Faster quantum and classical SDP approximations for quadratic binary optimization

arXiv:1909.04613

F.G.S.L. Brandão[1,2], R. Kueng[1] and D. Stilck França[3]

[1]Institute for Quantum Information and Matter (IQIM)
California Institute of Technology

[2]Amazon Web Services, AWS Center for Quantum Computing
Pasadena, CA

[3]QMATH, Department of Mathematical Sciences
University of Copenhagen

February 25, 2020

Table of Contents

Caltech

Table of Contents

Quantum algorithms for optimization

Caltech

- quantum algorithms for optimization tasks is a promising “new” area
- mild speedups, but many important applications many applications
- important example: semidefinite programming (SDPs)
- existing quantum algorithms don’t always yield improvements
- “open” challenge: **relaxations of binary quadratic problems**

ideas

- bundle many linear constraints together (convex constraints)
- develop **primal only** classical algorithm (mirror descent)
- embed quantum simulation as fast subroutine (Gibbs sampling)

Table of Contents

Caltech

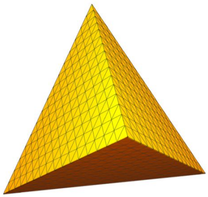
Binary quadratic optimization

Caltech

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{maximize}} \quad & \langle \boldsymbol{x}, \boldsymbol{A}\,\boldsymbol{x}\rangle = \mathrm{tr}(\boldsymbol{A}\,\boldsymbol{x}\boldsymbol{x}^*) \\ \text{subject to} \quad & \boldsymbol{x} \in \{\pm 1\}^n \end{aligned}$$

captures many important problems:

- i MaxCut and CutNorm
- ii community detection
- iii semi-discrete matrix factorization
- iv Ising model and spin glasses



SDP relaxation

$$\begin{aligned} \underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{maximize}} \quad & \mathrm{tr}\left(\boldsymbol{A}\, \boldsymbol{x}\boldsymbol{x}^*\right) \\ \text{subject to} \quad & \boldsymbol{x} \in \{\pm 1\}^n \end{aligned}$$

$$\begin{aligned} \underset{\boldsymbol{X} \in \mathbb{S}^n}{\text{maximize}} \quad & \mathrm{tr}\left(\boldsymbol{A}\, \boldsymbol{X}\right) \\ \text{subject to} \quad & \mathrm{diag}(\boldsymbol{X}) = \mathbf{1} \\ & \boldsymbol{X} \succeq \mathbf{0} \\ & \sout{\mathrm{rank}(\boldsymbol{X}) = 1} \end{aligned}$$

convex relaxation:

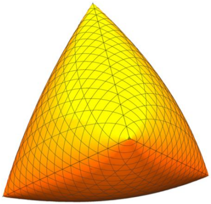
$f(\boldsymbol{X}) = \mathrm{tr}(\boldsymbol{A}\,\boldsymbol{X})$ is linear

$\boldsymbol{X} \in \mathcal{C}_1 \cap \mathcal{C}_2$ where

$\mathcal{C}_1 = \{\boldsymbol{X} : \mathrm{diag}(\boldsymbol{X}) = \mathbf{1}\}$ affine subspace

$\mathcal{C}_2 = \{\boldsymbol{X} : \boldsymbol{X} \succeq \mathbf{0}\}$ convex cone

actually a **SDP**, but $\mathrm{tr}(\boldsymbol{X}) = n$



Fundamental problem for this talk

Caltech

$$\begin{aligned} \underset{\boldsymbol{X}\in\mathbb{S}^n}{\text{maximize}} \quad & \operatorname{tr}\left(\boldsymbol{A}\,\boldsymbol{X}\right) \\ \text{subject to} \quad & \operatorname{diag}(\boldsymbol{X}) = \mathbf{1} \\ & \operatorname{tr}(\boldsymbol{X}) = n, \ \boldsymbol{X} \succeq \mathbf{0} \end{aligned}$$

Fundamental problem for this talk

Caltech

$$\begin{aligned} \underset{\boldsymbol{X}\in\mathbb{S}^n}{\text{maximize}} \quad & \operatorname{tr}\left(\frac{1}{\|\boldsymbol{A}\|}\boldsymbol{A}\,\boldsymbol{X}\right) \\ \text{subject to} \quad & \operatorname{diag}(\boldsymbol{X}) = \frac{1}{\boldsymbol{n}}\mathbf{1} && (\boldsymbol{X}\in\ \mathcal{C}_1) \\ & \operatorname{tr}(\boldsymbol{X}) = 1,\ \boldsymbol{X}\succeq \mathbf{0} && (\boldsymbol{X}\in\mathcal{S}) \end{aligned}$$

Table of Contents

Strategy

Caltech

phase I: optimization problem $\Rightarrow$ feasibility problem

phase II: develop quantum-inspired meta-algorithm

quantum boost: use quantum subroutines

inspiration: matrix multiplicative weights, mirror descent

Optimization $\Rightarrow$ feasibility

objective function $f(\boldsymbol{X})$ is *linear* and *bounded*

instead of optimizing $f(\boldsymbol{X})$ directly, choose $\lambda \in [-1,1]$ and ask:
is there a feasible $\boldsymbol{X}$ that obeys $f(\boldsymbol{X}) \leq \lambda$?

Binary search

$\mathcal{O}(2\log(1/\epsilon)) = \tilde{\mathcal{O}}(1)$ questions (with varying λ) nail down $f(\boldsymbol{X}_\sharp) \pm \epsilon$

Reformulate feasibility problem

task: for $\tilde{\boldsymbol{A}} = \frac{1}{\|\boldsymbol{A}\|}\boldsymbol{A}$ and $\lambda \in [-1, 1]$ solve

$$
\begin{aligned}
\text{find} \quad & \boldsymbol{X} \in \mathbb{S}^n \\
\text{subject to} \quad & \operatorname{tr}\left(\tilde{\boldsymbol{A}}\,\boldsymbol{X}\right) \leq \lambda && (\boldsymbol{X} \in \mathcal{A}_\lambda) \\
& \operatorname{diag}(\boldsymbol{X}) = \frac{1}{n}\boldsymbol{I} && (\boldsymbol{X} \in \mathcal{D}_n) \\
& \operatorname{tr}(\boldsymbol{X}) = 1, \ \boldsymbol{X} \succeq \boldsymbol{0} && (\boldsymbol{X} \in \mathcal{S}_n)
\end{aligned}
$$

- $\mathcal{A}_\lambda$ is half-space
- $\mathcal{D}_n$ is affine subspace
- $\mathcal{S}_n$ is the set of all density matrices

Quantum-inspired change of variables

$$
\boldsymbol{X} = \rho_{\boldsymbol{H}} = \frac{\exp(-\boldsymbol{H})}{\operatorname{tr}(\exp(-\boldsymbol{H}))} \in \mathcal{S}_n \quad \text{(Gibbs state)}
$$

Hamiltonian Updates

$\boldsymbol{X} \mapsto \boldsymbol{\rho_H} = \frac{\exp(-\boldsymbol{H})}{\mathrm{tr}(\exp(-\boldsymbol{H}))}$ automatically ensures $\boldsymbol{X} \in \mathcal{S}_n$

$$\begin{aligned} \text{find} \quad & \boldsymbol{H} \in \mathbb{S}^n \\ \text{subject to} \quad & \mathrm{tr}(\tilde{\boldsymbol{A}}\, \boldsymbol{\rho_H}) \leq \lambda & (\boldsymbol{\rho_H} \in \mathcal{A}_\lambda) \\ & \mathrm{diag}(\boldsymbol{\rho_H}) = \tfrac{1}{n} \boldsymbol{I} & (\boldsymbol{\rho_H} \in \mathcal{D}_n) \end{aligned}$$

Hamiltonian Updates:

1. start with $\boldsymbol{H} = \boldsymbol{0}$ ("infinite temperature")
2. check if $\boldsymbol{\rho_H} \in \mathcal{A}_\lambda$ and $\boldsymbol{\rho_H} \in \mathcal{D}_n$

   if true we are done
   else update $\boldsymbol{H}$ to penalize infeasible directions[a]
3. loop (at most) T times

[a] find separating hyperplane $\boldsymbol{P}$ and update $\boldsymbol{H} \leftarrow \boldsymbol{H} + \epsilon \boldsymbol{P}$

Illustration of Hamiltonian Updates

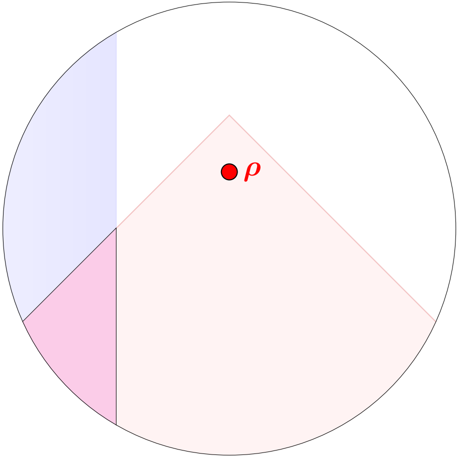


Illustration of Hamiltonian Updates

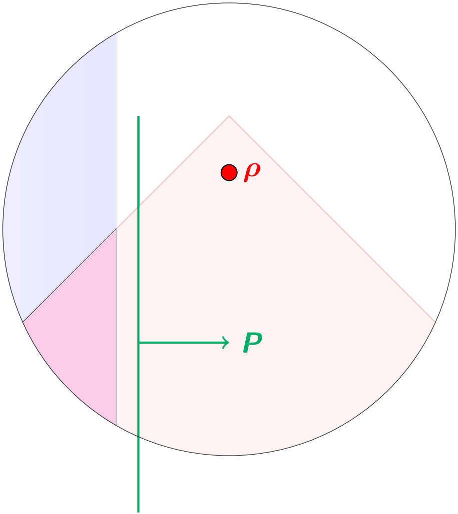


Illustration of Hamiltonian Updates

Caltech

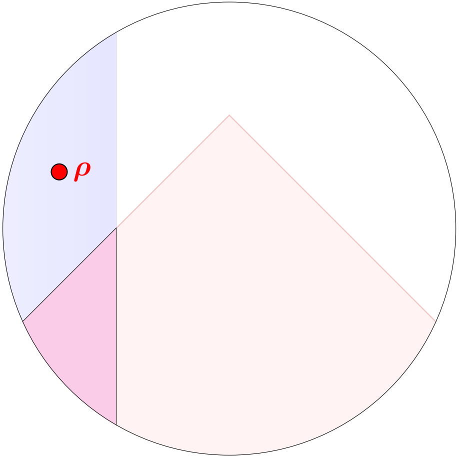


Illustration of Hamiltonian Updates

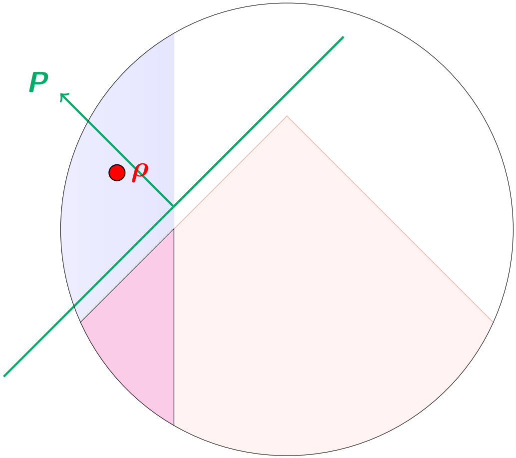


Illustration of Hamiltonian Updates

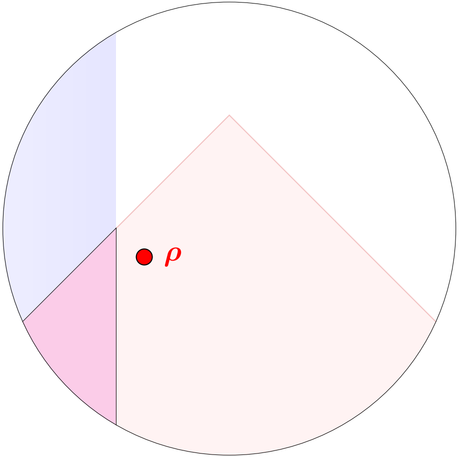


Illustration of Hamiltonian Updates

Caltech

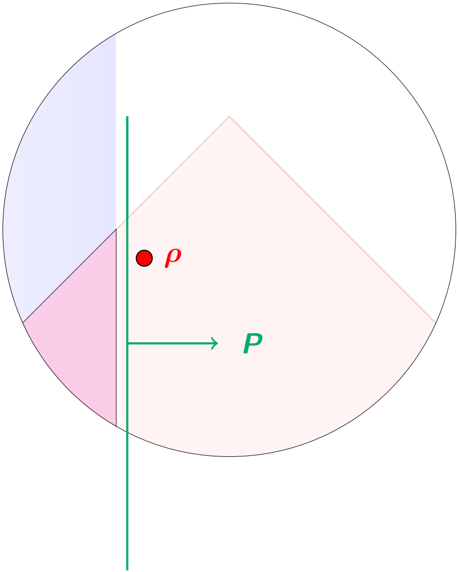


Illustration of Hamiltonian Updates

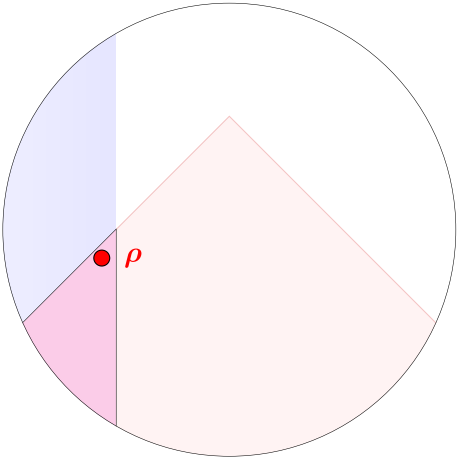


Table of Contents

Caltech

Hamiltonian Updates: convergence

Theorem (Brandão, RiK, França)

Hamiltonian Updates finds an approximately feasible point after (at most) $T = \lceil 16 \log(n)/\epsilon^2 \rceil + 1 = \tilde{\mathcal{O}}(1)$ *steps. Otherwise, the problem is infeasible.*

proof idea:

- relative entropy between $\rho_0 = \frac{1}{n}I$ and *any* feasible point ρ^* is $\leq \log(n)$
- show that each iteration makes constant progress in relative entropy:

$$S(\rho^* \| \rho_{t+1}) - S(\rho^* \| \rho_t) \leq -\frac{\epsilon^2}{16}$$

$\Rightarrow$ convergence after (at most) T steps, or $S(\rho^* \| \rho_T) < 0$

optimization context: mirror descent with von-Neumann entropy potential

Hamiltonian Updates: computational cost

- Hamiltonian Updates solves feasibility problem in $\mathcal{O}(\log(n)/\epsilon^2) = \tilde{\mathcal{O}}(1)$ steps
- each step requires three subroutines:
  - (i) compute $\boldsymbol{\rho_H} = \frac{\exp(-\boldsymbol{H})}{\mathrm{tr}(\exp(-\boldsymbol{H}))}$
  - (ii) $\boldsymbol{\rho_H} \in \mathcal{A}_\lambda$: check $\mathrm{tr}(\tilde{\boldsymbol{A}}\,\boldsymbol{\rho_H}) \leq \lambda$; output $\boldsymbol{P} = \tilde{\boldsymbol{A}}$
  - (iii) $\boldsymbol{\rho_H} \in \mathcal{D}_n$: check $\mathrm{diag}(\boldsymbol{\rho_H}) = \frac{1}{n}\mathbf{1}$; output $\boldsymbol{P} = \sum_i \mathbb{I}\left\{\langle \mathbf{e}_i, \boldsymbol{\rho_H}\boldsymbol{e}_i\rangle > \frac{1}{n}\right\} \mathbf{e}_i\mathbf{e}_i^t$
- naive cost:
  - (i) $\mathcal{O}(n^3)$
  - (ii) $\mathcal{O}(ns)$ $\quad s = \mathrm{(row)sparsity}(\tilde{\boldsymbol{A}})$
  - (iii) $\mathcal{O}(n)$
- naive total cost: $\tilde{\mathcal{O}}(n^4 s)$ (not very impressive yet)

Hamiltonian Updates: classical implementation

fact: Hamiltonian updates is designed to be robust

$\Rightarrow$ implementing subroutines up to accuracy ϵ still yields an approximately feasible solution (and correctly flags infeasibility)

classical boost: $\exp(-\boldsymbol{H}) \simeq \sum_{k=0}^{\ell} \frac{\boldsymbol{H}^k}{k!}$, $\ell = \mathcal{O}(\log(n)/\epsilon) = \tilde{\mathcal{O}}(1)$

Theorem (Brandão, RiK, França; 2019)

Hamiltonian Updates approximately solves quadratic SDP relaxations in classical runtime $\mathcal{O}\left(n^2 s \log(n)/\epsilon^{12}\right) = \tilde{\mathcal{O}}(n^2 s)$, *where* $s = (row)sparsity(\boldsymbol{A})$.

$$
\begin{aligned}
\text{maximize} \quad & \operatorname{tr}\left(\frac{1}{\|\boldsymbol{A}\|}\boldsymbol{A}\,\boldsymbol{X}\right) \\
\text{subject to} \quad & \operatorname{diag}(\boldsymbol{X}) = \mathbf{1} \\
& \boldsymbol{X} \succeq \mathbf{0}.
\end{aligned}
$$

1. best existing general algorithm: $\tilde{\mathcal{O}}(n^{2.5} s)$
2. approx. discrepancy: $\epsilon n \|\boldsymbol{A}\|$ vs. $\epsilon \|\boldsymbol{A}\|_{\ell_1}$
3. favorable for generic problem instances
4. no speedup for MaxCut

Hamiltonian updates: quantum implementation

Caltech

classical bottleneck: compute Gibbs states $\rho_H = \frac{\exp(-H)}{\mathrm{tr}(\exp(-H))}$

quantum speedup:

prepare copies of ρ_H on quantum computer	$\tilde{\mathcal{O}}(\sqrt{ns}s^{o(1)})$
estimate $\mathrm{tr}(\tilde{A}\,\rho_H)$ via phase estimation	$\mathcal{O}(1/\epsilon^2)$ copies
estimate $\mathrm{diag}(\rho_H)$ via computational basis measurements	$\mathcal{O}(n/\epsilon^2)$ copies

Theorem (Brandão, RiK, França; 2019)

Hamiltonian Updates approximately solves binary quadratic SDP relaxations in quantum runtime $\tilde{\mathcal{O}}(n^{1.5}(\sqrt{s})^{1+o(1)})$.

1. first quantum speedup for **combinatorial SDP relaxation**
2. beats classical runtimes $\tilde{\mathcal{O}}(n^2 s)$ and $\tilde{\mathcal{O}}(n^{2.5} s)$
3. classical access to (approx.) optimal Hamiltonian $\Rightarrow$ data processing

Details about quantum subroutine

important design feature: Hamiltonians are very structured:
$\boldsymbol{H} = \alpha \tilde{\boldsymbol{A}} + \beta \boldsymbol{D}$, $\alpha, \beta = \mathcal{O}(\log(n)/\epsilon)$

1. use [Poulin, Wojcan; 2009] to reduce task of preparing $\boldsymbol{\rho_H}$ to simulating time evolution ($\mathcal{O}(\sqrt{n})$ invocations)
2. use [Childs, Wiebe; 2012] to split up time evolution (negligible overhead)
3. [Low; 2019]: implementing $\exp(it\alpha\tilde{\boldsymbol{A}})$ costs $\tilde{\mathcal{O}}(\sqrt{s}^{1+o(1)})$
4. [Prakash; 2014] implementing $\exp(it\beta\boldsymbol{D})$ with quantum RAM costs $\tilde{\mathcal{O}}(n)$

$\Rightarrow$ total cost: $\tilde{\mathcal{O}}(n^{1.5}\sqrt{s}^{1+o(1)})$

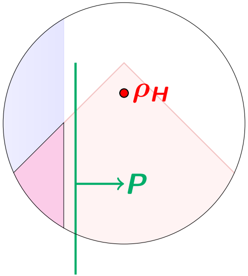
Table of Contents

Caltech

Conclusion

we established speedups for important problem class:

$$\begin{aligned} \underset{\boldsymbol{X}\in\mathbb{S}^n}{\text{maximize}} \quad & \mathrm{tr}(\boldsymbol{A}\,\boldsymbol{X}) \\ \text{subject to} \quad & \mathrm{diag}(\boldsymbol{X}) = \tfrac{1}{n}\mathbf{1} \\ & \mathrm{tr}(\boldsymbol{X}) = 1,\ \boldsymbol{X} \succeq \mathbf{0} \end{aligned}$$



our strategy:

(i) replace optimization by a sequence of feasibility problems

(ii) change of variables: $\boldsymbol{X} \leftarrow \rho_{\boldsymbol{H}} = \frac{\exp(-\boldsymbol{H})}{\mathrm{tr}(\exp(-\boldsymbol{H}))}$

(iii) iteratively penalize infeasible directions by Hamiltonian Updates $\boldsymbol{H} \leftarrow \boldsymbol{H} + \epsilon \boldsymbol{P}$

(iv) boost runtime by preparing each $\rho_{\boldsymbol{H}}$ on quantum computer

our result: we obtain approximate solutions faster than existing approaches: $\tilde{\mathcal{O}}(n^2 s)$ (classical) and $\tilde{\mathcal{O}}(n^{1.5}\sqrt{s}^{1+o(1)})$ (quantum) vs. $\tilde{\mathcal{O}}(n^{2.5} s)$ (classical)

Outlook

Caltech

1. improve runtime scaling in approximation accuracy ϵ
2. implementation on near-tearm devices or classical computers
3. adapt meta-algorithm to other important convex optimization problems:
   - quantum state tomography
   - semi-discrete matrix factorization

Thank you!