Security Games in the Field: Deployments on a Transit System

Francesco M. Delle Fave, Matthew Brown, Chao Zhang, Eric Shieh,
Albert X. Jiang, Heather Rosoff, Milind Tambe, and
J. P. Sullivan*

University of Southern California
CA 90049, Los Angeles, USA
http://teamcore.usc.edu/default.html
{dellefav,matthew.a.brown,zhan661,eshieh,jiangx,rosoff,tambe}@usc.edu
*Los Angeles County Sheriff's Department
CA 91754, Los Angeles, USA
jpsulliv@lasd.org

Abstract. This paper proposes the Multi-Operation Patrol Scheduling System (MOPSS), a new system to generate patrols for transit system. MOPSS is based on five contributions. First, MOPSS is the first system to use three fundamentally different adversary models for the threats of fare evasion, terrorism and crime, generating three significantly different types of patrol schedule. Second, to handle uncertain interruptions in the execution of patrol schedules, MOPSS uses Markov decision processes (MDPs) in its scheduling. Third, MOPSS is the first system to account for joint activities between multiple resources, by employing the well known SMART security game model that tackles coordination between defender's resources. Fourth, we are also the first to deploy a new *Opportunistic Security Game* model, where the adversary, a criminal, makes opportunistic decisions on when and where to commit crimes. Our fifth, and most important, contribution is the evaluation of MOPSS via real-world deployments, providing data from security games in the field.

Keywords: Security, Game-theory, Real-world deployment

1 Introduction

Research in Stackelberg security games has led to several real-world deployments to aid security at ports, airports and air transportation [16]. Such systems generate unpredictable security allocations (e.g., patrols and checkpoints), while carefully weighing each potential target, considering the scarcity of defender resources and the adversary's response. In a Stackelberg security game, the defender (e.g., the security agency) commits to her strategy first, taking into account the attacker's (e.g., a terrorist's) ability to conduct surveillance before launching his attack [5, 6].

Among the different applications of security games, the problem of patrolling a transit system has gathered significant interest [9, 17]. Due to the large volume of people using it every day, a transit system is a key target for illegal activities such as fare evasion (FE), terrorism (CT) and crime (CR). The security of such a system then, poses a number of challenges. The first challenge is multi-operation patrolling. Whereas most previous work in security games has focused on single threats which could be represented with a single adversary model (e.g., PROTECT, TRUSTS and IRIS)[16], the comprehensive security of a transit system requires different specialized security responses against three threats (FE, CT and CR). The second challenge is execution uncertainty. Security resources are often interrupted during their patrols (e.g., to provide assistance or arrest a suspect). Thus, traditional patrol schedules are often difficult to complete. Current research in security games has proposed the use of Markov decision processes (MDPs) to plan patrols under uncertainty [9]. However, such schedules were not actually deployed in the field, therefore, their real effectiveness has yet to be verified in the real-world. The fourth challenge is accounting for joint activities. In CT patrolling, security resources, such as explosive detective canine (EK9) teams, often patrol train lines in cooperation with other resources. By doing so, their effectiveness is increased. Recently, [14] proposed a new security game model, SMART (*Security games with Multiple coordinated Activities and Resources that are Time-dependent*), that *explicitly* represents jointly coordinated activities between defender's resources. [14]. Yet, similarly to the work of [9] discussed earlier, this framework has still not been deployed in the real-world. The fourth challenge is crime. Literature in criminology describes criminals as opportunistic decision makers [15]. At a specific location, they decide whether to commit a crime based on available opportunities and on the presence (or lack thereof) of security officers. Thus far, this type of adversary—less strategic in planning and more flexible in executing multiple attackes— has not been addressed in previous work, which has focused on strategic single shot attackers [16].

The fifth and most important challenge is that, despite earlier attempts [13], the actual evaluation of the deployed security games applications in the field is still a major open challenge. The reasons are twofold. First, previous applications focused on counter-terrorism, therefore controlled experiments against real adversaries in the field were not feasible. Second, the number of practical constraints related to real-world deployments limited the ability of researchers to conduct head-to-head comparisons

To address these challenges, this paper introduces five major contributions. Our first contribution is MOPSS, the first Multi-Operation Patrol Scheduling System for patrolling a train line. MOPSS provides an important insight: the multiple threats (FE, CT and CR) in a transit system require such fundamentally different adversary models that they do not fit into state-of-the-art multi-objective or Bayesian security game models suggested earlier [18, 4]. Instead, in MOPSS each of the three threats is modeled as a separate game with its own adversary model. These three game formulations provide security for the

same transit system, require data from the same transit system as input, use smart-phones to display the schedules and share several algorithmic insights. Our second contribution addresses execution uncertainty. We deployed MDP-based patrol schedules in the field, and used sampling-based cross-validation to handle model uncertainty in such MDPs [8]. Similarly, our third contribution is the deployment of coordinated schedules for CT patrolling. We incorporate the framework in [14] to MOPSS, and use it to generate patrols for counter-terrorism. Fourth, we address crime patrolling. Our contribution is the first ever deployment of opportunistic security games (OSGs). We model criminals as opportunistic players who decide whether to commit a crime at a station based on two factors, the presence of defender resources and the opportunities for crime at the station.

Our fourth contribution is the real world evaluation of MOPSS. This evaluation constitutes the largest scale evaluation of security games in the field in terms of duration and number of security officials deployed. As far as we know, it constitutes the *first* evaluation of algorithmic game theory in the field at such a scale. We carefully evaluated each component of MOPSS (FE, CT and CR) by designing and running field experiments. In the context of fare evasion, we ran a 21-day experiment, where we compared schedules generated using MOPSS against competing schedules comprised of a random scheduler augmented with officers providing real-time knowledge of the current situation. Our results show that our schedules led to statistically significant improvements over the competing schedules, despite the fact that the latter were improved with real-time knowledge. For counter-terrorism, we organized a full-scale exercise (FSE), in which 80 security officers (divided into 14 teams) patrolled 10 stations of a metro line for 12 hours. The purpose of the exercise was a head-to-head comparison of the MOPSS game-theoretic scheduler against humans. The comparison was in terms of the schedule generation process, as well as provide a thorough evaluation of the performance of both schedules as conducted by a number of security experts. Our results show that MOPSS game-theoretic schedules were able to perform at least equivalently to (and in fact better than those) generated by human schedulers. Finally, we ran a two-day proof-of-concept experiment on crime where two teams of officers patrolled 14 stations of a train line for two hours. Our results validate our OSG model in the real world, thus showing its potential to combat crime.

2 Transit Line Patrolling

The Los Angeles Sheriff's Department (LASD), the security agency responsible for the security of the Los Angeles Metro System (LA Metro), requested a multi-operation patrol scheduling system to improve and facilitate the comprehensive security of each train line. This system should generate randomized schedules for three different operations each addressing a fundamentally different threat:
Fare evasion patrols (FE): This type of patrol covers both the trains and the stations of a train line. The purpose is to capture as many fare evaders as

possible to improve the perception of order within an area. Thus, this type of patrolling should favor the locations with a large volume of riders because it would lead to a large number of fare evaders caught.

Counter-terrorism patrols (CT): This type of patrol covers the stations of a train line. Each station concentrates a large number of people at a specific place and time. In addition, in Los Angeles, several stations are located within key economic and cultural areas of the city (e.g., tourist locations, business and financial districts). Thus, the effects on society of any successful attack on the metro system would be catastrophic. Terrorists are then strategic adversaries who carefully study the weaknesses of a train line before committing an attack. To optimize security, this type of patrol should cover the different stations while favoring the stations either with large passenger volume and/or located in key areas.

Crime: This type of patrol covers the stations of a train line. Crimes can be of different types including robbery, assaults and drug dealing. Each of this crimes is a symptom that the train line's security is not sufficient. In addition, criminals behave differently than terrorists or fare evaders. They are opportunistic decision makers, they randomly traverse a train line, moving from station to station, seeking opportunities for crime (e.g., riders with smart-phones) [2, 15]. The key purpose of crime patrolling is then to patrol each of these stations, while favoring the stations representing "hot-spots" for crime (i.e., the most attractive stations from a criminal's perspective).

Given the three operations defined above, the LASD computes patrol schedules, manually, on a daily basis. This task, however, introduces a significant cognitive burden for the human expert schedulers. Thus, to generate more effective schedules in a timely fashion, we introduce MOPSS, described in the next section.

3 MOPSS

MOPSS addresses the *global* security of a transit system. Hence, it presents two key advantages for the LASD. First, it can be used to generate specialized patrols for substantially different threats and second it concentrates *all* the information relevant to the comprehensive security of each transit line (e.g., crime and ridership statistics). MOPSS is comprised of a centralized planner and a smart-phone application (shown as a demonstration in [11]). The system is shown in Figure 1. The core of MOPSS consits of the three game modules. Each module generates patrols for one operation (FE, CT or CR). Each operation deals with a fundamentally different adversary model (fare evaders, terrorists or criminals), therefore each operation is modeled as a different two-player security game (the defender's resources represent the security officers). Each module takes as input the information about the requested patrol (i.e., the number of officers, the starting time and the duration) and connects to a database to get the data necessary to build the security game model. Each game is cast as an optimization problem and sent to the SOLVER which contains three algorithms, one for each game

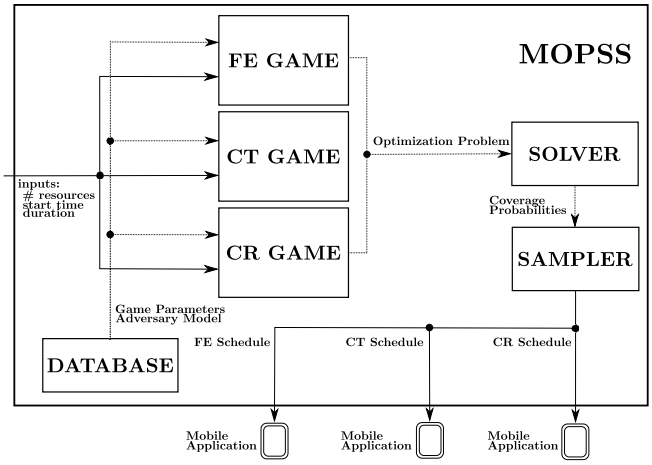


Fig. 1. The MOPSS system

[14, 20, 9]. Once the game is solved, the defender's mixed strategy is sent to the SAMPLER to produce the schedule which is uploaded into the application.

3.1 Fare Evasion Module

This module aims to generate the defender's (i.e., security officers') mixed strategies against fare evaders [9]. The idea is to use such strategy to derive patrol schedules that randomly favor the trains and the stations with a large volume of riders. Fare evaders are modeled as daily riders based on statistics.

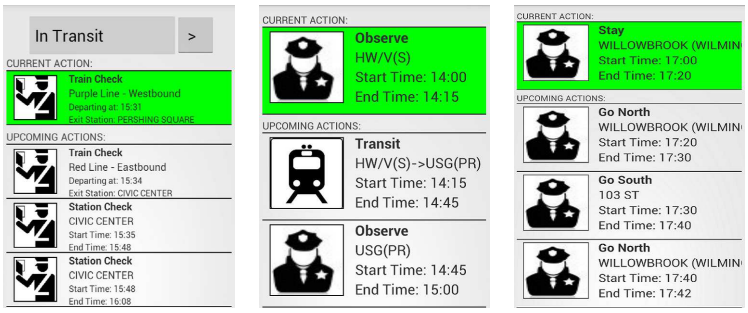
The key requirement of fare evasion patrolling is to be able to address execution uncertainty. To do so, in the FE module, the mixed strategy for each defender resource i is determined by an MDP denoted by a tuple $\langle S_i, A_i, T_i, R_i \rangle$ where: (i) S_i is a finite set of states ($s_i = (l, \tau)$ where l is a train or a station and τ is the time step); (ii) A_i is a set of two actions: perform a train or a station check (equivalently do a train or a station check) and (iii) $T_i(s_i, a_i, s_i')$ is the transition probability which can model execution uncertainty such as an officer being delayed while trying to conduct a fare check (e.g., due to arrests) and (iv) R_i is the immediate reward for transition (s_i, a_i, s_i'). Although this reward could potentially model more complex domains, it is unrelated to the game-theoretic payoffs, and is not considered in the remainder of this work.

The FE game is then represented as a two player Bayesian zero-sum game (see [9] for the definition of the linear program). Given a resource i and rider $\lambda \in \Lambda$ (i.e., defined by their daily itinerary in the train line), the objective is to maximize the expected utility of the defender, defined as $\max \sum_{\lambda \in \Lambda} p_\lambda u_\lambda$ where each utility u_λ is the defender's payoff against passenger type λ, which has a prior p_λ calculated using ridership statistics (calculated using ridership statistics). Each u_λ is calculated by the constraint $u_\lambda \leq \mathbf{x}^T U_\lambda \mathbf{e}_\alpha \forall \lambda, \alpha$ where each utility $U_\lambda(s_i, a_i, s_i', \alpha)$ represents the payoff that resource i will get for executing action

a_i in state s_i and ending up in s'_i, while the attacker plays action α (defined by the base vector $\mathbf{e}_\alpha$) and $\mathbf{x}$ is the marginal probability that the resource will actually go from s_i to s'_i. In other words, $\mathbf{x}$ represents the probability that the officer will overlap with a fare evader of type λ playing action α.

The optimization problem defined above is used by the SOLVER module to produce a mixed strategy represented as a Markov policy π_i. The SAMPLER then generates a single MDP patrol schedule that is loaded onto the handheld smartphone. An example of such a schedule is shown in Figure 2(a). The figure shows the schedule as it is visualized by the mobile application. The schedule contains two actions: train checks and station checks. Given that there is now a full MDP policy on the smartphone, a schedule can be updated whenever a security officer is delayed, by pushing the ">" button shown in Figure 2(a).

We next turn to instantiating the parameters in this game model for deployment. Fortunately, given fixed train fares and penalties for fare evasion, populating the payoff matrices is straightforward. Furthermore, via observations, we were able to set the transition function T_i. However, the delay length, whenever an office was interrupted, seemed to vary significantly, and modeling this variability became important. A continuous-time MDP or modeling multiple fine-grained delays are both extremely expensive. As a practical compromise we use a model considering a single delay whose value is chosen via cross-validation [8]. First, we randomly generate N MDPs, each of which assumes that resource i can experience delays of five different lengths: 6, 12, 18, 24 and 30 minutes (any delay longer than 30 minutes is considered to be beyond repair and a new schedule is generated). Second, we solve each MDP and obtain N Markov policies π_i^k corresponding to each MDP^k which we cross validate by running 100000 Monte Carlo simulations. In each simulation, we sample one strategy for the defender and calculate the resulting expected utility against all N MDPs. Finally, we pick the policy that maximizes the minimum. If the officer gets into a state not directly represented in the MDP, we pick the next available state at their current action.



(a) FE Schedule (b) CT Schedule (c) CR Schedule

Fig. 2. Three schedules for each threat of a transit system

3.2 Counter Terrorism Module

The counter-terrorism module aims to generate a defender mixed strategy that can be used to produce schedules that deter terrorists from attacking the stations of a train line [14]. Since stations are often composed of multiple levels, these schedules should then randomly patrol each of these stations while taking the levels into account and while favoring the most important stations. Terrorists are modeled as strategic adversaries who carefully observe the security of a train line before executing an attack.

The key requirement of CT patrolling is to represent joint activities. We achieve this by incorporating the SMART problem framework defined in [14] in the CT component of MOPSS. A SMART problem is a Security Game [10] such that each target $t \in T$ is assigned a reward $U_d^c(t)$ and a penalty $U_d^u(t)$ if t is covered and uncovered by a defender's resource. Similarly, each target is assigned a reward $U_a^c(t)$ and a penalty $U_a^u(t)$ for the attacker. The defender has a set of R resources. Each resource chooses an activity from the set $\mathcal{A} = \{\alpha_1, \alpha_2, \dots \alpha_K\}$ for each target $t \in T$. Each resource $r \in R$ is assigned a graph $G_r = (T, E_r)$, where the set of vertices T represents the set of targets to patrol and the set of edges E_r represents the connectivity between such targets. Each edge $e \in E_r$ is assigned a time value $\tau(e)$ representing the time that it takes to one defender resource r to traverse e.

The attacker's pure strategy space is the set of all targets, T. A pure strategy for the defender is a set of routes, one route X_i for each resource. Each route is defined as a sequence of activities α, conducted at a specific target t with specific duration γ. Joint activities are then represented when there exists two routes X_i and X_j such that $t_i = t_j$ and $|\gamma_i - \gamma_j| \leq W$, i.e. when two activities of two different resources overlap in space and time (within a time window W). For each activity α_i, $\texttt{eff}(\alpha_i)$ represents the individual effectiveness of the activity α_i, which ranges from 0% to 100%, and measures the probability that the defender will be able to successfully prevent an attack on target t. The effectiveness of the joint activity $\langle \alpha_i, \alpha_j \rangle$ is defined as $\texttt{eff}(\alpha_i, \alpha_j)$.

Given these parameters, the expected utilities $U_d(\mathbf{P}_i, t)$ and $U_a(\mathbf{P}_i, t)$ for both players, when the defender is conducting pure strategy $\mathbf{P}_i$ (defined as a joint pure strategy for multiple defender resources), and when the attacker chooses to attack target t is given as follows:

$$\omega_t(\mathbf{P}_i) = \max_{\substack{(t,\alpha,\gamma)\in\mathbf{P}_i \\ \{(t,\alpha_l,\gamma_l),(t,\alpha_m,\gamma_m)\}\subseteq\mathbf{P}_i, |\gamma_l-\gamma_m|\leq W}} \{\texttt{eff}(\alpha), \texttt{eff}(\alpha_l, \alpha_m)\} \tag{1}$$

$$U_d(\mathbf{P}_i, t) = \omega_t(\mathbf{P}_i) U_d^c(t) + (1 - \omega_t(\mathbf{P}_i)) U_d^u(t) \tag{2}$$

$$U_a(\mathbf{P}_i, t) = \omega_t(\mathbf{P}_i) U_a^c(t) + (1 - \omega_t(\mathbf{P}_i)) U_a^u(t) \tag{3}$$

Here $\omega_t(\mathbf{P}_i)$ defined in Equation (1) represents the *effective coverage* of the defender on target t when executing pure strategy $\mathbf{P}_i$.

To solve this problem, we use SMART_H, a branch-and-price, heuristic approach, which we incorporate in the SOLVER component of MOPSS. SMART_H is based on a branch-and-price framework, it constructs a branch-and-bound

tree, where for each leaf of the tree, the attacker's target is fixed to a different t'. Due to the exponential number of defender pure strategies, the best defender mixed strategy is determined using *column generation*, which is composed of a *master* and *slave* procedure, where the slave iteratively adds a new column (defender strategy) to the master. The objective of the pricing component is to find the best defender mixed strategy $\mathbf{x}$ at that leaf, such that *the best response of the attacker* to $\mathbf{x}$ is to attack target t'. The structure of the algorithm is illustrated in Figure 3. In the figure, the master solves the non-zero-sum game to get a defender mixed strategy over a small subset of joint patrol pure strategies. After solving the master problem, the duals are retrieved and used as inputs for the slave. The purpose of the slave is to generate a pure strategy which is then added to the master and the entire process is iterated until the optimal solution is found.

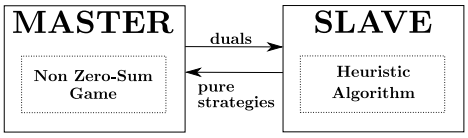


Fig. 3. The column generation algorithm

An example counter-terrorism schedule, as visualized by the mobile application, is shown in Figure 2(b). The schedule describes two actions, observe (patrol a station) and transit (go to a station) each with a specific time and duration. The key challenge to deploy CT schedules is to define an accurate SMART problem instance to accurately encompass the real-world problem. To achieve this, we had to define three types of features. First, we had to define the payoffs of the game[1]. We defined the payoffs for each target (32 in total) in discussions with security experts from the LASD. Each set of payoffs for each station was based on the number of people using the station every day and by the economic impact that losing this station would have on the city. The different levels of a single station had slightly different payoffs which were based on the number of persons present at each specific level of the station every weekday. Second, we had to define the defender different resources, i.e., the type of teams participating to the experiment, which we will refer to as type 1 to type 5[2]. Third, we had to define the single and joint effectiveness for both the observe and transit actions. All Transit actions were given a 0 effectiveness, since moving from one station to another (i.e., riding the trains or taking the car) will not have any effect on the security of the stations. Most teams were assigned the same positive individual effectiveness of 0.7, except one Type 3 which has a greater individual effectiveness because it is composed of officers from multiple agencies carrying

[1] We are not able to reveal the value of these payoffs due to an agreement with the LASD.

[2] The real name of each type is omitted as requested by the agencies participating to the FSE

heavy weapons. Resources of types 1, 2 and 3 typically work alone. Hence, to define their effectiveness values, their individual effectiveness is positive while their joint effectiveness is null (any joint effectiveness value below 0.7 would induce the same type of behavior, but we chose 0 since it is a clear indicator of the type of behavior that we want to obtain). Resources of type 4 are assigned a joint effectiveness greater then their individual effectiveness because they can perform all type of activites, but, typically, they prefer joint over individual activities. In contrast, resources of type 5 typically work only in cooperation with other teams, therefore they are assigned a null individual effectiveness and a positive joint effectiveness of 0.75.

3.3 Crime Module

The crime module aims to generate a defender mixed strategy to prevent crime on a train line. The idea is to generate schedules that take criminal behavior into account and attempt to predict the stations that are more likely to be affected by crime. Crime statistics are used to characterize the behavior of criminals and the attractiveness that they attribute to each station of the train line. The key difference with the previous modules is that criminals behave differently than fare evaders and terrorists. They are less strategic in planning crimes and more flexible in committing them than is assumed in a Stackelberg game. They opportunistically and repeatedly seek targets and react to real-time information at execution-time, rather than strategically planning their crimes in advance.

Crime schedules are computed using an OSG [20]. An OSG is similar to a Stackelberg game in that the defender commits to a patrol strategy first, after which the criminal chooses the station(s) to attack given their belief about the defender's deployment. In an OSG, the defender's actions are computed using a Markov chain, which assigns probabilities for how the defender should move through the train line. The criminal's behavior is defined by a quantal-biased random walk, i.e., the next station to visit for potentially committing a crime is determined according the quantal response model [16]. This model takes as input information the attractiveness $Att(i)$ of each station i and the criminal's belief about the defender's strategy which is updated using real-time observations. Station attractiveness is a measure based on crime statistics about the availability of opportunities for committing crime as well as how likely criminals are to seize such opportunities. The behavior models for both the defender and the criminal are combined to form a Markov chain with transition matrix T_s, which along with the rewards to the defender, define an OSG that can be solved to generate an optimal defender strategy. To solve an OSG, we iteratively calculate the defender expected utility V_d over all the possible states of the Markov chain for a number of crime opportunities k as follows:

$$\begin{aligned} Obj &= \lim_{K\to\infty} \sum_{k=0}^{K} V_d(k+1) \\ &= \mathbf{r}_d \cdot (I - (1-\alpha)T_s)^{-1}\mathbf{X}_1, \end{aligned} \tag{4}$$

where R_d is a vector defining the utility of each state of the Markov chain in terms of the payoff u_d for the defender and the attractiveness $Att(i)$; I is the identity matrix; α is the probability of leaving the train line after an attack and X_1 is the initial coverage probability over all the possible states of the Markov chain. By maximizing Obj (i.e., minimizing the total amount of crime in the metro), we obtain a transition matrix T_s^*. This matrix is then used to compute the defender's Markov strategy π.

The maximization of Equation 4 is a nonlinear optimization problem. Therefore, to scale up to the number of states necessary to represent a real train line we use the Compact OPportunistic security game State algorithm (COPS) [20] in the SOLVER module. COPS returns a number of coverage probabilities for the different stations of the train line. These are then sent to the SAMPLER module which generates a schedule. An example of a schedule for crime patrolling is shown in Figure 2(c). It describes three actions, go north (i.e., take the next northbound train), go south (i.e., take the next southbound train) and stay (i.e., patrol a specific station).

To deploy crime schedules, two key challenges had to be addressed. The first challenge deals with defining of the attractiveness parameter. In our work, we define the attractiveness $Att(i)$ of station i following the statistical model presented in [15]. Formally, $Att(i) = 1 - \exp^{-aN(i)}$, where $N(i)$ is the number of past crimes at station i (based on actual crime statistics received from the LASD) and a is a weighting coefficient. The second challenge is the parameterization of the criminal behavior model, which consists of defining the quantal-biased random walk. In our crime tests (Section 4.3), we defined the criminal behavior in collaboration with both security agencies and criminologists.

4 Real World Evaluation

In collaboration with the Los Angeles Sheriff's Department (LASD), we designed three types of real world tests, one for each of the three operations defined in Section 2. Each of these tests allows us to evaluate different aspects of game-theoretic patrolling. This evaluation introduces the following novelties: (i) in fare evasion, we present the first real world deployment of game-theoretic schedules and analyze their performance against real adversaries (fare evaders); similarly, (ii) in counter-terrorism, we present the first real world head-to-head comparison between game-theoretic and human generated schedules. Finally, (iii) in crime, we introduce the first deployment of OSGs. The crime tests provide the first real world data showing the benefits of game-theoretic scheduling when facing opportunistic attackers.

4.1 Fare Evasion Experiment

This experiment took place over 21 days during the months of July and August 2013. The organization of the experiment (e.g., train the security officers, design and organize the experiment in collaboration with the LASD) required

approximately two weeks. This experiment had two key purposes. The first was to validate the MDP model of the Fare Evasion module (Section 3.1) in the real world. The second was to run a head-to-head comparison between the game-theoretic approach calculated using MOPSS and a Markov policy that takes execution uncertainty into account, but that assign actions based on a uniform random probability. The uniform random Markov strategy (Markov UR) assigns, given a state $s \in S_i$ of the MDP defined in Section 3.1, a uniform probability to all the actions taken in s leading to another state $s \in S_i$. It was chosen because it constitutes is the approach that security agencies adopt when they are not using a game-theoretic approach for randomization. This section discusses the setup of the experiment and the results that we obtained.

Fig. 4. The map of the blue line of the LA Metro

Experiment Setup The fare evasion experiment took place on the Blue line of the LA Metro system (see Figure 4 for the map of the metro line). Other lines could not be tested, because the LASD only allowed us to use our strategies on the Blue line during our real-world test. This line consists of 22 different stations and is one of the biggest lines in the LA Metro system. It was selected by the LASD, which helped to organize the experiment (e.g., assign security officers and patrol times).

Each day, a team of two security officers (see Figure 5), was randomly selected by the LASD, to patrol the line for a duration of at most 120 minutes. Patrols were run during both the morning and the afternoon. Some days the tests ended early due to the officers being reassigned. One of the two officers acted as the leader of the team: he was given the smartphone, he had to read the schedule

Fig. 5. Two security officers performing fare checks on a train.

to the other officers, collect the data and eventually update it whenever a delay occurred. An update could be made either during a station-check, i.e., when checking riders at a station, or during a train-check, i.e., when checking riders on the coach of a train. In the latter case, the officers were required to leave the train at the next station to request an update. This was required because the Markov strategy is defined over each state of the MDP (i.e., station, time). Thus any new strategy has to be sampled from a specific state. Every week the team was provided with one of two types of schedules:

Game-theoretic schedules (GT): This type of schedule was generated using MOPSS' fare evasion component (Section 3.1).

Markov UR schedules (UR): This type of schedule was generated by modeling the problem as an MDP. However, the corresponding Markov strategy π_{s_i,a_i}, for each state s_i and action a_i was calculated assuming a uniform probability distribution.

The officers were not told which schedule they were using as not to bias their performance. Before the experiment, we anticipated that the officers might view some of the schedules as leading to low performance in terms of catching very few fare evaders. In such situation, the officers, in order to avoid poor performance, might end up voluntarily deviating from their given schedules to reach a better location because they were unsatisfied with the current one. In anticipation of such voluntary deviations, we augmented both the game-theoretic and UR schedules with the ability to perform updates. More specifically, we allowed the officers to request VOLUNTARY or INVOLUNTARY updates. VOLUNTARY updates consisted of the officers updating the current schedule because in their opinion, the current specified action was not fruitful as a venue to check fares. Officers were allowed to choose a new location that they considered more fruitful for catching fare evaders and request a new schedule from there. INVOLUNTARY updates consisted of the officers requesting a new schedule because they were delayed (e.g., from issuing citations or arresting a suspect) and were unable to perform the next action on their schedule. This type of update could be requested *anytime* an officer was delayed. As we will see below the officers

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	total
GT	60	60	90	60	90	10	90	110	90	90	105	855
UR	60	60	60	60	60	75	100	100	100	90		765

Table 1. Patrol duration over each of the 21 days.

used VOLUNTARY updates almost every day with the UR schedules, but never in the GT schedules.

Finally, it is important to notice that given the duration of our experiment, the game-theoretic schedules are essentially testing a maximin strategy. As discussed in Section 3.1, the fare evasion component computes a Stackelberg strategy, a strategy based on the assumption that the riders will conduct surveillance and observe the defender's mixed strategy. However, considering only 21 days of patrol whereby the officers could only patrol less than few hours per day, either in the morning or the afternoon, we cannot assume that the riders had sufficient time to conduct accurate surveillance, observe the mixed strategy and best respond to it. Nonetheless, the FE component in Section 3.1 solves a zero-sum game for which a Stackelberg equilibrium and the maximin strategy are known to be equivalent [19]. Thus, since the maximin strategy provides a guaranteed level of defender utility without making any assumption on the adversary's surveillance of the defender's mixed strategy, these experiments compare the benefit of using a maximin strategy against other (non-game-theoretic) approaches for generating patrol schedules.

Results During the 21 weekdays of our experiments, we were able to run GT schedules for 11 days of testing while UR schedules were deployed for 10 days, resulting in 855 and 765 patrol minutes, respectively. The schedules were compared using two different metrics. First, we counted the number of passengers checked and the number of captures at the end of each patrol. The captures were defined as the sum of the number of warnings, citations, and arrests. Passengers without a valid ticket could be given a warning or cited for a violation on the discretion of the officer. This metric was chosen because it would allow us to measure the performance of each schedule in the real world. Second, we counted the number of times that the update function was used voluntarily and involuntarily. While involuntary updates helped determine the value of using MDPs as discussed below, voluntary updates measured the human (officer) perception of quality of the schedules – the more such voluntary updates, the more the officers were dissatisfied with their given action. Table 1 shows the duration of each day of patrol for both GT and UR schedules[3].

As shown in the table, the actual duration of a daily patrol was often different over the 21 days of the experiment, for both GT and UR schedules. For this

[3] As shown in Table 1, each day of patrol correspond to a 2-day test where GT schedules were tested on the first day and UR schedules were tested on the second, both at identical times.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	total
GT	0	1	3	1	1	0	2	2	4	2	1	18
UR	0	2	1	1	1	2	2	2	3	2		16

Table 2. Number of INVOLUNTARY (delays) deviations for each day of patrol

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	total
GT	0	0	0	0	0	0	0	0	0	0	0	0
UR	1	0	1	1	1	0	1	1	1	1		8

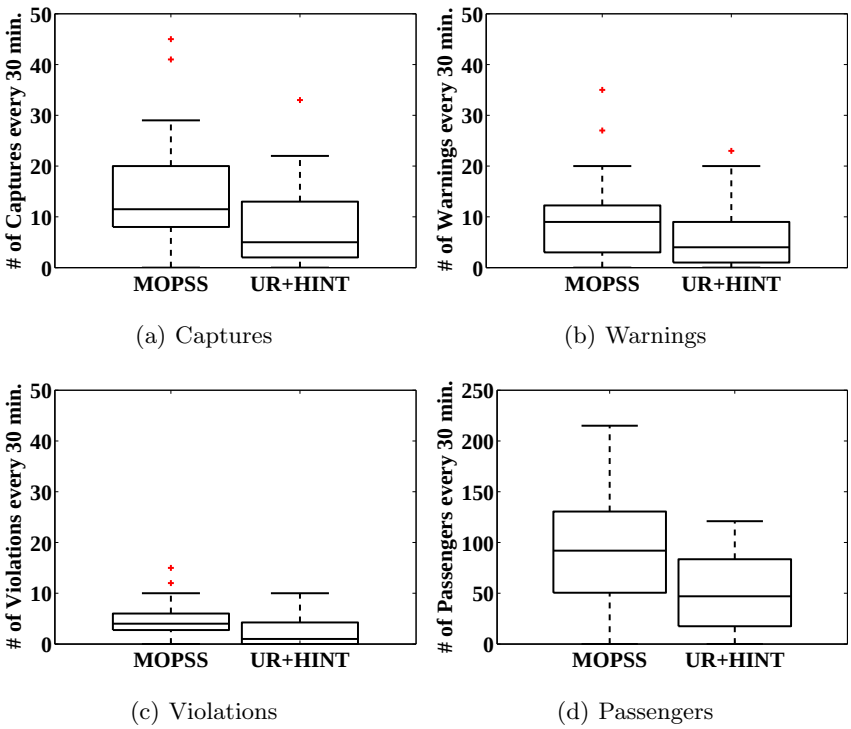
Table 3. Number of VOLUNTARY (updates) deviations for each day of patrol

reason, providing a comparison normalized over the days of the experiment was impossible. However, most of the days, we were able to collect data for multiples of 30 minutes (e.g., 60, 90 minutes). Hence, to properly compare our results, we divided our data in 30 minutes segments. More specifically, we considered all the train and station checks within a time window of 30 minutes and collected the data resulting from these actions[4]. Having defined the data points, we can now proceed to analyze our results.

Validation of the MDP model: As discussed at the beginning of this section Both GT and UR schedules were calculated by solving an MDP. For this reason both schedules could be updated to request a new schedule. Tables 2 and 3 then show, for each day of patrol, the number of VOLUNTARY and INVOLUNTARY deviations requested by the officers. In total, GT schedules were updated 18 times, all of which were *involuntary* deviations, i.e., delays. All these update requests confirm that the MDP model was able to provide schedules that could be updated whenever necessary.

All INVOLUNTARY deviations were due to the officers writing citations or helping people. The average delay length was of 12 minutes (the largest delay was of 20 minutes). In each case, as discussed at the beginning of this section, a new schedule was provided starting at the officers' current location and closest time. Finally, Table 3 shows that voluntary deviations were used only with UR schedules. This result strongly suggests that the officers were mostly satisfied with GT schedules. In addition, it means that GT schedules did not really compete against UR schedules only. Rather, the comparison was between UR schedules which were augmented with real-time human intelligence for most of the time (8 out of 10 days). We discuss the results of such comparison next.

[4] In so doing, the segments are also statistically independent. Within each segment the officers will check different people who are unable to affect each other. Each segment corresponds to a sample of different train riders taken at different times and locations. Not only do the officers never check the same rider twice but most importantly, during 30 minutes, they will visit different locations by riding the trains (roughly, one train every 6 minutes in the blue line) and inspecting the stations (on-station operations last no longer than 20 minutes).



(a) Captures (b) Warnings

(c) Violations (d) Passengers

Fig. 6. Results of the Fare Evasion tests

Game-Theory vs. Uniform Random: The results that we obtained are shown in Figure 6 and in Table 4. Figure 6 shows eight boxplots depicting the data that we collected during each patrol, using both GT and UR schedules. Respectively, the four figures present data collected on captures (Figure 6(a)), warnings (Figure 6(b)), violations (Figure 6(c)), and passengers checked (Figure 6(d)) per 30 minutes of patrolling[5]. For each boxplot, the top and bottom of the box represent the 75th and 25th percentiles, respectively, while the middle line indicates the median of the collected data. The "+" data points indicate statistical outliers, while the whiskers show the most extreme non-outlier data points. Each of the four figures (captures, warnings, violations and passengers checked) shows that the data collected using GT schedules had higher values than the data collected using UR schedules. As shown in Table 4, on average, GT schedules led to, respectively 15.52 captures, 10.42 warnings and 5.03 violations issued every 30 minutes against, respectively against 9.55 captures, 6.48 warnings and 3.07 violations obtained using UR schedules. To confirm the statistical significance of these results, we ran a number of weighted unpaired student

[5] GT schedules also led to two arrests on day 6. This is why the patrol only lasted 10 minutes.

	Days	avg. C	avg. W	avg. V	avg. P
GT	11	15.52	10.42	5.03	96.77
UR	10	9.55	6.48	3.07	60.85

Table 4. Average captures (C), warnings (W), violations (V) and passengers (P) based on the results obtained in Figure 6

t-tests ($p = 0.05$) [7, 1] and verified, for each metric, that the difference in the results was statistically significant. We used a weighted t-test because some data segments had a duration shorter than 30 minutes and we wanted to use all the available data for our analysis. As shown in Table 1, not all the patrol durations could be properly divided into a finite number of 30 minutes segments (e.g., UR: D6, D7, D8, D9 and GT: D6, D8, D11). Therefore, we calculated a weighted average for each of the metric defined above, whereby each segment was given a weight which was defined based on the segment's duration (longer segments corresponded to higher weights).

From a practical perspective, the magnitude of the difference between the two approaches is significant: cumulatively over a period of 21 days GT would capture a much larger total number of fare evaders. This result can be emphasized even further if we correlate it with the results shown in Tables 3 and 2. While running UR schedules the officers were requesting INVOLUNTARY deviations essentially every day, whereas no such deviations were requested while running GT schedules. In other words, they were using real-time situation awareness to augment the quality of the schedules, thus making the UR schedule more compelling.

The results in Table 4 also indicate that GT schedules led to 96.77 passengers checked every 30 minutes against 60.85 passengers checked by using UR schedules. As discussed in [9], GT schedules are generated by leveraging all the possible sequences of train and station checks and by taking into account key dimensions such as the train schedules, the officers' effectiveness and, most importantly the daily ridership statistics. This means that stations or trains with a higher presence of riders will be given a higher coverage probability since they are more likely to contain fare evaders. Hence, these results confirm the accuracy of the model as both Figure 6(d) and Table 4 show that GT schedules led the officers to check more passengers than UR schedules.

This raises the question of whether a static type of schedule, which only deploys the officers at the most crowded locations, would lead to similar or even better results than those obtained with GT. Given the limited amount of time that we had to conduct our experiments, we were unable to compare GT schedules against a static deployment – where the key weakness is predictability in the longer term. Indeed, effective randomization was one of the main reasons for LASD to collaborate on these experiments – security agencies know that static schedules become predictable in the long term[6]. After a certain amount of

[6] [16] discusses the benefits of randomization in detail.

time, the passengers would know where the officers are located and could exploit this information to avoid paying the fare.

4.2 Counter-Terrorism Experiment

The purpose of this experiment is to run a head-to-head comparison between MOPSS and a manual allocation, the standard methodology adopted by several security agencies. Security agencies refer to this type of experiment as a mass transit full scale exercise (FSE). A FSE is a training exercise where multiple security agencies analyze the way their resources cooperate to secure a specific area while simulating a critical scenario. This scenario typically describes a "high level" threat, e.g., intelligence reports confirming that a terrorist attack might take place in the Los Angeles Metro System. The FSE consists of simulating the response to this threat, i.e., increasing the number of resources patrolling a train line on a daily basis to improve the quality of the security.

Setup: The FSE consisted of patrolling 10 stations of one train line of the LA Metro system for 12 hours. Each station on the train line is composed of three levels (street level, platform level and mezzanine) except station 1 which is composed of 5 levels (2 more platform levels). The exercise involved multiple security agencies, each participating with a number of resources. Overall, 80 security personnel were involved. These resources were divided into 14 teams, each with different abilities (see Section 3.2).

The exercise was divided into 3 different "sorties", each consisting of three hours of patrolling and one hour of debriefing. Human-generated schedules were used during the first sortie while MOPSS schedules were used during the second and the third sorties. The first two sorties were used to run the head-to-head comparison. Hence, the sorties were ran under the same settings: the same number of officers had to cover the 10 stations for a cumulative time of 450 minutes. The two sorties were ran during off-peak times (9h00 to 12h00 and 13h00 to 16h00, respectively), hence the type and the number of riders of the train lines could be considered to be, approximately, the same. The purpose of Sortie 3 was to test whether the officers were capable of following MOPSS schedules for a longer period (900 minutes instead of 450) and during peak time. We found out that the officers were actually able to follow the schedules. Thus, since the purpose of this Sortie was unrelated to our comparison, we will focus on Sorties 1 and 2 in the remainder of this section. Each type of schedule was generated as follows:

MOPSS schedules: The schedules were generated by (i) instantiating a CT game using the specifics of the FSE discussed earlier; (ii) solving this problem instance using the SOLVER and (iii) sampling a pure strategy in the SAMPLER to generate the patrol schedule for each of the different resources involved. Specifically, we ran the SMART$_H$ in the SOLVER component, considering 14 resources and 32 targets. The algorithm produced a mixed strategy which was then sampled to generate a pure strategy in the SAMPLER. This pure strategy contains a schedule for each resource.

Manual Schedules: The schedules were generated by human expert schedulers of the LASD. They were generated using a two-step process. First, each station was assigned a coverage duration of 45 minutes (i.e., $\frac{1}{10}^{th}$ of the time). The idea was to have the officers perform *three observe* actions at each station. Second, the human expert schedulers assigned teams to each station so that each station was covered for exactly 45 minutes. Joint team activities were used 6 times in six different stations. This simple two-step process was adopted to avoid the cognitive burden involved with leveraging the effectiveness of each team to cover the different stations individually or while coordinating with other teams. Despite its simplicity, this process was difficult for the human expert schedulers. It involved several discussions and required one entire day of work.

Results: We first analyze the type of schedules generated as a result of using either MOPSS or manual scheduling. Then, we evaluate the results obtained by deploying the schedules during Sorties 1 and 2 and measuring their performance in the real-world.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Manual	3	3	3	2	3	2	2	2	2	2
MOPSS	2	2	3	3	2	2	2	3	3	2

Table 5. Count of Individual Activities

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Manual	0	0	0	1	0	1	1	1	1	1
MOPSS	1	0	0	0	0	2	0	1	1	1

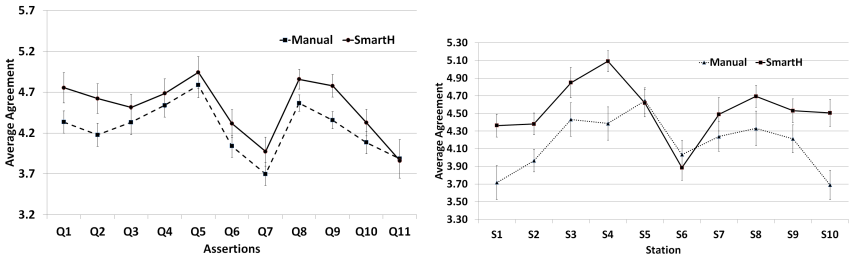
Table 6. Count of Joint Activities

The numbers of individual and joint activities for both the schedules generated during the FSE are shown in Tables 5 and 6. In both tables we can see that the number of individual (IA) and joint (JA) activities for both approaches are the same (IA: both 24; JA: both 6). All the joint activities in the MOPSS schedules are performed by CRM and EK9 teams, i.e., the teams with a positive joint effectiveness. This is similar to the behavior of the manual generated schedules, where joint activities are mostly performed by resources of types 4 and 5 (once by a team of resources of type 3). The remaining individual activities are performed by resources of type 1, 2 and 3.

There are two important differences between the two types of schedules. First, MOPSS sent the most effective type, type 3, to the most important stations because its individual effectiveness is greater than the effectiveness of other

teams. This was not seen in the human schedule. Second, the schedules generated using MOPSS assigned the different teams to cover all the different levels of the different stations, whereas manual schedules did not specify such levels. The reason for this is that human schedulers were not able to reach this level of detail and thus they preferred to leave the decision of which level to patrol to the teams once they were deployed. In addition, the effort required to generate the schedules using MOPSS was much lower than the effort required to generate manual schedules, which required one day of work due to its significant cognitive burden. Since typically such patrols would be conducted day-in and day-out for several days in high-threat periods, the savings of human effort achieved by game-theoretic schedulers are thus very significant.

Each type of security allocation (either manual or game-theoretic based on MOPSS) was evaluated by security experts. A team of security experts (SEs) was placed at each station for the entire length of the exercise. Their task was to observe and evaluate the officers' patrolling activity during each sortie, and determine how their behavior was affecting the quality of the security within each station. In what follows, we report the conclusions of their analysis. The SEs did not know what type of schedules (so as to not bias their evaluation). To translate the observers' observations into a comparable value, each observer was asked to fill out a questionnaire every 30 minutes. The objective was to define a number of key sentences that could help to qualify the way in which the security officers had been patrolling the station in the last 30 minutes. Each questionnaire contained 11 *assertions* about the level of security within the station. The assertions were defined in collaboration with a team of SEs from the LASD and with social scientists. Each SE had to determine his level of agreement with each assertion, which was defined in the integer interval {0,6}, where 0 meant a strong disagreement, whereas 6 meant a strong agreement.



(a) Assertions (b) Stations

Fig. 7. Evaluation of the FSE: average agreement over the different questions and stations.

Figures 7(a) and 7(b) show the results that we obtained. Figure 7(a) shows the weighted average agreement obtained for each assertion calculated over all the stations (the average was calculated considering each station's correspond-

ing weight). Figure 7(b) shows the average agreement obtained for each station calculated over all the assertions. The error bars in both figures show the standard error of the mean calculated for each specific assertion (in Figure 7(a)) and station (in Figure 7(b)). As we can see the difference between some data points of the two approaches do not seem to be statistically significant. A student t-test confirmed this trend. This is expected, since we were only able to collect data for few hours of a single day. Nonetheless, we can still acquire some interesting information about the performance of game-theoretic schedules in the field, by analyzing the results that are statistically significant.

In Figure 7(a), we can see that MOPSS schedules seem to yield a higher level of agreement than manual schedules over all questions. As shown in the figure, the difference is significant only for assertions Q_1, Q_2, Q_8 and Q_9.These four assertions correspond to very general statements about the security at each station which address the efficiency of the schedules, their ability to provide a strong feeling of safety and to allow the officers to patrol each area as much as needed.

Similarly, in Figure 7(b), we can see that the average agreement is higher for MOPSS schedules over Manual schedules for stations S_1, S_2, S_3, S_4, S_8, S_9 and S_{10}. Some of these stations (S_1, S_8 and S_9) are the ones assigned a higher set of payoffs, as discussed above. Hence, they correspond to the ones given a higher coverage by MOPSS.

These results indicate that game-theoretic schedules were evaluated as more effective than manual schedules. By analyzing the differences between the schedules, we can infer that this happened for two key reasons. First, as discussed earlier, manual schedules were generated by leaving the decision of which level of a station to patrol to each deployed team. The officers then, were not able to properly coordinate over the different levels to patrol and therefore they ended up patrolling the same levels. Second, MOPSS produced a schedule which more effectively scheduled resources of type 3, i.e., the team with the highest effectiveness (0.8) for covering each target. More specifically, the resources of type 3 patrolled *all* the most important stations at key levels. In contrast, manual schedules assigned the same type of resources, without accounting for their effectiveness. This made an impact on the security evaluators, which considered the game-theoretic allocation more effective than the manual allocation, because it was leveraging the abilties of the resources in a way that human experts could not achieve.

4.3 Crime Experiment

Our crime experiment was designed to be a proof-of-concept of MOPSS crime component. As discussed in Section 3.3, OSGs are a new framework to represent opportunistic adversaries. The purpose of our experiment is then to validate this new framework in the real world to ascertain its ability to generate effective schedules against crime. The experiment was organized as follows:

Setup: We ran tests for two days with each test consisting of a two hours patrol involving two teams of two security officers. Each team had to patrol

seven stations of a particular LA Metro train line using schedules generated using MOPSS. MOPSS generated the schedules by converting crime statistics into a set of coverage probabilities for the different stations. Figure 8 shows such probabilities and correlates them to the crime statistics for each of the 14 stations to patrol. In the figure, the x-axis enumerates the 14 stations to patrol. The bar graphs (y-axis on the right) show, for each station, the total number of crimes that happened during 2012 and 2013. Finally, the line graph shows the different coverage probabilities calculated for each station (y-axis on the left). In the figure, the stations with a larger coverage probability (stations 5 to 10) are either the stations with a large number of crimes (stations 5 and 8) or the adjacent stations (Stations 6, 7, 9 and 10). The latter stations are given a large coverage probability because the OSG model anticipates the possibility that criminals will choose stations 6, 7, 9 and 10 anticipating that stations 5 and 8 will be frequently patrolled by security officers [20]. Hence, these coverage probabilities show how game theory allows to build real world patrol schedules.
Results: During the tests, the officers were able to write 5 citations and make 2 arrests. In general, they were able to understand and follow the schedule easily. Overall, these tests indicate that the CR module in MOPSS can produce effective schedules that would work in the real world.

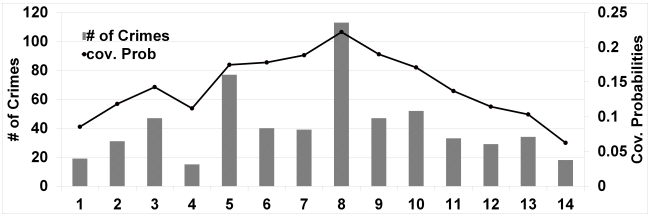


Fig. 8. Crime Statistics and Coverage Probabilities

5 Lessons learned

The work presented in this paper is the result of a long term collaboration between university researchers and personnel from different security agencies including decision makers, planners and operators. To interact with such security agencies, we took inspiration from the lessons presented in [13]. We discussed the strengths and weaknesses of every aspect of MOPSS and emphasized the requirement of learning from the field to ascertain the performance of our system. In addition, The field experience allowed us to discover two new insights regarding real-world applied research in security games: (i) testing this research in the field requires a period of "immersion" and (ii) users are a key factor when when running field experiments.

The first insight is a key lesson for running field experiments. Any real world test of a security game based system will involve real security officers protecting a critical area for a long period of time. To succeed in such an experiment,

researchers should immerse themselves in order to deeply understand the way officers and, more generally, a security agency operate every day. A period of "immersion", as we did for both the FE and the CT experiments, also ensures that the security agencies do not think researchers as ivory tower occupants leading to easier acceptance of technology. To test MOPSS, we spent several months observing the different security agencies patrolling the LA Metro to understand how they operate so as to set up effective field experiments.

The second insight comes from our interactions with the security personnel. These officers are the end users of our system. Thus, it is critical that they understand exactly the benefits of game-theoretic scheduling. Not doing this could severely affect the results of the evaluation. As an example, at the beginning of our FE tests (Section 4.1), the officers required a number of days to understand that their schedules could be updated without having to request a new allocation to the dispatch.

6 Summary

This paper steps beyond deployment to provide results on security games in the field, a challenge not addressed by existing literature in security games. Readers will notice that the paper does not contain any simulation results as all of our results are based on real world experiments. We presented MOPSS, a novel game-theoretic scheduling system for patrolling a train line. MOPSS introduced five contributions not addressed in previous applied systems, including both TRUSTS [18] and the system in [17].

The first contribution is multi-operation patrolling. Thus far, all existing game-theoretic scheduling systems [16] (in particular TRUSTS) and the system in [17] were focused on a single mission. In contrast, MOPSS is the *first deployed system* to use three significantly different adversary models to develop three different patrol schedules for the threats of fare evasion, terrorism and crime. In contrast with previous work suggesting such threats could be modeled as a multi-objective security game [4], A fundamental contribution of this paper is the insight that these different threat types lead to fundamentally different adversary models that cannot be folded into a single security game framework. MOPSS then is built upon these three adversary models. The second contribution deals with uncertain interruptions in the execution of patrol schedules. Existing systems, including TRUSTS [18], generated patrols that were often interrupted and left incomplete. This led to the use of MDPs for planning defender patrols in security games [9]. MOPSS exploits this idea to generate patrols for fare evasion. The third contribution is that MOPSS is the first system to generate patrols for counter-terrorism which accounts for joint coordinated activities between defender resources. This is achieved by incorporating the framework in [14] within both the SOLVER and the CT-Game in MOPSS. As a fourth contribution, MOPSS is the *first* system to deploy the *Opportunistic Security Game* model, where the adversary makes opportunistic decisions to commit crimes.

Finally, the fifth, and most important, contribution is the evaluation of MOPSS via real-world deployments. We ran three field experiments showing the benefits of game-theoretic scheduling in the real world. To the best of our knowledge, this evaluation constitutes the first evaluation of security games and, most importantly, the largest evalutation of *algorithmic* game theory, in the field. Existing literature on game theory in the field has focused on showing equilibrium concepts in the human and animal activities [12, 3]. Our work shares their enthusiasm of taking game theory to the field, but fundamentally focuses on algorithmic deployments and the impact of such algorithms. Most importantly, our work opens the door of applied research in security games to the realm of field evaluation. Given the maturity that such research has acquired in the recent years and its strong connection with real world patrolling problems, we argue that field deployment should become a key area for future research in security games.

Acknowledgements

The authors would like to acknowledge their appreciation for the collaboration of the Los Angeles Sheriffs Department (LASD), the Booz-Allen Hamilton Company and the Transportation Security Administrations (TSA) Intermodal Security Training and Exercise Program (I-STEP). The LASD provided us with exceptional support and preparation which allowed us to organize our experiments with great detail and accuracy. Booz-Allen managed TSA s I-STEP Los Angeles Mass Transit Full-Scale Exercise, thus allowing us to run experiments and collect data in very realistic and practical conditions. This research was supported by the United States Department of Homeland Security (DHS) through the National Center for Risk and Economic Analysis of Terrorism Events (CREATE) at the University of Southern California (USC) under Basic Ordering Agreement HSHQDC-10-A-BOA19, Task Order No. HST02-12-J-MLS151. 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, or the University of Southern California, or CREATE.

References

1. Martin J. Bland and Sally M. Kerry. Weighted comparison of means. *BMJ: British Medical Journal*, 316:125–129, 1998.
2. P. Brantingham and P. Brantingham. Criminality of place. *European Journal on Criminal Policy and Research*, 1995.
3. A. Brown, C. F. Camerer, and D. Lovallo. To review or not to review? limited strategic thinking at the movie box office. *American Economic Journal: Microeconomics*, (2), 2012.
4. M. Brown, B. An, C. Kiekintveld, F. Ordonez, and M. Tambe. An extended study on multi-objective security games. *JAAMAS*, 2013.

5. V. Conitzer. Computing game-theoretic solutions and applications to security. In *AAAI*, 2012.
6. N. Gatti. Game theoretical insights in strategic patrolling: Model and algorithm in normal form. In *ECAI*, 2008.
7. Lisa R. Goldberg, Alex N. Kercheval, and Kiseop Lee. t-statistics for weighted means in credit risk modeling. *Journal of Risk Finance*, 6(4):349–365, 2005.
8. R. Jaulmes, J. Pineau, and D. Precup. A formal framework for robot learning and control under model uncertainty. In *ICRA*, 2007.
9. A. X. Jiang, Z. Yin, C. Zhang, M. Tambe, and S. Kraus. Game-theoretic randomization for security patrolling with dynamic execution uncertainty. In *AAMAS*, 2013.
10. Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, pages 233–239, 2009.
11. S. Luber, Z. Yin, F. M. Delle Fave, A X. Jiang, M. Tambe, and J. P. Sullivan. Game-theoretic patrol strategies for transit systems: the trusts system and its mobile app (demonstration). In *AAMAS (Demonstrations Track)*, 2013.
12. R. Ostling, J. Wang, J. Tao-yi, E. Y. Chou, and C. F. Camerer. Testing game theory in the field: Swedish lupi lottery games. *American Economic Journal: Microeconomics*, 2011.
13. E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer. Protect: A deployed game theoretic system to protect the ports of the united states. In *AAMAS*, 2012.
14. E. Shieh, M. Jain, A. X. Jiang, and M. Tambe. Effciently solving joint activity based security games. In *IJCAI*, 2013.
15. M. B. Short, M. R D'Orsogna, V. B. Pasour, G. E. Tita, P. J. Brantingham, A. L. Bertozzi, and L. B. Chayes. A statistical model of criminal behavior. *Mathematical Models and Methods in Applied Sciences*, (supp01).
16. M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.
17. P. Varakantham, H. Chuin Lau, and Z. Yuan. Scalable randomized patrolling for securing rapid transit networks. In *IAAI*, 2013.
18. Z. Yin, A. Jiang, M. Johnson, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. Sullivan. Trusts: Scheduling randomized patrols for fare inspection in transit systems. In *IAAI*, 2012.
19. Zhengyu Yin and Milind Tambe. A unified method for handling discrete and continuous uncertainty in Bayesian stackelberg games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.
20. Chao Zhang, Albert Xin Jiang, Martin B. Short, Jeffrey P. Brantingham, and Milind Tambe. Towards a game theoretic approach for defendingagainst crime diffusion. In *AAMAS*, 2014.