Eric Du | Physics C191
Final Project Notes | **Introduction to Quantum Information** | April 22, 2024

Header styling inspired by CS 70: https://www.eecs70.org/

1 Poster Outline

- General theme around quantum advantage:

  Understanding complexity theory is extremely useful for gauging possible quantum advantage in optimization, but it is not necessary, nor sufficient when seeking a practical quantum advantage.

1.1 Background

1.1.1 Linear Programs (LP)

- Usually phrased in terms of finding the maximum value of an objective function $f : \mathbb{R}^n \to \mathbb{R}$ over all $x \in \mathbb{R}^n$.
- Linear programs in particular have specific structures: specifically, we want to minimize a linear function subject to linear inequalities on non-negative vectors.

  This means we want to minimize a quantity like $c^\top \vec{x}$ subject to some constraints $A\vec{x} \leq \vec{b}$.

1.1.2 Semidefinite Programming (SDP)

- Is a larger class of problem where we try to maximize:

$$\max_{X \succeq 0} \operatorname{tr}(CX)$$

  subject to the constraints:

$$\operatorname{tr}(A_i X) \leq b_i,\ \forall i \in [m]$$

  So here, C is a matrix like the objective function, and $A_i, \dots, A_m$ are the set of constraint matrices. These matrices must also be symmetric.

  What does the $X \succeq 0$ term refer to? Where did we introduce any "partial order" into the system?

  It's essentaially saying that all the m eigenvalues of X must be nonnegative.

- Linear programs are a specific kind of SDP, where all the given matrices are diagonal.
- In terms of quantum speedups, they usually trade off between n, m and the precision ϵ to which we solve the problem.
- **First Order Methods:** First order methods are generlly based on the Multiplicative Weights Update (MWU) algorithms, where we have solutions of the form:

$$X \propto \frac{\exp\left(\sum_{i=1}^m y_i A_i\right)}{\operatorname{tr}(\exp\left(\sum_{i=1}^m y_i A_i\right))}$$

  for some sparse vector $y \in \mathbb{R}^m$. Here, I think sparse refers to the case where most of the entries in y are zero.

  Then, we compute quantities such as $\operatorname{tr}(A\rho)$ where A is either one of the A_i's or C.

- The best classical algorithms for these problems were on the order of $\widetilde{\mathcal{O}}\left(mns\left(\frac{Rr}{\epsilon}\right)^4 + ns\left(\frac{Rr}{\epsilon}\right)^7\right)$. We generally write $\gamma = \frac{Rr}{\epsilon}$ so the runtime is written as $\widetilde{\mathcal{O}}\left(mns\gamma^4 + ns\gamma^7\right)$ but the best quantum algorithms achieve a runtime of $\widetilde{\mathcal{O}}\left((\sqrt{m} + \sqrt{n}\gamma)s\gamma^4\right)$
- Generally the parameter γ scales poorly with given parameters n and m.

- The issue with all the quantum approaches at the moment is that they all rely on something called QRAM, which is a memory bank (like RAM) but addressed by qubits. The issue with this is that such a structure is not physically possible at the moment, or at least not without much error. However, QRAM is very common throughout literature.
- **Second Order Methods:** These methods rely on interior point methods (IPMs), which use Newton's method, starting at a point on the interior of a convex "feature" denoted by $\chi \subset R^n$ to solve a problem of the form:

$$\min_{x \in \mathbb{R}^n} \eta \cdot \langle c, x \rangle + f(x)$$

  where $f : \mathrm{int}(\chi) \to \mathbb{R}$ is a constraint function that essentially defines the boundary of the feasible set χ. This process is then repeated until we find a minimum point.
- Quantum speedups are inspired by a quantum state tomography algorithm, which basically optimizes the computation to figure out the direction in which we need to iterate. The issue with this is that there is a $\frac{1}{\epsilon}$ dependence on the runtime, where ϵ is the precision to which we want to calculate. Overall, because of the mixed dependence on other parameters, it's hard to say whether quantum algorithms have an overall speedup over classical ones.

  There has been a recent advancement that speedsup the Newton step in computation, that is worth looking at.

1.1.3 Extension into Mixed Integer Programming (MIP)

- A class of optimization problem where some of the varaibles are constrained to being integers. These problems are at least as hard as either discrete or continuous optimization.
- There are multiple approaches to solving these problems, some of which guarantee speedup.

  Branch-and-Bound-and-Cut is the main way these problems are solved classcally. Essentially, this approach branches based on the discrete variables, and finds solutions to the continuous ones by building around it. The other variables are then solved through convex optimization, which we already know has seen much speedup recently.
- A lot of these approaches basically aim to reformulate the problem in terms of other problems that are equally as hard to solve, but have recently seen progress in finding speedups.
- There aren't any general provable advantages for MIP. Most quanutm algorithms known use a classical approach
- As of last week, MILP has seen an advancement by showing that we can transform any MILP into two problems: an Integer Master problem and a linear subproblem, the latter of which is solved using classical techniques. Altogether, this is what the authors call a "quantum-classical" approach.

1.2 Literature Review, look at what's been done

- Here, we investigate well-known demonstrations of quantum advantage. From the very basics, we look at concrete examples (Shor's algorithm and order-finding), but also I think it's useful to look at currently, how far do we think we can take this.

1.3 Two quantum optimizations over classical algorithms

- Specifically we will look at the quantum advantage regarding semidefinite programming (SDP) problems.

1.4 Theoretical Applications

2 Quantum speedups for Interior Point Method (IPM)

Focus on the paper by Apers, Gribling

2.1 Interior Point Method

- Reformulates the LP by asking to find the minimum of a function $f_\eta(x)$, where $f_\eta(x) = \eta c^\top x + f(x)$. Here, $f(x)$ characterizes the constraints, and η characterizes the step size.

- The overall runtime of the algorithm is based on the size of η, and it's been shown that we can iteratively change η to some η' while still guaranteeing that we reach the optimal solution. So, the challenge comes down to figuring out how much we can change $\eta \to \eta'$
- The number of times we need to increase η scales roughly as $\widetilde{O}(\sqrt{\vartheta_f}\log(1/\epsilon)$.
- Best classical algorithm: runtime of $\widetilde{O}(nd + d^3)$ whereas the best quantum algorithms:
  - Logarithmic Barrier: $\widetilde{O}(nd)$ queries, for a runtime of $\widetilde{O}(nd^2 r + \sqrt{n}d^\omega)$
  - Volumetric Barrier: $\widetilde{O}(\sqrt{\vartheta_f}\log(1/\epsilon)$ queries, with runtime $\widetilde{O}(n^{1/4}d^{1/4}(\sqrt{n}d^2 r + d^{\omega+1}))$
  -
- Basically, the speedup comes from modifying the form of $f(x)$, in order to generate a speedup. Essentially, $f(x)$ is a function that goes to infinity as we approach the boundary of a feasible region.

2.2 Spectral Approximation

- Utilizes a repeated halving algorithm in order to compute the spectral approximation.
- To do this, they use a combination of

2.3 Approximating Matrix-Vector Product

2.4 Approximating Hessians and Gradients

- As the solution using IPMs are heavily dependent on the speed of the Newton step, one way that this new algorithm demonstrates this is by approximating Hessian matrices and gradients to lighten up the Newton step.
- The paper calls this "approximate Newton steps". Essentially, from a feasible point $x \in \mathbb{R}^d$, they compute approximations $Q(x)$ and $g(x)$ satisfying
$$Q(x) \preceq H(x) \preceq CQ(x) \quad \text{and} \quad \|\widetilde{g}(x) - g(x)\|_{H(x)^{-1}} \leq \zeta$$
for some parameters C and ζ such that $C, \zeta^{-1} \in \widetilde{O}(1)$. These parameters are basically just constants that dictate how good we want our approximation to be.

  We say that $A \preceq B$ if and only if $0 \preceq B - A$, and $0 \preceq M$ is the notation to say that M is positive semidefinite. Positive semidefinite means that the matrix product $x^\top Mx \geq 0$ for every vector x.
- The Hessians are all the form $A^T S_x^{-1} W S_x^{-1} A$, and the only difference between the barriers is the form of W.
  - For the log-barrier, W is the identity
  - For the volumetric barrier, W is the diagonal matrix of leverage scores $S_x^{-1}A$
  - For the Lewis barrier, W is the diagonal matrix of lewis weights $S_x^{-1}A$.

  Does this mean that the volumetric barrier and Lewis barrier have the same form for W?
- To approximate the gradient, the quantum algorithm for approximate vector products is used. We use the fact that $g(x) = -A^T S_x^{-1} W\mathbf{1}$, where $\mathbf{1}$ denoting the all-ones vector. Then, you can write this as the vector product between $\sqrt{W} S_x^{-1} A$ and $v = \sqrt{W}\mathbf{1}$, which can then be computed using the approximation algorithm.