From Adversaries to Algorithms

Troy Lee
Rutgers University
Quantum Query Complexity

- As in classical complexity, it is difficult to show lower bounds on the time or number of gates required to solve a problem on a quantum computer.

- Query complexity is a simpler model which still captures the essence of quantum speedups—Grover’s $O(\sqrt{n})$ search algorithm, heart of Shor's efficient factoring algorithm.

- Moreover, in this model we can prove lower bounds!
  - Polynomial method, Quantum Adversary method
Classical Query Model

- For some function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, you want to compute $f(x)$.

- The function is known but you are only given black box access to input $x$. Can ask $x_i =$?

- How many queries are needed on the worst case $x$?
Quantum Query Model

• Model is similar, but now can ask queries in superposition.

• Querying $x$ at index $i$ is modeled by the operator $O_x|i⟩|z⟩ = (-1)^{xi}|i⟩|z⟩$.

• Query acts linearly: $O_x (∑_{i} \alpha_i|i⟩|z⟩) = ∑_{i}(-1)^{xi}\alpha_i|i⟩|z⟩$.

• In between queries can do unitary transformations. At the end of algorithm, measure the first bit of workspace.
Classical vs. Quantum Query Complexity: Examples

- **OR function**: Quantum $\Theta(\sqrt{n})$ [BBBV97, Grover96], Classically $\Omega(n)$.

- **Parity function**: Quantum $\Omega(n)$, Classically $\Omega(n)$.

- **Evaluating AND-OR trees**: Quantum $\Theta(\sqrt{n})$ [FGG08, ACRSZ07], Classically $\Theta(n^{0.753})$ [Saks-Wigderson86].
Sometimes Quantum is Easier

- Interesting case in point: For formulas with AND, OR, NOT gates with $n$ variables where every variable appears once, the quantum query complexity is resolved: $\tilde{\Theta}(\sqrt{n})$ [Barnum-Saks04, Reichardt09].

- For randomized query complexity it is still an open problem!

- Balanced AND-OR trees are believed to be the hardest case, but largest general lower bound is $\Omega(n^{0.51})$ [Heiman-Wigderson91].
Reichardt’s Recent Result

• Characterizes quantum query complexity in terms of a (relatively) simple semidefinite program.

• This semidefinite program, the general adversary method [Høyer-L-Špalek07], was known to be a lower bound.

• Reichardt considers the dual of this program, and uses a solution to the dual to construct an algorithm.

• Now one can construct quantum algorithms just by designing vector systems with certain properties!
• Adversary method developed by [Ambainis, 2002].

• Many competing formulations: weight schemes [Amb03, Zha05], spectral norm of matrices [BSS03], and Kolmogorov complexity [LM04].

• All these methods shown equivalent by [Špalek and Szegedy, 2006].
Reinventing the adversary

- General adversary method ADV\(^\pm\), while similar in form to previous methods, does not face their limitations.

- Previous adversary bounds used the principle that a successful algorithm must distinguish ‘yes’ inputs from ‘no’ inputs.

- General adversary method actually makes use of the fact that algorithm computes the function.
General adversary lower bound

- Considers a potential function. Shows that the potential is large at the start of the algorithm, and small at the end of a successful algorithm.

- Then it bounds how much the potential function can change in any one step—this is essentially bounding how much information the algorithm can learn in any one query.

- Reichardt’s result implies the algorithm is actually able to learn this much information in each step!
**General adversary lower bound**

- The potential function is defined by a symmetric matrix \( \Gamma \) with rows and columns indexed by inputs to \( f \). The only requirement is that \( \Gamma(x, y) = 0 \) if \( f(x) = f(y) \).

- The bound is given by

\[
\max_{\Gamma} \frac{\|\Gamma\|}{\max_{i \in [n]} \| \Gamma \circ D_i \|}.
\]

where \( D_i(x, y) = 1 \) if \( x_i \neq y_i \) and 0 otherwise.

- Intuitively, the principal eigenvector of \( \Gamma \) gives a “hard” distribution for the algorithm.
The $\Gamma$ matrix

<table>
<thead>
<tr>
<th></th>
<th>$f^{-1}(0)$</th>
<th>$f^{-1}(1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f^{-1}(0)$</td>
<td>0</td>
<td>A</td>
</tr>
<tr>
<td>$f^{-1}(1)$</td>
<td>$A^*$</td>
<td>0</td>
</tr>
</tbody>
</table>

Notice that the spectral norm of $\Gamma$ equals that of $A$. 
The $\Gamma \circ D_1$ matrix

The spectral norm of $\Gamma \circ D_1$ equals $\max\{\|B\|, \|C\|\}$. 
Example: OR function

We define the matrix:

\[
\begin{array}{c|cccc}
0000 & 1000 & 0100 & 0010 & 0001 \\
0000 & 1 & 1 & 1 & 1 \\
\end{array}
\]

The spectral norm of this matrix is $\sqrt{4}$, and the spectral norm of each $\Gamma \circ D_i$ is one.

Generalizing this construction we find $Q_2(\text{OR}_n) = \Omega(\sqrt{n})$. 
Dual program

• When you look at the dual of the adversary bound, you get the following.

• You want to find vectors $v_{x,i}$ for every input $x$ and index $i \in [n]$ such that for all $x, y$ with $f(x) \neq f(y)$

\[
\sum_{i : x_i \neq y_i} \langle v_{x,i}, v_{y,i} \rangle = 1
\]

• Objective is to minimize $\sum_i \|v_{x,i}\|^2$. 
Using these vectors, Reichardt designs a weighted bipartite graph.

If $f(x) = 1$ this graph will have a 0-valued eigenvector with support on special node.

If $f(x) = 0$ all eigenvectors with small eigenvalues will have small overlap with special node.

Gap is proportional to reciprocal of adversary bound. By doing a quantum walk on the graph, can distinguish which case you are in.
Graph Construction

![Graph Diagram]
Conclusions

- Quite remarkable to have such a meeting of upper and lower bounds.
- Role of duality.
- Still many interesting problems where quantum query complexity is unknown. Can studying these vector systems help?