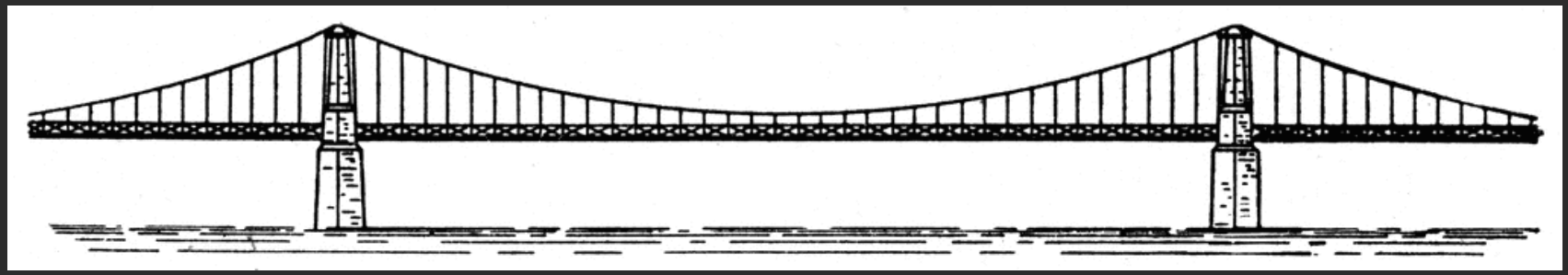
Towards a Theory AB Toolbox

Verifying Randomized Algorithms



Marco Gaboardi[1] and Justin Hsu[2]

[1]University of Dundee
[2]University of Pennsylvania

May 6th, 2015

A story

Alice wants to protect privacy

A story

Alice wants to protect privacy

Jointly Private Convex Programming

[redacted]

November 6, 2014

Abstract

In this paper we present an extremely general method for approximately solving a large family of convex programs where the solution can be divided between different agents, subject to joint differential privacy. This class includes multi-commodity flow problems, general allocation problems, and multi-dimensional knapsack problems, among other examples. The accuracy of our algorithm depends on the *number* of constraints that bind between individuals, but crucially, is *nearly independent* of the number of primal variables and hence the number of agents who make up the problem. As the number of agents in a problem grows, the error we introduce often becomes negligible.

We also consider the setting where agents are strategic and have preferences over their part of the solution. For any convex program in this class that maximizes *social welfare*, there is a generic reduction that makes the corresponding optimization *approximately dominant strategy truthful* by charging agents prices for resources as a function of the approximately optimal dual

Complex code

Algorithm 1 Joint Differentially Private Convex Solver: PrivDude$(\mathcal{O}, \sigma, \tau, w, \varepsilon, \delta, \beta)$

Input: Convex problem $\mathcal{O} = (S, v, c, b)$ with n agents and k coupling constraints, gradient sensitivity bounded by σ, a dual bound τ, width bounded by w, and privacy parameters $\varepsilon > 0, \delta \in (0,1)$, confidence parameter $\beta \in (0,1)$.

Initialize:

$$\lambda_j^{(1)} := 0 \text{ for } j \in [k], \qquad T := w^2, \qquad \varepsilon' := \frac{\varepsilon\sigma}{\sqrt{8T\ln(2/\delta)}}, \qquad \delta' := \frac{\delta}{2T},$$

$$\eta := \frac{2\tau}{\sqrt{T}\left(w + \frac{1}{\varepsilon'}\log\left(\frac{Tk}{\beta}\right)\right)}, \qquad \Lambda := \{\lambda \in \mathbb{R}_+^k \mid \|\lambda\|_\infty \le 2\tau\}.$$

for iteration $t = 1 \dots T$

  for each agent $i = 0 \dots n$

    Compute personal best response:

$$x_t^{(i)} := \operatorname*{argmax}_{x \in S^{(i)}} v^{(i)}(x) - \sum_{j=1}^{k} \lambda_j^{(t)} c^{(i)}(x).$$

  for each constraint $j = 1 \dots k$

    Compute noisy gradient:

$$\widehat{\ell}_j^{(t)} := \left(\sum_{i=1}^{n} c^{(i)}(x_t^{(i)})\right) - b_j + \mathcal{N}\left(0, \frac{2\sigma^2 \log(1.25/\delta')}{\varepsilon'^2}\right).$$

Complex proofs

Proof. Let ν_t denote the noise vector we have in round t, we can decompose the regret into several parts

$$\begin{aligned}
\mathcal{R}_T &= \frac{1}{T}\sum_{t=1}^{T}\langle p_t, x_t\rangle - \frac{1}{T}\min_{p\in\mathcal{P}}\sum_{t=1}^{T}\langle p, x_t\rangle \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle p_t, \widehat{x}_t\rangle - \frac{1}{T}\sum_{t=1}^{T}\langle p_t, \nu_t\rangle - \frac{1}{T}\left[\min_{p\in\mathcal{P}}\sum_{t=1}^{T}\langle p, x_t\rangle - \min_{\widehat{p}\in\mathcal{P}}\sum_{t=1}^{T}\langle \widehat{p}, \widehat{x}_t\rangle\right] - \frac{1}{T}\min_{\widehat{p}\in\mathcal{P}}\sum_{t=1}^{T}\langle \widehat{p}, \widehat{x}_t\rangle \\
&= \left[\frac{1}{T}\sum_{t=1}^{T}\langle p_t, \widehat{x}_t\rangle - \frac{1}{T}\min_{\widehat{p}\in\mathcal{P}}\sum_{t=1}^{T}\langle \widehat{p}, \widehat{x}_t\rangle\right] - \frac{1}{T}\sum_{t=1}^{T}\langle p_t, \nu_t\rangle - \frac{1}{T}\left[\min_{p\in\mathcal{P}}\sum_{t=1}^{T}\langle p, x_t\rangle - \min_{\widehat{p}\in\mathcal{P}}\sum_{t=1}^{T}\langle \widehat{p}, \widehat{x}_t\rangle\right] \\
&= \widehat{\mathcal{R}}_T - \frac{1}{T}\sum_{t=1}^{T}\langle p_t, \nu_t\rangle - \frac{1}{T}\left[\min_{p\in\mathcal{P}}\sum_{t=1}^{T}\langle p, x_t\rangle - \min_{\widehat{p}\in\mathcal{P}}\sum_{t=1}^{T}\langle \widehat{p}, \widehat{x}_t\rangle\right] \\
&\leq \widehat{\mathcal{R}}_T - \frac{1}{T}\min_{p\in\mathcal{P}}\sum_{t=1}^{T}\langle p, \nu_t\rangle - \frac{1}{T}\left[\min_{p\in\mathcal{P}}\sum_{t=1}^{T}\langle p, x_t\rangle - \min_{\widehat{p}\in\mathcal{P}}\sum_{t=1}^{T}\langle \widehat{p}, \widehat{x}_t\rangle\right].
\end{aligned}$$

We will bound the three terms separately. By the no-regret guarantee of online gradient descent in Lemma 13, we have the following the regret guarantee w.r.t the noisy losses if we set $\eta = \frac{\|\mathcal{P}\|}{\sqrt{T}\|\widehat{\mathcal{X}}\|}$

$$\widehat{\mathcal{R}}_T = \frac{1}{T}\sum_{t=1}^{T}\langle p_t, \widehat{x}_t\rangle - \min_{p\in\mathcal{P}}\frac{1}{T}\sum_{t=1}^{T}\langle p, \widehat{x}_t\rangle \leq \frac{\|\mathcal{P}\|^2}{2\eta T} + \frac{\eta\|\widehat{\mathcal{X}}\|^2}{2} = \frac{\|\mathcal{P}\|\|\widehat{\mathcal{X}}\|}{\sqrt{T}},$$

where $\|\mathcal{P}\|$ and $\|\widehat{\mathcal{X}}\|$ denote the bound on the ℓ_2 norm of the vectors $\{p_t\}$ and $\{\widehat{x}_t\}$ respectively.

Current practice

Paper proofs

- Produced by humans
- Major steps included
- Minor steps skipped

"Morally correct"

- Complex proofs checked by humans
- Sometimes bugs

Challenges in formalizing proofs

Complex properties

- Single run/multiple runs/???
- Quantitative: measure how performance scales with input

Challenges in formalizing proofs

Complex properties

- Single run/multiple runs/???
- Quantitative: measure how performance scales with input

Diverse proofs

- Variety of tools and proof structures, non-local reasoning
- Proof about a single program can be research contribution

Challenges in formalizing proofs

Complex properties

- Single run/multiple runs/???
- Quantitative: measure how performance scales with input

Diverse proofs

- Variety of tools and proof structures, non-local reasoning
- Proof about a single program can be research contribution

Probability theory

- Probabilities of events, expected values
- Very rich theory, too much to formalize

The overall idea

Imitate paper proofs

Bring patterns, abstractions, notations to formal verification

What's so great about paper proofs?

Probability theory: just the good parts

- Use useful properties and abstractions
- Avoid low-level probability theory

What's so great about paper proofs?

Probability theory: just the good parts

- Use useful properties and abstractions
- Avoid low-level probability theory

Concise, light reasoning

- Useful notations and high-level reasoning
- Major steps are evident, not buried in boilerplate
- Powerful patterns to structure proofs

Pattern: The union bound

$$\Pr[E_1 \vee \cdots \vee E_n] \leq \Pr[E_1] + \cdots + \Pr[E_n]$$

Pattern: The union bound

$$\Pr[E_1 \vee \cdots \vee E_n] \leq \Pr[E_1] + \cdots + \Pr[E_n]$$

Compositional reasoning

- Let events be different ways algorithm can fail

Pattern: The union bound

$$\Pr[E_1 \vee \cdots \vee E_n] \leq \Pr[E_1] + \cdots + \Pr[E_n]$$

Compositional reasoning

- Let events be different ways algorithm can fail

Pattern: The union bound

$$\Pr[E_1 \vee \cdots \vee E_n] \leq \Pr[E_1] + \cdots + \Pr[E_n]$$

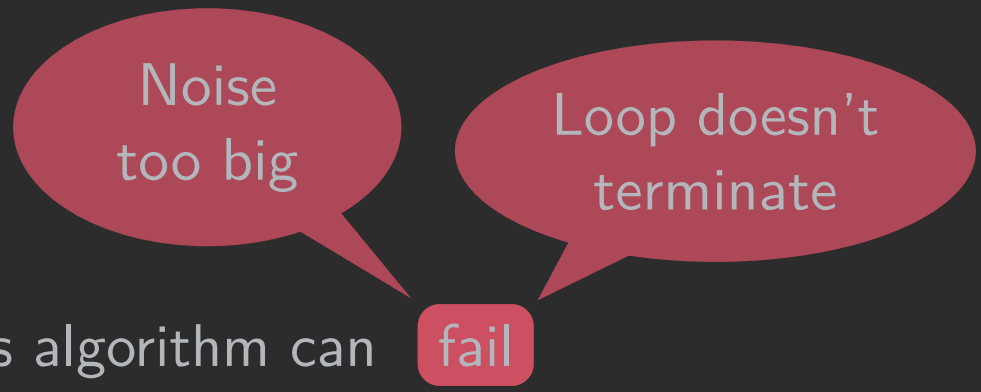
Noise too big

Compositional reasoning

- Let events be different ways algorithm can fail

Pattern: The union bound

$$\Pr[E_1 \vee \cdots \vee E_n] \leq \Pr[E_1] + \cdots + \Pr[E_n]$$

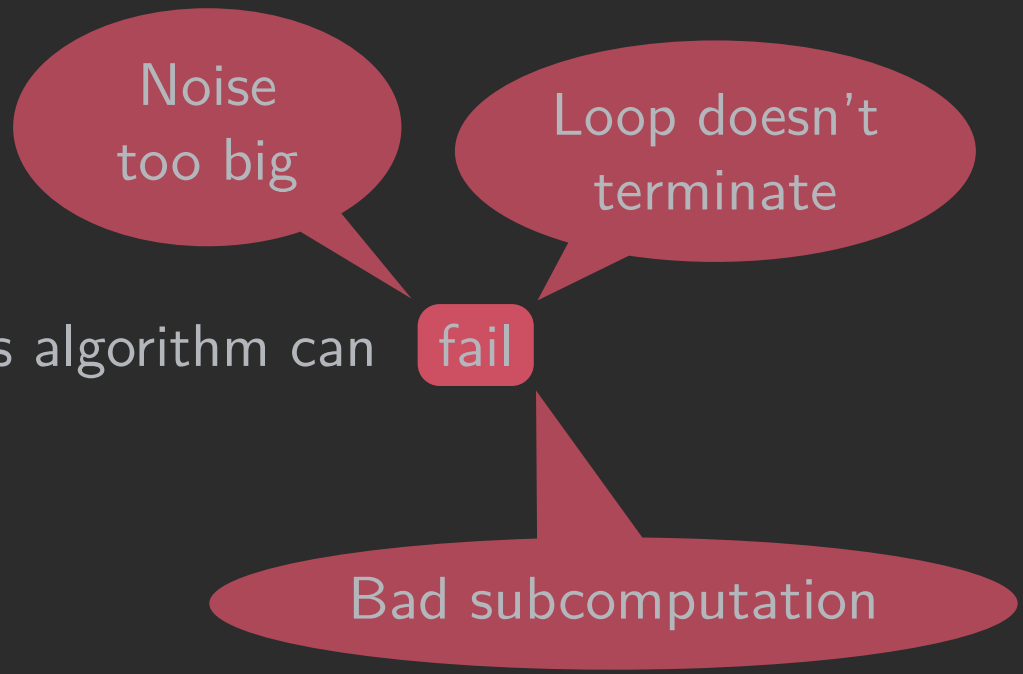


Compositional reasoning

- Let events be different ways algorithm can fail

Pattern: The union bound

$$\Pr[E_1 \vee \cdots \vee E_n] \leq \Pr[E_1] + \cdots + \Pr[E_n]$$

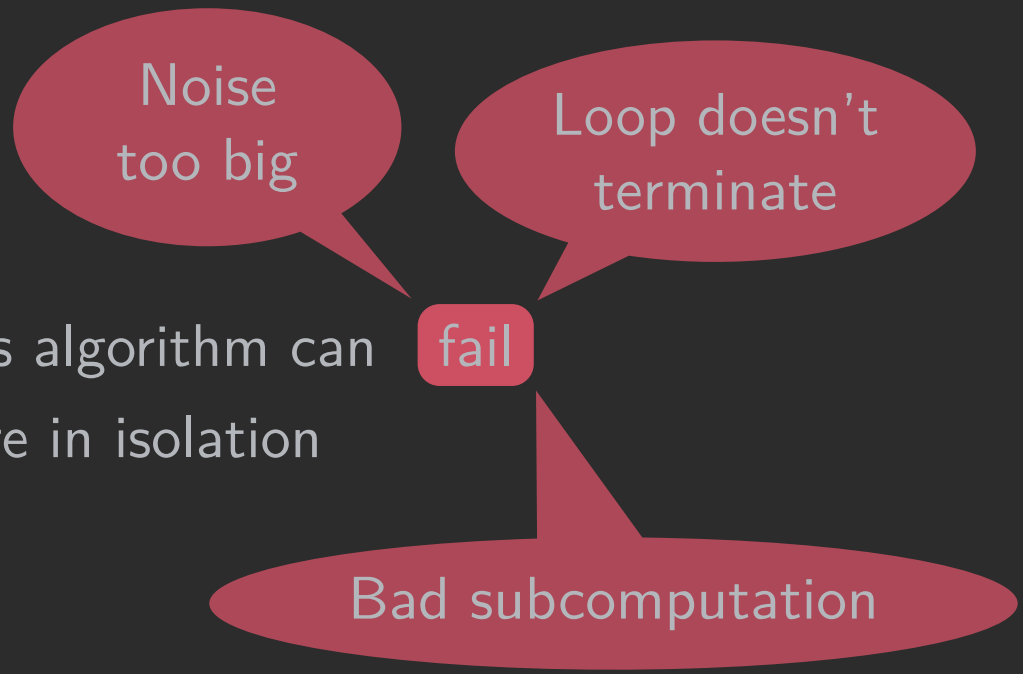


Compositional reasoning

- Let events be different ways algorithm can fail

Pattern: The union bound

$$\Pr[E_1 \vee \cdots \vee E_n] \leq \Pr[E_1] + \cdots + \Pr[E_n]$$



Compositional reasoning

- Let events be different ways algorithm can fail
- Analyze each possible failure in isolation

Work in progress

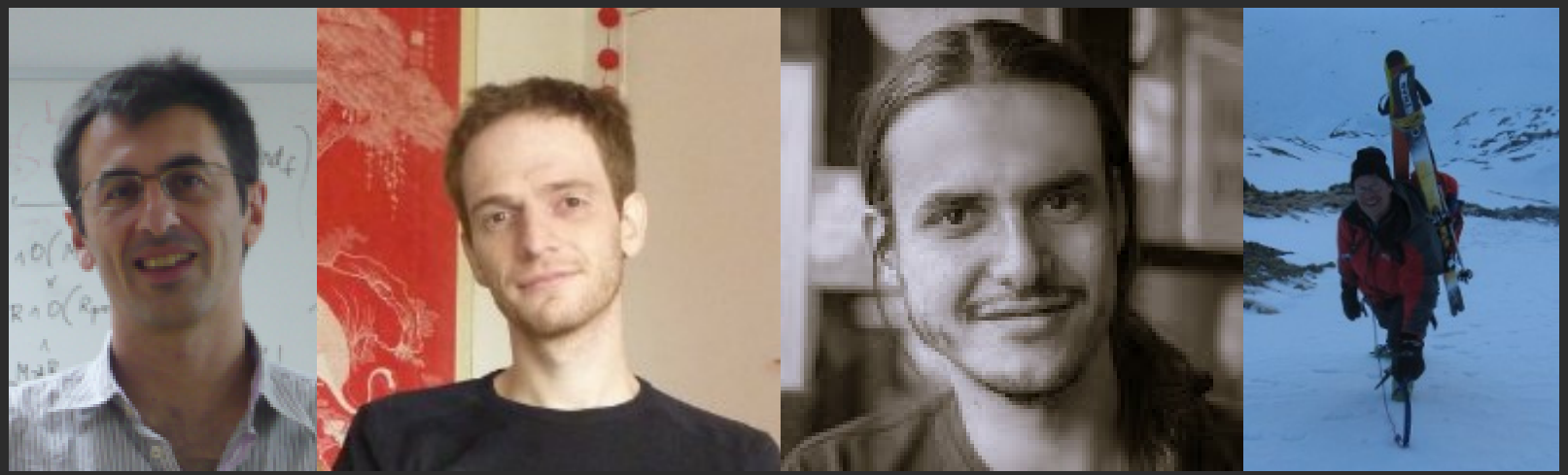
A probabilistic Hoare logic

- Assertions from paper proofs:

  $\Pr[X = 1] = 1/2, \qquad Y = \sum_{i=1}^{n} X_i, \qquad \#_{i=1}^{n} X_i, \qquad \ldots$

- Interactive: part of the EasyCrypt system
- Target: algorithms from recent STOC/FOCS/???

Fantastic collaborators



Towards a Theory AB

Towards a Theory AB

For Algorithms/Complexity Theory

- Computer verification of complex proofs
- Tools for different scales
- Theoretical tools (?)

Towards a Theory AB

For Algorithms/Complexity Theory

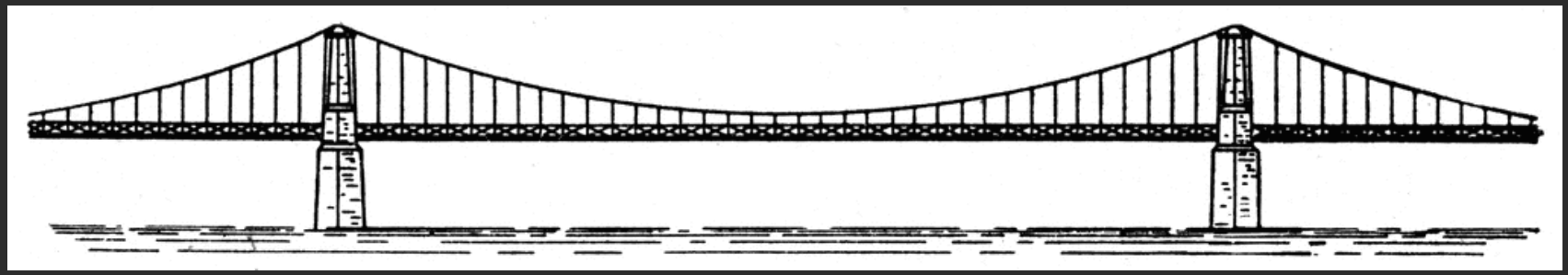
- Computer verification of complex proofs
- Tools for different scales
- Theoretical tools (?)

For our community

- Tons and tons of novel, challenging properties
- Different styles of proofs
- New abstractions?

Towards a Theory AB Toolbox

Verifying Randomized Algorithms



Marco Gaboardi[1] and Justin Hsu[2]

[1]University of Dundee
[2]University of Pennsylvania

May 6th, 2015