Faster quantum algorithm for evaluating game trees

Ben Reichardt
University of Waterloo
Problem: Evaluate the AND-OR formula with minimal queries to the input bits $x_i$. 

$$\varphi(x)$$
Motivations:

• Two-player games (Chess, Go, …)
  - Nodes ↔ game histories
  - White wins iff ∃ move s.t. ∀ responses, ∃ move s.t. …

• Decision version of min-max tree evaluation
  - inputs are real numbers
  - want to decide if minimax is ≥ 10 or not

• Model for studying effects of composition on complexity
Deterministic decision-tree complexity = Ω

Any deterministic algorithm for evaluating a read-once AND-OR formula must examine every leaf.

For balanced, binary formulas, α-β pruning is optimal ⇒ Randomized complexity $N^{0.754}$

$N^{0.51} ≤ \text{Randomized complexity} ≤ N$

[Snir '85, Saks & Wigderson '86, Santha '95]

[Heiman, Wigderson '91]

(see also K. Amano, Session 12B Tuesday)
Deterministic decision-tree complexity = $N$

$N^{0.51} \leq$ Randomized complexity $\leq N$

Quantum query complexity $= \sqrt{N}$

(very special case of the next talk)

This talk: What is the time complexity for quantum algorithms?
Farhi, Goldstone, Gutmann ’07 algorithm

- **Theorem** ([FGG ’07]): A balanced binary AND-OR formula can be evaluated in time $N^{\frac{1}{2}+o(1)}$. 
Theorem ([FGG '07]): A balanced binary AND-OR formula can be evaluated in time $N^{1/2+o(1)}$.

- Convert formula to a tree, and attach a line to the root
- Add edges above leaf nodes evaluating to one
Farhi, Goldstone, Gutmann ’07 algorithm

- **Theorem** ([FGG ’07]): A balanced binary AND-OR formula can be evaluated in time $N^{1/2+o(1)}$.
  - Convert formula to a tree, and attach a line to the root
  - Add edges above leaf nodes evaluating to one
\[ x_{11} = 1 \]

\[ \varphi(x) = 0 \]

\[ x_{11} = 0 \]

\[ \varphi(x) = 1 \]
\[ |\psi_t\rangle = e^{iA_G t} |\psi_0\rangle \]
\[ |\psi_t\rangle = e^{iA_G t} |\psi_0\rangle \]
What’s going on?

\[ \phi(x) = 0 \]

Observe: State inside tree converges to energy-zero eigenstate of the graph.
What’s going on?

Observe: State inside tree converges to \textit{energy-zero eigenstate} of the graph (supported on vertices that witness the formula’s value)

ϕ(x) = 0

○ = 0
● = 1
Energy-zero eigenvectors for AND & OR gadgets

AND:

Input adds constraints via dangling edges:

OR:

Together in a formula:

Input adds constraints via dangling edges:
Balanced AND-OR formula evaluation in $O(\sqrt{n})$ time

Squared norm = $1 + 2 + 2 + 4 + 4 + 8 + 8 + \cdots + 2^{\frac{1}{2} \log_2 n} = O(\sqrt{n})$
Effective spectral gap lemma

If $M \vec{u} \neq 0$, then $M \vec{u} \perp \text{Kernel}(M^\dagger)$

(by the SVD $M = \sum_\rho \rho |v_\rho\rangle\langle u_\rho|$)

projection of $M \vec{u}$ onto the span of the left singular vectors of $M$ with singular values $\leq \lambda$ \leq \lambda \|\vec{u}\|

\left( \text{since } \|\Pi M \vec{u}\|^2 = \sum_{\rho \leq \lambda} \rho^2 |\langle u_\rho | u \rangle|^2 \right)
Case $\varphi(x)=1$

Squared norm = $O(\sqrt{n})$

*Constant* overlap on root vertex
Case $\varphi(x)=1$

Eigenvalue-zero eigenvector with constant overlap on root vertex

Case $\varphi(x)=0$

Root vertex has $\Omega(1/\sqrt{n})$ effective spectral gap

Projection of $M\hat{u}$ onto the span of the left singular vectors of $M$ with singular values $\leq \lambda$
Case $\varphi(x)=1$

Eigenvalue-zero eigenvector with \textit{constant} overlap on root vertex

Case $\varphi(x)=0$

Root vertex has $\Omega(1/\sqrt{n})$ effective spectral gap

Quantum algorithm:
Run a quantum walk on the graph, for $\sqrt{n}$ steps from the root.

- $\varphi(x)=1 \Rightarrow$ walk is stationary
- $\varphi(x)=0 \Rightarrow$ walk mixes
Evaluating unbalanced formulas
[Ambainis, Childs, Reichardt, Špalek, Zhang ’10]

Proper edge weights on an unbalanced formula give $\sqrt{n \cdot \text{depth}}$ queries

```
\begin{circuit}
\node[and] at (0,0) (a1);
\node[or] at (1,0) (a2);
\node[and] at (2,0) (a3);
\node[or] at (3,0) (a4);
\node[and] at (4,0) (a5);
\node[or] at (5,0) (a6);
\node[or] at (6,0) (a7);
\node[and] at (7,0) (a8);
\node[or] at (8,0) (a9);
\end{circuit}
```

depth $n$, spectral gap $1/n$

“Rebalancing” Theorem:
For any AND-OR formula with $n$ leaves, there is an equivalent formula with

$n e^{\sqrt{\log n}}$ leaves, and depth $e^{\sqrt{\log n}}$

[Shouty, Cleve, Eberly ’91, Bonet, Buss ’94]

$\Rightarrow O(\sqrt{n} e^{\sqrt{\log n}})$ query algorithm

Today: $O(\sqrt{n \log n})$
Tensor-product composition

OR:

AND:

Direct-sum composition  Tensor-product composition
Direct-sum composition  Tensor-product composition

Diagram illustrating the concepts of direct-sum and tensor-product compositions.
Tensor-product composition

Properties

• **Depth** from root stays \( \leq 2 \)
  
  — \( 1/\sqrt{n} \) spectral gap

• **Graph stays sparse**—provided composition is along the maximally unbalanced formula

• **Middle vertices** ↔ **Maximal false inputs**
Final algorithm

- With direct-sum composition, large depth implies small spectral gap
- Tensor-product composition gives $\sqrt{n}$-query algorithm (optimal), but graph is dense and norm too large for efficient implementation of quantum walk
- Hybrid approach:
  - Decompose the formula into paths, longer in less balanced areas
  - Along each path, tensor-product
  - Between paths, direct-sum
- Tradeoff gives $1/(\sqrt{n} \log n)$ spectral gap, while maintaining sparsity and small norm
  \[ \Rightarrow \text{Quantum walk has efficient implementation} \]
  \[ \text{(poly-log } n \text{ after preprocessing)} \]

\[ \text{ACRŠZ '10} \quad \text{today} \]

\[ \sqrt{n} \quad e^{\sqrt{\log n}} \quad \sqrt{n} \log n \]