Guaranteed Feasibility in Differentially Private Linearly Constrained Convex Optimization

Alexander Benvenuti[1], Brendan Bialy[2], Miriam Dennis[2], Matthew Hale[1]

***Abstract*— Convex programming with linear constraints plays an important role in the operation of a number of everyday systems. However, absent any additional protections, revealing or acting on the solutions to such problems may reveal information about their constraints, which can be sensitive. Therefore, in this paper, we introduce a method to keep linear constraints private when solving a convex program. First, we prove that this method is differentially private and always generates a feasible optimization problem (i.e., one whose solution exists). Then we show that the solution to the privatized problem also satisfies the original, non-private constraints. Next, we bound the expected loss in performance from privacy, which is measured by comparing the cost with privacy to that without privacy. Simulation results apply this framework to constrained policy synthesis in a Markov decision process, and they show that a typical privacy implementation induces only an approximately 9% loss in solution quality.**

***Index Terms*— Differential Privacy, Optimization, Markov Processes**

I. Introduction

CONVEX optimization problems with linear constraints appear in many applications, such as power grids [1], transportation systems [2], and resource allocation problems [3]. The constraints in such problems can be sensitive, e.g., the load flow in a power grid, the time to travel between locations, and the costs of resources, which may reveal information about individuals and/or trade secrets. The solutions to these problems are necessary for these systems to operate; however, simply computing and using these solutions may reveal the sensitive constraints used to generate them.

Interest has therefore arisen in solving these problems while both (i) preserving the privacy of constraints, and (ii) ensuring that all constraints are still satisfied at a solution. In this paper we address an open question posed in [4], namely, protecting the privacy of constraints, specifically the constraint coefficient matrix, while maintaining feasibility with respect to the original, non-private constraints. For linear constraints $Ax \leq b$, the work in [4] privatized b while ensuring constraint satisfaction, and [4] identified the privatization of A with guaranteed constraint satisfaction as an open problem. That is the problem we solve.

To provide privacy to these constraints, we use differential privacy. Differential privacy is a statistical notion of privacy originally developed to protect entries in databases [5]. We use it in this work partly because of its immunity to post-processing [6], namely that arbitrary computations on private data do not weaken differential privacy. Therefore, we first privatize each constraint in the constraint coefficient matrix, then solve the resulting optimization problem, which is simply a form of post-processing private data. Thus, the solution preserves the privacy of the constraints, as do any downstream computations that use it.

Some common privacy mechanisms, e.g., the Gaussian and Laplace mechanisms [6], add noise with unbounded support. Here, such mechanisms can perturb constraints by arbitrarily far amounts, which can cause a solution not to exist. Therefore, we use the truncated Laplace mechanism [4], [7], which allows us to privatize constraints such that they only become tighter. We use this property to ensure that (i) a solution always exists for a private problem, and (ii) the solution to the private problem satisfies the constraints of the original, non-private problem. Then, we bound the change in optimal cost due to privacy, which directly relates privacy to performance. To summarize, our contributions are:

- We develop a differential privacy mechanism for the coefficient matrix in linear constraints (Theorem 1).
- We prove that privatized problems have solutions that satisfy the constraints of their corresponding original, non-private problems (Theorem 2).
- We bound the expected change in optimal cost due to privatizing constraints (Theorem 3).
- We empirically validate the performance of this method on constrained Markov decision processes (Section V).

A. Related Work

There is a large literature on differential privacy in optimization, specifically looking at privacy for objective functions [8]–[13]. We differ from these approaches since privacy for the objective function does not affect feasibility, and thus a new approach must be developed to maintain feasibility under privacy of constraints. Privacy for linear programming, a

[1]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA USA. Emails: {abenvenuti3,matthale}@gatech.edu.

[2]Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL USA. Emails: {brendan.bialy,miriam.dennis.1}@us.af.mil.

This work was partially supported by AFRL under grant FA8651-23-F-A008, NSF under CAREER grant 1943275, ONR under grant N00014-21-1-2502, AFOSR under grant FA9550-19-1-0169, and the NSF Graduate Research Fellowship under grant DGE-2039655. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of sponsoring agencies.

special case of convex optimization with linear constraints, was addressed in [4], [14], [15]. While [14], [15] both privatize constraints, they allow for constraint violation, which may be unacceptable, e.g., if constraints encode safety.

B. Notation

For $N \in \mathbb{N}$, we use $[N] := \{1, 2, \ldots, N\}$. We use $\phi(B)$ to be the set of probability distributions over a finite set B, and $|\cdot|$ denotes the cardinality of a set. $\mathrm{Tr}\,(M)$ denotes the trace of a square matrix M.

II. Background and Problem Formulation

A. Convex Optimization with Linear Constraints

We consider optimization problems of the form

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & g(x) \\ \text{subject to} \quad & Ax \le b, \quad x \ge 0, \end{aligned} \tag{P}$$

where $g : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz and concave, $A \in \mathbb{R}^{m \times n}$ is the "constraint coefficient matrix", and $b \in \mathbb{R}^m$ is the "constraint vector". We use $\mathscr{A}$ to denote the set of all possible A matrices.

Assumption 1. The set $\mathscr{A}$ is bounded and the bounds are publicly available.

Assumption 1 is quite mild since the entries of A may represent physical quantities that do not exceed certain bounds, e.g., with voltages in a power grid. We consider the case in which all constraints require privacy, though this approach can be applied as-is to any subset of constraints.

Assumption 2. The feasible region $\{x \in \mathbb{R}^n : Ax \le b\}$ has non-empty interior for all $A \in \mathscr{A}$.

Remark 1. If Assumption 2 fails, then any perturbation to the constraints can cause infeasibility, making such constraints fundamentally incompatible with privacy.

It has been observed [4], [14], [15] that the public release of the solution to Problem (P) may reveal the A matrix used to generate it, and thus we apply differential privacy to A.

B. Differential Privacy

Overall, the goal of differential privacy is to make "similar" pieces of data appear approximately indistinguishable. The notion of "similar" is defined by an adjacency relation. Many adjacency relations exist, and we present the one used in the remainder of the paper; see [6] for additional background.

Definition 1 (Adjacency)**.** For a constant $k > 0$, two vectors $v, w \in \mathbb{R}^n$ are said to be adjacent if there exists an index $j \in [n]$ such that (i) $v_i = w_i$ for all $i \in [n] \backslash \{j\}$ and (ii) $\|v - w\|_1 \le k$. If two vectors v and w are adjacent, we write $\mathrm{Adj}_k(v, w) = 1$; otherwise we write $\mathrm{Adj}_k(v, w) = 0$.

To make adjacent pieces of data appear approximately indistinguishable, we implement differential privacy, which is done using a randomized map called a "mechanism". In its general form, differential privacy protects a sensitive piece of data x by randomizing some function of it, say $f(x)$. In this work, the sensitive data we consider is the matrix A, and we privatize the output of the identity map acting on A, which privatizes A itself. This is known as "input perturbation", and next we define differential privacy for this approach.

Definition 2 (Differential Privacy; [6])**.** Fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let $k > 0$, $\delta \in [0, \frac{1}{2})$, and $\epsilon > 0$ be given. A mechanism $\mathscr{M} : \mathbb{R}^m \times \Omega \to \mathbb{R}^m$ is (ϵ, δ)-differentially private if for all $v, w \in \mathbb{R}^m$ that are adjacent in the sense of Definition 1, we have $\mathbb{P}[\mathscr{M}(v) \in T] \le e^{\epsilon}\mathbb{P}[\mathscr{M}(w) \in T] + \delta$ for all Borel measurable sets $T \subseteq \mathbb{R}^m$.

The strength of privacy is set by ϵ and δ, and smaller values of both imply stronger privacy. The value of ϵ quantifies the leakage of sensitive information, and typical values for it are 0.1 to 10 [16]. The value of δ can be interpreted as the probability that more information is leaked than ϵ allows, and typical values for δ range from 0 to 0.05 [17].

Lemma 1 (Immunity to Post-Processing; [6])**.** Let $\mathscr{M} : \mathbb{R}^n \times \Omega \to \mathbb{R}^m$ be an (ϵ, δ)-differentially private mechanism. Let $h : \mathbb{R}^m \to \mathbb{R}^p$ be an arbitrary mapping. Then the composition $h \circ \mathscr{M} : \mathbb{R}^n \to \mathbb{R}^p$ is (ϵ, δ)-differentially private.

In the context of convex programming with linear constraints, Lemma 1 implies that we can privatize the constraint coefficient matrix A, and the solution to the privatized optimization problem preserves the privacy of A.

C. Problem Statements

Consider Problem (P) . Computing x^* depends on the sensitive constraint coefficient matrix A, and simply computing and using x^* can reveal information about A. Therefore, we aim to develop a framework for solving problems in the form of Problem (P) that preserves the privacy of A while still satisfying the constraints in Problem (P) . This will be done by solving the following problems.

Problem 1. *Develop a privacy mechanism to privatize the constraint coefficient matrix.*

Problem 2. *Prove that the privacy mechanism produces constraints such that the solution to the privately generated optimization problem also satisfies the constraints of the original, non-private problem.*

Problem 3. *Bound the average change in the cost between the non-private solution and the privatized solution.*

III. Private Linear Constraints

In this section, we solve Problems 1 and 2. Specifically, we (i) detail our approach to implementing privacy for the A matrix, (ii) prove that is it (ϵ, δ)-differentially private, and (iii) show that solutions computed with private constraints also satisfy the corresponding non-private constraints. Entries of A that equal zero may represent that there is physically no relationship between a decision variable and a constraint. For example, in a Markov decision process that models a traffic system, a zero transition probability may indicate that one street does not connect to another, which is publicly known. Thus, we will only privatize the non-zero entries of A.

A. Implementing Differential Privacy

For a given A matrix, we use a_i^0 to denote the vector of non-zero entries in row i, and we use $a_{i,j}^0$ to denote the j^{th} entry in that vector. To implement privacy we will compute $\tilde{a}_{i,j}^0 = a_{i,j}^0 + s_i + z_{i,j}$, where $z_{i,j}$ is privacy noise; we add s_i to tighten the constraints.

We will add bounded noise to ensure that $z_{i,j}$ only tightens the constraints when privatizing A; if the constraints are only tightened, then privacy can only shrink the feasible region, and thus satisfaction of the privatized constraints implies satisfaction of the original, non-private constraints. We do this with the truncated Laplace mechanism.

Lemma 2 (Truncated Laplace Mechanism; [4], [7])**.** Let $\epsilon > 0$ and $\delta \in (0, \frac{1}{2}]$ be given, and fix the adjacency relation from Definition 1. The Truncated Laplace Mechanism takes sensitive data $x \in \mathbb{R}^m$ as input and outputs private data $z \in \mathbb{R}^m$, where $z_i \in \mathcal{S}$ and $z_i \sim \mathcal{L}_T(\sigma, \mathcal{S})$ for all $i \in [m]$. Here, $\mathcal{L}_T(\sigma, \mathcal{S})$ is the truncated Laplace distribution with density $f(z_i) = \frac{1}{Z_i} \exp\left(-\frac{1}{\sigma}|z_i|\right)$, where $\mathcal{S} := [-s, s]$ and the values of s and $-s$ are bounds on the support of the private outputs. We set $Z_i = \mathbb{P}(z_i \leq |s|)$. Then the truncated Laplace mechanism is (ϵ, δ)-differentially private if $\sigma \geq \frac{k}{\epsilon}$ and $s = \frac{k}{\epsilon} \log\left(\frac{m(e^{\epsilon}-1)}{\delta} + 1\right)$.

We apply this mechanism to each row of a constraint coefficient matrix A, which provides row-wise privacy. This approach in fact provides privacy to the entire A matrix.

Lemma 3 (Parallel Composition; [18])**.** Consider a database D partitioned into disjoint subsets $D_1, D_2, \ldots, D_N$, and suppose that there are privacy mechanisms $\mathscr{M}_1, \mathscr{M}_2, \ldots, \mathscr{M}_N$, where $\mathscr{M}_i$ is (ϵ, δ)-differentially private for all $i \in [N]$. Then the release of the queries $\mathscr{M}_1(D_1), \mathscr{M}_2(D_2), \ldots, \mathscr{M}_N(D_N)$ provides D with (ϵ, δ)-differential privacy.

We consider A and partition it into its rows, then privatize each row individually. Lemma 3 ensures that doing so provides (ϵ, δ)-differential privacy to the A matrix as a whole.

Along with privacy, we must also enforce feasibility. In order to guarantee that the privately obtained solution $\tilde{x}^*$ is feasible with respect to the non-private problem, it is clear that the two problems must have at least one feasible point in common. We state this formally in the following assumption.

Assumption 3 (Perturbed Feasibility)**.** The set $S = \bigcap_{A \subseteq \mathscr{A}} \{x : Ax \leq b\}$ is not empty.

In words, Assumption 3 says that there exists at least one point that satisfies the constraints produced by every realization of the constraint coefficient matrix A. With Assumption 3, we post-process $\tilde{a}_{i,j}^0 = a_{i,j}^0 + s_i + z_{i,j}$ to obtain $\bar{a}_{i,j}^0 = \min\left\{\tilde{a}_{i,j}^0, \sup_{\mathscr{A}} a_{i,j}^0\right\}$, and we do so for each (i, j) such that $a_{i,j}$ is non-zero. The output of these computations is the private constraint coefficient matrix, denoted $\tilde{A}$.

Remark 2. Taking the minimum in $\bar{a}_{i,j}^0$ ensures that each entry in $\tilde{A}$ appears in some matrix in $\mathscr{A}$. The supremum is finite since $\mathscr{A}$ is bounded and does not depend on sensitive information according to Assumption 1.

With this, we solve the optimization problem

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & g(x) \\ \text{subject to} \quad & \tilde{A}x \leq b, \quad x \geq 0. \end{aligned} \tag{DP-P}$$

Algorithm 1 provides a unified overview of our approach.

Algorithm 1: Privately Solving Convex Problems with Linear Constraints

```
Inputs: Problem (P), ϵ, δ, k;
Outputs: Privacy-preserving solution x̃*;
Set σ = k/ϵ;
for all i ∈ [m] do
    Count the non-zero entries in row i, namely n_i^0;
    Compute the support for the truncated Laplace
      mechanism, i.e., s_i = (k/ϵ) log( n_i^0(e^ϵ − 1)/δ + 1 );
    for all j ∈ [n] do
        Set ā_{i,j}^0 = min{ a_{i,j}^0 + s_i + z_{i,j}, sup_𝒜 a_{i,j}^0 };
    end
end
Form Ã by replacing each non-zero entry a_{i,j}^0 in A
  with ā_{i,j}^0;
Solve Problem (DP-P) (via any algorithm) to find x̃*
```

B. Characterizing Privacy

Next we prove that Algorithm 1 preserves the privacy of A. To do so, we show how Lemma 2 can be extended using Lemma 3 to develop a new privacy mechanism that generates a random matrix in which each entry is a truncated Laplace random variable.

Theorem 1 (Solution to Problem 1)**.** *Fix an adjacency parameter $k > 0$, let privacy parameters $\epsilon > 0$ and $\delta \in [0, \frac{1}{2})$ be given, and let Assumptions 1-3 hold. Then Algorithm 1 keeps A (ϵ, δ)-differentially private with respect to the adjacency relation in Definition 1.*

Proof. See Appendix A. □

Theorem 1 allows us to privatize each constraint individually, and the resulting constraint coefficient matrix $\tilde{A}$ will be (ϵ, δ)-differentially private. Solving Problem (DP-P) then preserves the privacy of A, and the solution $\tilde{x}^*$ can be released and acted on without harming privacy.

Theorem 2 (Solution to Problem 2)**.** *Fix an adjacency parameter $k > 0$, let privacy parameters $\epsilon > 0$ and $\delta \in [0, \frac{1}{2})$ be given, and let Assumptions 1-3 hold. Fix a constant vector b. Then the Problem* (DP-P) *produces a solution that also satisfies the constraints in Problem* (P).

Proof. See Appendix B. □

Theorem 2 guarantees that after privacy is implemented, the resulting problem is feasible. Since Algorithm 1 only tightens the constraints, this implies that a privatized solution $\tilde{x}^*$ always exists and is in the feasible set of the original, non-private problem. Previous works that sought to provide privacy to

$$\xi = \begin{cases} \sqrt{\sum_{j=1}^{m}\left(2m\left(\frac{k}{\epsilon}\right)^2 n_j^0 + \left(\frac{n_j^0 k}{\epsilon}\log\left(\frac{n_j^0(e^{\epsilon}-1)}{\delta}+1\right)\right)^2\right)} & \text{if } a_{i,j}+2s_i < \sup_{\mathscr{A}} a_{i,j} \text{ for all } i \text{ and } j \\ \left\|A-\bar{A}\right\|_F & \text{if there exists } i \text{ and } j \text{ s.t. } a_{i,j}+2s_i \geq \sup_{\mathscr{A}} a_{i,j}. \end{cases} \quad (1)$$

the A matrix, namely [14], [15], cannot guarantee constraint satisfaction. Thus, Theorem 2 sets Algorithm 1 apart as the only approach to the authors' knowledge that can guarantee feasibility under privacy for the A matrix.

IV. Solution Accuracy

In this section, we solve Problem 3. To do so, we compute an upper bound on the change in cost between the nominally generated solution and the privately generated solution. This bound depends on (i) the Lipschitz constant of the objective function, (ii) the largest feasible solution of the original, non-private constraint coefficient matrix, (iii) the largest possible constraint coefficients allowable from $\mathscr{A}$, and (iv) the "closeness" of the private and non-private feasible regions.

For (i), we state this as an assumption below.

Assumption 4. The objective function $g : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz with respect to the ℓ_2-norm on $\mathbb{R}^n$.

For (ii), we define

$$\bar{A} = [\sup_{\mathscr{A}} a_{i,j}]_{i\in[m],j\in[n]} \quad (2)$$

as the matrix where each entry is its largest value in the set of all constraint coefficient matrices $\mathscr{A}$. For (iii), we define

$$F(A) = \{x \in \mathbb{R}^n : Ax \leq b\} \quad (3)$$

as the feasible region of the original, non-private constraints given a choice of A. Then we define

$$\bar{x} \in \arg\max_{x\in F(A)} \|x\|_2, \quad (4)$$

which is an element in the feasible region of the original, non-private constraints which yields the largest 2-norm.

For (iv), the "closeness" of sets of linear inequalities have a bounded difference in their feasible regions.

Lemma 4 (Perturbation Bound; [19])**.** *Given $Ax \leq b$ with $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$, consider $F(A)$ from (3). For a matrix $\hat{A} \in \mathbb{R}^{m\times n}$ and vector $\hat{b} \in \mathbb{R}^m$, let $\hat{x}$ be such that $\hat{A}\hat{x} \leq \hat{b}$. Then there exists an $x \in F(A)$ such that $\|x-\hat{x}\|_2 \leq H_{2,2}(A)\left\|[(A-\hat{A})\hat{x}-(b-\hat{b})]^+\right\|_2$, where $[\cdot]^+$ is the projection onto the non-negative orthant of $\mathbb{R}^m$, and $H_{2,2}(A)$ is the Hoffman constant of A, i.e.,*

$$H_{2,2}(A) = \max_{J\in\mathcal{J}(A)} \frac{1}{\min\left\{\|A_J^T v\|_2 : v \in \mathbb{R}_+^J, \|v\|_2 = 1\right\}},$$

where (i) $\mathcal{J}(A) = \{J \subseteq [m] : \mathscr{A}_J(\mathbb{R}^n) + \mathbb{R}_+^{|J|} = \mathbb{R}^{|J|}\}$ is the set of all J such that the set-valued mapping $x \to A_J x + \mathbb{R}_+^{|J|}$ is surjective, (ii) A_J is the matrix formed by deleting all rows of A whose indices are not in J, and (iii) we define $\mathscr{A}_J(\mathbb{R}^n) = \{z \in \mathbb{R}^n : A_J z \leq b_J\}$, where b_J is formed by deleting entries of b whose indices are not in J.

Next, we bound the expected change in cost.

Theorem 3 (Solution to Problem 3)**.** *Fix an adjacency parameter $k > 0$, fix privacy parameters $\epsilon > 0$ and $\delta \in [0, \frac{1}{2})$, and let Assumptions 1-4 hold. Let x^* solve Problem* (P) *and $\tilde{x}^*$ solve Problem* (DP-P)*. Then $\mathbb{E}\left[|g(x^*) - g(\tilde{x}^*)|\right] \leq L\|\bar{x}\|_2 H_{2,2}(A)\xi$, where ξ is from* (1)*, $H_{2,2}(A)$ is from Lemma 4, $\bar{A}$ is from* (2)*, and $\bar{x}$ is from* (4)*.*

Proof. See Appendix C. □

Remark 3. In the range of ϵ values where $a_{i,j} + 2s_i < \sup_{\mathscr{A}} a_{i,j}$ for all i and j, increasing the strength of privacy (i.e., decreasing ϵ) will lead to a larger sub-optimality gap, indicated by increasing ξ. As privacy strengthens further, we eventually reach the worse-case scenario where $\tilde{A} = \bar{A}$, where we see ξ take on its maximum value, and we see no change in sub-optimality for increasing privacy's strength beyond that point. In terms of scalability, the bound on the sub-optimality gap grows with $\mathcal{O}(n^{3/2}\sqrt{m})$, implying that our algorithm will still perform well at scale.

Theorem 3 presents a tradeoff between the suboptimality gap and level of privacy (i.e., ϵ and δ). Users may utilize such a tradeoff to design their privacy parameters based on the worst-case loss in optimality.

Remark 4. The concavity of g guarantees the uniqueness of the optimal cost, which allows for the computation of the suboptimality bound. Our method keeps constraints private regardless of the concavity of the cost function, however we leave analysis of sub-optimality under non-concavity as an avenue for future work.

V. Application to Policy Synthesis

This section applies our developments to constrained Markov decision processes, which we define next.

Definition 3 (Constrained Markov Decision Process; [20])**.** A Constrained Markov Decision Process (CMDP) is the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, \mathcal{T}, \mu, f, f_0)$, where $\mathcal{S}$ and $\mathcal{A}$ are the finite sets of states and actions, respectively, and $|\mathcal{S}| = p$ and $|\mathcal{A}| = q$. Then, $r : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ is the reward function, $\mathcal{T} : \mathcal{S}\times\mathcal{A} \to \phi(\mathcal{S})$ is the transition probability function, $\mu \in \phi(s)$ is a probability distribution over the initial states, $f_i : \mathcal{S}\times\mathcal{A} \to [0, f_{\max,i}]$ for $i \in [N]$ are immediate costs, and $\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t f(s_t)\right] \leq f_0$, $f_0 \in \mathbb{R}^N$ are constraints.

We let $\mathcal{T}(s, \alpha, y)$ denote the probability of transitioning from state s to state y when taking action α. We consider CMDPs in which the constraint function f is linear. Then the constraints can be written as $AX - f_0 \leq 0$, where $A \in \mathbb{R}^{pq\times N}$, where p, q, and f_0 are from Definition 3 and X is the decision variable in policy synthesis, described next.

Solving an MDP, i.e., computing the optimal policy, or list of commands to take in each state, can be done with the linear program [21]

$$\begin{aligned}&\underset{x_\pi}{\text{maximize}} \quad \sum_{s\in S}\sum_{\alpha\in A} r(s,\alpha)x_\pi(s,\alpha)\\ &\text{s.t.} \quad x_\pi(s,\alpha)\geq 0, \quad f(x_\pi(s,\alpha))\leq f_0 \quad \forall s\in\mathcal{S},\alpha\in\mathcal{A},\\ &\sum_{\alpha'\in\mathcal{A}} x_\pi(s',\alpha') - \gamma\sum_{s\in\mathcal{S}}\sum_{\alpha\in\mathcal{A}} x_\pi(s,\alpha)\mathcal{T}(s,\alpha,s') = \mu(s') \ \forall s'\in\mathcal{S}.\end{aligned}$$

The optimal policy π^* can be calculated from the optimum $\{x^*_\pi(s,\alpha)\}_{s\in\mathcal{S},\alpha\in\mathcal{A}}$ via $\pi^*(\alpha \mid s) = \frac{x^*_\pi(s,\alpha)}{\sum_{\alpha'\in\mathcal{A}} x^*_\pi(s,\alpha')}$. Such a policy admits a value function v_π, which is defined as $v_\pi(s_0) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,\pi(s_t))\right]$, and is easily computable [21]. Constraints that may be encoded by A include enforcing a probability of reaching a goal state and safety, i.e., setting a maximum amount of visits to a set of hazardous states, with hazard values assigned to each state.

Remark 5. In [22], the authors privatize $r(s,a)$, which is equivalent to privatizing the objective function in the linear programming approach described above. As noted in Section I-A, such an approach need not be concerned with feasibility, and therefore the approach in [22] is insufficient to preserve the privacy of constraints and ensure feasibility.

This type of safety extends the example in [23] by allowing states to have varying "hazard" factors, which are the immediate costs for each state-action pair. Let $\mathcal{S}_H \subseteq \mathcal{S}$ be the set of hazardous states, and let $f(s_t) = \beta_s\mathbb{I}\{s_t \in \mathcal{S}_H\}$, where $\mathbb{I}$ is an indicator function, which encodes that the agent incurs a penalty of β_s for occupying state s. We then have the constraint that $\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\beta_{s_t}f(s_t)|s_0,\pi\right] \leq f_0$, or, equivalently, $f(x_\pi(s,\alpha)) = \beta_{s_t}\gamma\mathbb{I}\{x_\pi(s,\alpha) \in \mathcal{S}_H\} \leq f_0$. This takes the form $AX - f_0 \leq 0$, where A is a row vector with

$$a_i = \begin{cases}\beta_{s_t}\gamma & \text{if } x(s_t,\alpha)\in\mathcal{S}_H\\ 0 & \text{otherwise}\end{cases}.$$

These are the constraints that we will privatize. We will also empirically evaluate the "cost of privacy" using the metric in [22], [24], namely the percent decrease in the value function, equal to $\left((v_{\pi^*}(s_0) - v_{\tilde{\pi}^*}(s_0))/v_{\pi^*}(s_0)\right)\times 100\%$.

We apply Algorithm 1 to a CMDP representing the system in Figure 1, and we privatize the hazard values of each hazardous state. We set $\beta_i = 1$ for all $i\in\mathcal{S}_H$, and we define $\mathscr{A}$ so that $\sup_{\mathscr{A}} a_{i,j} = 3$. The cost of privacy for $\epsilon\in[0.01,5]$ and $\delta\in\{10^{-1},10^{-2},10^{-3}\}$ averaged over 100 samples is shown in Figure 2. With strong privacy, i.e., $\epsilon = 2$, $\delta = 10^{-2}$, we see a 18.35% reduction in performance, while with more typical privacy levels, i.e., $\epsilon = 3$, $\delta = 10^{-2}$, we see only a 9.45% reduction in performance, indicating that our method can simultaneously provide both desirable privacy protections and desirable levels of performance.

VI. Conclusion

We presented a differentially private method for keeping linear constraints private while ensuring that constraints are never violated in convex optimization. Future work will address the privatization of general convex, nonlinear, and stochastic constraints while ensuring their satisfaction as well. Additionally, future work will also consider simultaneous privatization of A and b, as well as the cost term c.

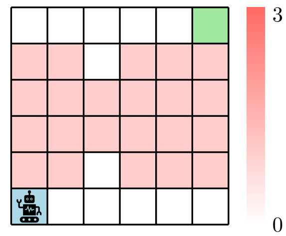

Fig. 1. Grid in which the agent starts at the blue state, its goal is the green state, and hazardous states are red.

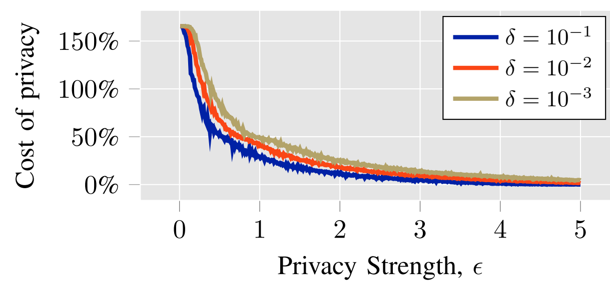

Fig. 2. Cost of privacy for the example shown in Figure 1 for a range of values of ϵ implemented with Algorithm 1 at various values of δ.

References

[1] B. Stott, J. Marinho, and O. Alsac, "Review of linear programming applied to power system rescheduling," in *Proceedings of the Power Industry Computer Applications Conference*, 1979, pp. 142–154.

[2] A. J. Hoffman, "On simple linear programming problems," in *Proceedings of Symposia in Pure Mathematics*, vol. 7, 1963, pp. 317–327.

[3] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[4] A. Munoz, U. Syed, S. Vassilvtiskii, and E. Vitercik, "Private optimization without constraint violations," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2557–2565.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[6] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[7] Q. Geng, W. Ding, R. Guo, and S. Kumar, "Tight analysis of privacy and utility tradeoff in approximate differential privacy," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 89–99.

[8] Y. Wang, M. Hale, M. Egerstedt, and G. E. Dullerud, "Differentially private objective functions in distributed cloud-based optimization," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 3688–3694.

[9] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proceedings of the 16th International Conference on Distributed Computing and Networking*, 2015, pp. 1–10.

[10] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 50–64, 2016.

[11] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via objective perturbation," in *2016 American control conference (ACC)*. IEEE, 2016, pp. 2061–2066.

[12] R. Dobbe, Y. Pu, J. Zhu, K. Ramchandran, and C. Tomlin, "Customized local differential privacy for multi-agent distributed optimization," *arXiv preprint arXiv:1806.06035*, 2018.

[13] Y.-W. Lv, G.-H. Yang, and C.-X. Shi, “Differentially private distributed optimization for multi-agent systems via the augmented lagrangian algorithm,” *Information Sciences*, vol. 538, pp. 39–53, 2020.
[14] J. Hsu, A. Roth, T. Roughgarden, and J. Ullman, “Privately solving linear programs,” in *Automata, Languages, and Programming: 41st International Colloquium*. Springer, 2014, pp. 612–624.
[15] R. Cummings, M. Kearns, A. Roth, and Z. S. Wu, “Privacy and truthful equilibrium selection for aggregative games,” in *Web and Internet Economics: 11th International Conference*, 2015, pp. 286–299.
[16] J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, B. C. Pierce, and A. Roth, “Differential privacy: An economic method for choosing epsilon,” in *2014 IEEE 27th Computer Security Foundations Symposium*. IEEE, 2014, pp. 398–410.
[17] C. Hawkins, B. Chen, K. Yazdani, and M. Hale, “Node and edge differential privacy for graph laplacian spectra: Mechanisms and scaling laws,” *IEEE Transactions on Network Science and Engineering*, 2023.
[18] N. Ponomareva *et al.*, “How to dp-fy ml: A practical guide to machine learning with differential privacy,” *Journal of Artificial Intelligence Research*, vol. 77, pp. 1113–1201, 2023.
[19] A. J. Hoffman, “On approximate solutions of systems of linear inequalities,” *Journal of Research of the National Bureau of Standards*, vol. 49, no. 4, 1952.
[20] E. Altman, *Constrained Markov decision processes*. Routledge, 2021.
[21] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
[22] A. Benvenuti, C. Hawkins, B. Fallin, B. Chen, B. Bialy, M. Dennis, and M. Hale, “Differentially private reward functions for markov decision processes,” in *2024 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2024, pp. 631–636.
[23] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, “A lyapunov-based approach to safe reinforcement learning,” *Advances in neural information processing systems*, vol. 31, 2018.
[24] P. Gohari, M. Hale, and U. Topcu, “Privacy-preserving policy synthesis in markov decision processes,” in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 6266–6271.

APPENDIX

A. Proof of Theorem 1

From Lemma 2, computing $\tilde{a}^0_{i,j} = a^0_{i,j} + s_i + z_{i,j}$ is (ϵ, δ)-differentially private if $z_{i,j} \sim \mathcal{L}_T(\frac{\Delta}{\epsilon}, \mathcal{S}_i)$, where $\mathcal{S}_i = [-s_i, s_i]$ and $s_i = \frac{k}{\epsilon}\log\left(\frac{n^0_i(e^\epsilon - 1)}{\delta} + 1\right)$. Additionally, computing $\bar{a}^0_{i,j} = \min\{\tilde{a}^0_{i,j}, \sup_{\mathscr{A}} a^0_{i,j}\}$ is merely post-processing of differentially private data, and thus by Lemma 1 it maintains (ϵ, δ)-differential privacy. Repeating this for each $i \in [m]$, each vector of non-zero constraint coefficients is (ϵ, δ)-differentially private, and from Lemma 3, replacing the non-zero elements in A with $\bar{a}_{i,j}$ to form $\tilde{A}$ is (ϵ, δ)-differentially private. From Lemma 1, it then follows that $\tilde{x}^*$ is (ϵ, δ)-differentially private by virtue of being post-processing of the differentially private quantity $\tilde{A}$.

B. Proof of Theorem 2

By definition, the constraint matrix $\tilde{A}$ is component-wise less than or equal to the matrix $\bar{A}$ in which $\bar{A}_{i,j} = \sup_{\mathscr{A}} a_{i,j}$ for all $i \in [n]$ and $j \in [m]$. Since $x \geq 0$ and the vector b is fixed, we have that the set $\{x : \bar{A}x \leq b\}$ is contained in $\{x : \tilde{A}x \leq b\}$ due to the fact that $a_{i,j} \leq \sup_{\mathscr{A}} a_{i,j}$, and as a result, we know that $\{x : \tilde{A}x \leq b\} \supseteq \{x : \bar{A}x \leq b\}$.

We will now show that the sets $\bigcap_{A\in\mathscr{A}}\{x : Ax \leq b\}$ and $\{x : \bar{A}x \leq b\}$ are equal. For any x in the first set, we know that $a_i \cdot x \leq b_i$ for all $A \in \mathscr{A}$. By definition of the supremum, it follows then that $\sup_{\mathscr{A}}(a_i \cdot x) \leq b_i$, and therefore $\bar{A}x \leq b$. As a result, if $x \in \bigcap_{A\in\mathscr{A}}\{z : Az \leq b\}$, then $x \in \{z : \bar{A}z \leq b\}$. We now show that the reverse is true. If $\bar{A}x \leq b$, then $Ax \leq b$ for all $A \in \mathscr{A}$ by definition of the supremum. Thus, if $x \in \{z : \bar{A}z \leq b\}$, then we also have $x \in \bigcap_{A\in\mathscr{A}}\{z : Az \leq b\}$.

Since we have $\{x : \tilde{A}x \leq b\} \supseteq \{x : \bar{A}x \leq b\}$ and $\{x : \bar{A}x \leq b\} = \bigcap_{A\in\mathscr{A}}\{x : Ax \leq b\}$, we know that $\{x : \tilde{A}x \leq b\} \supseteq \bigcap_{A\in\mathscr{A}}\{x : Ax \leq b\}$. From Assumption 3, the set $\bigcap_{A\in\mathscr{A}}\{x : Ax \leq b\}$ is non-empty, and therefore the set $\{x : \tilde{A}x \leq b\}$ is non-empty and thus is yields a feasible optimization problem.

C. Proof of Theorem 3

The Lipschitz property of g from Assumption 4 gives

$$\mathbb{E}\left[|g(x^*) - g(\tilde{x}^*)|\right] \leq L\mathbb{E}\left[\|x^* - \tilde{x}^*\|\right]. \tag{5}$$

Noting that b remains constant between the feasible regions $\{x : Ax \leq b\}$ and $\{x : \tilde{A}x \leq b\}$, we apply Lemma 4 and the linearity of the expectation to obtain to obtain

$$\mathbb{E}\left[\|x^* - \tilde{x}^*\|_2\right] \leq H_{2,2}(A)\mathbb{E}\left[\left\|[A - \tilde{A}]^+\right\|_F \|\tilde{x}^*\|_2\right]. \tag{6}$$

Since $\tilde{x}$ is in a subset of the feasible space in the non-private problem, the largest possible $\tilde{x}$ is bounded by the largest feasible $x \in F(A)$, where $F(A)$ is from (3). We denote this as $\bar{x} \in \arg\max_{x\in F(A)} \|x\|_2$. Then in (6) we may write the bound $\mathbb{E}\left[\left\|[A - \tilde{A}]^+\right\|_F \|\tilde{x}^*\|_2\right] \leq \mathbb{E}\left[\left\|[A - \tilde{A}]^+\right\|_F\right] \|\bar{x}\|_2$. Next we bound $\left\|[A - \tilde{A}]^+\right\|_F$. First, we define $Z = \tilde{A} - A$ and, using the non-expansive property of the projection onto the non-negative orthant, we obtain

$$\mathbb{E}\left[\left\|[A - \tilde{A}]^+\right\|_F\right] \|\bar{x}\|_2 \leq \mathbb{E}\left[\|Z\|\right] \|\bar{x}\|_2. \tag{7}$$

Using the definition of the Frobenius norm and Jensen's inequality, we use the preceding inequality to find the bound $H_{2,2}(A)\|\bar{x}\|_2 \mathbb{E}\left[\|Z\|\right] \leq H_{2,2}(A)\|\bar{x}\|_2 \sqrt{\mathrm{Tr}\left(\mathbb{E}\left[Z^TZ\right]\right)}$. Now we compute the diagonal entries of $\mathbb{E}\left[Z^TZ\right]$. We break Z down into two cases: the case where there exists an i and j such that $a_{i,j} + 2s_i \geq \sup_{\mathscr{A}} a_{i,j}$, and the case where $a_{i,j} + 2s_i < \sup_{\mathscr{A}} a_{i,j}$ for all i and j. Starting with the case $a_{i,j} + 2s_i < \sup_{\mathscr{A}} a_{i,j}$ for all i and j, we have

$$\begin{aligned}(Z^TZ)_{1,1} &= (s_1 + \eta_{1,1})(s_1 + \eta_{1,1}) + (s_2 + \eta_{2,1})(s_2 + \eta_{2,1}) \\ &+ \cdots + (s_m + \eta_{m,1})(s_m + \eta_{m,1}) = s_1^2 + 2s_1\eta_{1,1} + \eta_{1,1}^2 \\ &+ s_2^2 + 2s_2\eta_{2,1} + \eta_{2,1}^2 + \cdots + s_m^2 + 2s_m\eta_{m,1} + \eta_{m,1}^2.\end{aligned}$$

Each η_i^j has mean 0 and second moment $2\left(\frac{k}{\epsilon}\right)^2$. Thus, $\mathbb{E}\left[(Z^TZ)_{1,1}\right] = 2n_1^0\left(\frac{k}{\epsilon}\right)^2 + \sum_{i=1}^{n_1^0} s_i^2$. This pattern holds for each diagonal entry, so we have $\mathbb{E}\left[(Z^TZ)_{i,i}\right] = 2n_i^0\left(\frac{k}{\epsilon}\right)^2 + \sum_{i=1}^{n_i^0} s_i^2$. The trace is then the sum of these diagonal entries, and because $Z^TZ \in \mathbb{R}^{m\times m}$, there are m diagonal entries, and thus we may write the equality $\sqrt{\mathrm{Tr}\left(\mathbb{E}\left[Z^TZ\right]\right)} = \sqrt{\sum_{j=1}^m (2m\left(k/\epsilon\right)^2 n_j^0 + n_j^0 s_j^2)}$, where s_i is the support of the truncated Laplace mechanism for constraint i. In the event that there exists an i and j such that $a_{i,j} + 2s_i \geq \sup_{\mathscr{A}} a_{i,j}$, define $\bar{A} = [\sup_{\mathscr{A}} a_{i,j}]_{i\in[m], j\in[n]}$, that is, the matrix such that every entry in $\bar{A}$ takes the maximum possible value allowed by the set $\mathscr{A}$. In this case, $Z_{ij} \leq (\bar{A} - A)_{ij}$, and we have $H_{2,2}(A)\|\bar{x}\|_2 \sqrt{\mathrm{Tr}\left(\mathbb{E}\left[Z^TZ\right]\right)} \leq H_{2,2}(A)\|\bar{x}\|_2 \left\|A - \bar{A}\right\|_F$.

Defining ξ in (1), we substitute ξ into (7) to find $H_{2,2}(A)\left\|\bar{x}\right\|_2 \mathbb{E}\left[\left\|Z\right\|_F\right] \leq H_{2,2}(A)\left\|\bar{x}\right\|_2 \xi$, which we substitute into (5) to obtain the result.