Semidefinite Programming and Quantum Algorithms

Michael Czekanski & R. Teal Witter

Middlebury College

December 3, 2019

Overview

Motivation

As quantum computers someday soon maybe might become a reality...

What are the problems that quantum computers can solve more efficiently than classical computers? #QuantumSpeedUp

Houston, we have a problem:

Given Boolean function f...

What is the optimal quantum query complexity?

What is a query optimal quantum algorithm?

Query Complexity

Given bitstring x and function f, how many bits of x to determine $f(x)$ in the worst case?

Query Complexity

Given bitstring x and function f, how many bits of x to determine $f(x)$ in the worst case?

$$f : x_1 \wedge (x_2 \vee \bar{x_2}) \vee x_3$$

Query Complexity

Given bitstring x and function f, how many bits of x to determine $f(x)$ in the worst case?

$$f : x_1 \wedge (x_2 \vee \bar{x_2}) \vee x_3$$

$$\begin{aligned} x &\rightarrow f(x) \\ 000 &\rightarrow 0 \\ 010 &\rightarrow 0 \\ 011 &\rightarrow 1 \end{aligned}$$

Query Complexity

Given bitstring x and function f, how many bits of x to determine $f(x)$ in the worst case?

$$f : x_1 \vee x_2 \vee x_3$$

Query Complexity

Given bitstring x and function f, how many bits of x to determine $f(x)$ in the worst case?

$$f : x_1 \vee x_2 \vee x_3$$

$$\begin{aligned} x &\rightarrow f(x) \\ 000 &\rightarrow 0 \\ 010 &\rightarrow 1 \\ 011 &\rightarrow 1 \end{aligned}$$

Span Program

Given x and f, is $f(x)$ true?

$$\iff$$

Given input vectors $I(x)$ and target vector τ, do the input vectors span to the target vector?

Span Program

Given x and f, is $f(x)$ true?

$$\iff$$

Given input vectors $I(x)$ and target vector τ, do the input vectors span to the target vector?

For OR, $I(x) = x$ and $\tau = 1$.

Span Program

Given x and f, is $f(x)$ true?

$$\iff$$

Given input vectors $I(x)$ and target vector τ, do the input vectors span to the target vector?

For OR, $I(x) = x$ and $\tau = 1$.

I(000) = 0,0,0 does not span to τ :(

Span Program

Given x and f, is $f(x)$ true?

$$\iff$$

Given input vectors $I(x)$ and target vector τ, do the input vectors span to the target vector?

For OR, $I(x) = x$ and $\tau = 1$.

I(010) = 0,1,0 does span to τ :)

Our Approach, Generally

convex_optimization.pdf

Our Approach, Specifically [1]

$$f_{\text{bound}} = \min_{\mathbb{X}} \left(\max_{y \in D} \sum_{j \in [n]} \langle y, j | \, \mathbb{X} \, | y, j \rangle \right)$$

Subject to:

$$\mathbb{X} \succcurlyeq 0$$

$$\forall (y, z) \in F \sum_{j \in [n] : y_j \neq z_j} \langle y, j | \, \mathbb{X} \, | z, j \rangle = 1$$

What $\mathbb{X}$ Looks Like (2-bit OR)

$$\mathbb{X} = \begin{array}{c c} & \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} & \begin{bmatrix} X_{(00,00)} & X_{(00,01)} & X_{(00,10)} & X_{(00,11)} \\ X_{(01,00)} & X_{(01,01)} & X_{(01,10)} & X_{(01,11)} \\ X_{(10,00)} & X_{(10,01)} & X_{(10,10)} & X_{(10,11)} \\ X_{(11,00)} & X_{(11,01)} & X_{(11,10)} & X_{(11,11)} \end{bmatrix} \end{array}$$

What $\mathbb{X}$ Looks Like (2-bit OR)

$$\mathbb{X} = \left[\begin{array}{cc|cc|cc|cc} 0.7 & 0 & 0 & 0 & 1 & 0 & 0.5 & 0 \\ 0 & 0.7 & 0 & 1 & 0 & 0 & 0 & 0.5 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \sqrt{2} & 0 & 0 & 0 & 0.7 \\ \hline 1 & 0 & 0 & 0 & \sqrt{2} & 0 & 0.7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0.5 & 0 & 0 & 0 & 0.7 & 0 & 0.6 & 0 \\ 0 & 0.5 & 0 & 0.7 & 0 & 0 & 0 & 0.6 \end{array}\right]$$

What $\mathbb{X}$ Looks Like (2-bit OR)

We want to minimize the maximum (over inputs y) of:

$$\sum_{j \in [n]} \langle y, j | \, \mathbb{X} \, | y, j \rangle$$

$$\mathbb{X} = \left[\begin{array}{cc|cc|cc|cc}
0.7 & 0 & 0 & 0 & 1 & 0 & 0.5 & 0 \\
0 & 0.7 & 0 & 1 & 0 & 0 & 0 & 0.5 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & \sqrt{2} & 0 & 0 & 0 & 0.7 \\
\hline
1 & 0 & 0 & 0 & \sqrt{2} & 0 & 0.7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0.5 & 0 & 0 & 0 & 0.7 & 0 & 0.6 & 0 \\
0 & 0.5 & 0 & 0.7 & 0 & 0 & 0 & 0.6
\end{array}\right]$$

What $\mathbb{X}$ Looks Like (2-bit OR)

For y, z such that $f(y) \neq f(z)$:

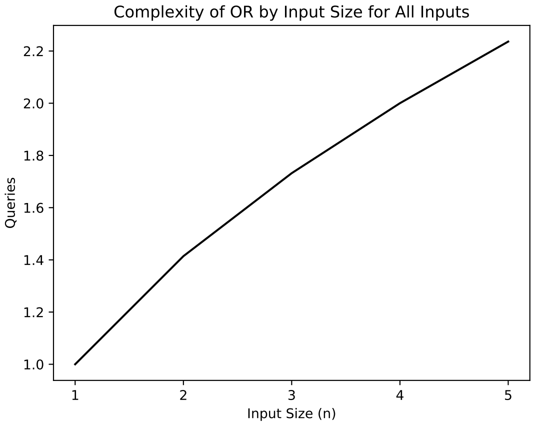
$$\sum_{j \in [n]: y_j \neq z_j} \langle y, j | \, \mathbb{X} \, | z, j \rangle = 1$$

$$\mathbb{X} = \left[\begin{array}{cc|cc|cc|cc}
0.7 & 0 & 0 & 0 & 1 & 0 & 0.5 & 0 \\
0 & 0.7 & 0 & 1 & 0 & 0 & 0 & 0.5 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & \sqrt{2} & 0 & 0 & 0 & 0.7 \\
\hline
1 & 0 & 0 & 0 & \sqrt{2} & 0 & 0.7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0.5 & 0 & 0 & 0 & 0.7 & 0 & 0.6 & 0 \\
0 & 0.5 & 0 & 0.7 & 0 & 0 & 0 & 0.6
\end{array}\right]$$

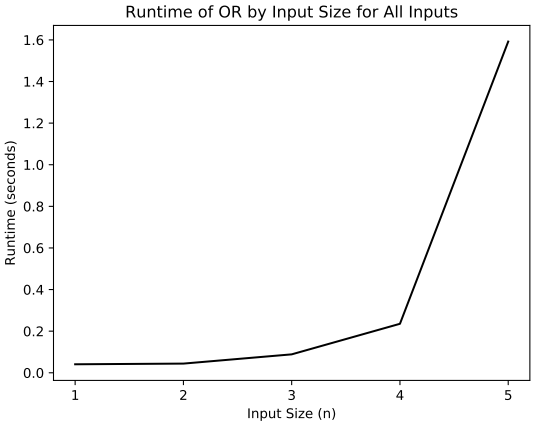
We use ADM [2]

- "Easy" to implement
- Exploits sparsity

Numerical Results: Very Accurate :)



Run Time Results: Too Many Inputs :(



In Conclusion...

- Our algorithm rocks
- Except on large inputs

Future Work

- Expand our algorithm beyond binary outputs
- Utilize a more efficient solver
- Memory efficiency of span programs

Thank you!

B. W. Reichardt.
Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function.
In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 544–551. IEEE, 2009.

Z. Wen, D. Goldfarb, and W. Yin.
Alternating direction augmented lagrangian methods for semidefinite programming.
Mathematical Programming Computation, 2(3-4):203–230, 2010.