Introduction to quantum algorithms

Gabriel Semanišin

Institute of Computer Science
P.J. Šafárik University, Faculty of Science
Košice, Slovakia
e-mail: gabriel.semanisin@upjs.sk

eduQUTE 2018

Motto

Michelangelo Buonarotti:

There is no greater loss than time which has been wasted

One real life problem

A producent

- has cca 9000 customers;
- every day has to realise approximately 900 orders;
- has 15 lorries of various capacity.

One real life problem

A producent

- has cca 9000 customers;
- every day has to realise approximately 900 orders;
- has 15 lorries of various capacity.

Every duty of its dispatcher:

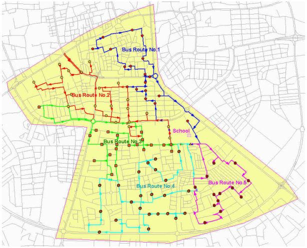
To find such a schedule that allows to satisfy all orders, respect all restrictions of customers and minimalise the transport costs.

VRP - Vehicle Routing Problem

The vehicle routing problem (VRP) is a combinatorial optimization and integer programming problem seeking to service a number of customers with a fleet of vehicles..

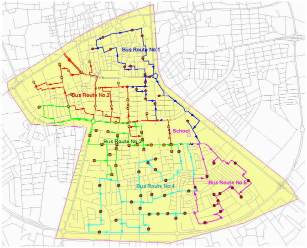
VRP - Vehicle Routing Problem

The vehicle routing problem (VRP) is a combinatorial optimization and integer programming problem seeking to service a number of customers with a fleet of vehicles..



VRP - Vehicle Routing Problem

The vehicle routing problem (VRP) is a combinatorial optimization and integer programming problem seeking to service a number of customers with a fleet of vehicles..



A **special case** of VRP is Travelling salesman problem (TSP).

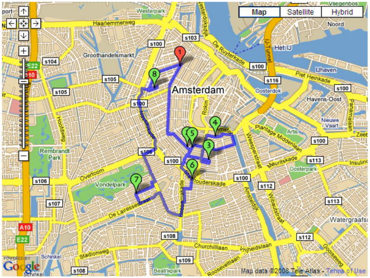
Travelling salesman problem

Given a list of cities and their pairwise distances, the task is to find the shortest possible route that visits each city exactly once and returns to the origin city.

Travelling salesman problem

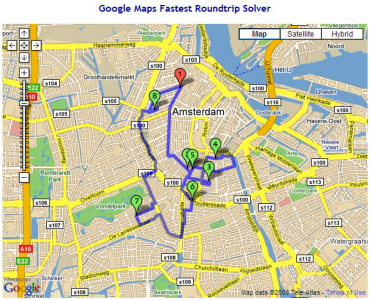
Given a list of cities and their pairwise distances, the task is to find the shortest possible route that visits each city exactly once and returns to the origin city.

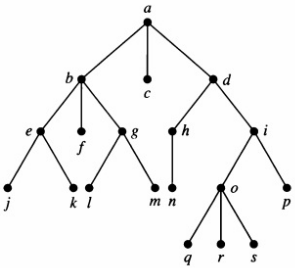
Google Maps Fastest Roundtrip Solver



Travelling salesman problem

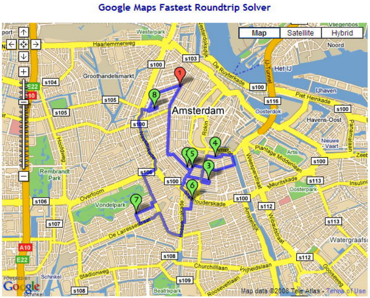
Given a list of cities and their pairwise distances, the task is to find the shortest possible route that visits each city exactly once and returns to the origin city.

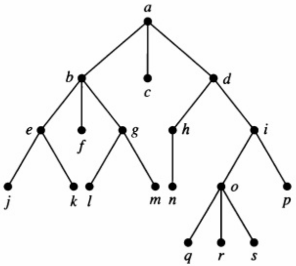




Travelling salesman problem

Given a list of cities and their pairwise distances, the task is to find the shortest possible route that visits each city exactly once and returns to the origin city.





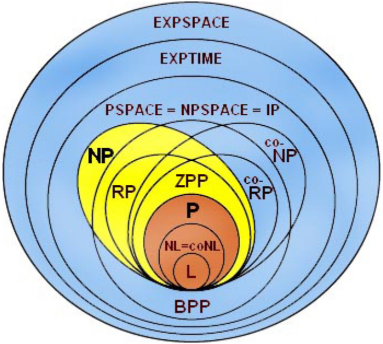
Problem: the number of branches can tend to $n!$

Are there some efficient algorithms for TSP?

Unfortunately, until now we know just **exponential time** algorithms.

Are there some efficient algorithms for TSP?

Unfortunately, until now we know just **exponential time** algorithms.

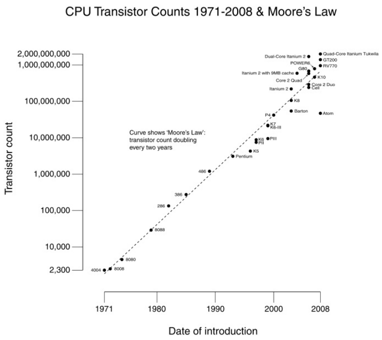


What is the behaviour of factorial function

$5! = 120 \quad 70! = 1,19.10^{100} \quad 1000! = 4,02.10^{2567}$

What is the behaviour of factorial function

$5! = 120 \quad 70! = 1,19.10^{100} \quad 1000! = 4,02.10^{2567}$



Moor's law

is a rule of thumb in the history of computing HW **whereby the number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years.**

How many instructions can computer perform during one human life

1 life $\approx 10^2$ years

How many instructions can computer perform during one human life

1 life $\approx 10^2$ years

1 life $\approx 10^5$ days

How many instructions can computer perform during one human life

1 life $\approx 10^2$ years

1 life $\approx 10^5$ days

1 life $\approx 10^7$ hours

How many instructions can computer perform during one human life

1 life $\approx 10^2$ years

1 life $\approx 10^5$ days

1 life $\approx 10^7$ hours

1 life $\approx 10^9$ minutes

How many instructions can computer perform during one human life

1 life $\approx 10^2$ years

1 life $\approx 10^5$ days

1 life $\approx 10^7$ hours

1 life $\approx 10^9$ minutes

1 life $\approx 10^{11}$ seconds

How many instructions can computer perform during one human life

1 life $\approx 10^2$ years

1 life $\approx 10^5$ days

1 life $\approx 10^7$ hours

1 life $\approx 10^9$ minutes

1 life $\approx 10^{11}$ seconds

1 life $\approx 10^{22}$ instructions

How many instructions can computer perform during one human life

1 life $\approx 10^2$ years

1 life $\approx 10^5$ days

1 life $\approx 10^7$ hours

1 life $\approx 10^9$ minutes

1 life $\approx 10^{11}$ seconds

1 life $\approx 10^{22}$ instructions

1000 orders

How many instructions can computer perform during one human life

1 life $\approx 10^2$ years

1 life $\approx 10^5$ days

1 life $\approx 10^7$ hours

1 life $\approx 10^9$ minutes

1 life $\approx 10^{11}$ seconds

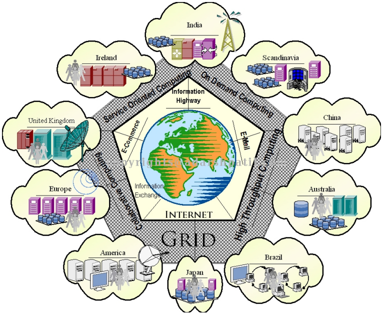
1 life $\approx 10^{22}$ instructions

1000 orders

$1000! \approx 4,02.10^{2567}$
branches of a computation

Classical computers does not provide a solution

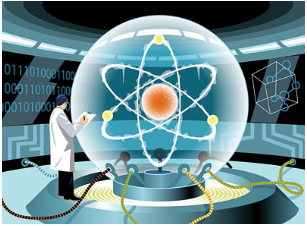
Classical processor will shortly reach **the natural physical limits**. (Although some other perspectives provides nanotechnology .)



For **hard problems** even distributed computing (e.g. grids, . . .) provides acceleration of very limited importance.

Quantum computing

Quantum computers provides an enormous power of **parallelism.**



Why quantum computing

- Why consider quantum computing at all?
- Can quantum computers do what classical ones cannot?
- Where lie the difference between the classical and quantum information processing?
- Can quantum computers solve some practically important problems much more effectively?
- Where does the power of quantum computing come from?

Why quantum computing

- Where are the drawbacks and bottlenecks of quantum computing?
- How feasible are (powerful) quantum computers and really important quantum information processing applications?
- Are not current computers quantum?
- Can quantum computers eventually replace classical ones?

Why it is quantum computation interesting for me

Interdisciplinarity

- Computer Science
- Mathematics
- Physics

Interdisciplinarity within Mathematics

- Mathematical Analysis
- Computational Complexity
- Linear Algebra
- Number Theory
- Geometry

Quantum Computation

Different areas related to Quantum computations

- Hardware for Quantum Computers
- ...
- Quantum Algorithms

Gedanken experiment?

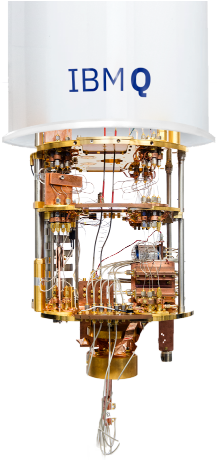
Gedanken experiment

A thought experiment (from the German term Gedankenexperiment) - in the broadest sense is the use of a hypothetical scenario to help us understand the way things actually are.

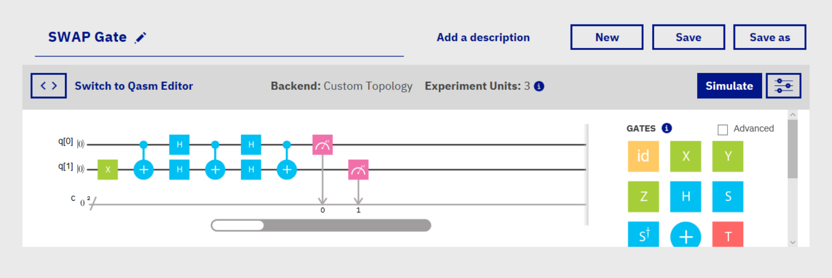
There are many different kinds of thought experiments. All thought experiments, however, employ a methodology that is a priori, rather than empirical, in that they do not proceed by observation or physical experiment.

Thought experiments have been used in a variety of fields, including philosophy, law, physics, and mathematics. In physics and other sciences, notable thought experiments date from the 19th, and especially the 20th Century, but examples can be found at least as early as Galileo.

IBM Q changes the situation



Computation on IBM Q



Quantum computer

A quantum computer is any device for computation that makes direct use of distinctively quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data.

The basic principle of quantum computation is that the

- quantum properties can be used to represent and structure data
- quantum mechanisms can be devised and built to perform operations with this data.

Qubit

Quantum information is physical information that is held in the state of a quantum system.

The most popular unit of quantum information is the qubit, a two-state quantum system.

However, unlike classical digital states (which are discrete), a two-state quantum system can actually be in a superposition of the two states at any given time.

Qubit implementation

An example of an implementation of qubits for a quantum computer could start with the use of particles with two spin states:

$$|\uparrow\rangle \text{ and } |\downarrow\rangle .$$

Qubit

But in fact any system possessing an observable quantity A which

- is conserved under time evolution
- has at least two discrete and sufficiently spaced consecutive eigenvalues,

is a suitable candidate for implementing a qubit, because any such system can be mapped onto an effective spin $\pm\frac{1}{2}$.

Quantum information processing

Quantum information differs from classical information in several respects, among which we note the following:

- It cannot be read without the state becoming the measured value.
- An arbitrary **state cannot be cloned.**
- The **state may be in a superposition of basis values.**

However, despite this, the amount of information that can be retrieved in a single qubit is equal to one bit.
The ability to manipulate quantum information enables us to perform tasks that would be unachievable in a classical context:

- unconditionally secure transmission of information
- efficient computation for some complex problems

Quantum Measurement

Quantum state of a system

is a mathematical object that fully describes the quantum system.

Once the quantum state has been prepared, some aspect of it is measured (for example, its position or energy). The expected result of the measurement is in general described not by a single number, but by a probability distribution. The measurement process is often said to be random and indeterministic.

Another important aspect of measurement is wavefunction collapse.

Quantum Measurement Postulate

It is a postulate of quantum mechanics that all measurements have an associated operator (called an observable operator, or just an observable), with the following properties:

- the observable is a Hermitian (self-adjoint) operator mapping a Hilbert space into itself
- the observable's eigenvalues are real and the possible outcomes of the measurement are precisely the eigenvalues of the given observable
- for each eigenvalue there are one or more corresponding eigenvectors, which will make up the state of the system after the measurement
- the observable has a set of eigenvectors which span the state space - it follows that each observable generates an orthonormal basis of eigenvectors (physically, this is the statement that any quantum state can always be represented as a superposition of the eigenstates of an

Classical vs. Quantum Computation

States

ordinary computer

$\square$ $\square$ $\cdots$ $\square$ bits

x_1 x_2 $\ldots$ x_n where $x_j \in \mathbb{B}$

quantum computer

$\square$ $\square$ $\cdots$ $\square$ qubits

basis: $|x_1, x_2, \ldots, x_n\rangle$

where $\sum_{x \in \mathbb{B}^n} c_x |x\rangle$ and $\sum_{x \in \mathbb{B}^n} |c_x|^2 = 1$

Transformations

ordinary computer

Transformations are functions from $\mathbb{B}^n$ to $\mathbb{B}^n$.

quantum computer

Transformations are unitary operators, i.e. operators that preserve the length $\sum_{x \in \mathbb{B}^n} |c_x|^2$ of each vector $\sum_{x \in \mathbb{B}^n} c_x |x\rangle$.

Inner-product space

Definition

An *inner-product space* H is a complex , equipped with an inner product $\langle\cdot|\cdot\rangle : H \times H \longrightarrow \mathbb{C}$ satisfying the following axioms for any vectors $\phi, \psi, \phi_1, \phi_2 \in H$, an any $c_1, c_2 \in \mathbb{C}$:

- $\langle\phi|\psi\rangle = \langle\psi|\phi\rangle^*$;
- $\langle\phi|\phi\rangle \geq 0$ and $\langle\phi|\phi\rangle = 0$ if and only if $\phi = 0$;
- $\langle\psi|c_1\phi_1 + c_2\phi_2\rangle = c_1\langle\psi|\phi_1\rangle + c_2\langle\psi|\phi_2\rangle$.

The inner product introduces on H the norm $||\psi|| = \sqrt{\langle\psi|\psi\rangle}$ and the metric (Euclidean distance) $\text{dist}(\phi, \psi) = ||\phi - \psi||$.

Hilbert space

Definition

An inner-product space H is called *complete*, if for any sequence $\{\phi_i\}_{i=1}^{\infty}$ with $\phi_i \in H$, and with the property that $\lim_{i,j\to\infty} ||\phi - \phi_i|| = 0$, there is a unique element $\phi \in H$ such that $\lim_{i\to\infty} ||\phi - \phi_i|| = 0$. A complete inner-product space is called a *Hilbert space*.

Definition

A *linear operator* on a Hilbert space H is a linear mapping $A : H \longrightarrow H$.

Dual Hilbert Space

For each $\phi \in H$ the mapping $f_\phi : H \longrightarrow \mathbb{C}$ defined by $f\phi(\psi) = \langle\phi|\psi\rangle$ is a linear mapping on H.

Theorem

To each continuous linear mapping $f : H \longrightarrow \mathbb{C}$ there exist a unique $\phi_f \in H$ such that $f(\psi) = \langle\phi_f|\psi\rangle$ for any $\psi \in H$.

The space of linear mapping (called also *functionals*) of a Hilbert space H forms again a Hilbert space, called *dual Hilbert space or conjugate Hilbert space*.

A vector ϕ of a Hilbert space is denoted $|\phi\rangle$ and referred as a *ket-vector*. The corresponding functional is denoted $\langle\psi|$ and referred as a *bra-vector*.

Unitary operators

Unitary matrix

is an n by n complex matrix U satisfying the condition

$$UU^{+} = I_n,$$

where I_n is the identity matrix and U^{+} is the conjugate transpose (also called the Hermitian adjoint) of U.

Note that a matrix U is unitary if and only if it has an inverse which is equal to its conjugate transpose U^{+}.

Important feature

Unitary matrix preserves inner-products, i.e. $\langle Ux|Uy\rangle = \langle x|y\rangle$.

Properties of unitary operators

Rather simple premises imply that there exists unitary mappings

$$U(t) : H_n \to H_n$$

which govern the time evolution in the following way: if

$$\psi(0) = c_0 \left|0\right\rangle + c_1 \left|1\right\rangle + \cdots + c_{n-1} \left|n-1\right\rangle$$

is the state of the system at time $t = 0$, then the state at time t is given by

$$\psi(t) = U(t)\psi(0).$$

Properties of unitary operators

Moreover, the unitary mappings $U(t)$ satisfy

$$U(t_1 + t_2) = U(t_1)U(t_2)$$

. If we further assume that the mapping $U(t)$ is continuous, it follows that there exists a self-adjoint mapping $H : H_n \to H_n$ such that

$$U(t) = e^{itH}.$$

Such a mapping H is called the Hamiltonian operator of the system and is of course determined by the physical conditions. A componentwise differentiation of the last equality implies that

$$i\frac{\mathrm{d}}{\mathrm{d}t}\psi(t) = H\psi(t).$$

This equation is called Schrődinger's equation of motion.

One qubit register

For one qubit register we have two possible states represented by the vectors:

$$|\uparrow\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |\downarrow\rangle = |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Very often we need also so-called dual (or Fourier) base:

$$|0'\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \quad |1'\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Hadamard transformation

Hadammard transformation is a unitary transformation that transform the standard base to the dual base. It is represented by the matrix

$$H_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

By an application of the Hadammard operator we have

$$|0'\rangle = H_1 |0\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$$

$$|1'\rangle = H_1 |1\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

Two qubit register

For two qubit register we have four possible states represented by the vectors:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and

$$|01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \ |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \ |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

n-dimensional Hadammard Transformation

n-dimensional Hadammard transformation has form

$$H_n = \otimes_{i=1}^{n} H_1$$

and an application of H_n to the n-dimensional state $|0^{(n)}\rangle$ yields

$$H_n \underbrace{|00\ldots0\rangle}_{n} = H_n \left|0^{(n)}\right\rangle = \left|0'^{(n)}\right\rangle = \underbrace{|0'0'\ldots0'\rangle}_{n},$$

where

$$\left|0'^{(n)}\right\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=1}^{n} |i\rangle .$$

Quantum parallelism

An application of an operator A to a state

$$|\phi\rangle = \sum_{i=0}^{2^n-1} c_i|i\rangle$$

yield to

$$A|\phi\rangle = \sum_{i=0}^{2^n-1} c_i A|i\rangle,$$

i.e. by a single application of the operator A (on a "single processor"), exponentially many operations on basis states are performed. This phenomenon is called *quantum parallelism*.

Quantum gates

A quantum gate or quantum logic gate is a basic quantum circuit operating on a small number of qubits.

They are the analogues for quantum computers to classical logic gates for conventional digital computers. Quantum logic gates are reversible, unlike many classical logic gates.

A set of universal quantum gates is any set of gates to which any operation possible on a quantum computer can be reduced, that is, any other unitary operation can be expressed as a finite sequence of gates from the set.

Some universal classical logic gates, such as the Toffoli gate, provide reversibility and can be directly mapped onto quantum logic gates. Quantum logic gates are represented by unitary

f-controlled NOT

Proposition

Given a function $f : \{0, 1, \ldots, 2^m - 1\} \longrightarrow \{0, 1, \ldots, 2^n - 1\}$*.*
There exists a unitary transformation U_f *such that*

$$|\phi\rangle = \frac{1}{\sqrt{2^m}} \sum_{i=0}^{2^m-1} |x, 0\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^m}} \sum_{i=0}^{2^m-1} |x, f(x)\rangle.$$

Amplitude sign changing operator

Given a function $f : \{0, 1, \ldots, 2^N - 1\} \longrightarrow \{0, 1\}$.

Using U_f we can construct a mapping V_f operating as $V_f |x\rangle = (-1)^{f(x)} |x\rangle$.

This operator is called sign-changing operator.

Inversion about the average D_n

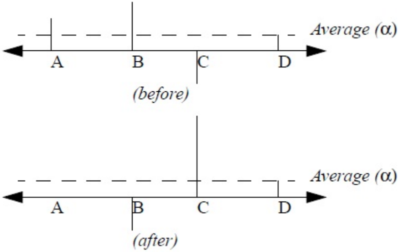
It acts in the following way:

$$D_n : \sum_{i=0}^{2^n-1} a_i \left|x_i\right\rangle \to \sum_{i=0}^{2^n-1} (2E - a_i) \left|x_i\right\rangle ,$$

where E is the average of the values $\{a_i : i = 0, 1, \ldots, 2^n - 1\}$.
The corresponding matrix has form:

$$\begin{pmatrix} -1 + \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \frac{2}{2^n} & -1 + \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & -1 + \frac{2}{2^n} \end{pmatrix},$$

How inversion about the average D_n acts



The composed operator $D_n.V_f$

Since V_f is diagonal matrix with diagonal $(-1, \underbrace{1, \ldots, 1}_{2^n-1})$, the operator $D_n.V_f$ has the form:

$$\begin{pmatrix} 1 - \frac{1}{2^{n-1}} & \frac{1}{2^{n-1}} & \frac{1}{2^{n-1}} & \cdots & \frac{1}{2^{n-1}} \\ -\frac{1}{2^{n-1}} & \frac{1}{2^{n-1}} - 1 & \frac{1}{2^{n-1}} & \cdots & \frac{1}{2^{n-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{2^{n-1}} & \frac{1}{2^{n-1}} & \frac{1}{2^{n-1}} & \cdots & \frac{1}{2^{n-1}} - 1 \end{pmatrix}.$$

Deutsch's problem

Informally, we want to guess whether a give coin is genuine (with head on one side and tail on the other) of fake (with both sides the same). The question is how many times we need to look at the coin to find out which case it is.

Problem (Deutsch's XOR problem)

Given a function $f : \{0, 1\} \longrightarrow \{0, 1\}$*, as a black box, the task is to determine whether* $f(0) \oplus f(1) = 0$*, or* 1 *(i.e. whether* f *is constant or balanced).*

Randomized algorithm

Algorithm (original randomized solution)

Let W_f be the unitary mapping of $|x, y\rangle$ into $|x, y \oplus f(x)\rangle$ - so-called f-controlled NOT. One application of W_f to the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle$ yields to the state $\frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle)$, which can be written in the standard and dual basis as follows: if f is constant $\frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle) = \frac{1}{\sqrt{2}}(|0', 0'\rangle + (-1)^{f(0)}|0', 1'\rangle)$ and if f is balanced:
$\frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle) = \frac{1}{\sqrt{2}}(|0', 0'\rangle + (-1)^{f(0)}|1', 1'\rangle)$.

If the measurement of the second qubit provides 0 we have lost all information about f. However, if the measurement yields 1, then the measurement of the first qubit yields the correct result.

Deterministic algorithm

Algorithm (deterministic solution)

1. *By an application of H_2 to $|0\rangle|1\rangle$ we get* $\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = \frac{1}{2}(|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle)$.
2. *By an application of U_f we obtain* $\frac{1}{2}(|0\rangle(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle) + |1\rangle(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)) = \frac{1}{2}\left(\sum_{x=0}^{1}(-1)^{f(x)}|x\rangle\right)(|0\rangle - |1\rangle) = (-1)^{f(0)}|(f(0) \oplus f(1))'\rangle|1'\rangle$.

By measuring of the first bit, with respect to the dual basis, we can immediately see whether f is constant or balanced.

Computation on IBM Q

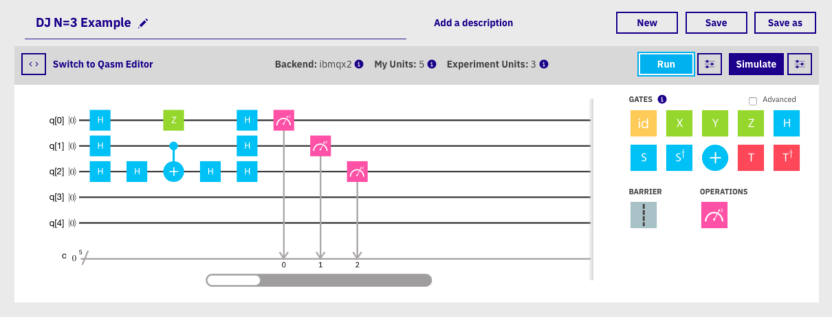


Figure: Deutsch-Jozsa for $f(x) = x_0 \oplus x_1 x_2$

Search problem

Problem (Unsorted database search)

Instance: *Given a positive integer* n *and an element* x^* *belonging to an unsorted database with* 2^n *elements equipped with a function* f *such that* $f(x) = 1$ *whenever* $x = x^*$ *and* $f(x) = 0$ *in all other cases.*
Goal: *Find the element* x^**.*

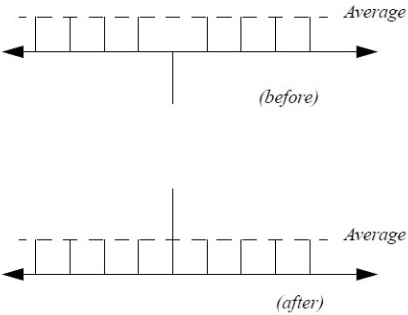
The time complexity for classical search algorithm is $O(2^n)$.

Grover's algorithm description

Algorithm (Grover's algorithm)

1. *Using Hadamard transformation H_n create the state* $|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$.
2. *Apply the sign-changing operator V_f to $|\phi\rangle$ to provide* $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle$.
3. *Apply the inversion about average operator D_n to the state received in the previous step.*
4. *Iterate $\lceil \frac{\pi}{4}\sqrt{2^n} \rceil$ times steps 2 and 3.*
5. *Measure the x-register to get x_0. If $f(x_0) \neq 1$ go to step 1.*

The correctness of Grover's algorithm



Estimation of the number of loops (1)

Let us label the qubits of the register X by x_i, $i = 0, 1, \ldots, 2^n - 1$. Without loss of generality we can assume that the register X is arranged so that the first qubit x_0 is equal to x^*. Due to the properties of the used operators (they act in the same manner on all qubits x_i, $i = 1, 2, \ldots, 2^n - 1$), one can quite easily see that in any stage of the computation, the actual state of the register can be expressed as:

$$\alpha \left| x_0 \right\rangle + \sum_{i=1}^{2^n-1} \beta \left| x_i \right\rangle ,$$

where $|\alpha|^2 + \sum_{i=1}^{2^n-1} |\beta|^2 = |\alpha|^2 + (2^n - 1)|\beta|^2 = 1$.

Estimation of the number of loops (2)

Since $f(x) = 1$ if and only if $x = x^*$ (i.e. $i = 0$) the sign-changing operator V_f produces a new state that is equal to

$$-\alpha |x_0\rangle + \sum_{i=1}^{2^n-1} \beta |x_i\rangle .$$

Estimation of the number of loops (3)

Let us denote by $(\alpha_t, \underbrace{\beta_t, \ldots, \beta_t}_{2^n-1})$ the vector of coefficients after t iterations, $t = 0, 1, \ldots$. The sign-changing operator V_f and the inversion about average operator D_n transforms the vector of coefficients $(\alpha_t, \underbrace{\beta_t, \ldots, \beta_t}_{2^n-1})$ to the new vector $(\alpha_{t+1}, \underbrace{\beta_{t+1}, \ldots, \beta_{t+1}}_{2^n-1})$ in the following way:

$$\begin{pmatrix} \alpha_{t+1} \\ \beta_{t+1} \\ \vdots \\ \beta_{t+1} \end{pmatrix} = D_n.V_f \begin{pmatrix} \alpha_t \\ \beta_t \\ \vdots \\ \beta_t \end{pmatrix}.$$

Estimation of the number of loops (4)

Since the operator $D_n.V_f$ has the form that was presented in the previous section, we have the following expressions for $\alpha_{t+1}, \beta_{t+1}$:

$$
\begin{aligned}
\alpha_{t+1} &= \left(1 - \frac{1}{2^{n-1}}\right)\alpha_t + \frac{2^n - 1}{2^{n-1}}\beta_t, \qquad (1) \\
\beta_{t+1} &= -\frac{1}{2^{n-1}}\alpha_t + \left(\frac{1}{2^{n-1}} - 1\right)\beta_t + \frac{2^n - 2}{2^{n-1}}\beta_t = \\
&= -\frac{1}{2^{n-1}}\alpha_t + \left(1 - \frac{1}{2^{n-1}}\right)\beta_t. \qquad (2)
\end{aligned}
$$

Estimation of the number of loops (5)

Since

$$\alpha_t^2 + (2^n - 1)\beta_t^2 = 1 \tag{3}$$

it is convenient to introduce the following substitution:

$$\alpha_t = \sin\phi_t \qquad \beta_t = \frac{1}{\sqrt{2^n - 1}}\cos\phi_t .$$

Estimation of the number of loops (6)

Let us try to describe the influence of the operator $D_n.V_f$ on behaviour of the value of the angle ϕ_t. Let us denote the change of ϕ_t by $2\delta_t$. Then

$$\alpha_{t+1} = \sin(\phi_t + 2\delta_t) \qquad \beta_{t+1} = \frac{1}{\sqrt{2^n - 1}} \cos(\phi_t + 2\delta_t).$$

Estimation of the number of loops (7)

Using the well know identities
$\sin(a+b) = \sin a \cos b + \cos a \sin b$ and
$\cos(a+b) = \cos a \cos b - \sin a \sin b$ we obtain

$$\alpha_{t+1} = \alpha_t \cos 2\delta_t + \sqrt{2^n - 1}\beta_t \sin 2\delta_t, \quad (4)$$

$$\beta_{t+1} = \beta_t \cos 2\delta_t - \frac{1}{\sqrt{2^n - 1}}\alpha_t \sin 2\delta_t. \quad (5)$$

Estimation of the number of loops (8)

By a combination of (4), (5) with (1), (2) we have

$$\alpha_t \cos 2\delta_t + \sqrt{2^n - 1}\beta_t \sin 2\delta_t = \left(1 - \frac{1}{2^{n-1}}\right)\alpha_t + \frac{2^n - 1}{2^{n-1}}\beta_t \tag{6}$$

$$\beta_t \cos 2\delta_t - \frac{1}{\sqrt{2^n - 1}}\alpha_t \sin 2\delta_t = -\frac{1}{2^{n-1}}\alpha_t + \left(1 - \frac{1}{2^{n-1}}\right)\beta_t. \tag{7}$$

If we multiply the equation (6) by α_t and the equation (7) by $(2^n - 1)\beta_t$ and we sum the results we get

$$\cos 2\delta_t = 1 - \frac{1}{2^{n-1}}$$

.

Estimation of the number of loops (9)

Now we utilise (3) and the formula
$\cos 2a = \cos^2 a - \sin^2 a = 1 - 2\sin^2 a$ and we obtain

$$\sin^2 \delta_t = \frac{1 - \cos 2\delta_t}{2} = \frac{1 - \frac{2^{n-1}-1}{2^{n-1}}}{2} = \frac{1}{2^n}.$$

Hence we see that the value of δ_t does not depend on the number of iterations t and it is constant. Since $\lim_{n\to\infty} \frac{1}{2^n} = 0$ and $\lim_{a\to 0} \frac{\sin a}{a} = 1$ we put

$$\delta_t = \delta \approx \frac{1}{\sqrt{2^n}}.$$

Estimation of the number of loops (10)

For t-th iteration, $t \geq 1$ we obtain

$$\alpha_t = \sin(\phi_0 + 2\delta t) \qquad \beta_t = \frac{1}{\sqrt{2^n - 1}} \cos(\phi_0 + 2\delta t).$$

But at the beginning of the loop we have the superposition of the basic states with the coefficient $\alpha_0 = \sin \phi_0$ equal to $\frac{1}{\sqrt{2^n}}$. Therefore $\phi_0 = \delta$ and

$$\alpha_t = \sin[(2t + 1)\delta] \qquad \beta_t = \frac{1}{\sqrt{2^n - 1}} \cos[(2t + 1)\delta].$$

Estimation of the number of loops (11)

We remind, that we are looking for the value $x_0 = x^*$ and we want to obtain a state when $\alpha_t = \sin[(2t+1)\delta]$ is close to 1 and $\beta_t = \frac{1}{\sqrt{2^n - 1}} \cos[(2t+1)\delta]$ is close to 0. This is true when $(2t+1)\delta = \arcsin 1 = \frac{\pi}{2}$ and

$$t = \frac{1}{2}\left(\frac{\pi}{2\delta} - 1\right) \approx \frac{\pi}{4}\sqrt{2^n}.$$

Hence, after $\frac{\pi}{4}\sqrt{2^n}$ repetitions of the loop described in the algorithm, the output of the algorithm is almost sure equal to x^*.

The power of Grover's algorithm

Theorem

Grover's algorithm is searching an unsorted database with $N = 2^n$ entries in $O(N^{1/2})$ time and using $O(\log N)$ storage space.

Grover's algorithm provides quadratic speedup. However, even quadratic speedup is considerable when N is large. According to result of C.H. Bennett et al. (*Strengths and weaknesses of quantum computing*, SIAM Journal on Computing **26** 1510 – 1523) the obtained result is the best possible.

Factorisation problem

Problem

Given an integer N. Find an integer p between 1 and N that divides N.

The algorithm is based on the following facts:

- factorisation of integers can be reduced to the problem of finding the period of a function,
- Fourier transform puts the period of any periodic function into multiples of the reciprocal of the period,
- Quantum Fourier Transform can be used to get efficiently approximations of the period,
- the exact period can be extracted from the available information.

Shor’s algorithm principle (1)

Shor’s algorithm consists of two parts:

- A reduction of the factoring problem to the problem of order-finding, which can be done on a classical computer.
- A quantum algorithm to solve the order-finding problem.

Shor's algorithm principle (2)

In order to illustrate the main ideas of Shor's algorithm, consider the quadratic equation

$$x^2 \equiv 1 \mod N,$$

which always has solutions $x = \pm 1 \mod N$, the so-called trivial solutions.
If N is an odd prime p, then these are the only solutions (since multiplication modulo p has inverses and $x^2 - 1 = (x^2 + 1)(x^2 - 1) = 0 \mod p$ implies $x + 1 \equiv 0 \mod N$, or $x - 1 \equiv 0 \mod p$).
However, if N is composite, then there are also pairs of nontrivial solutions of the form $x \equiv \pm a \mod N$ for some a.

Shor's algorithm principle (3)

If we have a nontrivial solution x of the studied equation we can efficiently find a nontrivial factor of N. We find such an x as follows.
Given N, choose a random $y < N$. If y and N are coprime then let r be the order of $y \mod N$. This is precisely the period of the function $F_N(a) = y^a \mod N$. Thus

$$y^r \equiv 1 \mod N.$$

If r is also even, then setting

$$x = y^{r/2}$$

we have $x^2 \equiv 1 \mod N$, so x is a candidate for our nontrivial solution of the studied equation.

An illustrion of the main idea of Shor's algorithm

a	Period r	$\gcd(15, a^{r/2} - 1)$	$\gcd(15, a^{r/2} + 1)$
1	1		
2	4	3	5
4	2	3	5
7	4	3	5
8	4	3	5
11	2	5	3
13	4	3	5
14	2	1	15

Shor's algorithm principle (4)

This provides the connection between **the periodicity of $F_N(a)$** and the **calculation of a nontrivial factor of N.**

Potential problem

The above process may fail if the **chosen value y has an odd order** r, or if r **is even but $y^{r/2}$ turns out to be a trivial solution the equation.**

However, **it can be proved that such a situation arise only with suitably small probability** if y is chosen at random.

Some useful auxiliary results

Let $n = p_1^{e_1}.p_2^{e_2}\dots p_k^{e_k}$, $k \geq 2$.

Lemma

Let $\phi(p^e) = 2^u v$, where $u \geq 1, 2 \not| v$ and $s \geq 0$ a fixed integer. Then the probability that a randomly (and uniformly) chosen element $a \in Z^*_{p^e}$ has an order of form $2^s t$ with $2 \not| t$ is at most $\frac{1}{2}$.

By a repetitive application of the previous lemma for a decomposition $(a_1, \dots, a_k) \in \mathbb{Z}^*_{p_1^{e_1}} \times \mathbb{Z}^*_{p_2^{e_2}} \times \mathbb{Z}^*_{p_k^{e_k}}$ of a

Lemma

The probability that $r = ord(a)$ is odd for a uniformly chosen $a \in Z^*_n$ is at most $\frac{1}{2^k}$.

Some useful auxiliary results II

Lemma

Let $n = p_1^{e_1} \cdots p_k^{e_k}$ *be a prime decomposition of an odd n and* $k \geq 2$*. If* $r = ord_n(a)$ *is even, then the probability that*

$$a^{\frac{r}{2}} \equiv -1 \pmod{n}$$

is at most $\frac{1}{2^k}$*.*

Lemma

Let $n = p_1^{e_1} \cdots p_k^{e_k}$ *be the prime factorisation of an odd n with* $k \geq 2$*. Then, for a random* $a \in Z_n^*$ *(chosen uniformly), the probability that* $r = ord_n(a)$ *is even and* $a^{\frac{r}{2}} \not\equiv -1 \pmod{n}$ *is at least* $(1 - \frac{1}{2^k})^2 \geq \frac{9}{16}$*.*

A subroutine for Shor's algorithm

Algorithm (Period-finding subroutine)

1. *Start with a pair of input and output qubit registers with* $\log_2 N$ *qubits each, and initialize them to* $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle |0\rangle$.
2. *Construct* $f(x)$ *as a quantum function and apply it to the above state, to obtain* $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |f(x)\rangle |0\rangle$.
3. *Apply the inverse quantum Fourier transform* F *on the input register. The inverse quantum Fourier transform on* N *points is defined by* $F|x\rangle \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{-2\pi ixy/N} |y\rangle$. *It results in the state*

$$\frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^{-2\pi ixy/N} |y\rangle |f(x)\rangle .$$

A subroutine for Shor's algorithm

Algorithm (Period-finding subroutine - cont.)

4. *Perform a measurement. We obtain some outcome y in the input register and $f(x_0)$ in the output register. Since f is periodic, the probability of measuring some pair y and $f(x_0)$ is given by*

$$\left| \frac{1}{N} \sum_{x:f(x)=f(x_0)}^{N-1} e^{-2\pi ixy/N} \right|^2 = \left| \frac{1}{N} \sum_{b} e^{-2\pi i(x_0+rb)y/N} \right|^2 .$$

   Analysis now shows that this probability is higher, the closer yr/N is to an integer.

5. *Turn y/N into an irreducible fraction, and extract the denominator r', which is a candidate for r.*

A subroutine for Shor's algorithm

Algorithm (Period-finding subroutine - cont.)

6. *Check if* $f(x) = f(x + r')$*. If so, we are done.*
7. *Otherwise, obtain more candidates for* r *by using values near* y*, or multiples of* r'*. If any candidate works, we are done.*
8. *Otherwise, go back to step 1 of the subroutine.*

Theorem

Shor's algorithm factors a number N *in* $O((\log N)^3)$ *time and* $O(\log N)$ *space.*

Some useful results II

Lemma

For $n \geq 100$ *the observation of* $(*)$ *will give a* $p \in Z_m$ *such that* $|pr - dm| \leq \frac{r}{2}$ *with a probability of not less than* $\frac{2}{5}$.

Theorem

For $r \geq 3$,

$$\frac{r}{\phi(r)} < e^{\gamma} \log\log r + \frac{2.50637}{\log\log r},$$

where $\gamma = 0,772156649\ldots$ *is the Euler's constant.*

Lemma

For $r \geq 19$, *the probability that, for a uniformly chosen* $d \in \{0, 1, \ldots, r-1\}$ $gcd(d, r) = 1$ *holds, is at least* $\frac{1}{4 \log\log n}$.

Shor's factoring algorithm

Algorithm (Shor's algorithm)

1. *Pick a random number* $a < N$.
2. *Compute* $\gcd(a, N)$. *This may be done using the Euclidean algorithm.*
3. *If* $\gcd(a, N) \neq 1$, *then there is a nontrivial factor of* N, *so we are done.*
4. *Otherwise, use the* period-finding subroutine *to find* r, *the period of the function* $f(x) = a^x \mod N$, *i.e. the smallest integer* r *for which* $f(x + r) = f(x)$.
5. *If* r *is odd, go back to step 1.*
6. *If* $a^{r/2} \equiv -1 (\mod N)$, *go back to step 1.*
7. *The factors of* N *are* $\gcd(a^{r/2} \pm 1, N)$. *We are done.*

Summary

We have already the following facts:

- The probability that, for a randomly (and uniformly) chosen $a \in Z_n$, the order r of a is even and $a^{\frac{r}{2}} \not\equiv -1 (\mod n)$ is at least $\frac{9}{16}$.
- The probability that observing (*) will give a p such that $p - d\frac{m}{r}| < \frac{1}{2}$ is at least $\frac{2}{5}$.
- The probability that $gcd(d, r) = 1$ is at least $\frac{1}{4 \log \log n}$.

By a combination of the previous facts we obtain:

Lemma

The probability that the quantum algorithm finds the order of an element of Z_n is at least $\frac{9}{160} \frac{1}{\log \log n}$.

Maximal Independent Set

Maximal Independent Set Problem: Given a graph $G = (V, E)$, compute a maximal independent set in G.

Theorem (S. Dőrn)

The quantum query complexity of the Maximal Independent Set algorithm is $O(n^{1.5})$ in the adjacency matrix model and $O(\sqrt{nm})$ in the adjacency list model.

Maximum Independent Set

Maximum Independent Set Problem: Given a graph $G = (V, E)$, compute an independent set $V' \subseteq V$ such that $|V'| = \alpha(G)$.

Theorem (S. Dőrn)

The expected quantum time complexity of the Maximum Independent Set algorithm is $O(2^{n/5}) = O(1.1488^n)$.

Graph Colouring

Vertex Colouring Problem: Given a graph $G = (V, E)$, compute a vertex coloring of G with k colours.

Strategy

1. determine a maximal independent set W of the graph G
2. assign all vertices of W with color i (at the beginning $i = 1$).
3. delete all the vertices of W from G and increase i; repeat this procedure as long as there are vertices in G.

Theorem (S. Dőrn)

The quantum time complexity of the vertex-coloring algorithm is $O(kn^{1.5} log^2 n)$ in the adjacency matrix model and $O(kp\sqrt{nm}\, log^2 n)$ in the adjacency list model.

Quantum cryptography principles

Quantum cryptography, or quantum key distribution, uses quantum mechanics to guarantee secure communication. It enables two parties to produce a shared random bit string known only to them, which can be used as a key to encrypt and decrypt messages.

An important and unique property of quantum cryptography is the ability of the two communicating users to detect the presence of any third party trying to gain knowledge of the key.

The security of quantum cryptography relies on the foundations of quantum mechanics, in contrast to traditional public key cryptography which relies on the computational difficulty of certain mathematical functions, and cannot provide any indication of eavesdropping or guarantee of key security.

Recommended reading I

A. Ekert and R. Jozsa
Quantum computation and Shor's factoring algorithm
Review of Modern Physics, Vol 68, No. 3., 733-753

L. Grover
A fast quantum mechanical algorithm for database search
Proceeding STOC '96 Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, 212 – 219.

J. Gruska
Quantum Computing
McGraw-Hill 1999, ISBN: 0-07-709503-0

Mika Hirvenslao
Quantum Computing
Springer Verlag 2001, ISBN: 354-066-783-0.

Recommended reading II

- Mika Hirvensalo
  Quantum computing - Facts and folklore
  Natural Computing **1** (2002) 135–155.
- G. Johnson
  A Shortcut Through Time: The Path to the Quantum Computer
  Alfred A. Knopf, New York 2003, ISBN 037-541-193-3
- A.Y. Kitaev, A.H. Shen, M.N. Vyalyi
  Classical and Quantum Computation
  American Mathematical Society 2002, ISBN 082-183-229-8
- M.A. Nielsen and I.L. Chuang
  Quantum Computation and Quantum Information
  Cambridge University Press 2000, ISBN 052-163-503-9

Thank you very much for your attention

Gabriel Semanišin

gabriel.semanisin@upjs.sk

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach