Is Quantum Search Practical?

George F. Viamontes

Igor L. Markov

John P. Hayes

University of Michigan, EECS

Outline

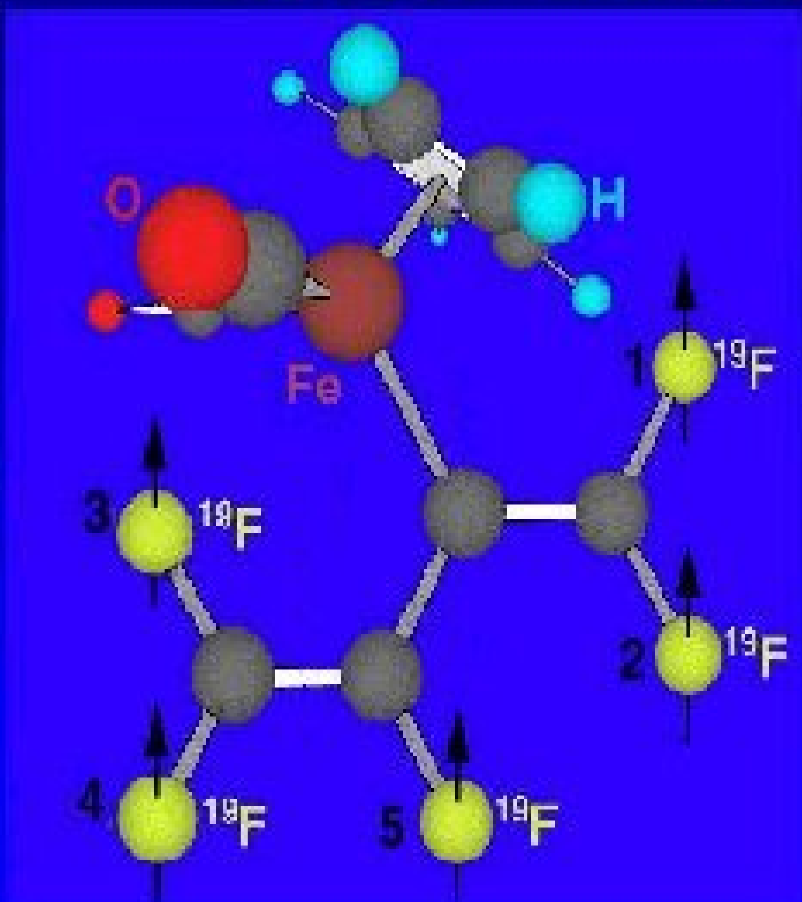
- Motivation
- Background
- Quantum search
- Practical Requirements
- Quantum search versus …
  - Classical simulation
  - Problem-specific algorithms
- Promising on-going work

Motivation

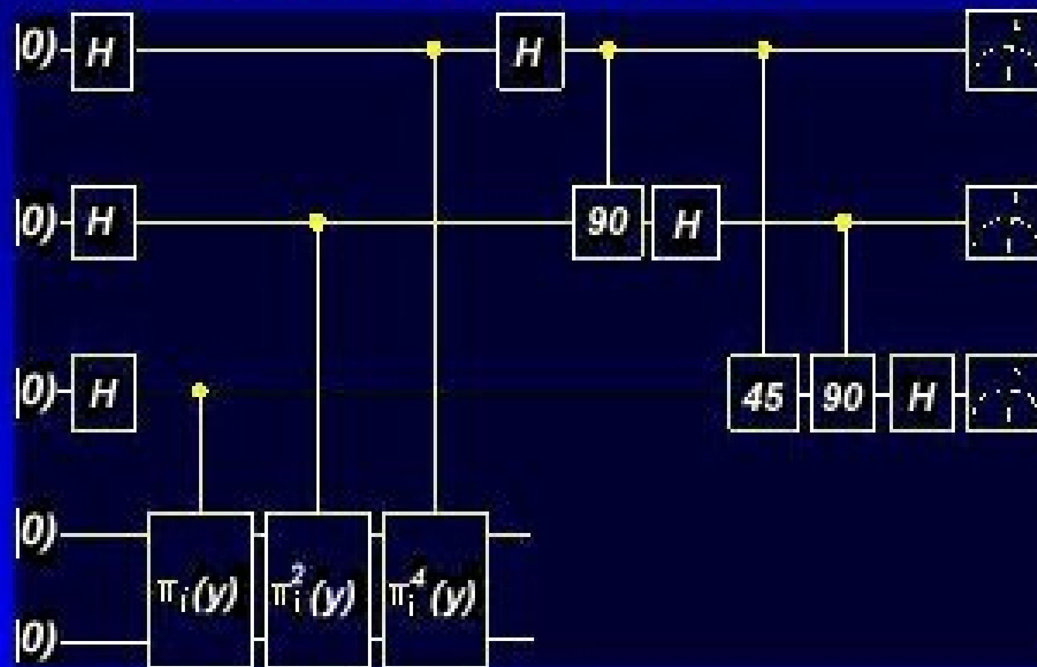
- Transistors are getting so small that quantum effects cannot be ignored
  - Why not harness them?
- Quantum circuits
  - A new model of computation: allows number-factoring in n^3 time
  - A different model of computation: allows provably faster search
  - Compare to SETs and QCAs, which perform traditional computation

5 qubit 215 Hz Q. Processor

(Vandersypen, Steffen, Breyta Yannoni, Cleve, and Chuang, 2000)



- The Molecule



- Quantum Circuit

$T_2 > 0.3$ sec ; ~ 200 gates

Source: IBM, Hot Chips Conference, 2000

Goals of This Work

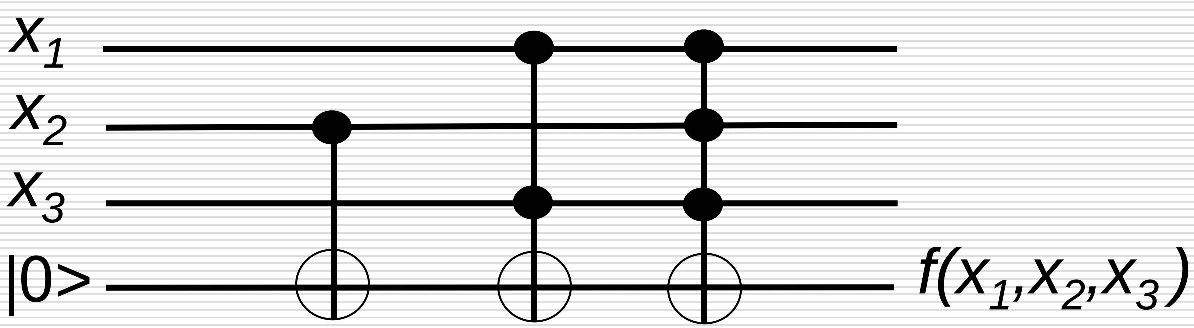
- Study one particular quantum algorithm for search
  - Here, algorithm = circuit family
- Look for practical applications
- **Note 1**: quantum communication circuits already have commercial applications
- **Note 2**: all known quantum circuits for similar problems are related

Background

- Reversible Circuits
- Linear Algebra and Probability
- Quantum Information
- Quantum Gates and Circuits
- Grover's Search Algorithm

Reversible Circuits

- Given a Boolean function $f(x_1,x_2,..,x_n)$ one can always construct a reversible circuit computing $f()$
  - Use Reed-Muller decomposition

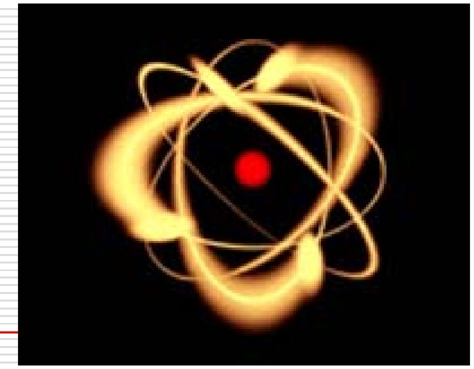


- Example: $f(x_1,x_2,x_3) = x_2 \oplus x_1 x_3 \oplus x_1 x_2 x_3$

Linear Algebra in 2^n Dimensions

- Basis vectors (basis-states) = bit-strings
  - $|00000\rangle$, $|01010\rangle$, $|11101\rangle$ etc
- Linear combinations are allowed
  - Can multiply by complex numbers, and add
  - Everything is normalized, e.g.,
    $(|0\rangle+|1\rangle)/\sqrt{2}$ and $(|00\rangle+|10\rangle)/\sqrt{2}$
- Bits & bit-strings are composed via tensor products
  - $|0\rangle \otimes |1\rangle = |01\rangle$
  - $(|0\rangle+|1\rangle)/\sqrt{2} \otimes |0\rangle = (|00\rangle+|10\rangle)/\sqrt{2}$
  - $(|00\rangle+|11\rangle)/\sqrt{2}$ is “entangled” (no decomposition)

Quantum Information



- Represents the physical state of
  - Photon polarizations, electron spins, etc
- Single-qubit: two-level quantum system
  - E.g., *spin-up* for $|0\rangle$ and *spin-down* for $|1\rangle$
  - When measuring $\alpha|0\rangle+\beta|1\rangle$, $\mathrm{Prob}|0\rangle=|\alpha|^2$
- Multiple qubits and quant. measurement
  - Linear combinations of bit-strings
  - E.g., $(|000\rangle+|010\rangle+|100\rangle+|110\rangle)/2$
  - Can only observe an individual bit-string

Classical Circuits versus Quantum

- 0-1 strings
  - E.g., one bit **{0,1}**
- Bool. Functions
  - Gates & circuits
- Primary outputs

- Lin. combinations of 0-1 strings
  - E.g., one qubit $\alpha|0\rangle+\beta|1\rangle$
- 2^n-by-2^n matrices
  - Gates & circuits
- Probabilistic measurement
  - Mostly ignored here

Quantum Gates and Circuits

- Quantum computations ***M*** are certain invertible matrices (called unitary)
- A conventional reversible gate/computation can be extended to quantum by linearity
  - E.g., a quantum inverter swaps **|0>** and **|1>**
  - Maps the state **(|0>+|1>)/√2** to itself

0	1
1	0

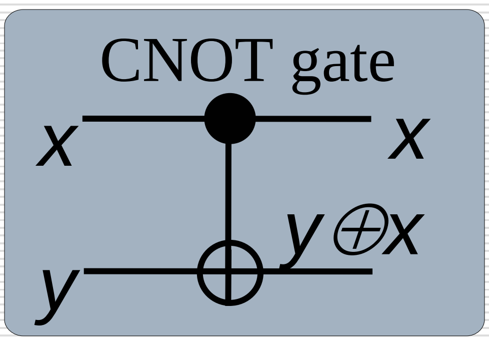
- Can apply an inverter on one of two qubits
  - E.g., **(|00>+*i*|11>)/√2** → **(|01>+*i*|10>)/√2**
- Hadamard gate: **|0>** → **(|0>+|1>)/√2**
  |1> → **(|0>-|1>)/√2**

1	1
1	-1

Quantum Circuits

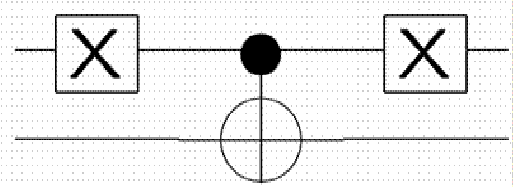
- Can apply an inverter on one of two qubits
  - E.g., **(|00>+*i*|11>)/√2** → **(|01>+*i*|10>)/√2**
- How do we describe this computation?
  - Tensor product: **Identity⊗NOT**
  - More generally: **A⊗B**

$$\begin{matrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{matrix} \qquad \begin{matrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$



$$(X \otimes \mathbf{1}) \circ (\mathtt{topCNOT}) \circ (X \otimes \mathbf{1})$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Unstructured (Database) Search

- One seeks 1 record out of N records
- One can look at single records, one at a time
- One can use a black box (oracle) that tells you whether a given record is good
- Goal: minimize the number of oracle queries
- Possible non-quantum strategies
  - Try record 1, record 2, etc … stop when rec. found
  - Or try records at random
- In the worst case, one must try all records
- On average, one will try half the records

Index Search

- If all items are indexed,
  you only need to find the right index
  - n bits for $N < 2^n$ records
- In this case, the oracle is just a Boolean function on n bits
  - This function is the input of search algorithm
- Example 1: Boolean SATisfiability
  - CNF formula represents an oracle
- Example 2: Picking locks / finding passwords
  - Verifying a password is easy

Quantum Search

- Now assume that the oracle can evaluate quantum queries
- Classical oracle: f(001)=Yes
- Quantum oracle:
  - f((|000>+|001>+|010>+|011>)/2)= (No+Yes+No+No)/2
- Can apply quantum gates before/after
- Must use quantum measurement
- Turns out: need only √N oracle queries

Grover’s Algorithm (Circuit)

- Input state: |000…0>   (n qubits)
  - Remember, the input of search algo is the oracle
- Apply a Hadamard gate on each qubit
  - Produces linear combination of *all* bit-strings
- √N identical iterations, one query in each
  - Amplify the bit-string (index) sought, by 1/√*N*, and de-amplify all remaining bit-strings
- Quantum measurement
  - Performed when the sought bit-string has highest probability of being observed

Requirements to Make this Practical

- R1: A search application *S* where classical methods do not provide sufficient scalability
- R2: An instantiation *Q*(*S*) of Grover's search for *S* with an asymptotic worst-case runtime which is less than that of any classical algorithm *C*(*S*) for *S*
- R3: A *Q*(*S*) with an actual runtime for practical instances of *S*, which is less than that of any *C*(*S*)

Application Scalability

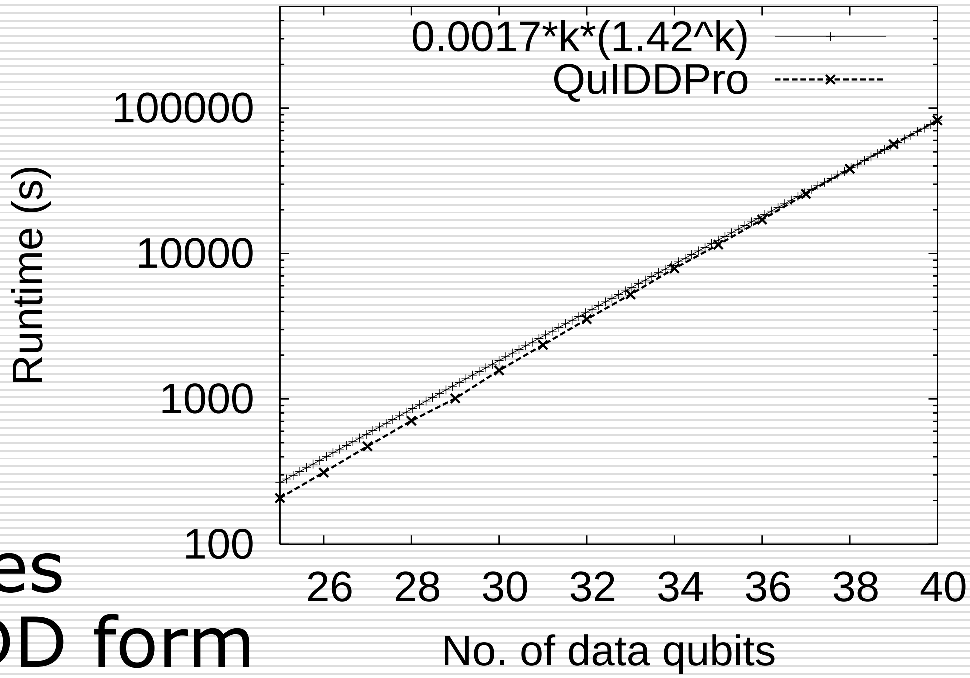
- Explicit databases
  - Store records explicitly
  - Customer support, transaction-processing
  - TeraBytes of data, distributed storage
  - Massively parallel (search is easy to ||-ze)
  - Classical methods scale so far (google.com)
- Implicit "databases" / index search
  - Combinatorial optimization & cryptography
  - Stronger demands for scalability

Oracle Implementation

- Complexity analysis proving quantum speed-up, assume oracle is a “black box”
  - I.e., absolutely no internal structure is known
- In applications, it is hard to avoid structure
  - Most “oracles” have small circuits
  - This may invalidate quantum speed-up
- Implementing the oracle can be difficult (requires circuit synthesis!)
  - If no small circuit exists/found,
    the oracle may dominate search time

Empirical Speed-up?

- Any successful quantum algorithm must outrun simulators
- Grover's search vs QuIDD Pro
  - Viamontes et al., *Quant Info Proc.*, October 2003



- This result assumes the oracle in QuIDD form
  - Creating QuIDD may be expensive

Comparing to Best Classical Algs.

- 3-SAT with *n* variables
  - Known randomized algorithm ~1.33^n
  - Grover's search ~1.41^n
- Graph 3-coloring solvable in ~1.37^n
- Grover's algorithm never finishes early
  - Classical algorithms often do
- Grover's algorithm cannot be improved w/o using structure

Presence and Use of Structure

- In many cases, structure is present but not immediately clear
  - In cryptography, no need for brute force
  - “Algebraic structure” in AES, etc
- Recent promising work on “structured” quantum search
  - *Roland and Cerf, Phys. Rev. A, Dec 2003*
  - Average-case time for 3-SAT estimated 1.31^n versus 1.33^n classical worst case

Conclusions

- Even if scalable quantum computers were available today, unstructured quantum search is not useful
  - Future breakthroughs may help
  - Will have to use structure
- Our analysis can be used for other problems touted for quantum computing, e.g., *graph isomorphism*
  - Up-coming DAC paper, new tool SAUCY

Thank you