Scaling-up Stackelberg Security Games Applications using Approximations

Arunesh Sinha[1], Aaron Schlenker[2], Donnabell Dmello[2], and Milind Tambe[2]

[1] University of Michigan, Ann Arbor, USA
[2] University of Southern California, Los Angeles, USA
arunesh@umich.edu, {aschlenk,ddmello,tambe}@usc.edu

Abstract. Stackelberg Security Games (SSGs) have been adopted widely for modeling adversarial interactions, wherein scalability of equilibrium computation is an important research problem. While prior research has made progress with regards to scalability, many real world problems cannot be solved satisfactorily yet as per current requirements; these include the deployed federal air marshals (FAMS) application and the threat screening (TSG) problem at airports. We initiate a principled study of approximations in zero-sum SSGs. Our contribution includes the following: (1) a unified model of SSGs called adversarial randomized allocation (ARA) games, (2) hardness of approximation for zero-sum ARA, as well as for the FAMS and TSG sub-problems, (3) an approximation framework for zero-sum ARA with instantiations for FAMS and TSG using intelligent heuristics, and (4) experiments demonstrating the significant 1000x improvement in runtime with an acceptable loss.

1 Introduction

The Stackelberg Security Game (SSG) model has been widely adopted in literature and in practice to model the defender-adversary interaction in various domains [20,11,6]. Over time SSGs have been used to model increasingly large and complex real world problems, hence an important research area within SSG research is the study of scalable Strong Stackelberg Equilibrium (SSE) computation algorithms, both theoretically and empirically. The scalability challenge has led to the development of a number of novel algorithmic techniques that compute the SSE of SSGs (see related work).

However, scalability continues to remain a pertinent challenge across many SSG applications. There are real world problems that even the best known approaches fail to scale up to, such as threat screening games (TSGs) and the Federal Air Marshals (FAMS) domain. The TSG model is used to allocate screening resources to passengers at airports and solves the problem for every hour (24 times a day). Yet, recent state-of-the-art approach for airport threat screening [4] scales only up to 110 flights per hour whereas 220 flights can depart per hour from the Atlanta Airport [9]. The FAMS problem is to allocate federal air marshals to US based flights in order to protect against hijacking attacks. Again, the best optimal solver for FAMS in literature [13] solves problems up to

200 flights (FAMS is a deployed application since 2011) and in our experiments a modified baseline approach scales up to 900 flights, whereas on average 3500 international flights depart from USA daily [22]. Further, the prior approaches are fundamentally limited by the hardness of computing the exact solution [23].

To overcome the computational hardness, and provide practical scalability we investigate approximation techniques for zero-sum SSGs. Towards that end, our *first contribution* is a *unified* model of SSGs that we name *adversarial randomized allocation* (ARA) games. ARA captures a large class of SSGs which we call linearizable SSGs (defined later) which includes TSGs and FAMS.

Our *second contribution* is a set of *hardness of approximation* results. For zero-sum ARAs, we show that the ARA equilibrium computation problem and the defender best response problem in the given ARA game have the same hardness of approximation property and in the worst case ARA is not approximable. Further, we show that subclasses of ARA problems given by FAMS and TSGs are hard to approximate to any sub-linear factor.

Our *third contribution* is a general *approximation framework* for finding the SSE of zero-sum ARAs. The approximation framework combines techniques from dependent sampling [21] with randomized rounding. However, the framework is not an out-of-the-box approach and requires specific insights for a successful application. As concrete instances, we instantiate the framework's for FAMS and TSGs family of problems by providing *intelligent heuristics*. We provide theoretical approximation bounds for both FAMS and TSGs.

Finally, as our *fourth contribution*, we demonstrate via experiments that we can solve FAMS problem up to 3500 flights and TSG problems up to 280 flights with runtime improvements up to 1000x over the current state of the art. Moreover, the loss for FAMS problems is less than 5% for 900 flights and the loss decreases with increasing flights. For TSGs, the loss is less than 1.5% across all cases upto the 110 flights that the state of the art could scale upto. Hence, our approach enables solving the real world FAMS and airport screening problem satisfactorily for a US wide deployment. All missing proofs are in the appendix.

2 Related Work

Two major approaches to scale up in SSGs include incremental strategy generation (ISG) and use of marginals. ISG uses a master slave decomposition, with the slave providing a defender or attacker best response [13]. All these approaches are limited by the NP hardness of finding an exact solution [15,23]. Use of marginals and directly sampling from marginals while faster suffers from the issue of non-implementable (invalid) marginal solutions [14,21]. Fixing the non-implementability again runs into complexity barriers [4]. Combination of marginals and ISG approaches has also been tried [3]. Our study stands in contrast to these approaches as we aim to approximate the SSE and not compute it exactly, providing a viable alternative to ISG and bypassing the non-implementability of marginals approach. Another line of work uses regret and endgame solving techniques [17,5] to approximately solve large scale sequential zero sum games. Our game does not have a sequential structure to exploit and the large action space precludes using a standard no-regret learning approach.

Our approximation is inspired by randomized rounding (RR) [18]. Previous work on RR with equality constraints address *only* equality constraints [10] or obtain an integral solution given an approximate fractional solution within a polyhedron with integral vertices [8]. However, our initial fractional solution may not lie within an integral polyhedron, and we have both equality and inequality constraints. Thus, we provide an approach that exploits the disjoint structure of equality constraints in TSGs and FAMS in order to use previous work on comb sampling [21] and then alters the output [2] to handle both equality and inequality constraints. Finally, our hardness of approximation results are the first such results for the classic FAMS and recent TSG problem.

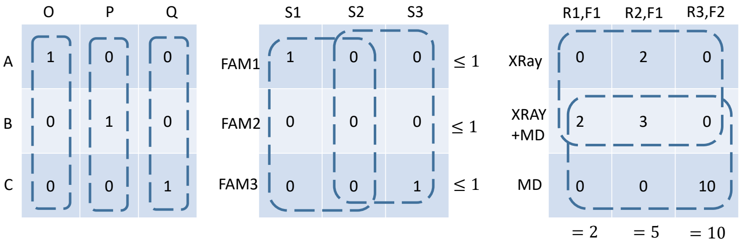
3 Model and Notation

We present a general abstract model of *adversarial randomized allocation* (ARA). ARA captures all *linearizable* SSGs, which is defined as those in which the probability c_t of defending a target t is linear in the defender mixed strategy; these include TSGs and FAMS. The ARA game model is a Stackelberg game model in which the defender moves first by committing to a randomized allocation and the adversary best responds. We start by presenting the defender's action space. There are k defense assets that need to be allocated to n objects to be defended. In this model, assets and objects are abstract entities and do not represent actual resources and targets in a security games. We will instantiate this abstract model with concrete examples of FAMS and TSG in the following sub-sections.

Defender's randomized allocation of resources: The allocation can be represented as a $k \times n$ matrix with the $(i, j)^{th}$ entry $x_{i,j}$ denoting the allocation of asset i to object j, and each $x_{i,j} \geq 0$. There is a set of *assignment constraints* on the entries of the matrix. Each assignment constraint is characterized by a set $S \subseteq \{1, \ldots, k\} \times \{1, \ldots, n\}$ of indexes of the matrix and the constraint is given by $n_s \leq \sum_{(i,j)\in S} x_{i,j} \leq N_S$, where n_s, N_S are non-negative integers. We will refer to each assignment constraint as S. Also for sake of brevity, we denote the vector of all the entries in the matrix as $\mathbf{x}$ and $\sum_{(i,j)\in S} x_{i,j}$ as $\mathbf{x}[S]$.

Pure strategies of the defender are *integral* allocations that respect the assignment constraints, i.e., *integral* $\mathbf{x}$'s such that $n_S \leq \mathbf{x}[S] \leq N_S$ for all assignment constraints S. See Figure 1 for an illustrative example of the assignment constraints and a valid pure strategy. Let the set of pure strategies be P and we will refer to a single pure strategy as $\mathbf{P}$. On the other hand, the space of marginal strategies MgS are those $\mathbf{x}$'s that satisfy the assignment constraints $n_S \leq \mathbf{x}[S] \leq N_S$ for all S; note that marginal strategies need not be integral.

Mixed strategies are probability distributions over pure strategies, e.g., probabilities $a_1, \ldots, a_m$ ($\sum_m a_m = 1$) over pure strategies $\mathbf{P}_1, \ldots, \mathbf{P}_m$. An expected (marginal) representation of a mixed strategy is $\mathbf{x} = \sum_m a_m \mathbf{P}_m$. Thus, the space of mixed strategies is exactly the *convex hull* of P, denoted as $conv(P)$. Typically, the space of marginal strategies is larger than $conv(P)$, i.e., $conv(P) \subset MgS$,



	O	P	Q
A	1	0	0
B	0	1	0
C	0	0	1

	S1	S2	S3	
FAM1	1	0	0	≤ 1
FAM2	0	0	0	≤ 1
FAM3	0	0	1	≤ 1

	R1,F1	R2,F1	R3,F2
XRay	0	2	0
XRAY +MD	2	3	0
MD	0	0	10
	= 2	= 5	= 10

Fig. 1. Three illustrations: (a) ARA with assets A,B,C and objects O,P,Q with 3 example assignment constraints (shown as dashed lines) with upper bound 1 on the columns. Shown also is an assignment that satisfies these constraints. (b) FAMS problem with 2 flights, 3 schedules and 3 FAMS. S1 and S2 share one flight and so do S2 and S3. The two assignment constraints (for the two flights) with upper bound 1 are represented by the two dashed lines. Additional constraints are present on each row, shown on the right of the matrix. The attacker chooses a flight to attack, hence the dashed lines also show the index set T of targets. A sample pure strategy fills the matrix. (c) TSG with the two assignment constraints (resource capacity) with upper bound 7 for XRay and 15 for Metal Detector (MD) represented by the two dashed lines. Additional equality constraints denoting the number of passengers in each passenger category (R,F) are present on each column, shown on the bottom of the matrix. A passenger category (column) is made from risk and flight. An adversary of type $R1$ can only choose the first column $R1, F1$ and $R2$ can choose from the other two columns. Thus, the index set T for targets corresponds to columns. A sample pure strategy fills the matrix.

hence every marginal strategies is not *implementable* as a mixed strategy. The conditions under which all marginal strategies are implementable (or not) has an easy interpretation in our model (see the implementability results in appendix).

Adversary's action: The presence of an adversary sets our model (and SSGs) apart from a randomized allocation problem [7] and makes ARA a game problem. The attacker's action is to choose a target to attack. In our abstract formulation a target t is given by a set $T \subset \{1, \ldots, k\} \times \{1, \ldots, n\}$ of indexes of the allocation matrix. In order to capture linearizable SSGs, the probability of successfully defending an attack on target t is $c_t = \sum_{i,j \in T} w_{i,j} x_{i,j}$, which is linear in $x_{i,j}$'s as the $w_{i,j}$'s are constants such that $w_{i,j} \leq 1/\max_{\mathbf{x} \in conv(P)} \sum_{i,j \in T} x_{i,j}$. The constraint on $w_{i,j}$ ensures that $c_t \leq 1$. We assume that the total number of targets is polynomial in the size of the allocation matrix. Then, as is standard for SSGs, the defender expected utility given $\mathbf{x}$ and t is

$$U_d(\mathbf{x}, t) = c_t U_s^t + (1 - c_t) U_u^t$$

where U_s^t (resp. U_u^t) is the defender's utility when target t is successfully (resp. unsuccessfully) defended. As we restrict ourselves to zero-sum games, the attacker's utility is negation of the above.[3]

[3] We remark that modeling-wise the extension to general-sum case, non-linearity in probabilities or exponentially many targets is straightforward; here we restrict the model as it suffices for the domains we consider.

The problem of Strong Stackelberg equilibrium computation can be stated as: $\max_{\mathbf{x},z,a_i} z$ subject to $z \leq U_d(\mathbf{x}, t)\ \forall t$ and $\mathbf{x} = \sum_{i:\mathbf{P}_i \in P} a_i \mathbf{P}_i$, where the last constraint represents $\mathbf{x} \in conv(P)$. Note that the inputs to the SSE problem are the assignment constraints, and the number of pure strategies can be exponential in this input. Thus, even though the above optimization is a LP, its size can be exponential in the input to the SSE computation problem. However, using the marginal strategies MgS instead of the mixed strategies $conv(P)$ results in a polynomial sized $marginalLP$:

$$\max_{\mathbf{x},z,c_t} z$$
$$\text{subject to } z \leq U(\mathbf{x}, t)\ \forall t \text{ and } n_s \leq \mathbf{x}[S] \leq N_S\ \ \forall S \text{ and } x_{i,j} \geq 0\ \ \forall i, j$$

But, as stated earlier $conv(P) \subset MgS$, and hence the solution to the optimization above may not be implementable as a valid mixed strategy. In our approximation approach we will solve the above $marginalLP$ as the first step obtaining marginal solution $\mathbf{x}^m$.

Bayesian Extension[4]: We also consider the following simple extension where we consider types of adversary $\theta \in \Theta$ and each adversary type θ attacks a set of targets $\mathcal{T}_\theta$ such that $\mathcal{T}_\theta \cap \mathcal{T}_{\theta'} = \phi$ for all $\theta, \theta' \in \Theta$. The adversary is of type θ with probability p_θ ($\sum_\theta p_\theta = 1$). Then, the exact SSE optimization can be written as: $\max_{\mathbf{x},z_\theta,a_i} p_\theta z_\theta$ subject to $z_\theta \leq U_d(\mathbf{x}, t)\ \forall \theta\ \forall t \in \mathcal{T}_\theta$ and $\mathbf{x} = \sum_{i:\mathbf{P}_i \in P} a_i \mathbf{P}_i$. A corresponding $marginalLP$ can be defined in the same way as for original ARA.

Implementability: Viewing the defender's action space as a randomized allocation provides an easy way to characterize non-implementability of mixed strategies across a wide range of SSGs, in contrast to prior work that have identified non-implementability for specific cases [15,16,4] . The details of this interpretation can be found in the appendix.

3.1 FAMS

We model zero-sum FAMS in the ARA model. The FAMS problem is to allocate federal air marshal (FAMS) to flights to and from US in order to prevent hijacking attacks. The allocation is constrained by the number of FAMS available and the fact that each FAMS must be scheduled on round trips that take them back to their home airport. Thus, the main technical complication arises from the presence of schedules. A schedule is a subset of flights that has to be defended together, e.g., flight f1 and f2 should be defended together as they form a round trip for the air marshal. Air marshals are allocated to schedules, no flight can have more than one air marshal and some schedules cannot be defended by some air marshals. The adversary attacks a flight.

Then, we capture the FAMS domain in the above model by mapping schedules in FAMS to objects (on columns) and air marshal in FAMS to assets (on

[4] Typically player types denotes different utilities but as Harsanyi [12] originally formulated, types capture any incomplete information including, as for our case, lack of information about adversary action space. The game is still zero-sum.

rows). See Figure 1 for an illustrative example. The assignment constraints include the constraint for each resource i: $\sum_j x_{i,j} \leq 1$, which states that every resource can be assigned at most once. If an air marshal i cannot be assigned to schedule j then add the constraint $x_{i,j} = 0$. A target t in the abstract model maps to a flight f in FAMS, and the set T are all the indexes for all schedules that include this flight: $\{(i,j) \mid \text{ flight f is in schedule } j\}$. The constraint that a flight cannot have more than one air marshal is captured by adding the *target allocation constraint* $\mathbf{x}[T] \leq 1$. The probability of defending a target (flight) is $c_t = \mathbf{x}[T]$, hence the weights $w_{i,j}$'s in ARA are all ones.

3.2 TSG

We model TSGs using the Bayesian formulation of ARA. The TSG problem is how to allocate screening resources to screenees in order to screen optimally, which we elaborate in the context of airline passenger screening. In TSGs, different TSG resources such as X-Rays and Metal Detector act in teams to inspect an airline passenger. The possible teams are given. Passengers are further grouped into passenger categories with a given N_c number of passengers in each category c. The allocation is of resource teams to passenger categories. There are *resource capacity constraints* for each resource usage (not on teams but on each resource). Further, all passengers need to be screened. Each resource team i has an effectiveness $E_i < 1$ of catching the adversary. Observe that, unlike SSGs, the allocation in TSGs is not just binary $\{0, 1\}$ but any positive integer within the constraints. The passenger category c is a tuple of risk level and flight (r, f); the adversary's action is to choose the flight f but he is probabilistically assigned his risk level.

Then, we capture the TSG domain in the above abstract model by mapping passenger categories in TSGs to objects (on columns) and resource teams in TSGs to assets (on rows). See Figure 1 for an illustrative example. The capacity constraint for each resource r is captured by specifying the constraint $\mathbf{x}[S] \leq N_S$ which contains all indexes of teams that are formed using the given resource r: $S = \{(i,j) \mid \text{ team } i \text{ is formed using resource } r\}$ with N_S equal to the resource capacity bound for resource r. For every passenger category j, the constraint $\sum_i x_{i,j} = N_j$ enforces that all passengers are screened. A target t in TSG is simply a passenger category j, thus, the set T is $\{(i,j) \mid j \text{ is given passenger category}\}$. The probability of detecting an adversary in category j is given by $\sum_{(i,j)\in T} E_i x_{i,j}/N_j$, hence the weights $w_{i,j}$ are E_i/N_j; since $E_i < 1$ it is easy to check that $\sum_{(i,j)\in T} w_{i,j} x_{i,j} \leq 1$ for any T. The adversary type is the risk level r, and each type r of adversary can choose a flight f, thus, choosing a target which is the passenger category (r, f). The probability of the adversary having a particular risk level is given.

4 Computation Complexity

In this section, we explore the *hardness of approximation* for ARAs, FAMS and TSGs. In prior work on computation complexity of SSGs, researchers [23] have focused on hardness of exact computation providing general results relating the hardness of defender best response (DBR) problem (defined below) to the hardness of exact SSE computation. In contrast, we relate the hardness of approximation of the DBR problem to hardness of approximation of ARAs. We also prove that special cases of ARA such as FAMS and TSGs are also hard to approximate.

First, we formally state the equilibrium computation problem in adversarial randomized allocation: given the assets, objects and assignment constraints of an adversarial randomized allocation problem as input, output the SSE utility and a set of pure strategies $P_1, \ldots, P_m$ and probabilities $p_1, \ldots, p_m$ that represents the SSE mixed strategy. We restrict m to be polynomial in the input size. This is natural, since a polynomial time algorithm cannot produce an exponential size output. Also, it is well known [23] that the size of the support set of any mixed strategy need not be more than $kn + 1$.

Next, as defined in prior literature [23], we state the DBR problem which aids in understanding the results. The DBR problem can be interpreted as the defender's best response to a given mixed strategy of the adversary. The DBR problem also shows up naturally as the slave problem in column generation based approaches to SSGs.

Definition 1. *The DBR problem is* $\max_{\mathbf{x} \in P} \mathbf{d} \cdot \mathbf{x}$ *where* $\mathbf{d}$ *is a vector of positive constants. DBR is a combinatorial problem that takes the assignment constraints as inputs, and not the set of pure strategies* P*.*

Next, we state the standard definition of approximation

Definition 2. *An algorithm for a maximization problem is* r*-approximate if it provides a feasible solution with value at least* OPT/r*, where* OPT *is the maximum.*

Note that lower r means better approximation. Depending on the best r possible, optimization problems are classified into various approximation complexity classes with increasing hardness of approximation in the following order PTAS, APX, log-APX, and poly-APX. We extensively use the well-known approximation preserving AP reduction between optimization problems for our results. AP reduction is analogous to reductions used for NP hardness but must also account for mapping of approximation ratios (and thus preserve hardness of approximation). AP reduction is among the strongest of all approximation preserving reductions as it preserves membership in most of the known approximation complexity classes. We do not delve into the formal definition of complexity classes or AP reduction here due to lack of space and these concepts being standard [1]. Our first result shows that the ARA's approximation complexity is same as that of the DBR problem and in the worst case cannot be approximated.

Theorem 1. *The following hardness of approximation hold for ARA problems: (1) ARA problems cannot be approximated by any bounded factor in poly time, unless* $P = NP$*; (2) if the DBR problem for given ARA problem lies in some given approximation class (APX, log-APX, etc.), then so does the ARA problem.*

Proof (Proof Sketch). The first result works by constructing a ARA from a NP hard unweighted ($\mathbf{d} = 1$) DBR problem such that the feasibility of the constructed ARA solves the DBR problem, thereby ruling out any approximation. Such unweighted DBR problems exist (e.g., for FAMS). The second part of the proof works by constructing an ARA problem with one target and showing that the solution yields an approximate value for a relaxed DBR with $\mathbf{x} \in conv(P)$. Moreover, this solution is an expectation over integral points (pure strategies), thus, at least one integral point in the support set output by ARA also provides an approximation for the corresponding combinatorial DBR.

As the above complexity result is a worst case analysis, one may wonder whether the above result holds for sub-classes of ARA problems. We show that strong versions of inapproximatibility also holds for FAMS and TSGs.

Theorem 2. *TSGs cannot be approximated to $O(n^{1-\epsilon})$ factor for any ϵ in poly time, unless P=NP.*

Proof (Proof Sketch). Using AP reduction from max independent set (MIS), the proof for TSG follows from an observation that a special case of the TSG problem is the MIS problem itself. MIS is known to be hard to approximate to any factor better than $n^{1-\epsilon}$ for any ϵ, unless P=NP.

Theorem 3. *FAMS problems cannot be approximated to $O(n^{1-\epsilon})$ factor for any ϵ in poly time, unless P=NP.*

Proof. We provide an AP reduction from max independent set (MIS). Given a MIS problem with vertices V and edges E construct the following FAMS problems, one for each k. Use $2n-k$ resources. All resources can be assigned to any schedule. Construct schedules $s_1, \ldots, s_n$ corresponding to the vertices $v_1, \ldots, v_n$. Construct target t_e corresponding to every edge $e = (u, v)$ such that $t_e \in s_u$ and $t_e \in s_v$. All t_e's have the same value for being defended or undefended and that value is $n+2$; thus, these targets do not need to covered but impose the constraint that s_u and s_v cannot be simultaneously defended. Thus, it is clear that any allocation of resources to $s_1, \ldots, s_n$ corresponds to an independent set. Next, consider additional $2n$ *valuable* targets and expand the set of targets of the schedules such that $t_i, t_{i+1} \in s_i$. Further, add $2n$ more singleton schedules $s_{n+1}, \ldots, s_{3n}$ with $t_i \in s_{n+i}$. All additional targets $t_1, \ldots, t_{2n}$ provide value k when defended and $k - 2n$ otherwise. Thus, the expected utility of defending a valuable target t given coverage c_t is $c_t(k) + (1 - c_t)(k - 2n) = 2n * c_t + k - 2n$.

For the given MIS problem, let the solution be k^*. Observe that for FAMS problems with resources $2n-k$ where $k \leq k^*$, all valuable (additional) targets can be covered by covering k^* schedules with $2k^*$ targets in $s_1, \ldots, s_n$ and using the remaining $\geq 2n - 2k^*$ resources to cover the remaining $2n - 2k^*$ valuable targets (via singleton schedules). This provides utility of k for the SSE. In particular, the utility with $2n - k^*$ resources is k^*. Also note that for every problem, there is always a trivial allocation of $2n - k$ resources to the $2n$ singleton schedules such that coverage of each target is $1 - k/2n$. This is deducible as the allocation to singleton schedules is unconstrained and can be implemented in poly time by Birkhoff-von Neumann result as provided in [15]. This trivial allocation provides an utility of 0.

Next, assume we have a poly time algorithm to approximately compute the SSE with approx factor r ($r > 1$). We will run this poly time algorithm with resources 2 to $2n - 1$ which is again a poly time overall, and also the overall output size is poly. We construct an approximation for the MIS problem.

Our construction relies on the following claim (proved in the next paragraph): given approximation factor r for the case with $2n - k^*$ resources then one of the pure strategy output for this case will have at least $k^* - l_{\min}$ schedules among $s_1, \ldots, s_n$ covered where $l_{\min} = \lfloor \text{argmin}_l \frac{k^*}{k*-l} \geq r \rfloor$. Note that by definition of $l_{\min}$, $\frac{k^*}{k*-(l_{\min}+1)} > r$ and $\frac{k^*}{k*-l_{\min}} \leq r$. As k^* is the max size of independent sets we obtain an approximation ratio r' for the max independent set problem such that $r' = \frac{k^*}{k^*-l_{\min}} \leq r$. Thus, we obtain an approximation r' for MIS as good as r approximation for the SSE. Thus, we have an AP reduction.

To prove the claim in last paragraph consider the contra-positive: suppose all pure strategies output cover at most $k^* - l_{\min} - 1$ schedules among $s_1, \ldots, s_n$, then in every pure strategy at least $l_{\min}+1$ valuable targets are not covered (since 2 valuable targets are covered for the $k^* - l_{\min} - 1$ schedules and rest of resources can cover only 1 valuable target). Then the coverage of the least covered target in the mixed strategy formed using such pure strategies is $\leq 1 - (l_{\min} + 1)/2n$ (this can be seen as sum of coverage of valuable targets must be at least $2n - l_{\min} - 1$, since that is true for every pure strategy). The utility for this least covered target is $\leq k^* - l_{\min} - 1$. The overall utility has to be lower than utility for any target, hence the utility is $\leq k^* - l_{\min} - 1$. The optimal utility is k^*. Thus, by definition of approximation ratio r we must have $k^* - l_{\min} - 1 \geq k^*/r$ or re-arranging $\frac{k^*}{k^*-l_{\min}-1} \leq r$ but by definition of $l_{\min}$ we must have $\frac{k^*}{k*-l_{\min}-1} > r$ hence a contradiction.

5 Approximation approach

Our approach to approximation first solves the *marginalLP*, which is quite fast in practice (see experiments) and provides an upper bound to the true value of the game. Then, we sample from the marginal solution, but unlike previous work [21], we alter the sampled value to ensure that the final pure strategy output is valid. We describe an abstract sampling and alteration approach for ARA in this part, which we instantiate for FAMS and TSGs in the subsequent sub-sections. Recall that a constraint is given by an index set S and the constraint is an equality if $n_S = N_S$. For our abstract approach we restrict our attention to ARAs with *partitioned equality assignment constraints*, which means the index set S for all equality constraints partitions the index set $\{1, \ldots, k\} \times \{1, \ldots, n\}$ of the allocation matrix. Further, for inequality constraints we assume $n_S = 0$. Call these problems as PE0-ARA; this class still includes FAMS and TSGs. For FAMS, which does not have equality constraints, we use dummy schedules s_i to get partitioned equalities $\sum_j x_{i,j} + s_i = 1$; $s_i = 1$ denotes that resource i is unallocated. Our abstract approximation approach for PE0-ARA is presented in Algorithm 1.

Algorithm 1: Abstract Approximation

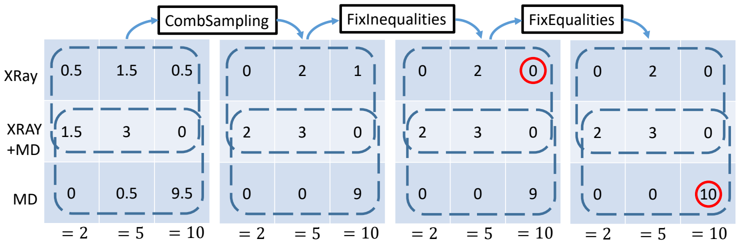
```
  Input: $\mathbf{x}^m$: the marginal solution
1 forall $S \in EqualityConstraints$ do
2     $\mathbf{x} \leftarrow CombSample(\mathbf{x}^m, S)$
3 $\mathbf{x} \leftarrow FixViolatedInequalityConstraints(\mathbf{x})$
4 $\mathbf{x} \leftarrow FixEqualityConstraints(\mathbf{x})$
```

The Algorithm takes as input the marginal solution $\mathbf{x}^m$ from $marginalLP$ and produces a pure strategy. The first for loop (line 1-2) performs comb sampling for each equality constraint S to produce integral values for the variables involved in S. Comb sampling was introduced in an earlier paper [21]; it provides the guarantee that $x^m_{i,j}$ is rounded up or down for all $(i, j) \in S$, the sample $x_{i,j}$ has expected value $E(x_{i,j}) = x^m_{i,j}$ for all $(i, j) \in S$ and equality S is still satisfied after the sampling. See Figure 2 for an example. Briefly, comb sampling works by creating Z buckets of length one each, where $Z = \sum_{(i,j)\in S}\{x^m_{i,j}\}$, where $\{.\}$ denotes fractional part. Each of the $\{x^m_{i,j}\}$ length fraction is packed into the bucket (in any order and some of the $\{x^m_{i,j}\}$ fraction may have to be split into two buckets), then a number between $[0, 1]$ is sampled randomly, say z, and for each bucket a mark is put at length z. Finally, the (i, j) whose $\{x^m_{i,j}\}$ fraction lies on the marker z for each bucket is chosen to be rounded up, and all other $x^m_{i,j}$ are rounded down.

Observe that in expectation the output of comb sampling matches the marginal solution, thus, providing the same expected utility as the marginal solution. Recall that this expected utility is an upper bound on the optimal utility. However, the samples from comb sampling may not be valid pure strategies. Thus, in case the output of comb sampling is not already valid, the two abstract methods in line 3 and 4 modify the sample strategy by first decreasing the integral values to satisfy the violated inequalities and then increasing the integral values to satisfy the equalities. Such modification of the sampled strategy to obtain a valid strategy is guided by the *principle* that the change in defender utility between the sampled and the resultant valid strategy should be small, which ensures that change in expected utility from the marginal solution due to the modification is small. As the expected utility of the marginal solution is an upper bound on the optimal expected utility this marginal expected utility guided modification leads the output expected utility to be close to the optimal utility. Thus, the two methods on line 3 and 4 need to be instantiated with carefully designed heuristics that aim to implement the principle of marginal expected utility guided modification. Below, we show the instantiation for the TSG and FAMS family of problems. A sample execution for TSGs is shown in Figure 2.

5.1 TSG

The heuristics for TSG are guided by three *observations*: (1) more effective resources are more constrained in their usage, (2) changing allocation for passenger



XRay	0.5	1.5	0.5	0	2	1	0	2	0	0	2	0
XRAY +MD	1.5	3	0	2	3	0	2	3	0	2	3	0
MD	0	0.5	9.5	0	0	9	0	0	9	0	0	10
	= 2	= 5	= 10	= 2	= 5	= 10	= 2	= 5	= 10	= 2	= 5	= 10

Fig. 2. (Left to right) Sample execution for the TSG example from Figure 1. Recall that the resource capacity is 7 for XRay and 15 for MD. The marginal solution is the left matrix which after CombSampling on each column becomes integral, e.g., 0.5 in the left column is rounded down to 0 and 1.5 rounded up to 2. Note that the CombSampling output satisfies all equalities, but exceeds the resource capacity 7 for X-ray. Next, allocation values are lowered (shown as red circle) to satisfy the X-Ray capacity but the equality constraint on third column is violated. Next, allocation values are increased (again red circle) to fix the equality which produces a valid pure strategy.

categories with higher number of passengers changes the probability of detection of adversary by a smaller amount than changing allocation for category with fewer passengers and (3) higher risk passenger categories typically have lower number of passengers.

Algorithm 2 shows the heuristic for TSG. Recall that for TSGs the inequalities are resource capacity constraints. Thus, for fixing violated inequalities we need to decrease allocation which decreases utility; we wish to keep the utility decrease small as that ensures that the expected utility does not move much further away from the upper bound marginal expected utility. Our approach for such decrease in allocation has the following steps: (a) [Line 1] prioritize fixing inequality of most violated (negative slack) resources first and (b) [Lines 2-10] for each such inequality we attempt to lower allocation for passenger category with higher number of passengers. In light of the observations for TSG above this approach aims to keep the change in expected utility small. Specifically, observation 1 makes it likely that constraints for more effective resources are fixed in step a above. Observation 3 suggests that the changes in step b happens for lower risk passengers. Thus, step (a) aims to keep the allocation of effective resources for high risk passengers unchanged. This keeps the utility change small as changing allocation for high risk passengers can change utility by a large amount. Next, by observation 2, step (b) aims to minimize the change in probability of detecting the adversary by a low amount so that expected utility change in small. For example in Figure 2, the inequality fix reduces the allocation for the third passenger category (column) which also has the highest number of passengers (15). Also, observe that within each passenger category in step (b) we reduce those variables that participate in most resource capacity inequality constraints (Line 7) just to ensure that more constraints are fixed with fewer changes.

Algorithm 2: TSG Pure Strategy Generation

```
   Input: $\mathbf{x}$ from Comb Sampling
1  $OrderedInequalityConstraints = Sort(InequalityConstraints, \mathbf{x})$ ascending
     by slack
2  forall $R \in$ OrderedInequalityConstraints do
3      $\mathbf{X}_j^R \leftarrow$ variables corresponding to passenger category $j$ in $R$ (thus, $\mathbf{X}_j^R$ is a
         set of variables)
4      $\mathbf{X}^R \leftarrow Sort(\{\mathbf{X}_j^R\}_{j=1,..})$ descending by no. of passengers in category $j$.
5      forall $\mathbf{X}_j^R$ in $\mathbf{X}^R$ do
6          while any variable in $\mathbf{X}_j^R$ is $> 0$ AND $R$ is violated do
7              $x_{i,j}$ = Positive variable participating in the most inequality
                 constraints among $\mathbf{X}_j^R$
8              $x_{i,j} = x_{i,j} - 1$
9          if $R$ is satisfied then
10             break
11 $OrderedEqualityConstraints = Sort(EqualityConstraints, \mathbf{x})$ ascending by
     no. of passengers in the category corresponding to each equality constraint
12 forall $C \in$ OrderedEqualityConstraints do
13     $\mathbf{X}_j \leftarrow$ variables in $C$ ($C$ corresponds to category $j$)
14     $\mathbf{X}_j^C \leftarrow Sort(\mathbf{X}_j)$ ascending by the min slack in the resource constraint of all
         resources that can inspect $x_{i,j} \in X_j$
15     forall component $x_{i,j}$ in $\mathbf{X}_j^C$ do
16         while $x_{i,j} \neq 0$ and $C$ is violated do
17             $x_{i,j} = x_{i,j} + 1$
18         if $C$ is satisfied then
19             break
```

Next, the equalities in TSGs are the constraints for every passenger category. For fixing equalities we need to increase allocation which increases utility; we wish to keep this utility increase high as it brings the expected utility closer to the upper bound marginal expected utility. Here we aim to do so by (a) [Line 11] prioritizing increase of allocation for categories with fewer people and (b) [Line 12-19] increasing allocation of those resources that have least slack in their resource capacity constraint (low slack means more utilized which could mean higher effectiveness). By Observation 1 low slack means that resource could be more effective and by Observation 2 fewer people means higher risk passengers. This ensures that higher risk passengers are screened more using more effective resources thereby raising the utility maximally. For example in Figure 2, the equality for the third column is fixed by using the only available resource MD.

Recall that, unlike FAMS, the allocation for TSGs are non-binary. This offers an advantage for TSGs with respect to approximation, as small fractional changes do not change the overall allocation by much (0.5 to 1 is a 50% change, but 4.5 to 5 is less than 10%). Thus, we assume here that the changes due to

Algorithm 3: FAMS Pure Strategy Generation

```
   Input: $\mathbf{x}$ from Comb Sampling
1  $\mathbf{X}_j \leftarrow$ variables corresponding to schedule $j$
2  $OrderedInequalityConstraints = Sort(InequalityConstraints)$ ascending by slack
3  forall $T \in OrderedInequalityConstraints$ do
4      $J \leftarrow$ schedules that $T$ belongs to
5      $J \leftarrow Sort(J)$ descending by the number of violated target allocation (inequality) constraints for schedule $j$
6      $\mathbf{X}_J \leftarrow \cup_{j \in J} \mathbf{X}_j$
7      while any variable in $\mathbf{X}_J$ is $> 0$ AND $T$ is violated do
8          $j \leftarrow 1stScheduleNoSatisifedTarget(J)$
9          if $j$ is -1 then
10             $j \leftarrow$ choose $j$ randomly from $J$
11         $x_{i,j} \leftarrow$ any variable from $\mathbf{X}_j$
12         $x_{i,j} = x_{i,j} - 1$
```

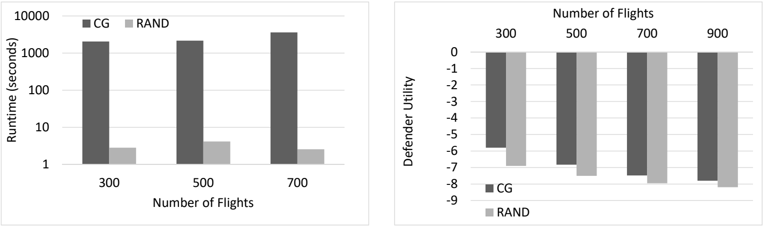
Algorithm 1 do not reduce the probability of detecting an adversary in any passenger category (from the marginal solution) by more than $1/c$ factor, where $c > 1$ is a constant. This restriction is realistic as it is very unlikely that any passenger category will have few passengers and we aim to change the allocation for passenger categories with a higher number of passengers. Hence we prove

Theorem 4. *Assume that Algorithm 2 successfully outputs a pure strategy and the change in allocation from the marginal strategy does not change the probability of detecting an adversary by more than $1/c$ factor. Then, the approximation approach above with the heuristic provides a c-approximation for TSGs.*

As a remark, the above result does not violate the inapproximatability of TSGs since the above holds for a restricted set of TSG problems. Also, the approximation for TSGs may sometimes fail to yield a valid pure strategy as satisfying the equalities may become impossible after using certain sequences of decreasing allocation. In our experiments we observe that the failure of obtaining a pure strategy for TSG after Algorithm 2 is rare and easily handled by repeating the Algorithm 1 (sampling and adjusting runs in milli-secs).

5.2 FAMS

Recall that for FAMS the inequalities are the target allocation constraints: $\mathbf{x}[T] \leq 1$ and fixing violations for these involves decreasing allocation. Algorithm 3 shows the heuristic for TSG. Our heuristic is simple: we fix the most violated constraints first (Line 2), the variables $x_{i,j}$ are set to zero (i.e., decreased) starting from those schedules j that contain the most number of targets for which target allocation constraint is violated (Line 5) and do not contain any target for which the target allocation constraint is satisfied (Line 8). We are



(a) Runtime (log-scale time)

(b) Solution quality

Fig. 3. CG and RAND Comparison

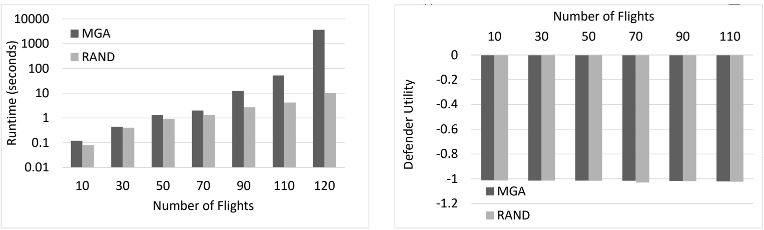
guaranteed to find a decrease in allocation that satisfies the constraint for T without changing the allocation for targets that already satisfy constraints in the cases when the target T with violated constraint (1) belongs to a schedule j that exclusively contains that target T ($x_{i,j}$ can be decreased without affecting any other constraint) or (2) T belongs to only one schedule (other targets in this schedule will violate their constraints). This approach ensures that we only work to fix the violated constraints and cause a minimal change in utility by leaving the satisfied constraints undisturbed. However, if in fixing a violated target allocation constraint for T it becomes necessary to reduce allocation for another already satisfied target constraint, then sample uniformly from the ≥ 2 schedules that T belongs to in order to choose the $x_{i,j}$ allocation to reduce (Line 10) till all inequality constraints are satisfied.

Then, we do nothing to fix equality constraints since we have only decreased $x_{i,j}$ and if any equality $\sum_j x_{i,j} + s_i = 1$ is not satisfied we can always set the dummy s_i to be one. Also, observe that since we only always decrease allocations, we always find a pure strategy for any sample from Algorithm 3 (unlike TSGs). We prove:

Theorem 5. *Let C_t be the number of targets that share a schedule with any target t, and $C = \max_t C_t$. The approximation approach above with the heuristic provides a $2^C k$-approximation for FAMS.*

6 Experimental Results

Our experimental results reveal the average case loss of our approximation. **Baseline**: Our set of experiments provide a comprehensive analysis of our approximation approach, which we name RAND. We compare RAND to the best know solver for zero sum TSGs called MGA; MGA [4] has been previously shown to outperform column generation based approaches by a large margin. A more recent work [19], called GATE, approximates general sum TSGs using MGA in a branch and bound tree. However, this work suffers from loss of more than 11% for problems that are zero-sum (Figure 7 in that paper) with runtime in 1000s of seconds compared to our loss of less than 1.5%. Moreover, the potential



(a) Runtime (log-scale time)

(b) Solution quality

Fig. 4. MGA and RAND Comparison

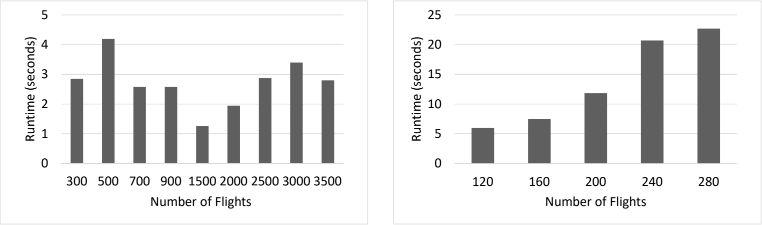
TSG application by Transport Security Administration (TSA) uses the zero-sum game version with MGA as the solver, which we confirmed through private communication with the authors of both these papers.

For the FAMS problem the best known solver in literature for the general sum case is ASPEN [13], which is a column generation based branch and price approach. Through private communication with the company (Avata Intelligence) managing the FAMS software, we know the FAMS problem is solved as a zero-sum problem for scalability using column generation. Even then the approach takes hours and is cut off without running to completion. On our end, for the zero-sum case we implemented a column generation (CG) solver for FAMS, since branch and price is an overkill for the zero sum case that we study.

All experimental results are averages over 30 randomly generated game instances. All game instances fix U_s^t to -1 and randomly select integral U_u^t between -2 and -10. The utility for RAND is computed by sampling 1000 pure strategies and taking their average as an estimate of the defender mixed strategy. All experiments were run with a Xeon 2.6 GHz processor and 4GB RAM.

For FAMS, we vary the number of flights, keeping the number of resources fixed at 10 and number of schedules fixed at 1000 and 5 targets/schedule. The runtime *in log scale* is shown in Figure 3. CG hits the 3600 seconds cut-off for 700 flights and the run time for RAND is much lower at only a few seconds. Next, we report the solution quality for RAND by comparing with the solution using CG. It can be seen that the solution gets better with increasing flights starting from 19% loss at 300 flights to 5% loss at 900 flights. An important point to note is that the approximation loss decreases with increasing number of flights, Thus, at 3500 flights (the number desired) we expect the loss percentage to be much lower, which we are unable to compare with CG as CG does not scale up. The numbers show that we obtain large speed-ups up to factor of 1000x and are still able to extract 95% utility for 900 flights beyond which CG does not scale.

For TSGs, we used six passenger risk levels, eight screening resource types and 20 screening team types. We vary the number of fights and we also randomly sample the team structure (how teams are formed from resources) for each of the 30 runs. The results in Figure 4 show runtime (in log scale) and defender utility values varying with number of flights (on x-axis). As can be seen, MGA



(a) Runtime FAMS (b) Runtime TSG

Fig. 5. Scalability of RAND

only scales up to 110 flights before hitting the cut-off of 3600 seconds, while RAND takes only 10 seconds for 110 flights. Also, the solution quality loss for RAND has a maximum averaged loss of 1.49%. Thus, we obtain at-least 360X speed-ups with very minor loss. We performed an additional experiment to show that the choices made by our heuristics are important. We change the heuristics in Algorithm 2 line 3 to sort ascending instead of descending and the modified RAND suffered 35% more loss over RAND for 110 flights. A figure showing the same with different number of flights is in the appendix.

Next, we test the scalability of RAND for FAMS and TSG, shown in Figure 5. As can be seen, the runtime for RAND is low even with the highest number of flights we tested: 280 for TSG and 3500 for FAMS. The maximum runtime for FAMS was under 5 seconds; the maximum runtime for TSG was under 25 secs.

7 Conclusion

We studied approximations in zero-sum SSGs both theoretically and practically. We provided approximation techniques to solve large scale zero-sum SSGs, which enables the application of already deployed application (FAMS) or applications under test (airport screening) at a national scale in USA. In fact, the number of international flights from USA was 2000 in 2010 [13] which has increased to 3500 [22] revealing the ever increasing trend. Our approach not only provide an avenue to solve the FAMS and airport screening problem for current problem sizes but is capable of scaling up to larger numbers in future.

References

1. Ausiello, G., Protasi, M., Marchetti-Spaccamela, A., Gambosi, G., Crescenzi, P., Kann, V.: Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties. Springer Science & Business Media (1999)
2. Bansal, N., Korula, N., Nagarajan, V., Srinivasan, A.: Solving packing integer programs via randomized rounding with alterations. Theory of Computing **8**(1), 533–565 (2012)

3. Bošanský, B., Jiang, A.X., Tambe, M., Kiekintveld, C.: Combining compact representation and incremental generation in large games with sequential strategies. In: AAAI (2015)
4. Brown, M., Sinha, A., Schlenker, A., Tambe, M.: One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats. In: AAAI (2016)
5. Brown, N., Sandholm, T.: Safe and nested subgame solving for imperfect-information games. In: NIPS. pp. 689–699 (2017)
6. Bucarey, V., Casorrán, C., Figueroa, Ó., Rosas, K., Navarrete, H., Ordóñez, F.: Building real stackelberg security games for border patrols. In: Decision and Game Theory for Security (2017)
7. Budish, E., Che, Y.K., Kojima, F., Milgrom, P.: Designing random allocation mechanisms: Theory and applications. The American Economic Review **103**(2), 585–623 (2013)
8. Chekuri, C., Vondrák, J., Zenklusen, R.: Dependent randomized rounding for matroid polytopes and applications. arXiv preprint arXiv:0909.4348 (2009)
9. FAA: Airport capacity profiles. https://goo.gl/YZvzsU (2014), accessed: 2018-05-15
10. Gandhi, R., Khuller, S., Parthasarathy, S., Srinivasan, A.: Dependent rounding and its applications to approximation algorithms. Journal of the ACM (JACM) **53**(3), 324–360 (2006)
11. Guo, Q., An, B., Vorobeychik, Y., Tran-Thanh, L., Gan, J., Miao, C.: Coalitional security games. In: AAMAS (2016)
12. Harsanyi, J.: Games with incomplete information played by bayesian players, i-iii part i. the basic model. Management Science **14**(3) (1967)
13. Jain, M., Kardeş, E., Kiekintveld, C., Tambe, M., Ordóñez, F.: Security games with arbitrary schedules: a branch and price approach. In: AAAI. pp. 792–797 (2010)
14. Kiekintveld, C., Jain, M., Tsai, J., Pita, J., Ordóñez, F., Tambe, M.: Computing optimal randomized resource allocations for massive security games. In: AAMAS (2009)
15. Korzhyk, D., Conitzer, V., Parr, R.: Complexity of computing optimal Stackelberg strategies in security resource allocation games. In: AAAI (2010)
16. Letchford, J., Conitzer, V.: Solving security games on graphs via marginal probabilities. In: AAAI (2013)
17. Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M., Bowling, M.: Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. Science (2017)
18. Raghavan, P., Thompson, C.D.: Randomized rounding: a technique for provably good algorithms and algorithmic proofs. Combinatorica **7**(4), 365–374 (1987)
19. Schlenker, A., Brown, M., Sinha, A., Tambe, M., Mehta, R.: Get me to my gate on time: Efficiently solving general-sum bayesian threat screening games. In: ECAI. pp. 1476–1484 (2016)
20. Tambe, M.: Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned. Cambridge University Press, New York, NY, USA (2011)
21. Tsai, J., Yin, Z., Kwak, J.y., Kempe, D., Kiekintveld, C., Tambe, M.: Urban security: Game-theoretic resource allocation in networked physical domains. In: AAAI (2010)
22. USDOT: Bureau of transportation statistics. https://goo.gl/Goz84L (2016), accessed: 2018-05-15
23. Xu, H.: The mysteries of security games: Equilibrium computation becomes combinatorial algorithm design. In: ACM-EC (2016)

Appendix

Implementability: Viewing SSGs as ARAs provides an easy way of determining implementability using results from randomized allocation [7]. First, we define *bi-hierarchical assignment constraints* as those that can be partitioned into two sets H_1, H_2 such that two constraints S, S' in the same partition (H_1 or H_2) it is the case that either $S \subseteq S'$ or $S' \subseteq S$ or $S \cap S' = \phi$. Further, as defined in [7], *canonical assignment constraints* are those that impose constraints on all rows and columns of the matrix. We obtain the following result

Proposition 1. *All marginal strategies are implementable, or more formally $conv(P) = MgS$, if the assignment constraints are bi-hierarchical. Given canonical assignment constraints, if all marginal strategies are implementable then the assignment constraints are bi-hierarchical.*

As Figure 1 reveals, both FAMS and TSG have non-implementable marginals due to overlapping constraints. The proof of the proposition is straightforward applications of Thm 1 and Thm 2 in Budish et al. [7].

Modified Heuristic is Bad: The modified RAND approach is compared to RAND in Figure 6. It can be seen that the loss increases a lot with almost 35% loss over RAND for 110 flights.

Proof of Theorem 1: First we define some problems related to the DB problem.

- DBR is the problem $\max_{\mathbf{x} \in P} \mathbf{d} \cdot \mathbf{x}$ where $\mathbf{d}$ is a vector of positive constants. DBR is a combinatorial problem.
- The continuous version of DBR is DBR-C: $\max_{\mathbf{x} \in conv(P)} \mathbf{d} \cdot \mathbf{x}$.
- The unweighted version of the DBR is DBR-U: $\max_{\mathbf{x} \in P} \mathbf{1} \cdot \mathbf{x}$.

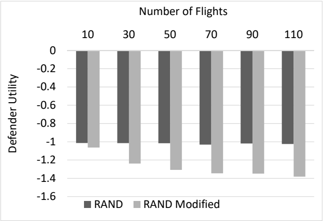


Fig. 6. RAND modified heuristic comparison

Proof. For the first part, given a NP hard DBR-U instance (for the decision version of DBR-U), we construct an ARA instance such that the feasibility problem for that ARA instance solves the hard DBR-U decision problem. Thus, as the feasibility is NP Hard, there exists no approximation. First, since the ARA problem is so general there exists DBR-U problems that are NP Hard. For example, the DBR-U problem for FAMS has been shown to be NP Hard [23]. Given the hard DBR-U problem, form an ARA problem with by adding the constraint $\mathbf{1} \cdot \mathbf{x} = k$. Also, let there be only one target t in the problem, so that the objective becomes $U(\mathbf{x}, t)$ instead of z and all constraints in the optimization are just the marginal space constraints and $\mathbf{1} \cdot \mathbf{x} = k$. Now, the existence of any solution of the optimization gives a feasible point $\mathbf{x} = \sum_m a_m \mathbf{P_m}$, where $\mathbf{P_m} \in P$ is integral. Also, it must be that $\mathbf{1} \cdot \mathbf{P_j} \geq \mathbf{1} \cdot \mathbf{x} = k$ for some j. Then, $\mathbf{P_j}$ is a solution to the decision version of the DBR-U problem, i.e., does there exist a solution of the DBR-U optimization problem with value $\geq k$? Thus, since finding the existence of any solution for ARA is NP Hard, thus, no approximation exists in poly time.

For the second part, we present a AP approximation preserving reduction (with problem mapping that does not depend on approximation ratio); such a reduction preserves membership in PTAS, APX, log-APX, Poly-APX (see [1]). Given any DBR problem, we construct the ARA problem with one target such that $T = \{1, \ldots, k\} \times \{1, \ldots, n\}$. Choose the weights $w_{i,j}$'s such that $w_{i,j} \propto d_{i,j}$ and $w_{i,j} \leq 1/\max_{\mathbf{x} \in MgS} \sum_{i,j} x_{i,j}$. Observe that $\max_{\mathbf{x} \in MgS} \sum_{i,j} x_{i,j}$ is computable efficiently and $\max_{\mathbf{x} \in MgS} \sum_{i,j} x_{i,j} \geq \max_{\mathbf{x} \in conv(P)} \sum_{i,j} x_{i,j}$, thus, the ARA is well-defined. Thus, due to just one target, the ARA optimization is same as $\max_{\mathbf{x} \in conv(P)} \mathbf{w} \cdot \mathbf{x}$. Suppose we can solve this problem with r approximation with the solution mixed strategy being $\mathbf{x}^{\epsilon} = \sum_{i=1}^{m} a_i \mathbf{P_i}$ for some pure strategies $\mathbf{P_i}$. Now, since $w_{i,j} \propto d_{i,j}$ we also know that this solution also provides r approximation for DBR-C. Let the optimal solution for DBR-C be OPT; note that OPT is also the optimal solution for DBR. $\mathbf{x}^{\epsilon}$ provides a solution value $\mathbf{w} \cdot \mathbf{x}^{\epsilon} \geq OPT/r$. Further, as the objective is linear in $\mathbf{x}$ and $\mathbf{x}^{\epsilon} = \sum_{i=1}^{m} a_i \mathbf{P_i}$, it must be the case that there exists a $j \in \{1, \ldots, m\}$ such that $\mathbf{w} \cdot \mathbf{P_j} \geq \mathbf{w} \cdot \mathbf{x}^{\epsilon} \geq OPT/r$. Thus, since $\mathbf{P_j} \in P$, $\mathbf{P_j}$ provides r approximation for DBR. Since, m the number of the pure strategies in support of $\mathbf{x}^{\epsilon}$ is polynomial, $\mathbf{P_j}$ can be found in polynomial time by a linear search.

Proof of Theorem 2:

Proof. Given an independent set problem with V vertices, we construct a TSG with $\{1, \ldots, V+1\}$ team types, where each team type in $1, \ldots, V$ corresponds to a vertex. The $V+1$ team is special in the sense that it does not correspond to any vertex and it is made up of just one resource with a very large resource capacity $2V$. Construct just one passenger category with passengers $N = V+1$. Since, there is just one passenger category (and target) we will use x_i as the matrix entries instead of $x_{i,j}$. Choose $U_s^t = V+1$ and $U_u^t = 0$ and efficiencies $E_i = 1$ for all teams, except $E_{V+1} = 0$. Then, the objective of the integer LP is $\sum_{i=1}^{V} x_i = \mathbf{1}_V \cdot \mathbf{x}$ where $\mathbf{1}_V$ is a vector with first V components as 1 and last component as 0.

Next, have resources for every edge $(i,k) \in E$ with resource capacity 1. This provides the inequality $\sum_{(i,k) \in E} x_i + x_j \leq 1$. Also, we have $x_{V+1} \leq 2V$. Inspection of every passengers provides the constraints $\sum_{i=1}^{V+1} x_i = V+1$. Treating x_{V+1} as a slack, we can see that the constraint $x_{V+1} \leq 2V$ and $\sum_{i=1}^{V+1} x_i = V+1$ are redundant. For the left over constraints $\sum_{(i,k) \in E} x_i + x_j \leq 1$, we can easily check that any valid integral assignment (pure strategy) is an independent set. Moreover, the objective $\sum_{i=1}^{V} x_i$ tries to maximize the independent set. The optimal value of this optimization over $conv(P)$ is an extreme point which is integral and equal to the maximum independent set OPT. Thus, suppose a solution $\mathbf{x}^{\epsilon}$ to the SSE problem with value $\geq OPT/r$. Further, as the objective is linear in $\mathbf{x}$ and $\mathbf{x}^{\epsilon} = \sum_{i=1}^{m} a_i \mathbf{P_i}$, it must be the case that there exists a $j \in \{1, \ldots, m\}$ such that $\mathbf{1}_V \cdot \mathbf{P_j} \geq \mathbf{1}_V \cdot \mathbf{x}^{\epsilon} \geq OPT/r$. Thus, since $\mathbf{P_j} \in P$, $\mathbf{P_j}$ provides r approximation for maximum independent set. Since, m the number of the pure strategies in support of $\mathbf{x}^\epsilon$ is polynomial, $\mathbf{P_j}$ can be found in poly time by a linear search.

Proof of Theorem 5:

Proof. Consider the event of a target t having an infeasible assignment after the comb sampling. Call this event E_t. Let $C_{t,i}$ be the event that resource i covers this target t. Then, $P(E_t) = \sum_i P(E_t|C_{t,i})P(C_{t,i})$. From the guarantees of comb sampling we know that $P(C_{t,i}) = \sum_{j:(i,j)\in T} x^m_{i,j} \leq 1$ and $P(x_{i,j} = 1) = x^m_{i,j}$. Also, by comb sampling if $x_{i,j} = 1$ then $x_{i,j'} = 0$ for any $j' \neq j$. Next, we know that $P(E_t|C_{t,i})$ is the probability that the any of the other $x_{i',j}$ is assigned a one, which is $1-$ the probability that all other $x_{i',j}$ are assigned 0. Thus,

$$P(E_t|C_{t,i}) = 1 - \prod_{i'\neq i}(1 - P(C_{t,i}))$$

Let $p_{t,i} = P(C_{t,i})$. Considering the fact that $\prod_i(1 - p_{t,i}) > 1 - \sum_i p_{t,i}$, we get

$$1 - \prod_{i'\neq i}(1 - P(C_{t,i})) \leq \sum_{(i',j):i'\neq i \wedge (i',j)\in T} x^m_{i',j} \leq 1 - \sum_j x^m_{i,j}$$

where the last inequality is due to the fact that $\sum_{(i,j)\in T} x^m_{i,j} \leq 1$.

Thus, $P(E_t) \leq \sum_i (1 - p_{t,i})p_{t,i} \leq \sum_i p_{t,i} - \sum_i (p_{t,i})^2$. Next, we know from standard sum of squares inequality that $\sum_i (p_i)^2 \geq (\sum_i p_i)^2/k$. Thus, we get $P(E_t) \leq (\sum_i p_i)(1 - \sum_i p_i/k)$ The RHS is maximized when $\sum_i p_i = 1$, thus, $P(E_t) \leq 1 - 1/k$. Also, then $P(\neg E_t) \geq 1/k$

Now consider the coverage of target t: $x^m_t = \sum_{(i,j)\in T} x^m_{i,j}$. According to our algorithm the allocation for target t continues to remain 1 with probability $(1/2)^C$ if its allocation is already feasible after comb sampling (and we always obtain a pure strategy). This is because this target shares schedules with C other targets and thus in the worst case may be reduced with 1/2 probability for each of the C targets. We do a worst case analysis and assume that no resource is allocated to a target when the sampled allocation is infeasible for that target. Thus, let y_t denote the random variable denoting that target t is covered. Thus, $E(y_t) = P(y_t = 1) = P(y_t = 1|E_t)P(E_t) + P(y_t = 1|\neg E_t)P(\neg E_t)$. Now, $P(y_t = 1|\neg E_t)$ is same as $x^m_t/2^C$ and we assumed the worst case of $P(y_t = 1|E_t) = 0$. Thus, we have $E(y_t) \geq x^m_t/2^C k$. As the utilities are linear in y_t, we have the utility for t as $U_t \geq U^m_t/2^C k$, where U^m_t is the utility under the marginal $\mathbf{x}^m$. Thus, if t^* is the choice of adversary under the marginal $\mathbf{x}^m$ we know that $U^m_{t^*}$ is the lowest utility for the defender over all targets t. Hence, we can conclude that the utility with the approximation is at least $U^m_{t^*}/2^C k$

Proof of Theorem 4:

Proof. The main assumption in the proof is that the steps after after comb sampling changes the probability of detecting an adversary in passenger category j by at most $1/c$. Also, by assumption of the theorem since Algorithm 1 does not fail ever, the change in utility for any passenger category j is at most a factor of $1/c$. By similar reasoning as for FAMS, we conclude that this provides a c-approximation.