Randomized Sampling for Large Zero-Sum Games

Shaunak D. Bopardikar Alessandro Borri João Hespanha Maria Prandini Maria D. Di Benedetto

***Abstract*— This paper addresses the solution of large zero-sum matrix games using randomized methods. We provide a procedure by which a player can compute mixed policies that, with a high probability, are security policies against an adversary that is also using randomized methods to solve the same game. The computational savings result from solving subgames that are much smaller than the original game and we provide bounds on how large these subgames should be to guarantee the desired high probability. We propose two methodologies to solve this problem. The first provides a game-independent bound on the size of the subgames that can be computed a priori. The second procedure is useful when computation limitations prevent a player from satisfying the first game-independent bound and provides a high-probability bound on how much the outcome of the game can violate the precomputed security level. All our probabilistic bounds are independent of the size of the original game and could, in fact, apply to games with continuous action spaces. To demonstrate the usefulness of these results, we apply them to solve a hide-and-seek game that exhibits exponential complexity.**

I. Introduction

While a large number of robust design problems can be formulated as zero-sum matrix games, in practice, such games lead to extremely large — often infinite — matrices. This is the case in any combinatorial problem, where the decision makers are faced with a number of possible options that increases exponentially with the size of the problem; for example, in path planning problems where the number of paths increases combinatorially with the number of points [1]. Very large zero-sum matrix games also arise in partial information feedback games wherein optimal strategies are functions of the players' actions and thus the number of possible strategies grows exponentially with the size of the players' action spaces.

Inspired by the use of randomized approaches to solve optimization problems, we consider an approach to solve very large zero-sum matrix games by using randomized sampling. Each player reduces her search space by taking a random sample of the available actions to construct a much smaller version of the original game. Players then solve these smaller games and utilize the saddle-point policies so obtained against each other. We call this procedure the *sampled saddle-point* (SSP) algorithm. Since each player only considers a small submatrix of the original game and the two players typically consider very different submatrices, the saddle-point policies obtained by this process will generally not be security policies for the whole game. This means that a player can obtain an outcome that is strictly worse than the value computed based on her submatrix. However, we show that this happens with low probability if the size of the submatrix is sufficiently large.

In this framework, a reasonable notion of security policy for a player is that the outcome of the game should not unpleasantly surprise the player with high probability. In particular, one wants the outcome of the game not too be much worse than what the player expects based on the computation of the value of her submatrix. In this paper, we analyze this sampling procedure for zero-sum games and provide conditions under which it leads to a security policy with high probability.

Related Work

Two-player zero-sum matrix games have been studied extensively over the past decades [2]. The classical Mini-Max theorem guarantees the existence of an optimal pair of strategies for the two players, each of which is a security policy for the corresponding player. However, when the matrix is of large size, the computation of the optimal strategies involves solving optimization problems with a large number of variables and constraints. Using probabilistic analysis, the existence of simple, near-optimal strategies over a subset with logarithmically smaller size of the original matrix game was established in [3].

Randomized methods have also been successful in providing efficient solutions to complex control design problems with probabilistic guarantees. [4] adopts a probabilistic approach to show the existence of randomized algorithms with polynomial complexity to solve complex robust stability analysis problems. [5] proposes a randomized method to determine the minimum number of samples that provide a probabilistic guarantee of the level of worst-case controller performance. [6, 7] demonstrate the use of randomized algorithms in statistical learning theory to solve control design problems and a number of well known complex problems in matrix theory. In [8–10], the authors introduce the scenario approach to solve convex optimization problems with an infinite number of constraints, and discuss possible applications of the approach to systems and control. The

This work has been partially supported by the Center of Excellence for Research DEWS, University of L'Aquila, Italy.

Shaunak D. Bopardikar and João Hespanha are with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, CA, USA, shaunak@engineering.ucsb.edu, hespanha@ece.ucsb.edu. Alessandro Borri and Maria D. Di Benedetto are with the Department of Electrical Engineering, University of L'Aquila, Italy {alessandro.borri,mariadomenica.dibenedetto}@univaq.it. Maria Prandini is with the Dipartimento di Elettronica e Informazione of the Politecnico di Milano, prandini@elet.polimi.it.

results in these papers are instrumental to establish several of the results in the present paper.

Contributions

Our contributions are three-fold. Based on results from the scenario approach in [8–10], we show that when the sizes of the subgames solved by each player are sufficiently large, the SSP algorithm provides security policies for both players with some pre-specified high probability $1-\delta$. The lower bounds on the sizes of the subgames are *game independent* and are easily computable a-priori. Not surprisingly, they grow with the desired confidence level $1-\delta$. However, they are *independent of the size of the original matrix game*, which could, in fact, be even infinite and not even have a value.

We also propose a procedure that provides an a-posteriori, high-probability bound on the deviation of the outcome of the game from the pre-computed security level. In particular, regardless of the size of the subgames solved by each player, we provide a high probability bound on how much a player can expect the outcome of the game to violate the value computed based on the submatrix used to determine her saddle-point equilibrium. This bound is computed after a player selects and solves her particular subgame.

Third and finally, we apply our procedure to efficiently solve a hide-and-seek game, in which one player hides a treasure in one of N points and the other player searches for the treasure by visiting each of the points. This is formalized as a zero-sum game in which the player that hides the treasure wants to maximize the distance that the other player needs to travel until the treasure is found. To determine the optimal strategy for this game, it is required to solve a matrix game whose size is $N \times N!$. Thus, exact solutions to this problem require computation that scales exponentially with the number of points N. Our approach is *completely independent of the size of the game* and therefore the total number of points plays no role in the amount of computation required by it. This is possible because each player concentrates on a subset of her action set, and probabilistic guarantees rather than deterministic guarantees on the quality of the solution with respect to the actual game outcome are provided.

Organization

This paper is organized as follows. The problem formulation and the sampled saddle-point algorithm are described in Section II. The game-independent a-priori bounds are established in Section III. The a-posteriori bounds are established in Section IV. The hide-and-seek problem and the implementation of our procedure are described in Section V.

II. Sampled Saddle-Point Algorithm

Consider a zero-sum matrix game defined by an $M \times N$ matrix A, in which player P_1 is the minimizer and selects rows and player P_2 is the maximizer and selects columns. We are interested in problems for which the matrix A is too large to permit the computation of mixed saddle-points and therefore the players are forced to consider only submatrices of A to select their policies. This scenario motivates the following procedure, which we henceforth call the *sampled saddle-point (SSP) algorithm*.

1) Each player P_k, $k \in \{1,2\}$ randomly selects m_k rows and n_k columns of A, which uses to construct a $m_k \times n_k$ submatrix A_k of A. Denoting by $\mathcal{B}^{k\times\ell}$ the set of $k\times\ell$ left-stochastic $(0,1)$-matrices (i.e., matrices whose entries belong to the set $\{0,1\}$ and whose columns add up to one), we can express the process of constructing each submatrix A_k by randomly selecting two random matrices $\Gamma_k \in \mathcal{B}^{M\times m_k}$ and $\Pi_k \in \mathcal{B}^{N\times n_k)}$ and then computing the product:
$$A_k = \Gamma_k' A \Pi_k.$$
2) Each player P_k, $k \in \{1,2\}$ computes the mixed security value and the corresponding security policy for her submatrix A_k:
$$\bar{V}(A_1) = \max_{z\in\mathcal{S}_{n_1}} y_1^{*\prime} A_1 z = \min_{y\in\mathcal{S}_{m_1}} \max_{z\in\mathcal{S}_{n_1}} y' A_1 z$$
$$\underline{V}(A_2) = \min_{y\in\mathcal{S}_{m_2}} y' A_2 z_2^* = \max_{z\in\mathcal{S}_{n_2}} \min_{y\in\mathcal{S}_{m_2}} y' A_2 z$$
where $\mathcal{S}_{m_k}$ and $\mathcal{S}_{n_k}$ denote the probability simplexes of appropriate dimensions. We call $\bar{V}(A_1)$ and $\underline{V}(A_2)$ the *sampled security values of the game* for players P_1 and P_2, respectively.
3) Player P_1 selects a row according to the distribution y_1^*, whereas P_2 selects a column according to the distribution z_2^*, which correspond to the following policies for the original game
$$y^* := \Gamma_1 y_1^*, \qquad z^* := \Pi_2 z_2^*$$
and the following game outcome
$$y^{*\prime} A z^* = y_1^{*\prime} \Gamma_1' A \Pi_2 z_2^*.$$
We call y^* and z^* the *sampled security policies* for players P_1 and P_2, respectively.

We say that *the SSP algorithm is ϵ-secure for player P_1 with confidence* $1-\delta$ if
$$\mathrm{P}_{\Gamma_1,\Pi_1,\Gamma_2,\Pi_2}\left(y^{*\prime} A z^* \le \bar{V}(A_1) + \epsilon\right) \ge 1-\delta. \qquad (1)$$
Here and in the sequel, we use a subscript in the probability measure P to emphasize which random variables define the events that is being measured. In essence, condition (1) states that the probability that the outcome of the game will violate P_1's sampled security value by more than ϵ is smaller than δ. Similarly, we say that *the SSP algorithm is ϵ-secure for player P_2 with confidence* $1-\delta$ if
$$\mathrm{P}_{\Gamma_1,\Pi_1,\Gamma_2,\Pi_2}\left(y^{*\prime} A z^* \ge \underline{V}(A_2) - \epsilon\right) \ge 1-\delta. \qquad (2)$$
The previous definitions guarantee that the two players will be surprised with low probability when playing with policies obtained from a one-shot solution to the SSP algorithm. However, no specific guarantee is given to the inherent safety of the policies/values obtained using this algorithm. So, e.g.,

player P_1 computes y^* once using the SSP algorithm and then plays this policy multiple times against a sequence of policies z^* that P_2 obtained by running the SSP algorithm multiple times, P_1 could conceivably be surprised with high probability. This would happen if she was "unlucky" and got a particular (low probability) y^* that is particularly bad or a value $\bar{V}(A_1)$ that is particularly optimistic. To avoid this scenario, we introduce notions of security that refer to specific policies/values: We say that *the policy y^* with value $\bar{V}(A_1)$ is ϵ-secure for player P_1 with confidence $1-\delta$* if

$$\mathrm{P}_{\Gamma_2,\Pi_2}\left(y^{*\prime}Az^* \le \bar{V}(A_1)+\epsilon \mid y^*,\ \bar{V}(A_1)\right) \ge 1-\delta \quad (3)$$

and that *the policy z^* with value $\underline{V}(A_2)$ is ϵ-secure for player P_2 with confidence $1-\delta$* if

$$\mathrm{P}_{\Gamma_1,\Pi_1}\left(y^{*\prime}Az^* \ge \underline{V}(A_2)-\epsilon \mid z^*,\ \underline{V}(A_1)\right) \ge 1-\delta. \quad (4)$$

So far, we have not specified the joint distribution of the row/column extraction matrices $\Gamma_1,\Gamma_2,\Pi_1,\Pi_2$, but this distribution will clearly affect the outcomes of the algorithm. In the context of noncooperative games, one should presume the extractions of the two players to be independent of each other. For simplicity, we will further assume that players extract rows and columns independently, as stated in the following assumption:

Assumption 1 (Independence): The four random matrices $\Gamma_1,\Pi_1,\Gamma_2,\Pi_2$ are statistically independent. Moreover, each of these matrices have independent and identically distributed columns. □

Remark 1 (Non-unique security policies): When the matrices A_1 and A_2 have multiple security policies, the SSP algorithm does not specify *which* of these should be used to define the sampled security policies. However, the choice of security policy may have a significant effect on the value of the probabilities in (1) and (2). In view of this, any useful probabilistic guarantee for ϵ-security should hold independently of which security policy is used in the SSP algorithm. This is the case of all results presented in this paper. □

Remark 2 (Non-matrix games): The results in this paper do not depend on the fact that the original game is a matrix game. They would extend trivially to any cost-function $J(u,d)$, $u\in\mathcal{U}$, $d\in\mathcal{D}$ where $\mathcal{U}$ and $\mathcal{D}$ denote the sets of policies for the minimizer and maximizer, respectively. In fact, it is not even necessary that the original game has saddle-point policies since all that the SSP algorithm uses is the fact that when we take finite samples of the sets of policies, we obtain finite matrix games. □

III. Game-Independent Probabilistic Guarantees

The main result of this section provides a bound on the size of the submatrices for the players that guarantees ϵ-security with $\epsilon=0$.

Theorem 1 (Game independent bounds): Suppose that Assumption 1 holds. Then

1) If Π_1 and Π_2 have identically distributed columns and

$$n_1 = \left\lceil \frac{m_1}{\delta} - 1 \right\rceil \bar{n}_2 \quad (5)$$

for some $\bar{n}_2 \ge n_2$, then the SSP algorithm is $\epsilon=0$-secure for P_1 with confidence $1-\delta$. If one further increases n_1 to satisfy

$$n_1 = \left\lceil \frac{2}{\delta}\Big(\ln\frac{1}{\beta}+m_1+1\Big)\right\rceil \bar{n}_2 \quad (6)$$

then, with probability[1] higher than $1-\beta$, the policy y^* with value $\bar{V}(A_1)$ is $\epsilon=0$-secure for P_1 with confidence $1-\delta$.

2) If Γ_1 and Γ_2 have identically distributed columns and

$$m_2 = \left\lceil \frac{n_2}{\delta} - 1 \right\rceil \bar{m}_1 \quad (7)$$

for some $\bar{m}_1 \ge m_1$, then the SSP algorithm is $\epsilon=0$-secure for P_2 with confidence $1-\delta$. If one further increases m_2 to satisfy

$$m_2 = \left\lceil \frac{2}{\delta}\Big(\ln\frac{1}{\beta}+n_2+1\Big)\right\rceil \bar{m}_1 \quad (8)$$

then, with probability[2] higher than $1-\beta$, the policy z^* with value $\underline{V}(A_2)$ is $\epsilon=0$-secure for P_2 with confidence $1-\delta$. □

In words, this results states that it is always possible to guarantee $\epsilon=0$-security for P_1, provided that she constructs her submatrix A_1 utilizing a sufficiently large number of columns. In particular, she always needs to choose a number of columns n_1 larger than the number of columns n_2 that P_2 is considering for her mixed policies. The additional number of columns P_1 needs to consider is a function of the number m_1 of rows that P_1 wants to consider for her mixed policy and the desired confidence levels. The result for P_2 is analogous.

In the probabilistic guarantees provided by Theorem 1 with (5), the confidence $1-\delta$ refers to the extraction of all the row/column matrices $\Gamma_1,\Gamma_2,\Pi_1,\Pi_2$ as in (1). However, for the probabilistic guaranteed with (6), the confidence $1-\delta$ refers to the extraction of Γ_2,Π_2 as in (3), whereas the confidence $1-\beta$ refers solely to the extraction of the matrix Π_1 and holds for any given matrix Γ_1 (as shown in the proof).

Note that only the logarithm of the confidence level β appears in bounds regarding the security of y^* and z^*. One can therefore make β extremely small with a relatively small additional computational cost.

Remark 3 (P_1's knowledge of n_2): According to Theorem 1, for player P_1 to enjoy guaranteed $\epsilon=0$-security with confidence $1-\delta$, she must know an upper bound $\bar{n}_2$ on the number of columns that P_2 is using to construct her submatrix A_2. Even if P_1 does not know $\bar{n}_2$ precisely and, e.g., underestimates $\bar{n}_2$ by a certain percentage, then (5) and (6) are still useful in that it predicts that the performance degradation in the confidence level δ should grow roughly by the same percentage. This is because the bounds in (5)

[1] The confidence level β for P_1 refers solely to the extraction of the matrix Π_1 and holds for any given matrix Γ_1.

[2] The confidence level β for P_2 refers solely to the extraction of the matrix Γ_2 and holds for any given matrix Π_2.

and (6) essentially scale with $\bar{n}_2/\delta$. An analogous comment could be made regarding the bounds (7) and (8) and about P_2's knowledge of m_1. □

Remark 4 (P_1's knowledge of the distribution of Π_2): To apply Theorem 1, the distributions of Π_1 and Π_2 must match, which means that P_1 must sample the columns of the matrix A using the same distribution as P_2. Suppose now that (5) holds but that the distributions of Π_1 and Π_2 do not match. Expanding the probability in (1) and using $\mathfrak{E}$ to abbreviate the event $y^{*\prime}Az^* \leq \bar{V}(A_1)+\epsilon$, we conclude that

$$\begin{aligned}
&\mathrm{P}\big(y^{*\prime}Az^* \leq \bar{V}(A_1)+\epsilon\big)\\
&\quad= \sum_W \mathrm{P}\left(\mathfrak{E} \mid \Pi_2 = W\right)\mathrm{P}(\Pi_2 = W)\\
&\quad= \sum_W \mathrm{P}\left(\mathfrak{E} \mid \Pi_2 = W\right)\mathrm{P}(\Pi_1 = W)\frac{\mathrm{P}(\Pi_2 = W)}{\mathrm{P}(\Pi_1 = W)}\\
&\quad\geq \Big(\min_W \frac{\mathrm{P}(\Pi_2 = W)}{\mathrm{P}(\Pi_1 = W)}\Big)\sum_W \mathrm{P}\left(\mathfrak{E} \mid \Pi_2 = W\right)\mathrm{P}(\Pi_1 = W)\\
&\quad\geq (1-\delta)\min_W \frac{\mathrm{P}(\Pi_2 = W)}{\mathrm{P}(\Pi_1 = W)},
\end{aligned}$$

where in the last step we were able to use Theorem 1 because the conditional probability on Π_2 was weighted by the distribution of Π_1. While the above bound is not particularly tight, it still allow us to conclude that a degradation in the confidence level is to be expected if Π_1 and Π_2 do not match, but this degradation should be small if the ratio of the distributions remains point-wise close to one. □

Proof of Theorem 1. We only prove the statement 1, since the proof of statement 2 can be obtained by symmetry. By definition of the security value $\bar{V}(A_1)$, we have that

$$\begin{aligned}
\bar{V}(A_1) &= \min_{y\in\mathcal{S}_{m_1}} \max_{z\in\mathcal{S}_{n_1}} y'\Gamma_1' A\Pi_1 z\\
&= \min_{y\in\mathcal{S}_{m_1}} \max_{j\in\{1,\dots,n_1\}} y'\Gamma_1' A\Pi_1 e_j\\
&= \min_{\theta\in\Theta}\big\{v : y'\Gamma_1' A\Pi_1 e_j \leq v,\ \forall j\in\{1,\dots,n_1\}\big\},
\end{aligned} \tag{9}$$

where e_j denotes the jth element of the canonical basis of $\mathbb{R}^{n_1}$, $\theta := (y,v)$, and $\Theta := \mathcal{S}_{m_1}\times\mathbb{R}$.

Since n_1 is an integer multiple of $\bar{n}_2$, i.e., $n_1 = K\bar{n}_2$ with $K = \left\lceil \frac{m_1}{\delta} - 1\right\rceil$, we can take the $K\bar{n}_2$ columns of $\Pi_1 \in \mathcal{B}^{N\times K\bar{n}_2}$ to construct K i.i.d. matrices $\Delta_1,\Delta_2,\dots,\Delta_K$, each in the set $\mathcal{B}^{N\times\bar{n}_2}$. If we then define the function

$$f(\theta,\Delta) = \max_{j\in\{1,\dots,\bar{n}_2\}} y'\Gamma_1' A\Delta e_j - v,$$

$\forall\theta := (y,v)\in\Theta$, $\Delta\in\mathcal{B}^{N\times\bar{n}_2}$, we can rewrite (9) as

$$\bar{V}(A_1) = \min_{\theta\in\Theta}\big\{v : f(\theta,\Delta_i)\leq 0,\ \forall i\in\{1,\dots,K\}\big\},$$

Let the minimum above be achieved for some $\theta^* = (y_1^*, \bar{V}(A_1))$. For any given realization of the matrix Γ_1 (which is independent of the Δ_i by Assumption 1) we conclude from [9, Proposition 3] that the (conditional) probability that another matrix Δ sampled independently from the same distribution as the Δ_i satisfies the constraint $f(\theta^*,\Delta)\leq 0$ can be lower-bounded as follows:

$$\mathrm{P}_{\Pi_1,\Delta}\left(f(\theta^*,\Delta)\leq 0 \mid \Gamma_1\right) \geq \frac{K-m_1}{K+1} \geq 1-\delta, \tag{10}$$

where the second inequality is a consequence of (5). Using the definition of f and θ^*, we can re-write (10) as

$$\begin{aligned}
&\mathrm{P}_{\Pi_1,\Delta}\big(y_1^{*\prime}\Gamma_1' A\Delta e_j \leq \bar{V}(A_1),\\
&\qquad\qquad \forall j\in\{1,\dots,\bar{n}_2\} \mid \Gamma_1\big) \geq 1-\delta.
\end{aligned}$$

Since $n_2\leq\bar{n}_2$, we further conclude that

$$\begin{aligned}
&\mathrm{P}_{\Pi_1,\Delta}\big(y_1^{*\prime}\Gamma_1' A\Delta e_j \leq \bar{V}(A_1),\\
&\qquad\qquad \forall j\in\{1,\dots,n_2\} \mid \Gamma_1\big) \geq 1-\delta.
\end{aligned}$$

Under Assumption 1, when the columns of Π_1 and Π_2 are identically distributed, the matrix consisting of the first n_2 columns of Δ can be viewed as the matrix Π_2 and we conclude from the inequality above that

$$\begin{aligned}
&\mathrm{P}_{\Pi_1,\Pi_2}\big(y_1^{*\prime}\Gamma_1' A\Pi_2 e_j \leq \bar{V}(A_1),\\
&\qquad\qquad \forall j\in\{1,\dots,n_2\} \mid \Gamma_1\big) \geq 1-\delta.
\end{aligned}$$

Since

$$\begin{aligned}
&y_1^{*\prime}\Gamma_1' A\Pi_2 e_j \leq \bar{V}(A_1),\ \forall j\in\{1,\dots,n_2\} \Rightarrow\\
&\qquad y_1^{*\prime}\Gamma_1' A\Pi_2 z \leq \bar{V}(A_1),\ \forall z\in\mathcal{S}^{n_2},
\end{aligned}$$

we conclude that

$$\mathrm{P}_{\Pi_1,\Gamma_2,\Pi_2}\left(y_1^{*\prime}\Gamma_1' A\Pi_2 z_2^* \leq \bar{V}(A_1) \mid \Gamma_1\right) \geq 1-\delta.$$

Since we have shown that this bound holds for an arbitrary realization of Γ_1, it also holds for the unconditional probability, which shows that the SSP algorithm is $\epsilon = 0$-secure for P_1 with confidence $1-\delta$.

If instead of applying [9, Proposition 3] we apply [10, Theorem 1] and using (6), we conclude that

$$\mathrm{P}_{\Delta}\left(f(\theta^*,\Delta)\leq 0 \mid \Gamma_1,\theta^*\right) \geq 1-\delta$$

with probability higher than $1-\beta$, where the confidence level $1-\beta$ refers to the extraction of $\Pi_1 = [\Delta_1,\dots,\Delta_K]$ that defines θ^* (given Γ_1). The proof can now proceed exactly as before, but with (10) replaced by the inequality above that now involves a probability conditioned to y^* and $\bar{V}(A_1)$. This shows that, with probability higher than $1-\beta$, the policy y^* with value $\bar{V}(A_1))$ is $\epsilon = 0$-secure for P_1 with confidence $1-\delta$. ■

IV. A-POSTERIORI PROBABILISTIC GUARANTEES

Suppose that, due to computational limitations, player P_1 cannot satisfy the bounds in Theorem 1 to obtain $\epsilon = 0$-security for a given level of confidence $1-\delta$. One option to overcome this difficulty would be to settle for a lower level of confidence until the bounds in Theorem 1 hold for a value of n_1 that is computationally acceptable for P_1. However, one may desire to maintain the same high level of confidence, and instead accept a larger value for ϵ. In this section, we explore this option, which is not covered by Theorem 1. For short, we only consider the SSP algorithm from the perspective of P_1.

Consider the following procedure for P_1:

1) Pick some value for n_1 and use the SSP algorithm to compute a sampled security policy y^* and the corresponding sampled security value $\bar{V}(A_1)$.
2) Using the column distribution of Π_1, independently extract k_1 columns of A into a matrix $\bar{\Pi}_1 \in \mathcal{B}^{N\times k_1}$ and compute the row vector
$$\bar{v} := \max_{j\in\{1,\ldots,k_1\}} y^{*\prime} A\bar{\Pi}_1 e_j, \tag{11}$$
where e_j denotes the jth element of the canonical basis of $\mathbb{R}^{k_1}$.

The following result provides an a-posteriori guarantee on this procedure.

Theorem 2 (A-posteriori bounds): Suppose that Assumption 1 holds. If Π_1 and Π_2 have identically distributed columns and
$$k_1 = \left\lceil \frac{1}{\delta} - 1 \right\rceil \bar{n}_2, \tag{12}$$
for some $\bar{n}_2 \geq n_2$, then the SSP algorithm is *ϵ-secure for P_1 with confidence* $1-\delta$ for any
$$\epsilon \geq \bar{v} - \bar{V}(A_1). \tag{13}$$
If one further increases k_1 to satisfy
$$k_1 = \left\lceil \frac{\ln(1/\beta)}{\ln(1/(1-\delta))} \right\rceil \bar{n}_2, \tag{14}$$
then, with probability higher than $1-\beta$, the policy y^* with value $\bar{V}(A_1)$ is ϵ-secure for P_1 with confidence $1-\delta$. □

In the probabilistic guarantee provided by Theorem 2 with (12), the confidence $1-\delta$ refers not only to the extraction of the row/column matrices $\Gamma_1, \Gamma_2, \Pi_1, \Pi_2$, but also to the test matrix $\bar{\Pi}_1$ since ϵ depends on it, i.e., (1) should be understood as
$$\mathrm{P}_{\Gamma_1,\Pi_1,\Gamma_2,\Pi_2,\bar{\Pi}_1}\left(y^{*\prime} A z^* \leq \bar{V}(A_1) + \epsilon\right) \geq 1-\delta. \tag{15}$$
For the probabilistic guarantee with (14), the confidence $1-\delta$ refers to the extraction of Γ_2, Π_2, i.e., (3) should be understood as
$$\mathrm{P}_{\Gamma_2,\Pi_2}\left(y^{*\prime} A z^* \leq \bar{V}(A_1) + \epsilon \mid y^*,\ \bar{V}(A_1),\ \epsilon\right) \geq 1-\delta$$
whereas the confidence $1-\beta$ refers solely to the extraction of the matrix $\bar{\Pi}_1$.

Proof of Theorem 2. From the definition of $\bar{v}$ and (13), we conclude that
$$\bar{V}(A_1) + \epsilon \geq \bar{v} = \max_{j\in\{1,\ldots,K\bar{n}_2\}} y^{*\prime} A\bar{\Pi}_1 e_j, \tag{16}$$
where $K := \left\lceil \frac{1}{\delta} - 1 \right\rceil$. Partitioning the columns of $\bar{\Pi}_1 \in \mathcal{B}^{N\times K\bar{n}_2}$ to construct K i.i.d. matrices $\Delta_1, \Delta_2, \ldots, \Delta_K$, each in the set $\mathcal{B}^{N\times\bar{n}_2}$ and defining
$$f(\Delta) = \max_{j\in\{1,\ldots,\bar{n}_2\}} y^{*\prime} A\Delta e_j, \quad \forall \Delta \in \mathcal{B}^{N\times\bar{n}_2}$$
we can rewrite (16) as
$$\bar{V}(A_1) + \epsilon \geq \max_{i\in\{1,\ldots,K\}} f(\Delta_i). \tag{17}$$
For any given realizations of y^* and $\bar{V}(A_1)$ (which are independent of the Δ_i), we conclude from [9, Proposition 4] that the (conditional) probability that another matrix Δ, sampled independently from the same distribution as the Δ_i, satisfies the constraint $f(\Delta) \leq \max_{i\in\{1,\ldots,K\}} f(\Delta_i)$ can be lower-bounded as follows:
$$\mathrm{P}_{\bar{\Pi}_1,\Delta}\left(f(\Delta) \leq \max_{i\in\{1,\ldots,K\}} f(\Delta_i) \mid y^*, \bar{V}(A_1)\right) \geq \frac{K}{K+1} \geq 1-\delta, \tag{18}$$
where the second inequality is a consequence of (12). Using the definition of f and (17), we conclude from (18) that
$$\mathrm{P}_{\bar{\Pi}_1,\Delta}\left(\max_{j\in\{1,\ldots,\bar{n}_2\}} y^{*\prime} A\Delta e_j \leq \bar{V}(A_1) + \epsilon \mid y^*, \bar{V}(A_1)\right) \geq 1-\delta,$$
and therefore
$$\mathrm{P}_{\bar{\Pi}_1,\Delta}\left(y^{*\prime} A\Delta e_j \leq \bar{V}(A_1) + \epsilon,\ \forall j \in \{1,\ldots,\bar{n}_2\} \mid y^*, \bar{V}(A_1)\right) \geq 1-\delta.$$
Since $n_2 \leq \bar{n}_2$, we further conclude that
$$\mathrm{P}_{\bar{\Pi}_1,\Delta}\left(y^{*\prime} A\Delta e_j \leq \bar{V}(A_1) + \epsilon,\ \forall j \in \{1,\ldots,n_2\} \mid y^*, \bar{V}(A_1)\right) \geq 1-\delta.$$
Under Assumption 1, when the columns of Π_1 and Π_2 are identically distributed, the matrix consisting of the first n_2 columns of Δ can be viewed as the matrix Π_2 and we conclude from the inequality above that
$$\mathrm{P}_{\bar{\Pi}_1,\Pi_2}\left(y^{*\prime} A\Pi_2 e_j \leq \bar{V}(A_1) + \epsilon,\ \forall j \in \{1,\ldots,n_2\} \mid y^*, \bar{V}(A_1)\right) \geq 1-\delta.$$
Given that
$$y^{*\prime} A\Pi_2 e_j \leq \bar{V}(A_1) + \epsilon,\ \forall j \in \{1,\ldots,n_2\} \quad \Rightarrow \quad y^{*\prime} A\Pi_2 z \leq \bar{V}(A_1) + \epsilon,\ \forall z \in \mathcal{S}^{n_2},$$
we get that
$$\mathrm{P}_{\Gamma_2,\Pi_2,\bar{\Pi}_1}\left(y^{*\prime} A\Pi_2 z_2^* \leq \bar{V}(A_1) + \epsilon \mid y^*, \bar{V}(A_1)\right) \geq 1-\delta.$$
Since we have shown that this bound holds for arbitrary realizations of y^* and $\bar{V}(A_1)$, it also holds for the unconditional probability, from which (15) follows.

If instead of applying [9, Proposition 4] we use (14) and apply [11, Theorem 1], we conclude that

$$\mathrm{P}_{\Delta}\left(f(\Delta) \leq \max_{i\in\{1,\ldots,K\}} f(\Delta_i) \mid y^*, \bar{V}(A_1), \epsilon\right) \geq 1-\delta,$$

with probability higher than $1-\beta$, where the confidence level $1-\beta$ refers to the extraction of $\bar{\Pi}_1 = [\Delta_1, \ldots, \Delta_K]$ that defines ϵ. The proof can now proceed exactly as before, but with (18) replaced by the inequality above that now involves a probability conditioned to y^*, $\bar{V}(A_1)$, and ϵ. This shows that, with probability higher than $1-\beta$, the policy y^* with value $\bar{V}(A_1)$ is ϵ-secure for P_1 with confidence $1-\delta$. ■

V. Hide-and-seek matrix game

Consider a zero-sum game where P_1 hides a non-moving object (treasure) in one of N points $\{p_1, \ldots, p_N\} \subset \mathbb{R}^2$ on the plane and P_2 has to find the treasure with minimum cost, by traveling from point to point until she finds it. The game is played over the set of mixed policies:

- P_1 chooses a probability distribution $z \in \mathcal{S}_N$ for the treasure over the N points, and
- P_2 chooses a probability distribution $y \in \mathcal{S}_M$ over the set $\mathcal{R} := \{r_j : j = 1, \ldots, M\}$ of $M := N!$ routes that start at P_1's initial position $p_0 \in \mathbb{R}^2$ and go through all possible permutations of the points.

Each route is assigned a cost equal to its total Euclidean length:

$$c(r_j) = \sum_{k=1}^{N} \|r_j(k) - r_j(k-1)\|,$$

where $r_j(0) := p_0$ and each subsequent $r_j(k) \in \mathbb{R}^2$, $k \in \{1, \ldots, N\}$ denotes the kth point in route r_j. When P_1 chooses to hide the treasure at point i and P_2 selects route r_j, the outcome of the game is equal to the cost of route r_j from its initial point until the point p_i where the treasure lies. Namely,

$$A_{ij} = -\sum_{k=1}^{k^*_{ij}} \|r_j(k) - r_j(k-1)\|, \tag{19}$$

where the summation ends at the index k^*_{ij} for which $r_j(k^*_{ij})$ corresponds to the point i where the treasure is hidden. The minus sign in (19) is needed to maintain consistency with the formulation in the first part of the paper, where P_1 is the minimizer. Indeed, P_1 hides the treasure to maximize the distance and therefore to minimize the entries of A.

The exact computation of the optimal mixed strategies is intractable because the size of the matrix A is $N \times N!$. However, the results in this paper regarding the SSP algorithm have a computational complexity that is completely *independent of the size of the game,* which means that we can provide probabilistic guarantees for games with an arbitrarily large number of points.

In this particular game, only the player P_2 that chooses paths has a very large number of options ($M = N!$) so we can assume that both players consider all possible N locations where P_1 can hide the treasure (all rows of A), but randomly select only a small number of paths (columns of A) to construct their submatrices. This means that the player P_2 that selects the paths will never be surprised since she always considers all options for the actions of P_1. However, the player P_1 that hides the treasure should respect the bounds provided by Theorems 1 and 2 to avoid unpleasant surprises.

In our numerical experiments, we considered N points distributed uniformly randomly in a square region.

For a fixed value of $\bar{n}_2$, β, and δ, we run Monte Carlo simulations of the procedure with the a-posteriori guarantees described in Section IV using the bound in (14), and studied the outcome $\bar{v}$ in (11) for increasing values of n_1 up to the corresponding a-priori bound (6), indicated by an arrow in Figure 1. Since $\bar{v}$ is obtained through a randomized procedure, it is a random variable and takes different values in the different Monte Carlo simulations. In Figure 1, we show the dotted 90 (resp. dashed 50) percentile curve such that 90% (resp. 50%) of the realizations of $\bar{v}$ were below this curve.

We then repeated the experiments with the same values of δ and $\bar{n}_2$, but using the a-posteriori bound in (12), and studied the outcome $\bar{v}$ in (11) for increasing values of n_1 up to the corresponding a-priori bound (5). The obtained solid 90 (resp. dashed-dot 50) percentile curves are plotted in Figure 1.

We observe that all of these curves are reasonably ”flat”, implying that with the choice of n_1 that is a few orders of magnitude lower than the a-priori bound, one can obtain a security strategy that has a relatively small value of ϵ with high probability. For example, from Figure 1, we conclude that with a value of n_1 upto 40 times lower than the a-priori bound (6) needed for $\epsilon = 0$-security of the policy y^*, in 90% (resp. 50%) of the simulations, the resulting empirically obtained strategy y^* is ϵ-secure with high probability, where $\epsilon \leq 5$ (resp. $\epsilon \leq 3$).

Figure 2 summarizes numerical results obtained with a higher value of $\bar{n}_2$. Akin to Figure 1, we observe that all of the curves are reasonably flat and compared to Figure 1, the outcome $\bar{v}$ increases on average, as is expected because player P_2 is allowed to select a larger number of columns and hence, to possibly achieve a larger value of the outcome of the game. (e.g., for $n_1 = 1000$, the 90 percentile curve is higher than in Figure 1 by at most 5 units). Both the a-priori and the a-posteriori bounds increase linearly with $\bar{n}_2$. Thus with an increased value of ϵ, it suffices for P_1 to sample much fewer columns than the corresponding a-priori bound to obtain an ϵ-secure strategy with high probability.

VI. Conclusions and Future Directions

We addressed the solution of large zero-sum matrix games using randomized techniques. We provided a procedure by which each player samples a submatrix, computes mixed policies for the submatrix and uses the resulting optimal strategy to play against the other player. We proposed the notion of security policies and levels for each player, and

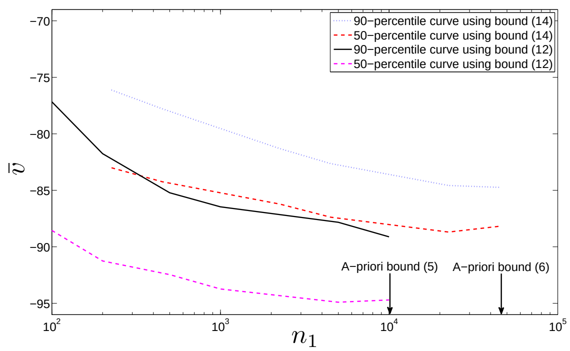

Fig. 1. Numerically determined values of the a-posteriori outcome $\bar{v}$ (cf. Section IV) for different values of n_1. In these experiments, the number of points is $N = 10$, side length of the square region is 50 units, $m_1 = \bar{n}_2 = 10$, $\delta = 0.01$, $\beta = 10^{-5}$, and the rows and the columns were drawn uniformly randomly.

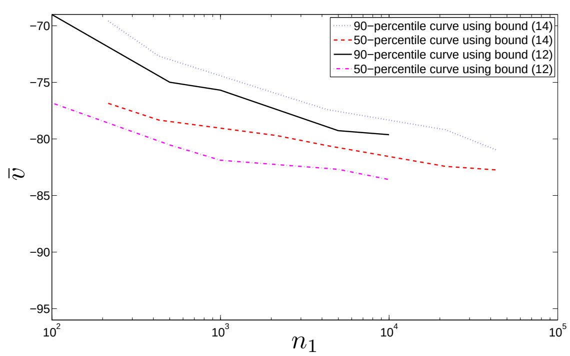

Fig. 2. Numerically determined values of the a-posteriori outcome $\bar{v}$ (cf. Section IV) for different values of n_1. In these experiments, $N = 10$, $m_1 = 10, \bar{n}_2 = 1000$, $\delta = 0.01$, $\beta = 10^{-5}$, and the rows and the columns were drawn uniformly randomly.

derived a-priori game-independent bounds on the size of the submatrices that guarantees a security policy with high probability. We also presented an a-posteriori bound on how much the outcome of the game can violate the precomputed security level if the size of the submatrices do not satisfy the a-priori bounds. Finally, we applied the technique to solve a combinatorial hide-and-seek game.

We are currently exploring the possibility to adopt incremental optimization techniques which could possibly reduce the bound on the size of the submatrices. Future directions of research include the application of randomized techniques to feedback games. It would be interesting to analyze closed-loop versions of the hide-and-seek game that involve the minimizer taking measurements of the location of the treasure as it moves from point to point.

References

[1] R. Bellman, "Dynamic programming treatment of the traveling salesman problem," *J. Assoc. Comput. Mach.*, vol. 9, pp. 61–63, 1962.

[2] T. Basar and G. J. Olsder, *Dynamic Non-Cooperative Game Theory*. Philadelphia, PA, USA: SIAM, 1999.

[3] R. J. Lipton and N. E. Young, "Simple strategies for large zero-sum games with applications to complexity theory," in *Twenty-sixth annual ACM Symposium on Theory of Computing*, 1994, pp. 734–740.

[4] P. Khargonekar and A. Tikku, "Randomized algorithms for robust control analysis and synthesis have polynomial complexity," in *Proceedings of the 35th IEEE Conference on Decision and Control*, vol. 3, Kobe, Japan, Dec. 1996, pp. 3470–3475.

[5] R. Tempo, E. W. Bai, and F. Dabbene, "Probabilistic robustness analysis: Explicit bounds for the minimum number of samples," *Systems and Control Letters*, vol. 30, no. 5, pp. 237–242, 1997.

[6] M. Vidyasagar, "Statistical learning theory and randomized algorithms for control," *IEEE Control Systems Magazine*, vol. 18, no. 6, pp. 69–85, Dec. 1998.

[7] M. Vidyasagar and V. D. Blondel, "Probabilistic solutions to some np-hard matrix problems," *Automatica*, vol. 37, no. 9, pp. 1397–1405, Sep. 2001.

[8] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, May 2006.

[9] M. C. Campi and G. C. Calafiore, "Notes on the scenario design approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 382–385, February 2009.

[10] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annual Reviews in Control*, vol. 33, no. 2, pp. 149–157, Dec. 2009.

[11] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of robust convex programs," *SIAM Journal on Control and Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.