New methods for quantum algorithms

Andris Ambainis

University of Latvia

AQIS’2011, August 25, 2011
Shor’s algorithm [1994]

- Factoring: given $N=pq$, find $p$ and $q$.
- Best algorithm - $2^{O(n^{1/3})}$, $n$ – number of digits.
- Quantum algorithm - $O(n^3)$ [Shor, 94].
- Cryptosystems based on hardness of factoring/discrete log become insecure.
Grover's search [1996]

- Find $i$ such that $x_i = 1$.
- Queries: ask $i$, get $x_i$.
- Classically, $N$ queries required.
- Quantum: $O(\sqrt{N})$ queries [Grover, 96].

Speeds up any search problem.
Element distinctness [A, 2004]

- Numbers \(x_1, x_2, \ldots, x_N\).
- Determine if two of them are equal.
- Classically: \(N\) queries.
- Quantum: \(O(N^{2/3})\).
Evaluating logic formulas [Farhi, et al., 2007, Reichardt, 2011]

- Formula of size $N$.
- $O(\sqrt{N})$ quantum algorithms.
From algorithms to methods

- **Element distinctness:**
  - Search by a quantum walk (quantization of Markov chains);

- **Formula evaluation:**
  - Span programs;

Mathematical structures
Quantum query model

- Input data $x_1, \ldots, x_N$ given by a black box that answers queries;

- Complexity = number of queries.

- Examples: Grover, element distinctness, etc.
Quantum algorithm

\[ |\psi_s\rangle \rightarrow \mathbb{Q} \rightarrow U_1 \rightarrow \mathbb{Q} \rightarrow \ldots \rightarrow |\psi_f\rangle \]

Start state

End state
Part 1

Search by quantum walk
Generic search problem

1. Finite search space.
2. Some states marked.
3. Task: find a marked state.
4. Solution: random walk on search space.
Search by a random walk

- Start in random state;
- Perform a random walk, until finding a marked state.
- $T$ – expected number of steps.

Szegedey, 2004: Quantum walk detects a marked state in $O(\sqrt{T})$ steps.
In more detail...

- Random walk:
  - $T$ walking steps to find a marked state;
  - Time complexity: $S + TW$:
    - $S$ – time to generate a random starting state;
    - $W$ – time to perform one walking step;

Szegedy, 2004: Quantum walk, $O(S + \sqrt{TW})$ steps.
Matrix multiplication [Buhrman, Špalek, 05]

- A, B, C – n×n matrices.
- Finding C=AB: $O(n^{2.37...})$ steps;
- Given A, B and C, we can test AB=C in:
  - $O(n^2)$ steps by a probabilistic algorithm;
  - $O(n^{5/3})$ steps by a quantum algorithm.
Triangle finding [Magniez, Santha, Szegedy, 05]

- Graph $G$ with $n$ vertices.
- $n^2$ variables $x_{ij}$; $x_{ij} = 1$ if there is an edge $(i, j)$.
- Does $G$ contain a triangle?
- Classically: $O(n^2)$.
- Quantum: $O(n^{1.3})$. 
Forbidden subgraph properties
[Childs, Kothari, 2011]

- \( P = \text{"G contains one of several subgraphs } H_1, \ldots, H_k \text{"} \).
- \( P \) – sparse if any \( G \) not containing \( H_1, \ldots, H_k \) has \( O(n) \) edges
- Any sparse \( P \) can be decided in \( o(n^{3/2}) \) steps.
Search vs. finding

- What can we do in time $O(S+\sqrt{T}W)$?
- [Szegedy, 2004] Quantum algorithm for detecting whether a marked state exists.
- [Krovi et. al., 2010] Quantum algorithm for finding a marked state*.

*Assumes knowledge of $p, p \approx p_{\text{marked}}, p_{\text{marked}}$ – fraction of marked states.
Part 2

Quantum algorithms for formula evaluation
Evaluating AND-OR trees

- Variables $x_i$ accessed by queries to a black box:
  - Input $i$;
  - Black box outputs $x_i$.

- Quantum case:
  \[
  \sum_{i} a_i \ket{i} \rightarrow \sum_{i} a_i (-1)^{x_i} \ket{i}
  \]

- Evaluate T with the smallest number of queries.
Results (up to 2007)

- Full binary tree of depth $d$.
- $N = 2^d$ leaves.
- Deterministic: $\Omega(N)$.
- Randomized $[SW,S]$: $\Theta(N^{0.753...})$.
- Quantum?
- Easy q. lower bound: $\Omega(\sqrt{N})$. 
New results

- [Farhi, Gutman, Goldstone, 2007]: $O(\sqrt{N})$ time algorithm for evaluating full binary trees in Hamiltonian query model.

- [A, Childs, Reichardt, Spalek, Zhang, 2007]: $O(N^{1/2+o(1)})$ time algorithm for evaluating any formulas in the usual query model.

Augmented tree

Finite “tail” in one direction
Farhi et al. algorithm

Starting state:

\[ |\Psi_{\text{start}}\rangle = \sum_{j} (-1)^{j} a |2j\rangle \]

Hamiltonian \( H \),

\( H \) – adjacency matrix
What happens?

- If $T=0$, the state stays almost unchanged.
- If $T=1$, the state “scatters” into the tree.

Run for $O(\sqrt{N})$ time, check if the state $|\Psi\rangle$ is close to the starting state $|\Psi_{\text{start}}\rangle$. 
When is the state unchanged?

- $H$ – forces acting on the system.
- $(\text{State } |\Psi\rangle \text{ unchanged}) \iff H |\Psi\rangle = 0.$
What does $H\ket{\Psi} = 0$ mean?

$H$ – adjacency matrix

$H\ket{\Psi} = \sum_{(i,j)\text{-edge}} a_j = 0$

$a_1 \ a_2 \ a_3 \ldots$
Example

Formula

Augmented tree
\[ H | \Psi \rangle = 0 \text{ state} \]

Such state can be constructed whenever \( T=0 \).
Leaves with non-zero $a_i$ force $T=0$. 
\( T=1 \) case

No \( |\Psi\rangle \) with \( H|\Psi\rangle = 0 \).

Cannot place non-zero value here
Span programs [Reichardt, Špalek, 2008]

Logic formula of size $T$

Span program with witness size $T$

$O(\sqrt{T})$ query quantum algorithm

Far-reaching generalization of formula evaluation
Span programs [Karchmer, Wigderson, 1993]

- **Target vector** $v$.  

- **Input** $x_1, \ldots, x_N \rightarrow$ vectors $v_1, \ldots, v_M$.  

- **Output** $F(x_1, \ldots, x_N) = 1$ if there exist $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ such that 

  $$v = v_{i_1} + v_{i_2} + \ldots + v_{i_k}.$$
Span programs [Reichardt, 2011]

Span program with witness size $T$

$\mathcal{O}(\sqrt{T})$ query quantum algorithm
**Adversary bound [A, 2000, Hoyer, Lee, Špalek, 2007]**

- **Boolean function** $f(x_1, \ldots, x_N)$;
- **Inputs** $x = (x_1, \ldots, x_N)$;
- **Matrix A**: $A[x, y] \neq 0$ only if $f(x) \neq f(y)$
- **Theorem** Computing $f$ requires

  $$\Omega\left(\frac{\lambda(A)}{\max \lambda(A \cdot D_i)}\right)$$

  quantum queries
Span programs [Reichardt, 2009]

Optimal adversary bound

Semidefinite program (SDP)

Dual SDP

Optimal span program
Span programs [Reichardt, 2011]

Span program with witness size $T$ $\ll O(\sqrt{T})$ query quantum algorithm
What can we do with span programs?
Example

- $\text{MAJ}(x_1, x_2, x_3, x_4) = 1$ if at least 2 $x_i$ are equal to 1.
- Formula size: 8.
- Span program: 6.
Iterated thresholds

d levels – formula of size $8^d$, span program $6^d$.

$O(\sqrt{6^d})$ quantum algorithm
Singularity testing

- Matrix $A$;
- Promise $A$ is singular or all singular values of $A$ are at least $\lambda_{\text{min}}$.
- Task: distinguish between the two cases.
Singularity testing

- $\lambda_1, \lambda_2, ..., \lambda_N$ - singular values of $A$.

- **Theorem** [Belovs, 2011] Quantum algorithm for testing singularity in time $\tilde{O}\left(\frac{T\sqrt{N}}{\lambda_{av}}\right)$ where

$$
\lambda_{av} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \max\left(|\lambda_i|^2, |\lambda_{\text{min}}|^2\right)}
$$

$T$ – time to implement $A$ as a Hamiltonian
Triangle finding

- Graph G with n vertices.
- Does G contain a triangle?
- [Belovs, 2011]: $O(n^{1.29...})$. 