Limits and Future of Computing

Where do we go from here?

Notes for CSC 100 - The Beauty and Joy of Computing
The University of North Carolina at Greensboro

Back to Algorithms...

Recall that algorithms provide computational solutions for problems

- Problems can be solved by multiple algorithms
- We can "rank" problems by the fastest algorithm that solves them

Some problems are *efficiently solvable*

- Algorithms solve them with time: constant, logarithmic, linear, quadratic
- In general, "polynomial time" - time bounded by n^c for some constant *c*

What about problems for which *we don't know efficient solutions*?

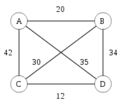
- Are there limits to what we can compute efficiently?

But there are some hard problems...

Example: The Traveling Salesman Problem (TSP)

Given a map, what is the shortest route that visits all cities and returns home?

A small example:



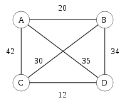
Question: What order to visit the cities (start from and return to "A")?

But there are some hard problems...

Example: The Traveling Salesman Problem (TSP)

Given a map, what is the shortest route that visits all cities and returns home?

A small example:

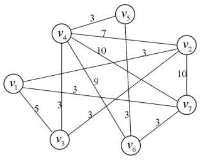


Can try all possibilities:
A-B-C-D-A = 97
A-B-D-C-A = 108
A-C-B-D-A = 141
A-C-D-B-A = 108
A-D-B-C-A = 141
A-D-C-B-A = 97

Question: What order to visit the cities (start from and return to "A")?

What happens when the number of cities grows?

What about 7 cities?



For a complete map of 7 cities, there are 6 choices for first city to visit, then 5 remaining cities for the second city, then 4, then 3, ... *So there are 6*5*4*3*2*1 = 720 routes*

Cities	Number of Routes
10	9! = 362,880
15	14! = 87,178,291,200
20	19! = 121,645,100,408,832,000

Testing 1 billion routes/sec would take 121,645,100 seconds...

... or over 3.85 years

NP-complete Problems

Some problems that share a common computational structure

Is there an algorithm that efficiently solves the TSP?

We don't know!!!

TSP (in yes/no form) is an *NP-complete problem*

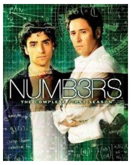
- Many important problems (thousands!) are NP-complete
- They share some common properties
  - *Can verify solutions efficiently*
- If can solve any NP-complete problem efficiently, can solve them *all* efficiently

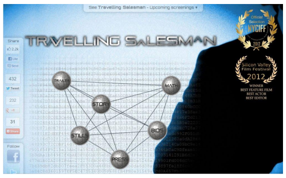
This is known as the P vs NP problem, and is the biggest unsolved problem in computer science

Clay Institute: $1 million "bounty" for a solution to this problem!

Some Awareness in Popular Media

Numb3rs TV show: Often deals with difficult problems, including NP Completeness (e.g., episode 2)





Movie with a plot revolving around TSP

http://www.travellingsalesmanmovie.com/

Beyond NP-hard Problems

Some problems are known to be solvable, but not efficiently (known!)

- "Generalized checkers": Computing optimal checkers strategy for an *nxn* checkers board

Some problems do not have algorithmic solutions at all!

The "Halting Problem"

- Programs are just bits stored in files, just like any other file
- Therefore, programs can be inputs to other programs
- The Halting Problem: Given a program to run with a specific input, will it eventually halt and give an answer?

Obviously would be great if we could solve (no more programs that hang!)

Unfortunately, *the Halting Problem is undecidable (uncomputable)*: no algorithm, no matter how clever or complex, can solve the Halting Problem for all inputs (i.e., for all programs)

Coping with NP-hard Problems

Lots of very important, practical problems are NP-hard

Is it just hopeless*?

Let's look at some strange cutting-edge research directions...

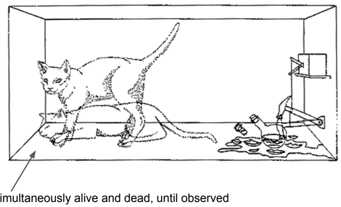
* If you want exact answers, that is. Approximation algorithms are sometimes "good enough"!

Quantum Computing

The physics of the matter...

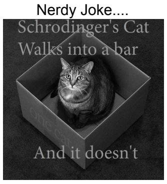
In the strange world of quantum physics, particles/matter can be in multiple states simultaneously - in *quantum superposition*.

Classic physics thought experiment: Schroedinger's cat



Simultaneously alive and dead, until observed
(*so the cat isn't an observer? but that's not really the point...*)

Nerdy Joke....



Quantum Computing

So what does this mean for computing?

In standard computing, bits are 0 or 1

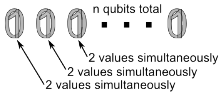
In quantum computing, qubits can simultaneously be both

- *Computations work on a superposition of values until observed*

So what?

- If working with data in many states simultaneously, can potentially do many calculations simultaneously!

A really over-simplified view of quantum computing power



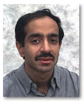
Taken altogether: 2^n simultaneous states

Quantum Computing

What can you do with a quantum computer?

Grover's Algorithm
for database searching

Invented in 1996



Lov K. Grover

Problem: Searching an unsorted list
(*like "contains" in BYOB!*)

Classical: Requires linear (n) time

Quantum: Grover's algorithm works in $n^{1/2}$ (square root of n) time.

Searching for a 64-bit crypto key:
Classical: 2^{64} steps (584 years @1GHz)
Quantum: 2^{32} steps (4 seconds @1GHz)

Shor's Algorithm
for factoring

Invented in 1994

Peter Shor

Problem: Factor a large number
("large" can mean hundreds of digits or more)

Importance: If you can factor, you can break RSA encryption

Classical: Worse than polynomial
("trial division" is exponential)

Quantum: Proportional to n^3

Quantum is an exponential improvement!

Quantum Computing

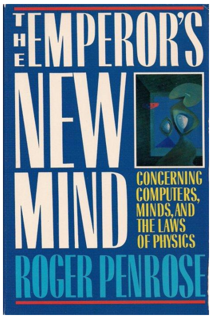
An interesting read...

The Emperor's New Mind
by Roger Penrose

Won the 1990 Science Book Prize

Central claim: Human consciousness is non-algorithmic, and quantum physics plays a key role in human consciousness.

So... are quantum computers essential to "real AI"?

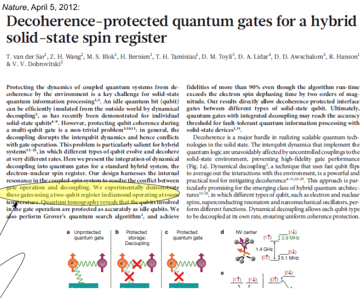


Quantum Computing

So, is this real or just mathematical games?

In the past few years quantum computing has gotten a lot of attention due to practical advancements...

From *Nature*, April 5, 2012:



Decoherence-protected quantum gates for a hybrid solid-state spin register

T. van der Sar[1], Z. H. Wang[2], M. S. Blok[1], H. Bernien[1], T. H. Taminiau[1], D. M. Toyli[3], D. A. Lidar[4], D. D. Awschalom[3], R. Hanson[1] & V. V. Dobrovitski[2]

Protecting the dynamics of coupled quantum systems from decoherence by the environment is a key challenge for solid-state quantum information processing[1,2]. An idle quantum bit (qubit) can be efficiently insulated from the outside world by dynamical decoupling[3], as has recently been demonstrated for individual solid-state qubits[4–9]. However, protecting qubit coherence during a multi-qubit gate is a non-trivial problem[3,10,11]: in general, the decoupling disrupts the interqubit dynamics and hence conflicts with gate operation. This problem is particularly salient for hybrid systems[12–22], in which different types of qubit evolve and decohere at very different rates. Here we present the integration of dynamical decoupling into quantum gates for a standard hybrid system, the electron–nuclear spin register. Our design harnesses the internal resonance in the coupled-spin system to resolve the conflict between gate operation and decoupling. We experimentally demonstrate these gates using a two-qubit register in diamond operating at room temperature. Quantum tomography reveals that the qubits involved in the gate operation are protected as accurately as idle qubits. We also perform Grover's quantum search algorithm[1], and achieve fidelities of more than 90% even though the algorithm run-time exceeds the electron spin dephasing time by two orders of magnitude. Our results directly allow decoherence-protected interface gates between different types of solid-state qubit. Ultimately, quantum gates with integrated decoupling may reach the accuracy threshold for fault-tolerant quantum information processing with solid-state devices[1,11].

Decoherence is a major hurdle in realizing scalable quantum technologies in the solid state. The interqubit dynamics that implement the quantum logic are unavoidably affected by uncontrolled couplings to the solid-state environment, preventing high-fidelity gate performance (Fig. 1a). Dynamical decoupling[3], a technique that uses fast qubit flips to average out the interactions with the environment, is a powerful and practical tool for mitigating decoherence[4–11,23–25]. This approach is particularly promising for the emerging class of hybrid quantum architectures[12–22], in which different types of qubit, such as electron and nuclear spins, superconducting resonators and nanomechanical oscillators, perform different functions. Dynamical decoupling allows each qubit type to be decoupled at its own rate, ensuring uniform coherence protection.

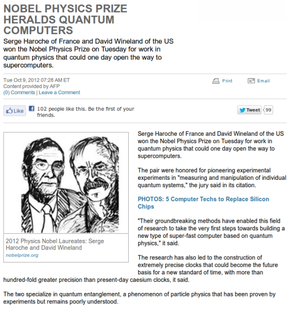
A two-qubit register... not useful, but a breakthrough nonetheless...

Quantum Computing

So, is this real or just mathematical games?

October 2012 announcement of Nobel Prize for Physics - for work that could help build quantum computers....

From "Discovery News", Oct 9, 2012:



NOBEL PHYSICS PRIZE HERALDS QUANTUM COMPUTERS

Serge Haroche of France and David Wineland of the US won the Nobel Physics Prize on Tuesday for work in quantum physics that could one day open the way to supercomputers.

Tue Oct 9, 2012 07:28 AM ET
Content provided by AFP
(0) Comments | Leave a Comment

2012 Physics Nobel Laureates: Serge Haroche and David Wineland
nobelprize.org

Serge Haroche of France and David Wineland of the US won the Nobel Physics Prize on Tuesday for work in quantum physics that could one day open the way to supercomputers.

The pair were honored for pioneering experimental experiments in "measuring and manipulation of individual quantum systems," the jury said in its citation.

PHOTOS: 5 Computer Techs to Replace Silicon Chips

"Their groundbreaking methods have enabled this field of research to take the very first steps towards building a new type of super-fast computer based on quantum physics," it said.

The research has also led to the construction of extremely precise clocks that could become the future basis for a new standard of time, with more than hundred-fold greater precision than present-day caesium clocks, it said.

The two specialize in quantum entanglement, a phenomenon of particle physics that has been proven by experiments but remains poorly understood.

DNA Computing

Basic idea: DNA is just a set of instructions on how to build a living organism, and constructing that organism is "executing the code"

So: Can we synthesize instruction sequences in DNA to compute a solution to a non-biological problem?

Why: DNA has incredibly high storage density!

One cubic centimeter of DNA holds more information than a trillion CDs.

DNA Computing

Are these real?

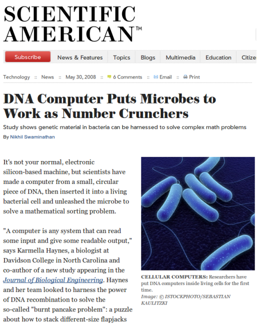
Yes, they can be built!

Existing DNA computers, like the one reported in 2008, are very simplistic ("two-pancake" problem, similar to "two-qubit" quantum computer).

- Used genetically engineered E. coli bacteria
- Not useful as computing systems yet, but interesting "proof of concept"

The potential (using real/realistic numbers):

- 1000 operations per second,
- With 100 billion in parallel,
- Gives 100 trillion operations per second.



Technology :: News :: May 30, 2008 :: 6 Comments :: Email :: Print

DNA Computer Puts Microbes to Work as Number Crunchers

Study shows genetic material in bacteria can be harnessed to solve complex math problems

By Nikhil Swaminathan

It's not your normal, electronic silicon-based machine, but scientists have made a computer from a small, circular piece of DNA, then inserted it into a living bacterial cell and unleashed the microbe to solve a mathematical sorting problem.

"A computer is any system that can read some input and give some readable output," says Karmella Haynes, a biologist at Davidson College in North Carolina and co-author of a new study appearing in the *Journal of Biological Engineering*. Haynes and her team looked to harness the power of DNA recombination to solve the so-called "burnt pancake problem": a puzzle about how to stack different-size flapjacks

CELLULAR COMPUTERS: Researchers have put DNA computers inside living cells for the first time.
Image: © ISTOCKPHOTO/SEBASTIAN KAULITZKI

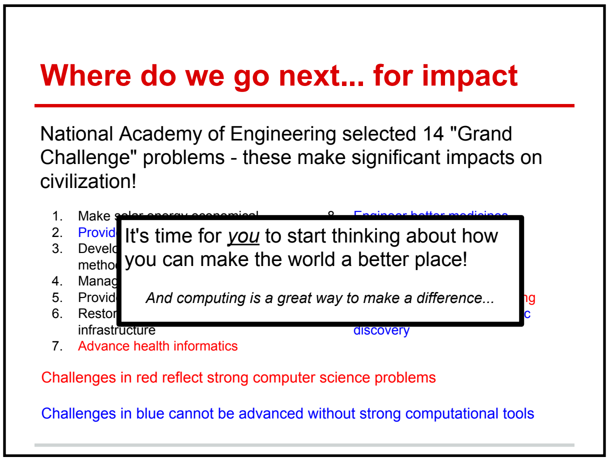
Where do we go next... for impact

National Academy of Engineering selected 14 "Grand Challenge" problems - these make significant impacts on civilization!

1. Make solar energy economical
2. Provide energy from fusion
3. Develop carbon sequestration methods
4. Manage the nitrogen cycle
5. Provide access to clean water
6. Restore and improve urban infrastructure
7. Advance health informatics
8. Engineer better medicines
9. Reverse-engineer the brain
10. Prevent nuclear terror
11. Secure cyberspace
12. Enhance virtual reality
13. Advance personalized learning
14. Engineer the tools of scientific discovery

Challenges in red reflect strong computer science problems

Challenges in blue cannot be advanced without strong computational tools



Where do we go next... for impact

National Academy of Engineering selected 14 "Grand Challenge" problems - these make significant impacts on civilization!

1. Make solar energy economical
2. Provid
3. Develo metho
4. Manag
5. Provid
6. Restor infrastructure
7. Advance health informatics

8. Engineer better medicines

ng

c discovery

It's time for *you* to start thinking about how you can make the world a better place!

And computing is a great way to make a difference...

Challenges in red reflect strong computer science problems

Challenges in blue cannot be advanced without strong computational tools