DCFS 2015

Quantum queries on permutations

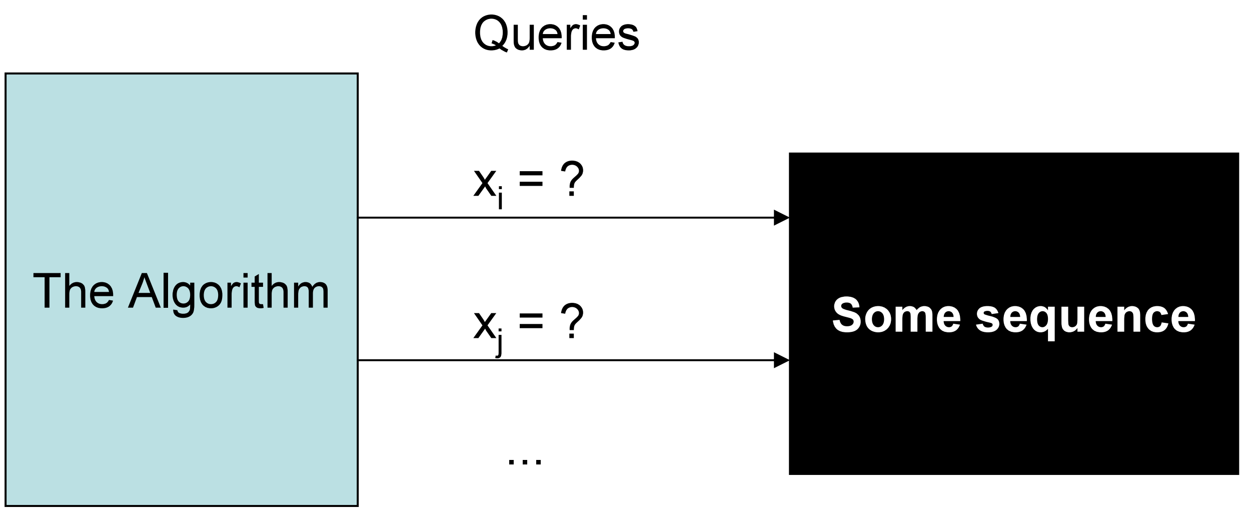
Taisia Mischenko-Slatenkova
Alina Vasilieva
Iļja Kucevalovs*
Rūsiņš Freivalds

Faculty of Computing
Latvijas Universitāte (University of Latvia)

Domain

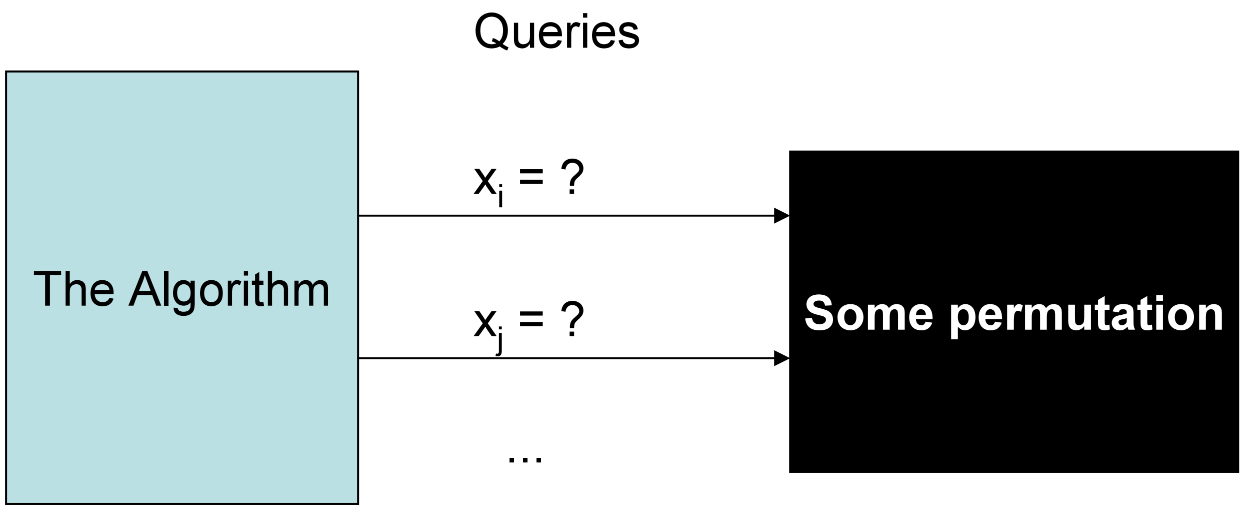
- Quantum vs. deterministic query algorithm complexity
  - The black box contains a permutation

Black box – The explanation



Based on the query results, the algorithm determines a certain Boolean property of the sequence.
The min number of queries needed to determine it is the **algorithm complexity**.

A permutation problem



Previous work

- Rūsiņs Freivalds, Kazuo Iwama. Quantum Queries on Permutations with a Promise. *Lecture Notes in Computer Science*, vol. 5642, p. 208–216, 2009.
  - Algorithms for deciding parity of permutations: Quantum vs. deterministic
  - Attempted to prove: Quantum algorithms need 2x less queries compared to deterministic ones
  - Proved: Quantum algorithms need
    - ***m*** queries for **2*m***-permutations
    - ***m*+1** queries for (**2*m*+1**)-permutations
    - More than ½ compared to deterministic algorithms

This paper

- A permutation problem
- Quantum algorithms need 2x less queries than deterministic

The problem

- Given a 5-permutation, does it belong to the group GR?

GR = {

01234 12340 23401 34012 40123
02413 13024 24135 30241 41302
03142 14203 20314 31420 42031
04321 10432 21043 32104 43210
}

The result

- To solve the problem,
  - no less than 4 queries are needed for a deterministic algorithm
  - only 2 queries are needed for a quantum algorithm

The deterministic case

- GR = {
  01234 12340 23401 34012 40123
  02413 13024 24130 30241 41302
  03142 14203 20314 31420 42031
  04321 10432 21043 32104 43210
  }
- Suppose 3 queries are enough
  - 012.. is received
  - 01234 $\in$ GR
  - 01243 $\notin$ GR
- Hence, **at least 4 queries are needed**

The quantum case: The result

- The algorithm enters 20 states (in the way of quantum parallelism), with equal amplitudes $^{1}/_{\sqrt{20}}$
- In each state, one of the 20 possible query pairs (x_i, x_j) is asked
  - i, $j \in \{0,1,2,3,4\}$ and $i \neq j$
  - Upon receiving the result, the amplitude is multiplied by (-1) or (+1) according to a specifically designed table
- The table is constructed so that:
  - If the permutation $\in$ GR, then all the 20 multipliers are equal
  - If the permutation $\notin$ GR, then half of the multipliers are (-1) and half are (+1)
- Hence, **2 queries are enough**

The construction

- A numbering of pairs (a, b) such that a, b $\in$ {0,1,2,3,4} and $a \neq b$:

  (0,1) (1,2) (2,3) (3,4) (4,0)
  (0,2) (2,4) (4,1) (1,3) (3,0)
  (0,4) (4,3) (3,2) (2,1) (1,0)
  (0,3) (3,1) (1,4) (4,2) (2,0)

- $D_r[(a,b),(u,v)]$ = RowNo$[(a,b)]$ – RowNo$[(u,v)]$ mod 4
- $D_c[(a,b),(u,v)]$ = ColNo$[(a,b)]$ – ColNo$[(u,v)]$ mod 5

The construction explained

- (0,1) (1,2) (2,3) (3,4) (4,0)
  (0,2) (2,4) (4,1) (1,3) (3,0)
  (0,4) (4,3) (3,2) (2,1) (1,0)
  (0,3) (3,1) (1,4) (4,2) (2,0)
- This corresponds to the linear functions
  x x + 1 x + 2 x + 3 x + 4
  2x 2x + 1 2x + 2 2x + 3 2x + 4
  4x 4x + 1 4x + 2 4x + 3 4x + 4
  3x 3x + 1 3x + 2 3x + 3 3x + 4
- Each row = previous row * 2 mod 5
- Each column = previous column + 1 mod 5

The construction explained (2)

- The permutations from GR themselves can be represented as linear functions modulo 5:
- x x + 1 x + 2 x + 3 x + 4
  2x 2x + 1 2x + 2 2x + 3 2x + 4
  3x 3x + 1 3x + 2 3x + 3 3x + 4
  4x 4x + 1 4x + 2 4x + 3 4x + 4
- GR = {
  01234 12340 23401 34012 40123
  02413 13024 24130 30241 41302
  03142 14203 20314 31420 42031
  04321 10432 21043 32104 43210
  }

Multiplier table

- i, j are the zero-based indices of the permutation elements to be queried ($i, j \in \{0,1,2,3,4\}$ and $i \neq j$)
- a_i, a_j are the results of the respective queries

$D_r[(i,j),(a_i,a_j)]$	$D_c[(i,j),(a_i,a_j)]$	$D_r[(j,i),(a_j,a_i)]$	$D_c[(j,i),(a_j,a_i)]$	Multiplier
0	0	0	0	+1
0	1	0	4	+1
...	...			...
3	0	3	0	–1
...	...			...

Example

- The permutation in the black box is 03241
- We query the elements #2 and #4
- The results are 2 and 1
  - (0,1) **(1,2)** (2,3) (3,4) (4,0)
    (0,2) **(2,4)** (4,1) (1,3) (3,0)
    (0,4) (4,3) (3,2) **(2,1)** (1,0)
    (0,3) (3,1) (1,4) **(4,2)** (2,0)
- $D_r[(2,4),(2,1)] = 2 – 1 \bmod 4 = 1$
- $D_c[(2,4),(2,1)] = 3 – 1 \bmod 5 = 2$
- $D_r[(4,2),(1,2)] = 0 – 3 \bmod 4 = 1$
- $D_c[(4,2),(1,2)] = 1 – 3 \bmod 5 = 3$
- Hence we need to search for the table row (1, 2, 1, 3)
  - In our table, the multiplier in this row is +1

Lemma 1

- If the permutation in the black box is from the group GR then all 20 multipliers are equal
- Proof:
  - If the permutation corresponds to the function $ax + b$ where $a = 1$ or $a = 2$, then the multiplier equals (+1)
  - If the permutation corresponds to the function $ax + b$ where $a = 3$ or $a = 4$, then the multiplier equals (−1)

Lemma 2

- If the permutation in the black box is one of the following:
  01243, 01342, 01423, 01324, 01432
  then exactly 10 multipliers equal (− 1) and exactly 10 multipliers equal (+1)
- Proof:
  - By explicit counting

Lemma 3

- If the permutation in the black box $f(x)$ can be obtained from a permutation $g(x)$ from the set

  { 01243, 01342, 01423, 01324, 01432 }

  as

  $$f(x) \equiv ag(x) + b \pmod 5$$

  then
  - exactly 10 multipliers equal (−1) and
  - exactly 10 multipliers equal (+1)
- Proof:
  - The definition of the values of multipliers depend only on the distances D_r but not on the distances D_c
  - Application of a linear function $at + b$ does not change the distance D_r

Lemma 4

- If the permutation in the black box is not from the group GR then
  - exactly 10 multipliers equal (−1) and
  - exactly 10 multipliers equal (+1)
- Proof:
  - The group G_5 of all 5-permutations consists of 120 elements
  - GR is a subgroup of G_5 consisting of 20 elements
  - Lagrange's theorem on finite groups: G_5 is subdivided into 6 cosets of equal size, one of the cosets being GR
  - The other 5 cosets GC_1, GC_2, GC_3, GC_4, GC_5 can be described as the set of all permutations f(x) such that
    - $f(x) \equiv ag(x) + b (\text{mod } 5)$ and
    - $g(x) \in GC_i$
  - From Lemma 3: exactly 10 multipliers equal (−1) and exactly 10 multipliers equal (+1)

Main theorem

- The algorithm enters 20 states (in the way of quantum parallelism), with equal amplitudes $1/\sqrt{20}$
- In each state, one of the 20 possible query pairs (x_i, x_j) is asked
  - $i, j \in \{0,1,2,3,4\}$ and $i \neq j$
  - Upon receiving the result, the amplitude is multiplied by (-1) or (+1) according to a specifically designed table
- The table is constructed so that:
  - If the permutation $\in$ GR, then all the 20 multipliers are equal
  - If the permutation $\notin$ GR, then half of the multipliers are (-1) and half are (+1)
- Hence, **there is an exact quantum query algorithm deciding the membership in the group GR with two queries**

Conclusion and future work

- There is a permutation problem, for which quantum algorithms need 2x less queries than deterministic ones
- In future, we hope to show a similar separation:
  - For a parity problem for permutations
  - For *n*-permutations, where *n* can be arbitrarily large

Thank you for your attention

Appendix: Application of *at* + *b* does not change the distance D_r

- The permutation in the black box is 03241
- We query the elements #2 and #4
- The results are 2 and 1
  - (0,1) **(1,2)** (2,3) (3,4) (4,0)
    (0,2) **(2,4)** (4,1) (1,3) (3,0)
    (0,4) (4,3) (3,2) **(2,1)** (1,0)
    (0,3) (3,1) (1,4) **(4,2)** (2,0)
- $D_r[(2,4),(2,1)] = 2 – 1 \bmod 4 = 1$
- $D_c[(2,4),(2,1)] = 3 – 1 \bmod 5 = 2$
- $D_r[(4,2),(1,2)] = 0 – 3 \bmod 4 = 1$
- $D_c[(4,2),(1,2)] = 1 – 3 \bmod 5 = 3$

Appendix: Application of *at* + *b* does not change the distance D_r (2)

- The permutation in the black box is 14302
- We query the elements #2 and #4
- The results are 3 and 2
  - (0,1) (1,2) **(2,3)** (3,4) (4,0)
    (0,2) **(2,4)** (4,1) (1,3) (3,0)
    (0,4) (4,3) **(3,2)** (2,1) (1,0)
    (0,3) (3,1) (1,4) **(4,2)** (2,0)
- D_r[(2,4),(3,2)] = 2 – 1 mod 4 = 1
- D_c[(2,4),(3,2)] = 2 – 1 mod 5 = 1
- D_r[(4,2),(2,3)] = 0 – 3 mod 4 = 1
- D_c[(4,2),(2,3)] = 2 – 3 mod 5 = 4

Appendix: Application of *at* + *b* does not change the distance D_r (3)

- The permutation in the black box is 20413
- We query the elements #2 and #4
- The results are 4 and 3
  - (0,1) (1,2) (2,3) **(3,4)** (4,0)
    (0,2) **(2,4)** (4,1) (1,3) (3,0)
    (0,4) **(4,3)** (3,2) (2,1) (1,0)
    (0,3) (3,1) (1,4) **(4,2)** (2,0)
- $D_r[(2,4),(4,3)] = 2 – 1 \bmod 4 = 1$
- $D_c[(2,4),(4,3)] = 1 – 1 \bmod 5 = 0$
- $D_r[(4,2),(3,4)] = 0 – 3 \bmod 4 = 1$
- $D_c[(4,2),(3,4)] = 1 – 1 \bmod 5 = 0$