case outcome. This worst case assumption contrasts with those of other Stackelberg solvers that given a tie the follower will choose a strategy that favors the leader (Conitzer & Sandholm 2006; Paruchuri *et al.* 2007; 2008). The following MILP maximizes the minimum reward that we obtain given such a worst-case assumption.

In the following MILP, we use the same variable notation as in MILP (1). In addition, the variables h_j^l identify the optimal strategy for follower type l with a value of a^l in constraints 3 and 4. Variables q_j^l represent all ε optimal strategies for follower type l; the second constraint now allows selection of more than one policy per follower type. The fifth constraint ensures that $q_j^l = 1$ for every action j such that $a^l - \sum_{i \in X} C_{ij}^l < \varepsilon$, since in this case the middle term in the inequality is $< \varepsilon$ and the left inequality is then only satisfied if $q_j^l = 1$. The sixth constraint helps define the worst objective value against follower type l, γ_l, which has to be lower than any leader reward for all actions $q_j^l = 1$. Note that in this case we do not have a quadratic objective so no linearization step is needed.

$$
\begin{array}{ll}
\max_{x,q,h,a,\gamma} & \sum_{l \in L} p^l \gamma_l \\
\text{s.t.} & \sum_{i \in X} x_i = 1 \\
& \sum_{j \in Q} q_j^l \geq 1 \\
& \sum_{j \in Q} h_j^l = 1 \\
& 0 \leq (a^l - \sum_{i \in X} C_{ij}^l x_i) \leq (1 - h_j^l) M \\
& \varepsilon (1 - q_j^l) \leq a^l - \sum_{i \in X} C_{ij}^l x_i \leq \varepsilon + (1 - q_j^l) M \\
& M(1 - q_j^l) + \sum_{i \in X} R_{ij}^l x_i \geq \gamma_l \\
& h_j^l \leq q_j^l \\
& x_i \in [0 \ldots 1] \\
& q_j^l, h_j^l \in \{0, 1\} \\
& a \in \Re
\end{array}
\tag{3}
$$

Proposition 1 *The optimal reward of BRASS is decreasing in ε.*

Proof sketch: Since the fifth constraint in (3) makes $q_j^l = 1$ when that action has a follower reward between $(a^l - \varepsilon, a^l]$, increasing ε would increase the number of follower strategies set to 1. Having more active follower actions in constraint 6 can only decrease the minimum value γ_l. ■

BOSS

BOSS considers the case where the follower may deviate from the optimal response because it obtains garbled or limited observations. Thus, the follower's model of the leader's strategy may deviate by δ_i from the exact strategy x_i that the leader is playing causing a non-optimal response. Using the robust approach we consider that the follower could select the strategy that degrades the leader reward the most out of the strategies possible because of the observational uncertainty. The following MILP finds the optimal leader strategy given a bounded error δ_i in the observations:

$$
\begin{array}{ll}
\max & \sum_{l \in L} p^l \gamma_l \\
\text{s.t.} & \sum_{i \in X} x_i = 1 \\
& z_i^k = x_i + \delta_i^k \\
& \sum_{j \in Q} q_j^{lk} = 1 \\
& 0 \leq a^{lk} - \sum_{i \in X} C_{ij}^l z_i^k \leq (1 - q_j^{lk}) M \\
& M(1 - q_j^{lk}) + \sum_{i \in X} R_{ij}^l x_i \geq \gamma_l \\
& x_i \in [0 \ldots 1] \\
& q_j^{lk} \in \{0, 1\} \\
& a \in \Re
\end{array}
\tag{4}
$$

MILP (4) is a heuristic approach that discretely samples the space of strategies that the follower may observe given its obseration errors; vector z^k represents the k-th erroneous observation by the follower. The errors in observation for the k-th observation are pre-defined and are specified by the vector δ^k, where $\sum_i \delta_i^k = 0$. Constraint 4 in the above MILP sets q_j^{lk} to 1 if and only if j is the best response of follower type l to the observed leader strategy z^k. As in BRASS, γ helps us define the worst objective value against follower type l. Then, in the fifth constraint, we maximize against all the q_j^{lk}'s that have been set to 1, thus maximizing our reward against any of the possible best follower responses for various possibly erroneous observations.

MAXIMIN

If we combine the uncertainty in the follower's response due to the follower's bounded rationality and limited observability, the uncertainty over the follower's response grows significantly — the follower might potentially take one of a very large set of actions. The MAXIMIN approach considers the possibility that the follower may indeed choose any one of its actions. The objective of the following LP is to maximize the minimum reward γ the leader will obtain irrespective of the follower's action.

$$\begin{array}{ll} \max & \sum_{l \in L} p^l \gamma_l \\ \text{s.t.} & \sum_{i \in X} x_i = 1 \\ & \sum_{i \in X} R_{ij}^l x_i \geq \gamma_l \\ & x_i \in [0 \ldots 1] \end{array} \tag{5}$$

Complexity: BRASS and BOSS, like DOBSS, require the solution of a MILP, whereas MAXIMIN is a linear programming problem. Therefore the complexity of MAXIMIN is polynomial. BRASS and BOSS on the other hand, like DOBSS, face an NP-hard problem (Conitzer & Sandholm 2006). In practice, a number of effective solution packages for MILP can be used, however their performance depends on the number of integer variables. We note that DOBSS considers $|Q|\,|L|$ integer variables, while BRASS doubles that, and BOSS has $|Q|\,|L|\,K$ integer variables, with K the total number of discrete samples of the strategy space. Thus we anticipate MAXIMIN to be the most efficient, followed by DOBSS with BRASS close behind, and BOSS even slower depending on the sample size K.

Proposition 2 *If $\frac{1}{3}\varepsilon \geq C \geq |C_{ij}^l|$ for all i, j, l, then BRASS is equivalent to MAXIMIN.*

Note that $|a^l|$ in (3) $\leq C$. The leftmost inequality of constraint 5 in (3) shows that all q_j^l must equal 1, which makes BRASS equivalent to MAXIMIN. Suppose some $q_j^l = 0$, then that inequality states that $-C \leq \sum_{i \in X} C_{ij}^l x_i \leq a^l - \varepsilon < C - 3C = -2C$ a contradiction. ■

Experiments

We now present results comparing runtimes and quality of BRASS, BOSS, and MAXIMIN with DOBSS. The goal of our new MILPs was to address followers that may be boundedly rational or have limited observations. To that end, experiments were set up to play against human subjects (students) as followers, with varying observability conditions.

First, we constructed a domain inspired by the security domain (Paruchuri *et al.* 2008), but converted it into a pirate-and-treasure theme. The domain had three pirates — jointly acting as the leader — guarding 8 doors, and each individual subject acted as a follower. The subject's goal was to steal treasure from behind a door without getting caught. Each of the 8 doors gave a different positive reward and penalty to both the subjects as well as to the pirates — a non zero-sum game. If a subject chose a door that a pirate was guarding, the subject would incur the penalty and the pirate would receive the reward, else vice-versa. This setup led to a Stackelberg game with $\binom{8}{3} = 56$ leader actions, and 8 follower actions. Subjects were given full knowledge of their rewards and penalties and those of the pirates in all situations.

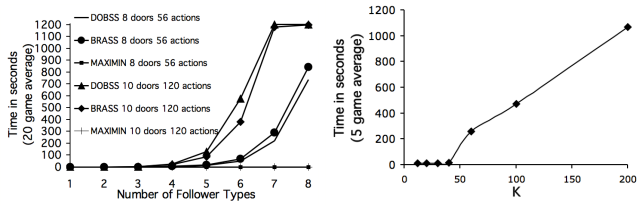
Runtime Results

For our run-time results, in addition to the original 8-door game, we constructed a 10-door game with $\binom{10}{3} = 120$ leader actions, and 10 follower actions. To average our runtimes over multiple instances, we created 19 additional reward structures for each of the 8-door and 10-door games. Furthermore, since our algorithms handle Bayesian games, we created 8 variations of each of the resulting 20 games to test scale-up in number of follower types.

In Figure 1(a), we summarize the runtime results for our Bayesian game using DOBSS, BRASS and MAXIMIN. The *x*-axis in Figure 1(a) varies the number of follower types the leader faces, from 1 to 8. The *y*-axis of the graph shows the runtime of each algorithm in seconds. Experiments were run using CPLEX 8.1. All experiments that were not concluded in 20 minutes (1200 seconds) were cut off since these runtimes are unusable in a real world setting. The results show that while, as anticipated, DOBSS was faster than BRASS in the 8-door domain, in the 10-door domain, BRASS (despite its larger number of integer variables) was faster on average. For example, for 6 follower types in the 10-door case, DOBSS ran for 577.3 seconds on average, while BRASS ran for 383.4 seconds. Overall trends suggest that DOBSS and BRASS should run within the same time frame, with neither one strictly dominating the other. MAXIMIN is also shown on this graph, however, it appears as a straight line along the *x*-axis in Figure 1(a), with maximum runtime of 0.054 seconds on average in the 10-door case. BOSS in contrast showed a surprising slowness even with one follower type. Figure 1 (b) shows the runtime of BOSS when the number of erroneous observations(K) is increased from 12 to 200. The *y*-axis shows the average runtime in seconds over 5 different reward structures and the *x*-axis denotes the value of k. For example, when k is 60, BOSS takes 257.40 seconds on average. Given this extreme slowness, BOSS was excluded for further experiments.

Quality Comparison

We now compare four algorithms, DOBSS, BRASS, MAXIMIN and UNIFORM using the 8-door 3-pirate domain.



(a) DOBSS, BRASS, MAXIMIN (b) BOSS

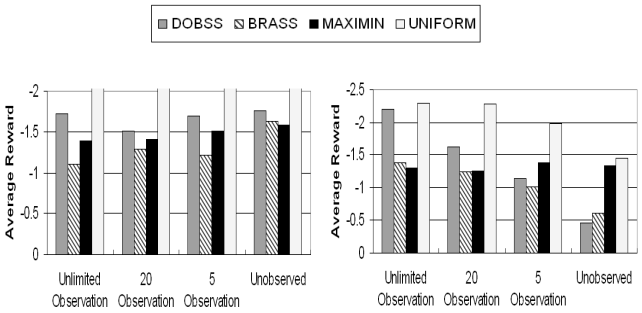
Figure 1: Comparing Runtimes

UNIFORM is our baseline uniformly random strategy; if the algorithms performed worse than UNIFORM, then the mixed strategies computation provided no benefits. In all of the experiments the value of ε for BRASS was set to 3. Each algorithm was tested with two reward structures for the 8-door domain. The second reward structure relaxed the penalty structure for the leader — to test its effect on our robust algorithms. Each combination of algorithm and reward structure was tested with four separate observability conditions where the subject observed the pirates' strategy under the current condition and then made their decision. A single observation consisted of seeing where the three pirates were stationed behind the eight doors, having the doors close, and then having the pirates restation themselves according to their mixed strategy. The four different observation conditions tested were: (i) The subject does not get any observations; (ii) Get 5 observations; (iii) Get 20 observations; (iv) Get infinite observations — simulated by revealing the exact mixed strategy of the pirate to the subject. In all cases the subject was given both their's and the pirates' full reward structure.

Each of our 32 game settings (two reward structures, four algorithms, four observability conditions) were played by 25 subjects, i.e. in total there were 800 total trials involving 57 subjects. Each subject played a total of 14 unique games and the games were presented in random orderings to avoid any order bias. For a given algorithm we computed the expected leader reward for each follower action, i.e. for each choice of door by subject. We then found the average expected reward for a given algorithm using the actual door selections from the 25 subject trials. For each game, the objective of a subject was to choose the door that would maximize his/her reward; and once a door was chosen that game was over and the subject played the next game. Starting with a base of 8 dollars, each reward point within the game was worth 15 cents for the subject and each penalty point deducted 15 cents. This was incorporated to give the subjects incentive to play as optimally as possible. On average, subjects earned $13.81.

Figure 2(a) shows the average expected leader reward for our first reward structure, with each data-point averaged over 25 runs. Figure 2(b) shows the same for the second reward structure (In both figures that *a lower bar is better* since all strategies have a negative average). In both figures, the *x*-axis shows the amount of observations allowed for each strategy and *y*-axis shows the average expected reward each strategy obtained. Examining Figure 2(a) for instance we can see in the unlimited observation case, BRASS scores an average expected leader reward of -1.10, whereas DOBSS suffers a 56% degradation of reward, obtaining an average score of -1.72.

We can observe the following from Figure 2. First, BRASS outperforms DOBSS under all conditions except for the unobserved condition of reward structure 2. Thus, given boundedly rational followers with any amount of observation — 5, 20 or unlimited — BRASS appears superior to DOBSS. Second, while MAXIMIN outperforms DOBSS in most conditions, BRASS outperforms MAXIMIN in all except two cases. In these two cases, BRASS is within 6% of MAXIMIN's rewards, while MAXIMIN in the worst case is 200% worse than BRASS (unobservability condition of reward structure 2). Combined with the earlier observation, BRASS thus appears to be the best among our new algorithms. Third, all of DOBSS, BRASS and MAXIMIN outperform UNIFORM, illustrating the benefits of mixed strategies over simple non-weighted randomization.



(a) Reward Structure One (b) Reward Structure Two

Figure 2: Expected Average Reward

Significance: Since our results critically depend on significant differences among DOBSS, BRASS and MAXIMIN, we ran the Friedman test (Friedman 1937); the non-normal distribution of our data precluded other tests such as ANOVA. Given that we have 2 reward structures the Friedman test is well suited — it's a two ways non-parametric test for when one of the ways (in our case, the strategy) is nested in the other way (the structure) — to test for significant differences in group means. The *p*-value obtained for the unlimited observations case was 0.0175, for the 20 observation was 0.0193 and for the 5 observation 0.002, indicating *significant differences*.

	Unlimited	20	5	Unobserved
DOBSS	.08	.12	.20	.16
BRASS	.96	.80	.92	.56

Table 2: Follower optimal choice percentage

Why does BRASS outperform DOBSS? DOBSS provides a leader strategy that maximizes the leader's expected reward, assuming an optimal follower response. Focusing on

reward structure 1, for DOBSS's leader strategy, assuming the follower will choose the optimal response of door 3, the leader obtains an expected reward of 0.79. If the follower's response is not door 3, the leader's expected reward decreases substantially (minimum of -4.21). BRASS maximizes the leader's expected reward assuming that the follower may choose the worst among its ε-optimal strategies. In reward structure 1, given BRASS's leader strategy, four of the doors (door 0,2,3,6) may give the follower a reward within ε of the follower's optimal strategy (door 3). If a follower chooses any of these four doors, BRASS guarantees the leader a minimum expected reward of -1.09. Thus, if all our subjects chose the optimal, DOBSS would obtain 0.79 in average expected reward, and BRASS -1.09 — DOBSS would win with perfectly rational followers. However, Table 2 shows the percentage of times subjects followed the optimal strategy in DOBSS vs the ε-optimal strategy in BRASS for our four observability conditions. For example, even with unlimited observability, where subjects could have computed their maximum expected utility, they choose their optimal strategy only 8% of times when playing the game with DOBSS's randomization reducing the reward in DOBSS to -1.72; but they choose BRASS's ε-optimal strategies 96% of times, keeping BRASS's rewards close to -1.09. These results are as expected due to the bounded rationality and preferences of humans.

Summary and Related Work

Stackelberg games are crucial in many multiagent applications, and particularly for security applications; the DOBSS algorithm is applied for security scheduling at the Los Angeles International Airport (Brown *et al.* 2006; Paruchuri *et al.* 2008). In such applications automated Stackelberg solvers may create an optimal leader strategy. Unfortunately, the bounded rationality and limited observations of the (human) followers in a Stackelberg game challenge a critical assumption — that followers will act optimally — in DOBSS or any other existing Stackelberg solver, demolishing their guarantee of optimality of leader strategy. To apply Stackelberg games to any setting with people, this limitation must be addressed. This paper provides the following key contributions to address this limitation. First, it provides three new robust algorithms, BRASS, BOSS and MAXIMIN, to address followers with bounded rationality and limited observation power. Second, it provides run-time analysis of these algorithms. Third, it tests these algorithms with humans, in 800 games played with 57 students over 4 observability conditions and two rewards structures, and shows that BRASS outperforms optimal Stackelberg solvers in quality.

In terms of related work, we earlier discussed other algorithms for Stackelberg games. Here we first discuss related work in robust game theory, first introduced for nash equilibria in (Aghassi & Bertsimas 2006) and adapted to wardrop network equilibria in (Ordóñez & Stier-Moses 2007). These prior works show that an equilibrium exists and how to compute it when players act robustly to parameter uncertainty. Another area of related work is approaches to bounded rationality in game theory (Rubinstein 1998) — the key question remains how to precisely to model it in game theoretic settings (Simon 1969). In addition to mathematical models (Rubinstein 1998), empirical analysis shows that people dont play equilibrium strategies. For example, the winners at the International World Championships conducted by the World Rock Paper Scissors Society were never equilibrium players (Shoham, Powers, & Grenager 2007). We complement these works via robust solutions for Bayesian Stackelberg game.

References

Aghassi, M., and Bertsimas, D. 2006. Robust game theory. *Math. Program.* 107(1-2):231–273.

Brown, G.; Carlyle, M.; Salmern, J.; and Wood, K. 2006. *Defending Critical Infrastructure*. Interfaces.

Cardinal, J.; Labbé, M.; Langerman, S.; and Palop, B. 2005. Pricing of geometric transportation networks. In *17th Canadian Conference on Computational Geometry*.

Conitzer, V., and Sandholm, T. 2006. Computing the optimal strategy to commit to. In *EC*.

Friedman, M. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. volume 32 No. 100, 675–701.

Fudenberg, D., and Tirole, J. 1991. *Game Theory*. MIT Press.

Harsanyi, J. C., and Selten, R. 1972. A generalized Nash solution for two-person bargaining games with incomplete information. *Management Science* 18(5):80–106.

Korilis, Y. A.; Lazar, A. A.; and Orda, A. 1997. Achieving network optima using stackelberg routing strategies. In *IEEE/ACM Transactions on Networking*.

Murr, A. 2007. Random checks. *Newsweek National News* http://www.newsweek.com/id/43401.

Nilim, A., and Ghaoui, L. E. 2004. Robustness in markov decision problems with uncertain transition matrices. In *NIPS*.

Ordóñez, F., and Stier-Moses, N. E. 2007. Robust wardrop equilibrium. In *NET-COOP*, 247–256.

Paruchuri, P.; Pearce, J. P.; Tambe, M.; Ordóñez, F.; and Kraus, S. 2007. An efficient heuristic approach for security against multiple adversaries. In *AAMAS*.

Paruchuri, P.; Marecki, J.; Pearce, J.; Tambe, M.; and Kraus, S. 2008. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *AAMAS*.

Rubinstein, A. 1998. *Modeling Bounded Rationality*. MIT Press.

Shoham, Y.; Powers, R.; and Grenager, T. 2007. If multiagent learning is the answer, what is the question? In *AIJ 171(7)*, 365–377.

Simon, H. 1956. Rational choice and the structure of the environment. volume 63, 129–138.

Simon, H. 1969. *Sciences of the Artificial*. MIT Press.

von Stengel, B., and Zamir, S. 2004. Leadership with commitment to mixed strategies. In *CDAM Research Report LSE-CDAM-2004-01, London School of Economics*.