Effective Solutions for Real-World Stackelberg Games: When Agents Must Deal with Human Uncertainties

James Pita, Manish Jain, Fernando Ordóñez, Milind Tambe
University of Southern California, Los Angeles, CA 90089

Sarit Kraus* and Reuma Magori-Cohen
Bar-Ilan University, Ramat-Gan 52900, Israel and
*Institute for Advanced Computer Studies,
University of Maryland, College Park, MD 20742

ABSTRACT

How do we build multiagent algorithms for agent interactions with human adversaries? Stackelberg games are natural models for many important applications that involve human interaction, such as oligopolistic markets and security domains. In Stackelberg games, one player, the leader, commits to a strategy and the follower makes their decision with knowledge of the leader's commitment. Existing algorithms for Stackelberg games efficiently find optimal solutions (leader strategy), but they critically assume that the follower plays optimally. Unfortunately, in real-world applications, agents face human followers (adversaries) who — because of their bounded rationality and limited observation of the leader strategy — may deviate from their expected optimal response. Not taking into account these likely deviations when dealing with human adversaries can cause an unacceptable degradation in the leader's reward, particularly in security applications where these algorithms have seen real-world deployment. To address this crucial problem, this paper introduces three new mixed-integer linear programs (MILPs) for Stackelberg games to consider human adversaries, incorporating: (i) novel *anchoring* theories on human perception of probability distributions and (ii) robustness approaches for MILPs to address human imprecision. Since these new approaches consider human adversaries, traditional proofs of correctness or optimality are insufficient; instead, it is necessary to rely on empirical validation. To that end, this paper considers two settings based on real deployed security systems, and compares 6 different approaches (three new with three previous approaches), in 4 different observability conditions, involving 98 human subjects playing 1360 games in total. The final conclusion was that a model which incorporates both the ideas of robustness and anchoring achieves statistically significant better rewards and also maintains equivalent or faster solution speeds compared to existing approaches.

General Terms

Algorithms, Experimentation, Security, Human Factors

Keywords

Security of Agent Systems, Game Theory, Bayesian and Stackelberg Games

Cite as: Effective Solutions for Real-World Stackelberg Games: When Agents Must Deal with Human Uncertainties, James Pita, Manish Jain, Fernando Ordóñez, Milind Tambe, Sarit Kraus, Reuma Magori-Cohen, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 369–376

1. INTRODUCTION

In Stackelberg games, one player, the leader, commits to a strategy publicly before the remaining players, the followers, make their decision [8]. There are many multiagent security domains, such as attacker-defender scenarios and patrolling, where these types of commitments are necessary by the security agent [3, 4, 15, 10] and it has been shown that Stackelberg games appropriately model these commitments [14, 16]. For example, security personnel patrolling an infrastructure decide on a patrolling strategy first, before their adversaries act taking this committed strategy into account. Indeed, Stackelberg games are at the heart of the ARMOR system deployed at the Los Angeles International Airport (LAX) for over a year to schedule security personnel [14, 16] and have recently seen application for the Federal Air Marshals [10]. Moreover, these games have potential applications for network routing, pricing in transportation systems and many others [5, 11].

Existing algorithms for Bayesian Stackelberg games find optimal solutions considering an *a priori* probability distribution over possible follower types [6, 14]. Unfortunately, to guarantee optimality, these algorithms make strict assumptions on the underlying games, namely that the players are perfectly rational and that the followers perfectly observe the leader's strategy. However, these assumptions rarely hold in real-world domains, particularly when dealing with humans. Of specific interest are the security domains mentioned earlier (e.g. LAX) — even though an automated program may determine an optimal leader (security personnel) strategy, it must take into account a human follower (adversary). Such human adversaries may not be utility maximizers, computing optimal decisions. Instead, their decisions may be governed by their bounded rationality [19] which causes them to deviate from their expected optimal. Humans may also suffer from limited observability of the security personnel's strategy, giving them a false impression of that strategy. Thus, a human adversary may not respond with the game theoretic optimal choice, causing the leader to face uncertainty over the gamut of adversary's actions. Therefore, in general, the leader in a Stackelberg game must commit to a strategy considering three different types of uncertainty: (i) adversary response uncertainty due to his bounded rationality where the adversary may not choose the utility maximizing optimal strategy; (ii) adversary response uncertainty due to his limitations in appropriately observing the leader strategy; (iii) adversary 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 [6, 14], these models can give a severely under performing strategy when the adversary deviates because of the first two types of uncertainty. This degradation in leader rewards may be unacceptable in certain domains.

To overcome this limitation, this paper proposes three new algorithms based on mixed-integer linear programs (MILPs). The major contribution of these new MILPs is in providing a fundamentally novel integration of key ideas from: (i) previous best known algorithms from the multiagent literature for solving Bayesian Stackelberg games; (ii) robustness approaches from robust optimization literature [2, 13]; (iii) anchoring theories on human perception of probability distributions from psychology [18]. While the robustness approach addresses human response imprecision, anchoring, which is an expansion of general support theory [20] on how humans attribute probabilities to a discrete set of events, addresses limited observational capabilities. To the best of our knowledge, the effectiveness of the combination of these ideas has not been explored in the context of Stackelberg games (or any other games). By uniquely incorporating these ideas our goal is to defend against the sub-optimal choices that humans may make due to bounded rationality or observational limitations. These new MILPs complement the prior algorithms for Bayesian Stackelberg games, handling all three types of uncertainty mentioned.

Since these algorithms are centered on addressing non-optimal and uncertain human responses, traditional proofs of correctness and optimality are insufficient: it is necessary to experimentally test these new approaches against existing approaches. Experimental analysis against human subjects allows us to show how these algorithms are expected to perform against human adversaries compared to previous approaches. To that end, we experimentally tested our new approaches to determine their success by considering two settings based on real deployed security systems. In both settings, 6 different approaches were compared (three new approaches, one existing approach, and two baseline approaches), in 4 different observability conditions. These experiments involved 98 human subjects playing 1360 games in total and yielded statistically significant results showing that one of our new algorithms substantially outperformed existing methods when dealing with human adversaries. Runtime results were also gathered from our new algorithms against previous approaches showing that their solution speeds are equivalent to or faster than previous approaches. Based on these results we concluded that, while theoretically optimal, existing algorithms for Bayesian Stackelberg games may need to be significantly modified for real-world security domains. They are not only outperformed by one of our new algorithms, which incorporates both robustness approaches and anchoring theories, but also may be outperformed by simple baseline algorithms in certain cases. This is an important conclusion since existing algorithms have seen real deployment such as at Los Angeles International Airport (LAX) [16]. Indeed our new algorithms for addressing human adversaries in Stackelberg games suggest significant potential improvements for wherever existing algorithms are deployed in real-world domains and leaders will face human adversaries.

2. BACKGROUND

Stackelberg Game: In a Stackelberg game, a leader commits to a strategy first, and then a follower optimizes his 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. If this were a simultaneous move game, the 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, in this Stackelberg game 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. This paper considers a Bayesian Stackelberg game that was inspired by a security domain presented for LAX [16]. This game has two agents, the leader and the follower. It is assumed 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 $u_n : L \times \sigma_1 \times \sigma_2 \to \Re$. The goal is *to find the optimal mixed strategy* for the leader given that the follower knows this strategy when choosing his 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 [6], researchers have continued to provide practical improvements. DOBSS is currently the most efficient algorithm for such games [14] and in use for security scheduling at the Los Angeles International Airport. It operates directly on the compact Bayesian representation, giving speedups over the multiple linear programs method [6] which requires conversion of the Bayesian game into a normal-form game by the Harsanyi transformation [9].

We now discuss DOBSS, which provides the optimal mixed strategy for the leader while considering an *optimal* follower response for this leader strategy. Note that it needs 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. The leader's mixed strategy is denoted by x, a probability distribution over the 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 strategy. The vector of strategies of follower $l \in L$ is denoted by q^l. The index sets of leader and follower type l's pure strategies are denoted by X and Q respectively. The payoff matrices of the leader and each of the followers l is indexed by the matrices R^l and C^l. DOBSS assumes *a priori* probabilities p^l, with $l \in L$ of facing each follower type. Considering auxiliary variable $z^l_{ij} = x_i q^l_j$, DOBSS computes the leader's optimal decision problem using the following MILP formulation [14]:

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

For future discussion it is important to understand the following set of constraints. The fourth and eighth constraints 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 the fifth constraint ensure that $q_j^l = 1$ only for a strategy j that is optimal for follower type l. Therefore, in the current formulation each follower type l is allowed to choose exactly one optimal action from his set of possible actions.

Baseline Algorithms: For completeness this paper includes both a uniformly random strategy and a MAXIMIN strategy against human opponents as a baseline against the performance of both existing algorithms, such as DOBSS, and our new algorithms. Algorithms must outperform the two baseline algorithms to provide benefits.

UNIFORM: UNIFORM is the most basic method of randomization which just assigns an equal probability of taking each action $i \in X$ (a uniform distribution).

MAXIMIN: MAXIMIN is a traditional approach which assumes the follower may take any of the available actions. The objective of the following LP is to maximize the minimum reward γ the leader will obtain irrespective of the follower's action.

$$\begin{array}{ll} \max & \sum_{l\in L} p^l \gamma_l \\ \text{s.t.} & \sum_{i\in X} x_i = 1 \\ & \sum_{i\in X} R_{ij}^l x_i \geq \gamma_l \\ & x_i \in [0\ldots 1] \end{array} \tag{2}$$

3. ROBUST ALGORITHMS

There are two fundamental assumptions underlying current algorithms for Stackelberg games, including DOBSS. First, the follower is assumed to act with infallible utility maximizing rationality, choosing the absolute optimal among his strategies. Second, if the follower faces a tie in his 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 [21]. Unfortunately, in many real-world domains, agents can face human followers who may not respond optimally: this may be caused by their bounded rationality, or their uncertainty regarding the leader strategy. 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. 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 [2, 12], as well as psychological support theory for human decision making when they are given a discrete set of actions and an unknown probability function over those actions [18, 20]. In the presented Stackelberg problem, the leader will make a robust decision by considering that the boundedly rational follower could choose a strategy from his range of possible responses that degrades the leader rewards the most or that he could choose a strategy that is based on limited observations. This approach differs from standard robust optimization methodology in that it makes predictions about how and why the human adversary's response will deviate and robustly guards against those predictions, as opposed to considering arbitrary deviations in the responses. This paper introduces three mixed-integer linear programs (MILPs) to that end. The first MILP, BRASS (Bounded Rationality Assumption in Stackelberg Solver) addresses the uncertainty that may arise from human imprecision in choosing the expected optimal strategy due to bounded rationality. The second MILP, GUARD (Guarding Using Alpha Ranges in DOBSS) utilizes the anchoring biases to protect against limited observation conditions. The third MILP, COBRA (Combined Observability and Rationality Assumption), provides a robust response for all three types of uncertainty previously mentioned. We first describe in depth the key ideas behind our new approaches and then define the MILPs that use them.

Bounded Rationality: Some of our new algorithms assume that the follower is boundedly rational and may not strictly maximize utility. As a result, the follower may select an 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 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 [6, 14], making this new approach novel for human followers.

Anchoring Theory: Support theory is a theory of subjective probability [20] and has been used to introduce anchoring biases [18]. An anchoring bias is when, given no information about the occurrence of a discrete set of events, humans will assign an equal weight to the occurrence of each event (a uniform distribution). It has been shown that humans are particularly susceptible to anchoring on the uniform distribution before they are given any information and that, once given information, they are slow to update away from this assumption [18]. Thus they leave some weight, $\alpha \in [0\ldots 1]$, on the uniform distribution and the rest, $1-\alpha$, on the occurrence they have actually viewed. As humans become more confident in what they are viewing this bias begins to diminish, decreasing the value of α. Models have been proposed to address this bias and predict what probability a human will assign to a particular event x from a set of events X. One proposed model is written in odds form as $R(x, X\backslash x) = (|x|/|X\backslash x|)^\alpha * (P(x)/P(X\backslash x))^{1-\alpha}$, however, a linear model is also possible [1, 20]. The linear model introduces a new term $P(x')$, which is the probability the human assigns to event x as opposed to the real probability of event x occurring: $P(x') = (1/|X|) * (\alpha) + (1-\alpha) * P(x)$. The parameter α dictates how much support the human will give to the uniform probability distribution and how much support he will give to the real probability $(P(x))$. The end result is the predicted probability the human will assign to event x. We commandeer this anchoring bias for Stackelberg games to determine how a human follower may perceive the leader strategy. For example, in the game shown in Table 1, suppose the leader strategy was to play *a* with a probability of 0.8 and *b* with 0.2. Anchoring bias would predict that in the absence of any information ($\alpha = 1$), humans will assign a probability of 0.5 to each of *a* and *b*, and will only update this belief (alter the value of α) after observing the leader strategy for some time.

3.1 BRASS

BRASS considers the case of a boundedly rational follower, where it maximizes the minimum reward it obtains from any ε-optimal response. In the following MILP, we use the same variable notation as in DOBSS. In addition, the variables h_j^l identify the optimal strategy for follower type l with a value of a^l in the third and fourth constraints. Variables q_j^l represent all ε-optimal strategies for follower type l; the second constraint now allows selection of more than one strategy 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 less than ε and the left inequality is then only satisfied if $q_j^l = 1$. This robust approach required the design of a new objective and additional constraint. The sixth constraint helps define the objective value against follower type l, γ_l, which must be lower than any leader reward for all actions $q_j^l = 1$, as opposed to the DOBSS formulation which has only one action $q_j^l = 1$. Setting γ_l to the minimum leader reward allows BRASS to robustly guard against the worst case scenario. The new MILP is as follows:

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

3.2 GUARD

GUARD considers the case where the human follower is perfectly rational, but faces limited observations. GUARD draws upon the theory of anchoring biases mentioned previously to help address the human uncertainty that arises from such limited observation. It deals with two strategies: (i) the real leader strategy (x) and (ii) the leader strategy the follower believes (x'), where x' is defined by the linear model presented earlier. Given the follower's belief strategy, x_i is replaced in the third constraint with x_i' and x_i' is accordingly defined as $x_i' = (1/|X|) * (\alpha) + (1 - \alpha) * x_i$. The justification for this replacement is as follows. First, this particular constraint ensures that the follower maximizes his reward. Since the follower believes x_i' to be the leader strategy then he will choose his strategy according to x_i' and not x_i. Second, given this knowledge, the leader can find the follower's responses based on x_i' and optimize his actual strategy x_i against this strategy. Since x_i' is a combination of x_i and the bias toward the uniform probability distribution GUARD is able to find a strategy x_i that will maximize the leader's reward based on how the follower will update his beliefs. The new MILP then is as follows:

$$\begin{array}{ll}
\max_l & \sum_{l\in L} p^l \gamma_l \\
\text{s.t.} & \sum_{i\in X} x_i = 1 \\
 & \sum_{j\in Q} q_j^l = 1 \\
 & 0 \leq (a^l - \sum_{i\in X} C_{ij}^l * x_i') \leq (1 - q_j^l)M \\
 & M(1 - q_j^l) + \sum_{i\in X} R_{ij}^l x_i \geq \gamma_l \\
 & x_i \in [0 \ldots 1] \\
 & q_j^l \in \{0, 1\} \\
 & a \in \Re \\
 & x_i' = (1/|X|) * (\alpha) + (1 - \alpha) * x_i
\end{array} \tag{4}$$

3.3 COBRA

COBRA is an MILP that combines both a bounded rationality assumption and an observational uncertainty assumption. This is achieved by incorporating the alterations made in BRASS and GUARD into a single MILP. Namely, COBRA includes both the ε parameter and the α parameter from MILP (3) and MILP (4) respectively. The MILP that follows is identical to MILP (3) except that in the fourth and fifth constraints, x_i is replaced with x_i' as it is in MILP (4). The justification for this replacement is the same as in MILP (4). The new MILP then is as follows:

$$\begin{array}{ll}
\max_{x,q,h,a,\gamma} & \displaystyle\sum_{l\in L} p^l \gamma_l \\
\text{s.t.} & \sum_{i\in X} x_i = 1 \\
 & \sum_{j\in Q} q_j^l \geq 1 \\
 & \sum_{j\in Q} h_j^l = 1 \\
 & 0 \leq (a^l - \sum_{i\in X} C_{ij}^l * x_i') \leq (1 - h_j^l)M \\
 & \varepsilon(1 - q_j^l) \leq a^l - \sum_{i\in X} C_{ij}^l * x_i' \leq \varepsilon + (1 - q_j^l)M \\
 & M(1 - q_j^l) + \sum_{i\in X} R_{ij}^l x_i \geq \gamma_l \\
 & h_j^l \leq q_j^l \\
 & x_i \in [0 \ldots 1] \\
 & q_j^l, h_j^l \in \{0, 1\} \\
 & a \in \Re \\
 & x_i' = (1/|X|) * (\alpha) + (1 - \alpha) * x_i
\end{array} \tag{5}$$

PROPOSITION 1. *When $\varepsilon = 0$ and $\alpha = 0$ then MILPs (1) and (5) are equivalent.*

Proof sketch: It follows from the definition of x_i' that when $\alpha = 0$ then $x_i' = x_i$ since the follower is assumed to once again perfectly observe and believe the leader strategy x_i. Note that if $\varepsilon = 0$ the inequality in the fifth constraint of (5) is the same expression as the inequality in the fourth constraint with q_j^l substituted for h_j^l. We will show that the two problems attain the same optimal objective function value.

To show that solution to (5) $\geq$ solution to (1), consider (q, z, a) a feasible solution for (1). We define $\bar{x}_i = \sum_{j\in Q} z_{ij}^l$, $\bar{q} = \bar{h} = q$, $\bar{a} = a$, and $\bar{\gamma}_l = \sum_{i\in X} \sum_{j\in Q} R_{ij}^l z_{ij}^l$. From the first through third constraints and the sixth constraint in (1) we can show that $z_{ij}^l = 0$ for all j such that $q_j^l = 0$ and thus that $\bar{x}_i = z_{ij}^l$ for all j such that $q_j^l = 1$. This implies that $\bar{\gamma}_l = \sum_{i\in X} R_{ij}^l \bar{x}_i$ for the j such that $q_j^l = 1$ and it is then easy to verify that $(\bar{x}, \bar{q}, \bar{h}, \bar{a}, \bar{\gamma})$ is feasible for (5) with the same objective function value of (q, z, a) in (1).

For solution to (1) $\geq$ solution to (5), consider (x, q, h, a, γ) feasible for (5). Define $\bar{q} = h$, $\bar{z}_{ij}^l = x_i h_j^l$, and $\bar{a} = a$. Then we can show that $(\bar{q}, \bar{z}, \bar{a})$ is feasible for (1) by construction. Since $h_j^l \leq q_j^l$ it follows that $\gamma_l \leq \sum_{i\in X} R_{ij}^l x_i$ for the j such that $h_j^l = 1$. This implies that $\gamma_l \leq \sum_{i\in X} \sum_{j\in Q} R_{ij}^l \bar{z}_{ij}^l$ and that the objective function value of $(\bar{q}, \bar{z}, \bar{a})$ in (1) greater than or equal to the objective value of (x, q, h, a, γ) in (5). ■

The key implication of the above proposition is that when $\varepsilon = 0$, COBRA loses its robustness feature, so that once again when the follower faces a tie, it selects a strategy favoring the leader, as in DOBSS. Based on this proposition, a few observations that can be made surrounding the COBRA algorithm are the following: (i) if $\alpha = 0$, COBRA is equivalent to BRASS, (ii) if $\varepsilon = 0$, COBRA is equivalent to GUARD, (iii) if both $\alpha = 0$ and $\varepsilon = 0$, COBRA is equivalent to DOBSS. Based on these observations the propositions presented in this paper can be generalized to the other three algorithms (DOBSS, GUARD, and BRASS) accordingly.

PROPOSITION 2. *When α is held constant, the optimal reward COBRA can obtain is decreasing in ε.*

Proof sketch: Since the fifth constraint in (5) 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 the sixth constraint can only decrease the minimum value γ_l. ■

PROPOSITION 3. *Regardless of α, if $\frac{1}{3}\varepsilon \geq C \geq |C^l_{ij}|$ for all i, j, l, then COBRA is equivalent to MAXIMIN.*

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

Deciding α and ε: To decide the value of ε we employed a heuristic where ε is decided based on how close to the optimal response the follower is expected to come, e.g. if we expect human followers to play within 20% of the optimal, we set ε to 20% of the optimal reward. We try two different techniques to determine α, leading to two different versions of COBRA. The first approach is to vary α based on the number of observations that human followers are anticipated to have. This standard version of COBRA implies that when deploying it, α is adjusted per anticipated observation capability. In this case, if a human follower has had zero observations, we assume that he would be entirely guided by the anchoring bias to uniform probability, and hence set $\alpha = 1$, i.e. $x' = 1/|X|$. In contrast, if a follower has infinite observations, he would correctly determine the actual leader strategy, i.e. $x' = x$, and hence $\alpha = 0$. When a follower has only a limited number of observations, we heuristically select α, decreasing it with increasing number of follower's observations — choosing the right α remains an issue for future work. The second approach is to assume a constant α, leading to a version of COBRA that we will refer to as COBRA-C (COBRA with constant α). We discuss the choice of α for COBRA-C in Section 4.1

Complexity: DOBSS, BRASS, GUARD and COBRA require the solution of a MILP, whereas MAXIMIN is a linear programming problem. Therefore the complexity of MAXIMIN is polynomial while DOBSS, BRASS, GUARD and COBRA face an NP-hard problem [6]. A number of effective solution packages for MILPs can be used, but their performance depends on the number of integer variables. DOBSS and GUARD consider $|Q|\,|L|$ integer variables, while BRASS and COBRA double that. Thus we anticipated MAXIMIN will have the lowest running time per problem instance, followed by DOBSS and GUARD with BRASS and COBRA close behind. However, as shown in runtime results, this was not the final result.

4. EXPERIMENTS

We now present results comparing the quality and runtime of strategies introduced in the previous two sections. The goal of our new algorithms was to improve interactions between agents and humans by addressing the bounded rationality that humans may exhibit and the limited observations they may experience in real-world settings. To that end, experiments were set up to play against human subjects as followers (adversaries), with varying observability conditions.

First, we constructed a domain inspired by the security domain at LAX [14, 16], 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 an adversary. The subject's goal was to steal treasure from behind a door without getting caught. Each of the 8 doors would have a unique reward and penalty associated with it for both the subjects as well as the pirates – a non zero-sum game. If a subject chose a door that a pirate was guarding, the subject would incur the unique subject penalty for that door and the pirate would receive the unique pirate reward for that door, else vice-versa. This setup led to a Stackelberg game with $\binom{8}{3} = 56$ leader actions, and 8 follower actions.

4.1 Quality Comparison

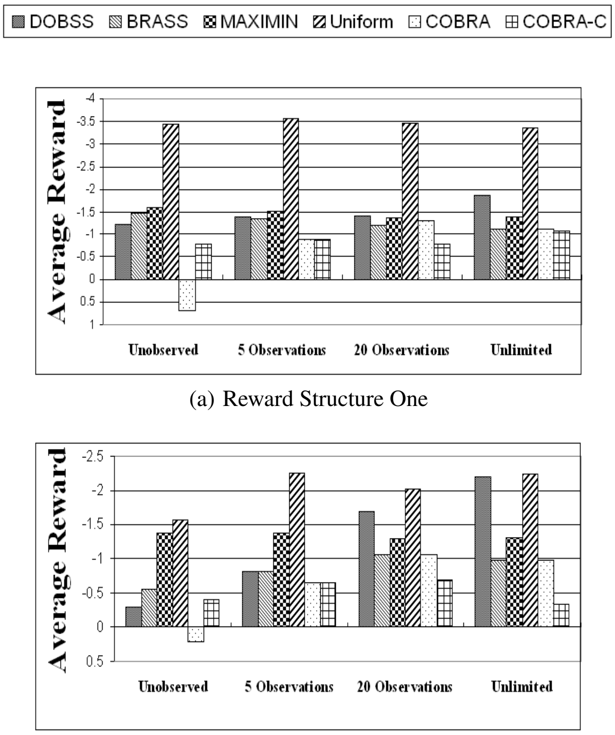
Experimental Structure and Setup: Given the 8-door 3-pirate domain described, we constructed two unique reward structures corresponding to the eight doors. The second reward structure increased the penalty structure for the leader — to test its effect on our robust algorithms. For each reward structure there were also four separate observability conditions that the subjects were exposed to. The subject observed the pirates' strategy under the current observability condition and reward structure and then was allowed to make his 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) the subject gets 5 observations; (iii) the subject gets 20 observations; (iv) the subject gets infinite observations — simulated by revealing the exact mixed strategy of the pirate to the subject. Subjects were given full knowledge of their rewards and penalties and those of the pirates in all situations.

Algorithms: These experiments only compare DOBSS, BRASS, COBRA, MAXIMIN, and UNIFORM. We reiterate that GUARD refers to a special case of COBRA, where ε is set to zero. On closer examination it is clear that GUARD is dominated by COBRA: (i) GUARD is equivalent to DOBSS when $\alpha = 0$; thus when $\alpha = 0$, our results will show that COBRA is superior to DOBSS and consequently to GUARD; (ii) On the other extreme when $\alpha = 1$ in the unobserved observation condition it has also been concluded by experimental tests that GUARD once again performs worse than COBRA, obtaining an expected reward of -.65 in reward structure one and -2.15 in reward structure two compared to the expected reward .205 and .7 obtained by COBRA. Furthermore, in both cases these results were statistically significant. Since at both extremes GUARD is dominated by COBRA, we do not include GUARD in our experimental analysis and results. We could make a similar argument for not including BRASS, however, it is important to include either BRASS or GUARD to demonstrate that the results obtained by COBRA are not only due to handling human bounded rationality but to handling both human bounded rationality and limited observation conditions.

For these experiments ε was set to 2.5. This choice for ε was made because the follower's reward for each door ranged from 1 to 10 and we wanted to robustly guard against boundedly rational strategies within 25% of the optimal strategy. We employed our heuristic for deciding the α parameters of COBRA, which was explained in Section 3. For COBRA-C α was set to the same α value as the 5 observation cases from the two reward structures with the expectation that it would perform poorly in higher observation conditions since it was not appropriately adjusted.

Experiments: Each of our 48 game settings (two reward structures, six algorithms, and four observability conditions) were played by 40 subjects, i.e. in total there were 1360 total trials. Notice that the unobserved case only needed to be played by one set of 40 subjects as the choices made without any observation would be similar regardless of the algorithm. This follows from the fact that the subject had no information regarding the strategy he was facing and thus his decisions for this particular condition were solely based on the reward structure. Given this setup, each subject played a total of 14 unique games and the games were presented in random orderings to avoid any order bias. In total there were 98 different subjects that played. 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 40 subject trials. For each game, the objective of a subject was to earn as many points as possible by choosing the highest value door he thought would be unguarded; 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 $13.13.

Results: Figure 1(a) shows the average expected leader reward for our first reward structure, with each data-point averaged over 40 human responses. Figure 1(b) shows the same for the second reward structure. Notice that *a lower bar is better* since all strategies have a negative average with the exception of COBRA in the unobserved case. In both figures, the *x*-axis shows the observation condition for each strategy and *y*-axis shows the average expected reward each strategy obtained. For example, examining Figure 1(b) in the unlimited observation case, COBRA-C scores an average expected leader reward of -0.33, whereas DOBSS suffers a 663% degradation of reward, obtaining an average score of -2.19.



(a) Reward Structure One

(b) Reward Structure Two

Figure 1: Expected Average Reward

Statistical Significance: Since our results critically depend on significant differences among DOBSS, BRASS, COBRA, MAXIMIN, and UNIFORM, we ran the Friedman test for repeated observations [7] in the unobserved case and Yuen's test for comparing trimmed means [22] for the 5, 20, and infinite observation cases[1]. For our tests we used a standard 20% trimmed mean to test for significant differences in group means. The maximum p-value obtained for COBRA-C versus any other strategy was .033 showing that under all conditions the results obtained for COBRA-C are statistically significantly different than the results obtained by other strategies. COBRA also obtained statistical significance in all cases against other strategies except the 20 observation case with a maximum p-value of .029. It is evident from these values and the results presented that COBRA-C is statistically significantly better than all other strategies in every observation condition except the unobserved case.

Conclusions and Analysis: Analysis of the reported results yields the following conclusions: (i) COBRA, which adjusts its strategy based on observations, performs significantly better when dealing with humans than DOBSS. The main implication being that if we know approximately how many observations the adversary will obtain, then we can exploit the variable α in COBRA to our advantage. (ii) Dealing with both bounded rationality and limited observations are important when designing an algorithm that performs well against humans. Our results demonstrate that only utilizing α or ε is not enough, but rather the combination of the two is necessary for superior performance under all observation conditions. (iii) COBRA-C surprisingly performs better than COBRA under high observation conditions. This finding is particularly important since in many real-world domains the observational limitations may be unknown making it difficult to decide α. (iv) COBRA and COBRA-C both perform better than our baseline algorithms making the extra computation worthwhile.

Next we discuss the key implications of these conclusions and why they were reached. We include two tables for reference in the following discussion, Tables 2 and 3. Table 2 shows the percentage of times the follower chose a response that the current algorithm predicted he would choose for different observation conditions in reward structure one, which we will refer to as a *predicted response*. The predicted responses are the ones the leader optimized against. Table 3 shows the expected rewards (for a subset of the algorithms tested) the leader should obtain for each door selection by the follower in reward structure one. For instance, if the follower selected Door 2 when playing against DOBSS the leader would expect to obtain a reward of -.97.

Structure One	Unobserved	5	20	Infinite
DOBSS	20%	7.5%	17.5%	12.5%
BRASS	65%	65%	65%	70%
COBRA	57.5%	92.5%	72.5%	70%
COBRA-C	92.5%	92.5%	87.5%	95%
MAXIMIN	100%	100%	100%	100%

Table 2: Percentage of Times Follower Chose a Leader Predicted Response in Reward Structure One

	DOBSS	BRASS	MAXIMIN	COBRA-C COBRA-5	COBRA-20
Door 1	-5	-4.58	-1.63	-5	-4.61
Door 2	-.97	-.42	-1.63	-.30	-.37
Door 3	.36	-.36	-1	-.30	-.37
Door 4	-1.38	-.79	-1.63	-.30	-.73
Door 5	.06	-.36	-1.63	-.30	-.37
Door 6	-1	-.86	-1	-1	-.87
Door 7	.39	-.36	-1.63	-.30	-.37
Door 8	-4.57	-3.69	-1.63	-3.32	-3.67

Table 3: Leader Expected Rewards for Reward Structure One

Why does COBRA perform better than DOBSS? The simple

[1] Yuen's test was run on the combined data from both reward structures since a two-way Friedman test reveals that structure is insignificant to the results.

answer is that by incorporating a bounded rationality assumption along with anchoring theory for limited observation conditions COBRA more accurately predicts human responses. If followers played according to the expectations of DOBSS, it would be the superior strategy, however, they do not. Looking at Table 2 for instance, we see that in the 5 observation case of DOBSS the follower chooses a predicted response only 7.5% of the time while in COBRA he chose a predicted response 92.5% of the time. The predicted response by DOBSS is that the follower plays door 7 and for COBRA it is all doors where it obtains -.3. Notice in Table 3 if the human follower had played the predicted response of Door 7 100% of the time then DOBSS would have obtained a reward of .39 while COBRA in the 5 observation case can only obtain a meager -.30. Further examination of Table 3, however, reveals that DOBSS can suffer tremendously depending on what non-optimal response is chosen. In Door 1 for example, DOBSS can obtain a reward of -5. This shows why DOBSS can suffer if followers stray from the predicted response. Since followers rarely stray from the predicted response in COBRA we expect to obtain a reward around the predicted reward of -.30 and indeed COBRA in the 5 observation condition gives an expected reward of -.65, lower than expected, but much better than the -.81 that DOBSS obtained compared to the predicted of .39. In fact, under high observation conditions, DOBSS is seen performing even worse than our simple baseline of MAXIMIN.

Now we examine why dealing with both bounded rationality and observational limitations are necessary for performance. BRASS is equivalent to COBRA with $\alpha = 0$, showing how COBRA performs without an α parameter. As shown in Figure 1(a), BRASS is outperformed by COBRA (obtains lower expected rewards) in the unobserved and 5 observation cases. This demonstrates that by varying α COBRA has significantly improved its strategies and expected rewards in limited observation conditions. Of course when observation is perfect, COBRA and BRASS are equivalent and both outperform DOBSS in the infinite observation conditions. This demonstrates that ε is also important even when α is not present (since $\alpha = 0$ in this case). These results clearly show that dealing with both bounded rationality and observational limitations are necessary to achieve a superior performing algorithm.

Why does COBRA-C outperform COBRA? The simple answer is that COBRA-C utilizes its resources better, by being better able to predict human responses. Looking at Table 2, COBRA-C accurately predicts human responses 87.5% of the time in the worse case. COBRA-C makes use of the concept that even though the human follower may not have seen a guard on a particular door he will still attribute some probability, even if it is low, that a guard may appear on that door at some point. Although the strategies are not presented here, in the 20 observation case COBRA assigns a guard to Doors 1 and 6 7% of the time. COBRA-C on the other hand uses this 14% and distributes it among other choices assuming the follower will assign some probability to these doors regardless of what the actual strategy is. Thus, COBRA-C increases the expected value of other doors (-.3 rather than -.37 for COBRA-20). Even in the infinite observation case, COBRA-C is found to be a better predictor of human responses with followers choosing a predicted response 95% of the time as opposed to the 70% against COBRA. Although this was not expected, it was a welcome surprise.

Why do COBRA and COBRA-C perform better than our baseline algorithms? The main reason is they make more intelligent use of the resources available. UNIFORM is a naive strategy that does not make use of the reward structure and MAXIMIN is too defensive, trying to make all doors of equal value so it can be safe regardless of the follower's choice. COBRA and COBRA-C exploit game theoretic reasoning to solve the problem at hand, utilizing their resources to better deal with the imprecise decisions of humans, but not trivially wasting resources as in MAXIMIN and UNIFORM.

Given the analysis presented, COBRA and COBRA-C, with appropriately chosen α values, appear to be the best performing among our new algorithms. The performance of DOBSS in these experiments also illustrates the need for the novel approaches presented in this paper for dealing with humans. Indeed, with DOBSS having been deployed for over a year at Los Angeles International Airport (LAX) [16], these results show that security at LAX could potentially be improved by incorporating our new methods for dealing with human adversaries.

4.2 Runtime Results

For our runtime 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 run-times 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. We assume each follower occurs with a 10% probability except the last which occurs with $1 - .10 * (n - 1)$ where n is the number of follower types. Experiments were run using CPLEX 8.1 on an Intel(R) Xeon(TM) CPU 3.20GHz processor with 2 GB RDRAM.

In Figure 2, we summarize the runtime results for our Bayesian game using DOBSS, BRASS, COBRA and MAXIMIN. The 8-door results are marked with solid figures and the 10-door results are marked with open figures. The value of α was varied to show the impact on solution speed. We include $\alpha = .25$ and $\alpha = .75$ in the graph, denoted by COBRA_25 and COBRA_75 respectively. The x-axis in Figure 2 varies the number of follower types from 1 to 8. The y-axis of the graph shows the runtime of each algorithm in seconds. All experiments that were not concluded in 1200 seconds were cut off. As expected, MAXIMIN is the fastest among the algorithms with a maximum runtime of 0.054 seconds on average in the 10-door case. Not anticipated was the approximately equivalent runtime of DOBSS and BRASS and even more surprising were the significant speedups of COBRA over DOBSS and BRASS depending on the value of α. As shown in Figure 2 as α increases, the runtime of COBRA decreases. For example, in the 10-door 8 follower type case when $\alpha = .25$ COBRA is unable to reach a solution within 1200 on average, however, when we increase α to .75 COBRA is able to find a solution in 327.5 seconds on average. In fact, every strategy except COBRA with $\alpha = .75$ reached the maximum runtime in the 10-door 8 follower type domain.

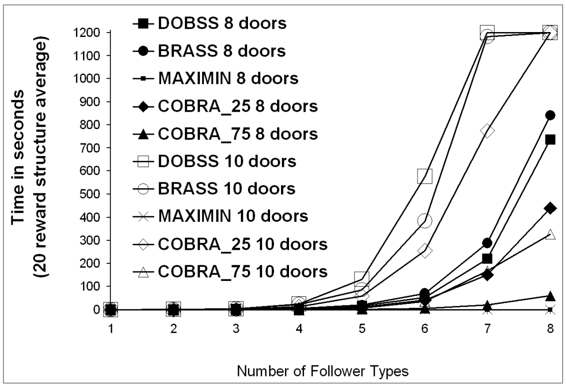


Figure 2: Comparing Runtimes

5. SUMMARY AND RELATED WORK

Stackelberg games are crucial in many multiagent applications, and particularly for security applications [4, 14]; the DOBSS algorithm, for instance, is applied for security scheduling at the Los Angeles International Airport [16]. In such applications automated Stackelberg solvers may create an optimal leader strategy. Unfortunately, the bounded rationality and limited observations of human followers challenge a critical assumption — that followers will act optimally — in DOBSS or any other existing Stackelberg solver, which may lead to a severely under performing strategy when the follower deviates from the optimal strategy. To apply Stackelberg games to any setting with people, this limitation must be addressed. This paper provides the following key contributions. First, it provides three new robust algorithms, BRASS, GUARD and COBRA, based on two key ideas: (i) human anchoring biases drawn from support theory; (ii) robust approaches for MILPs to address human imprecision. To the best of our knowledge, the effectiveness of each of these key ideas against human adversaries had not been explored in the context of Stackelberg games. These algorithms take a robust approach to solving Stackelberg games according to predictions on how and why human adversaries' responses will deviate from the optimal. Second, this paper provides experimental evidence that these new algorithms, in particular COBRA, perform statistically significantly better than existing algorithms and baseline algorithms when dealing with human adversaries as followers. These conclusions are drawn from experiments done on two settings based on real deployed security systems, in 4 different observability conditions, involving 98 human subjects playing 1360 games in total. These results show that COBRA is likely better suited for real-world applications dealing with human adversaries. Lastly, runtime analysis is provided for these algorithms showing that they maintain equivalent solution speeds compared to existing approaches.

In terms of related work, other non-game theoretic models have also been explored for security. The patrolling problem itself has received significant attention in multi-agent literature due to its wide variety of applications ranging from robot patrol to border patrolling of large areas [3, 10, 15]. We complement these works by applying Bayesian Stackelberg games to these domains. In particular, we turn to robust game theory, which was first introduced for Nash equilibria [2] and adapted to Wardrop network equilibria [13]. These prior works show that an equilibrium exists and how to compute it when players act robustly to parameter uncertainty. We also draw inspiration from approaches to bounded rationality in game theory [17] — the key question remains how to precisely model it in game theoretic settings. Limited observability provides a different challenge which we addressed via support theory [20]. Related work in support theory has shown that people exhibit anchoring biases and that they are slow to update away from these biases [18]. Combining these concepts in a novel context (Stackelberg games) we are able to address human adversaries as followers.

6. ACKNOWLEDGMENTS

This research was supported by the United States Department of Homeland Security through the Center for Risk and Economic Analysis of Terrorism Events (CREATE) under grant number 2007-ST-061-000001. However, any opinions, findings, and conclusions or recommendations in this document are those of the authors and do not necessarily reflect views of the United States Department of Homeland Security. This work was also supported in part by the National Science Foundation grant number IIS0705587 and the Israel Science Foundation.

7. REFERENCES

[1] Fox, C. personal communication.

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

[3] N. Agmon, V. Sadov, S. Kraus, and G. Kaminka. The impact of adversarial knowledge on adversarial planning in perimeter patrol. In *AAMAS*, 2008.

[4] G. Brown, M. Carlyle, J. Salmerón, and K. Wood. *Defending Critical Infrastructure*. Interfaces, 2006.

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

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

[7] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32 No. 100:675–701, 1937.

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

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

[10] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, M. Tambe, and F. Ordóñez. Computing Optimal Randomized Resource Allocations for Massive Security Games. In *AAMAS*, 2009.

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

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

[13] F. Ordóñez and N. E. Stier-Moses. Robust wardrop equilibrium. In *NET-COOP*, 2007.

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

[15] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus. Security in multiagent systems by policy randomization. In *AAMAS*, 2006.

[16] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed armor protection: The application of a game theoretic model for security at the los angeles international airport. In *AAMAS*, 2008.

[17] A. Rubinstein. *Modeling Bounded Rationality*. MIT Press, 1998.

[18] K. E. See, C. R. Fox, and Y. S. Rottenstreich. Between ignorance and truth: Partition dependence and learning in judgment under uncertainty. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32:1385–1402, 2006.

[19] H. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63:129–138, 1956.

[20] A. Tversky and D. J. Koehler. Support thoery: A nonextensional representation of subjective probability. *Psychological Review*, 101:547–567, 1994.

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

[22] K. K. Yuen. The two-sample trimmed t for unequal population variances. *Biometrika*, 61:165–170, 1974.