Lower Bounds for Query Complexity of Some Graph and Matrix Problems

Lelde Lāce and Rūsiņš Freivalds

Institute of Mathematics and Computer Science University of Latvia
Raina bulvāris 29, Riga, LV-1459, Latvia
E-mail addresses: (Lelde.Lace, Rusins.Freivalds)@mii.lu.lv

Abstract The paper [4] by H.Buhrman and R. de Wolf contains an impressive survey of solved and open problems in quantum query complexity, including many graph problems. We use recent results by A.Ambainis [1] to prove higher lower bounds for some of these problems. Some of our new lower bounds do not close the gap between the best upper and lower bounds. We prove in these cases that it is impossible to provide a better application of Ambainis’ technique for these problems.

Keywords. Quantum algorithm, lower bound, graph theory

1. Introduction

Recently it has become clear that a quantum computer could, in principle, solve certain problems faster than a conventional computer. A quantum computer is a device, which takes full advantage of quantum mechanical superposition and interference. Building an actual quantum computer is probably far off in the future.

Boolean decision trees model is the simplest model to compute Boolean functions. In this model the primitive operation made by an algorithm is evaluating an input Boolean variable. The cost of a (deterministic) algorithm is the number of variables it evaluates on a worst-case input.

The black-box model of computation arises when one is given a black-box containing an N-tuple of Boolean variables X=(x₁,x₂,...,x₅). The box is equipped to output xᵢ on input i. We wish to determine some property of X, accessing the xᵢ only through the black-box. Such a black-box access is called a query. A property of X is any Boolean function that depends on X, i.e. a property is function f :{0,1}ᴺ→{0,1}. We want to compute such properties using as few queries as possible.

Consider, for example, the case where the goal is to determine whether or not X contains at least one 1, so we want to compute the property OR(X) = x₁ ∨ x₂ ∨ ... ∨ xᴺ. It is well known that the number of queries required to compute OR by any classical (deterministic or probabilistic) algorithm is O(N).Grover [6] discovered a

---

1 Research supported by Grant No.01.0354 from the Latvian Council of Science and by the European Commission, Contract IST-1999-11234 (QAIP)
remarkable quantum algorithm that, making queries in superposition, can be used to compute OR with small error probability using only $O(\sqrt{N})$ queries.

On the other hand, quantum algorithms are in a sense more restricted. For instance, only unitary transformations are allowed for state transitions. Hence rather often a problem arises whether or not the needed quantum automaton exists. In such a situation lower bounds of complexity are considered. It is proved in [3] that Grover database search algorithm is the best possible. It is proved in [3] that no quantum query algorithm exists for PARITY with $\Omega(N)$ queries, etc.

We use a result by A.Ambainis [1] to prove lower complexity bounds for quantum query algorithms. Currently, this is the most powerful method to prove lower bounds of complexity for quantum query algorithms. In some cases there still remains a gap between the upper and the lower bounds of the complexity. In these cases we prove additionally that Ambainis’ method cannot provide a better lower bound for this problem.

2. Query model

In the query model, the input $x_1,\ldots,x_N$ is contained in a black box and can be accessed by queries to the black box. In each query, we give $i$ to the black box and the black box outputs $x_i$. The goal is to solve the problem with the minimum number of queries. The classical version of this model is known as decision trees [4].

There are two ways how to define the query box in the quantum model. The first is the extension of the classical query (Figure 1).

<table>
<thead>
<tr>
<th>$i$</th>
<th>$a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
</tr>
</tbody>
</table>

Figure 1. Quantum black box.

It has two inputs $i$, consisting of $\left\lceil \log N \right\rceil$ bits and $b$ consisting of 1 bit. If the input to the query box is a basic state $|i\rangle|b\rangle$, the output is $|i\rangle|b\oplus x_i\rangle$. If the input is a superposition $\sum a_{i,b}|i\rangle|b\rangle$, the output is $\sum a_{i,b}|i\rangle|b\oplus x_i\rangle$. Notice that this definition applies both to case when $x_i$ are binary and to the case when they are k-valued. In the k-valued case, we just make $b$ to consist of $\left\lceil \log_2 k \right\rceil$ bits and take $b\oplus x_i$ to be bitwise XOR of $b$ and $x_i$.

In the second form of quantum query (which only applies to problems with $\{0,1\}$-valued $x_i$), the black box has just one input $i$. If the input is a state $\sum a_i|i\rangle$, the output is $\sum (-1)^{a_i}a_i|i\rangle$. While this form is less intuitive, it is very convenient for the use in quantum algorithms, including Grover’s search algorithm [6]. A query of second type can be simulated by a query of first type [6].
A quantum query algorithm with $T$ queries is just a sequence of unitary transformations

$$U_0 \rightarrow O \rightarrow U_1 \rightarrow O \rightarrow \ldots \rightarrow U_{T-1} \rightarrow O \rightarrow U_T$$

on some finite-dimensional space $C^k$. $U_0$, $U_1$, ..., $U_T$ can be any unitary transformations that do not depend on the bits $x_1$, ..., $x_N$ inside the black box. $O$ are query transformations that consist of applying the query box to the first $\log N+1$ bits of the state. That is, we represent basic states of $C^k$ as $|i,b,z\rangle$. Then, $O$ maps $|i,b,z\rangle$ to $|i,b\otimes x_i,z\rangle$. We use $O_x$ to denote the query transformation corresponding to an input $x=(x_1,\ldots,x_N)$.

The computation starts with state $|0\rangle$. Then, we apply $U_0$, $O_x$, ..., $O_x$, $U_T$ and measure the final state. The result of the computation is the rightmost bit of the state obtained by the measurement (or several bits if we are considering a problem where the answer has more than 2 values).

The quantum algorithm computes a function $f(x_1,\ldots,x_N)$ if, for every $x=(x_1,\ldots,x_N)$ for which $f$ is defined, the probability that the rightmost bit of $U_T O_x U_{T-1} \ldots O_x U_0 |0\rangle$ equals $f(x_1,\ldots,x_N)$ is at least $1-\varepsilon < \frac{1}{2}$.

The query complexity of $f$ is the smallest number of queries used by a quantum algorithm that computes $f$. We denote it $Q(f)$.

Our proofs use the following results by A. Ambainis.

**Theorem A1** [1] Let $f(x_1, x_2, x_3, \ldots, x_n)$ be a function of $n$ $\{0,1\}$-valued variables and $A \subset \{0,1\}^n$, $B \subset \{0,1\}^n$ be such that $f(A)=1$, $f(B)=0$ and

- for every $x=(x_1, x_2, x_3, \ldots, x_n)\in A$, there are at least $m$ values $i \in \{1,\ldots,n\}$ such that $(x_1, \ldots, x_{i-1}, 1-x_i, x_{i+1}, \ldots, x_n) \in B$,
- for every $x=(x_1, x_2, x_3, \ldots, x_n)\in B$, there are at least $m'$ values $i \in \{1,\ldots,n\}$ such that $(x_1, \ldots, x_{i-1}, 1-x_i, x_{i+1}, \ldots, x_n) \in A$.

Then $Q(f) = \Omega(\sqrt{mm'})$.

**Theorem A2** [1] Let $f(x_1, x_2, x_3, \ldots, x_n)$ be a function of $n$ $\{0,1\}$-valued variables and $A$, $B$ be two sets of inputs such that $f(x) \neq f(y)$ if $x \in A$ and $y \in B$. Let $R \subset A \otimes B$ be such that

- for every $x \in A$ there exist at least $m$ different $y \in B$ such that $(x,y) \in R$,
- for every $y \in B$ there exist at least $m'$ different $x \in A$ such that $(x,y) \in R$,
- for every $x \in A$ and $i \in \{1,\ldots,n\}$ there are at most $l$ different $y \in B$ such that $(x,y) \in R$ and $x_i \neq y_i$,
- for every $y \in B$ and $i \in \{1,\ldots,n\}$ there are at most $l'$ different $x \in A$ such that $(x,y) \in R$ and $x_i \neq y_i$.

Then $Q(f) = \left\lceil \frac{mm'}{\max(l,l')} \right\rceil$.

**Definition** For any Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ and any $x=(x_1, x_2)$, $ND(f,x)$ is the number of queries needed by nondeterministic algorithms on the values $x=(x_1, x_2)$. 


**Definition** For any Boolean function \( f : \{0,1\}^N \rightarrow \{0,1\} \)
ND_0(f)= \( \max_{f(i)=0} ND(f,x) \) and \( ND_1(f)= \max_{f(i)=1} ND(f,x) \).

**Theorem A3** [2] Whatever the sets A and B, Theorem 1 cannot prove a better lower bound for the query complexity \( Q(f) \) than \( \sqrt{\text{ND}_0(f) \cdot \text{ND}_1(f)} \).

We consider the following problems in our paper.

3. Problems

**Problem 1 Partition into cliques**
Instance: Graph \( G=(V,E) \), with \(|V|=qk\) for fixed integer \( q>1 \) and some integer \( k \).
Question: Can the vertices of \( G \) be partitioned into \( k \) disjoints sets \( V_1,V_2,..V_k \) such that, for \( 1\leq i \leq k \), the subgraph induced by \( V_i \) is a complete graph and \(|V_i|=q\)?

**Problem 2 Partition into triangles**
Instance: Graph \( G=(V,E) \), with \(|V|=3k\) for some integer \( k \).
Question: Can the vertices of \( G \) be partitioned into \( k \) disjoint sets \( V_1,V_2,..V_k \), each containing exactly 3 vertices, such that for each \( V_i=\{u_i,v_i,w_i\}, 1\leq i \leq k \), all three of the edges \( \{u_i,v_i\}, \{u_i,w_i\}, \{v_i,w_i\} \) belong to \( E \)?

**Problem 3 Matching**
Instance: Graph \( G=(V,E) \), |\( V | =n \).
Question: Can the vertices of \( G \) be partitioned into \( n/2 \) disjoints pairs \( P_1,P_2,..P_{n/2} \) such that for each \( P_i=\{u_i,v_i\}, 1\leq i \leq n/2 \), edge \( \{u_i,v_i\} \) belong to \( E \)?

**Problem 4 Parity**
Instance: Matrix \( M \in \mathbb{R}^{2n \times 2n} \), \( M_{ij} \in \{0,1\} \).
Question: \( \sum_{i=1,2n} \text{PARITY}(M_i) = n? \)

**Problem 5 Hamiltonian circuit**
Instance: Graph \( G=(V,E) \).
Question: Does \( G \) contain Hamiltonian circuit?

**Problem 6 Vertex cover**
Instance: Graph \( G=(V,E) \), positive integer \( k \leq |V| \).
Question: Is there a vertex cover of size \( k \) or less for \( G \), i.e. a subset \( V' \subseteq V \) with \(|V'| \leq k \) such that for each edge \( \{u,v\} \in E \) at least one of vertices \( u \) and \( v \) belongs to \( V' \)?

**Problem 7 Dominating set**
Instance: Graph \( G=(V,E) \), positive integer \( k \leq |V| \).
Question: Is there a dominating set of size \( k \) or less for \( G \), i.e. a subset \( V' \subseteq V \) with \(|V'| \leq k \) such that for all \( u \in V-V' \) there is a \( v \in V' \) for which \( \{u,v\} \in E \)?
Problem 8 Chromatic number
Instance: Graph $G=(V,E)$, positive integer $k \leq |V|$
Question: Does there exist a function $f: V \rightarrow \{1,2,\ldots,k\}$ such that $f(u) \neq f(v)$ whenever $\{u,v\} \in E$?

Problem 9 Monochromatic triangle
Instance: Graph $G=(V,E)$.
Question: Is there a partition of $E$ into two disjoint sets $E_1, E_2$ such that neither $G_1=(V,E_1)$ nor $G_2=(V,E_2)$ contains a triangle?

4. Main results

4.1. Partition into cliques

Lemma L1: If there are $k+1$ mutually not connected vertices in the graph $G=(V,E)$, $|V|=kq$, then Partition into cliques problem is not solvable.

Proof: If there is a solution for Partition into cliques, we get $k$ disjoint sets. Since there is $k+1$ mutually not connected vertices, there is at least one subset containing two mutually not connected vertices and Partition into cliques problem is not solvable.

Lemma L2: If a graph $G=(V,E)$, $|V|=kq$, satisfies the following requirements: there are $k/2$ mutually not connected (red) vertices, there are $k$ green vertices not connected with red ones, green vertices are grouped in pairs and each pair is connected by edge, subgraph induced by all the rest vertices (black) is a complete graph and all black vertices are connected to all red and green vertices, then Partition into cliques problem is solvable.

Proof: Vertices are grouped in subsets in accordance with the following: each red vertex is put in a separate subset ($k/2$ subsets), each pair of green vertices is put in a separate subset ($k/2$ subsets), black vertices are added as follows: $q-1$ to red and $q-2$ to green vertices. Such a distribution satisfies Partition into cliques problem.

Lemma L3: If graph $G=(V,E)$, $|V|=kq$, satisfies the following requirements: there are $k/2+2$ mutually not connected (red) vertices, there are $k-2$ green vertices not connected with red ones, green vertices are grouped in pairs and each pair is connected by edge, subgraph induced by all the rest vertices (black) is a complete graph and all black vertices are connected to all red and green vertices, then Partition into cliques problem is not solvable.

Proof: If we take red vertices and one from each pair of green vertices then we get $k/2+2+(k-2)\geq k+1$ vertices. These vertices are not mutually connected. The Partition into cliques problem is not solvable because the set of selected vertices satisfies the requirements of Lemma L1.
Theorem T1 Partition into cliques requires $\Omega(n^{1.5})$ quantum queries.

Proof: We construct the sets $A$ and $B$ for the usage of Theorem A1.

The set $A$ consists of all graphs $G$ satisfying the requirements of Lemma L2. The set $B$ consists of all graphs $G$ satisfying the requirements of Lemma L3.

From each graph $G \in A$, we can obtain $G' \in B$ by disconnecting any one of the edges, which connect the green vertices. Hence $m=k/2=O(k)$. From each graph $G \in B$, we can obtain $G' \in A$ by connecting any two red vertices. Hence $m'=(k/2+2)(k/2+1)/2=O(k^2)$.

Since $q$ is fixed, it follows that $k=O(n)$. By Theorem A1, the quantum query complexity is $\Omega(\sqrt{n \cdot n^2})=\Omega(n^{1.5})$.

The same idea proves the following two theorems.

Theorem T2 Matching requires $\Omega(n^{1.5})$ quantum queries.

Theorem T3 Partition into triangles requires $\Omega(n^{1.5})$ quantum queries.

Theorem T4 The lower bound for Partition into cliques cannot be improved by Ambainis’ method.

Proof: We use Theorem A3. Let the Boolean function $f$ describe Partition into cliques.

$ND_1(f) = O(n)$, because it suffices to ask the edges for all the guessed subsets of vertices; all the subsets are of constant size.

$ND_0(f) = O(n^2)$, because it suffices to exhibit a subset of $k+1$ vertices connected by no edges. Since $k=O(n)$, $(k-1)k/2 = O(n^2)$.

Hence $\sqrt{ND_1(f) \cdot ND_0(f)} = O(n^{1.5})$.

4.2. Parity problem

Theorem T5 Parity problem requires $\Omega(n^2)$ quantum queries.

Proof: We construct the sets $A$ and $B$ for the usage of Theorem A1.

The set $A$ consists of all matrices $M$ with $n$ rows containing $n$ symbols “1” per row plus $n$ rows containing $n+1$ symbols “1” per row. The set $B$ consists of all matrices $M'$ with $n-1$ rows containing $n$ symbols “1” per row plus $n+1$ rows containing $n+1$ symbols “1” per row.

Every matrix $M \in A$ can be transformed into a matrix $M' \in B$ by taking an arbitrary row with $n$ symbols “1” and transforming an arbitrary “0” into “1”. Hence $m = n^2$. Every matrix $M \in B$ can be transformed into a matrix $M \in A$ by taking an arbitrary row with $n+1$ symbols “1” and transforming an arbitrary symbol “1” into “0”. Hence $m' = n^2$.

By Theorem A1, the quantum query complexity is $\Omega(\sqrt{n^2 \cdot n^2}) = \Omega(n^2)$. This is the maximum possible lower bound.
4.3. Hamiltonian circuit problem

**Lemma L4** If a graph \( G(V,E) \), \( |V|=5n \), satisfies the following requirements:
- there are \( n \) mutually not connected (red) vertices,
- there are \( 2n \) green vertices not connected with red ones, green vertices are grouped in pairs and each pair is connected by edge,
- all rest \( 2n \) vertices (black) are connected to all red and green vertices,
then Hamiltonian circuit problem is solvable.

**Proof:** We make sequence as follows: one red vertex, one black, pair of green, one black, one red, etc. This sequence satisfies Hamiltonian circuit problem. \( \square \)

**Lemma L5** If a graph \( G(V,E) \), \( |V|=5n \), satisfies the following requirements:
- there are \( n+2 \) mutually not connected (red) vertices,
- there are \( 2n-2 \) green vertices not connected with red ones, green vertices are grouped in pairs and each pair is connected by edge,
- all rest \( 2n \) vertices (black) are connected to all red and green vertices,
then Hamiltonian circuit problem is not solvable.

**Proof:** The red vertices and the pairs or green vertices are mutually not connected. The only way to get from one red vertex to another (or from one green pair to another) is through some black vertex. There are \( 2n \) black vertices in the graph, but \( n+2 \) red vertices and \( n-1 \) green pair makes altogether \( 2n+1 \). So at least one of the black vertices will be used twice, which is not allowed in Hamiltonian circuit. \( \square \)

**Theorem T6** Hamiltonian circuit problem requires \( \Omega(n^{1.5}) \) quantum queries.

**Proof:** We construct the sets \( A \) and \( B \) for the usage of Theorem A1.

The set \( A \) consists of all graphs \( G \) satisfying the requirements of Lemma L4. The set \( B \) consist of all graphs \( G' \) satisfying the requirements of Lemma L5.

From each graph \( G \in A \), we can obtain \( G' \in B \) by disconnecting any one of the edges, which connect the green vertices. Hence \( m=n=O(|V|) \). From each graph \( G' \in B \), we can obtain \( G \in A \) by connecting any two red vertices. Hence \( m'=(n+2)(n+1)/2=O(|V|^2) \).

By Theorem A1, the quantum query complexity is \( \Omega(\sqrt{n} \cdot n^2) = \Omega(n^{1.5}) \). \( \square \)

**Theorem T7** The lower bound for Hamiltonian circuit cannot be improved by Ambainis’ method.

**Proof:** We use Theorem A3. Let the Boolean function \( f \) describe Hamiltonian circuit.

\( ND_1(f) = O(n) \), because it suffices to guess the sequence of vertices and ask the edge for every pair of subsequent vertices.

\( ND_0(f) = O(n^2) \), because it suffices to check all edges.

Hence \( \sqrt{ND_1(f) \cdot ND_0(f)} = O(n^{1.5}) \). \( \square \)
4.4. Vertex cover problem

**Lemma L6** If a graph \( G=(V,E) \), \(|V|=n\), \( K=n/4 \), satisfies the following requirements:
- there are \( n/2 \) black vertices, not mutually connected
- there are \( n/2 \) red vertices, pairwisely connected,
then **Vertex cover** problem is solvable.

**Proof:** We take in subset one vertex of each red pair and give \( n/4 \) vertices, which satisfies **Vertex cover** problem.

**Lemma L7** If a graph \( G=(V,E) \), \(|V|=n\), \( K=n/4 \), satisfies the following requirements:
- there are \( n/2-2 \) black vertices, not mutually connected
- there are \( n/2+2 \) red vertices, pairwisely connected,
then **Vertex cover** problem is not solvable.

**Proof:** We take in subset one vertex of each red pair and give \( n/4+1 \) vertices, which are more than \( K \) and doesn’t satisfy **Vertex cover** problem.

**Theorem T8** **Vertex cover** problem requires \( \Omega(n^{1.5}) \) quantum queries.

**Proof:** We construct the sets \( A \) and \( B \) for the usage of Theorem A1.

The set \( A \) consists of all graphs \( G \) satisfying the requirements of Lemma L6. The set \( B \) consist of all graphs \( G' \) satisfying the requirements of Lemma L7.

From each graph \( G \in A \), we can obtain \( G' \in B \) by disconnecting any one of the edges, which connect the red vertices. Hence \( m=n/2=O(n) \). From each graph \( G' \in B \), we can obtain \( G \in A \) by connecting any two black vertices. Hence \( m'=(n/2-2)/(n/2-3)/2=O(n) \).

By Theorem A1, the quantum query complexity is \( \Omega \sqrt{n \cdot n^2} = \Omega(n^{1.5}) \).

The same idea proves the following theorem.

**Theorem T9** **Dominating set** problem requires \( \Omega(n^{1.5}) \) quantum queries.

4.5. Chromatic number problem

**Lemma L8** If a graph \( G1 \) contains one circuit and a graph \( G2 \) contains two circuits, then graph \( G1 \) and graph \( G2 \) separation problem requires \( \Omega(n^{1.5}) \) quantum queries.

**Proof:** We construct the sets \( A \) and \( B \) for the usage of Theorem A2.

The set \( A \) consists of all graphs \( G=(V,E), |V|=n \), containing Hamiltonian circuit. The set \( B \) consists of all graphs \( G' \). Each graph \( G' \) is constructed to merge two graphs \( G''=(V'',E''), |V''|=O(n) \), containing Hamiltonian circuit.

![Figure 2. Disconnection of Graph G](image)
From each graph $G \in A$, we can obtain $G' \in B$ as follows: disconnecting any one of the edges (n) (Figure 2. top) and one of edges in figure 2. bottom (n/3). Then we make two circuits. From each graph $G' \in B$, we can obtain $G \in A$ as follows: disconnecting one edge in each circuit (n$^2$) and building up one circuit. Hence $m=n \cdot n/3 = \Omega(n^2)$ and $m'= O(n^0)$.

Now we’ll find max($l \cdot l'$). For any edge, which we disconnect in first step $l= n/3$, because we can do that in n/3 combinations with other edges, and $l'=\text{const}$, because we can connect this edge in fixed count combinations.

For any edge, which we connect in first step $l= \text{const}$, because we can do that in fixed count combinations, and $l'=O(n)$, because we can disconnect this edge in combinations with all edges in another circuit. Thus $\text{max}(l \cdot l')= \Omega(n)$.

By Theorem A2, the quantum query complexity is $\Omega(n^{1.5})$.

Lemma L9 If a graph $G=(V,E)$, $|V|=2n$, contains only Hamiltonian circuit then Chromatic number problem for two colors is solvable.

Proof: We color one vertex in first color, next vertex in Hamiltonian circuit in second color, next in first color, etc. This coloring solve Chromatic number problem for two colors.

Lemma L10 If a graph $G=(V,E)$, $|V|=2n+1$, contains only Hamiltonian circuit then Chromatic number problem for two colors is not solvable.

Proof: Regardless of how we color vertices, at least two vertices in one color will always be connected.

Theorem T10 Chromatic number problem requires $\Omega(n^{1.5})$ quantum queries.

Proof: We construct the sets $A$ and $B$ for the usage of Theorem A1.

The set $A$ consists of all graphs $G=(V,E)$ satisfying the requirements of Lemma L9. The set $B$ consists of all graphs $G'$. Each graph $G'$ is constructed to merge two graphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$, which both satisfy the requirements of Lemma L10 and $|V_1|+|V_2|=|V|$ and $O(|V_1|)=O(|V|)$ and $O(|V_2|)=O(|V|)$.

By Lemma L8, the quantum query complexity is $\Omega(n^{1.5})$.

4.6. Monochromatic triangle problem

We construct graphs $G=(V,E)$, $|V|=3n$ as follows. All vertices in graph are partitioned into three. Each three is connected by another three in one of the following ways (Figure 3.). Each three in this graph does not contain triangle. We put in one subset all vertices from one three.

Figure 3. Connections of threes
Lemma L11 If Monochromatic triangle problem is not solvable in this distribution into threes, then problem is not solvable in another distribution into threes.

Proof: We cannot mix vertices 1,2,3 with 4,5,6. If we do it three will contain triangle and Monochromatic triangle problem will not be solvable. If we mix vertices 1,2,3 with 1a, 2a, 3a then Monochromatic triangle problem solution will not change.

![Graph G and matrix M](image)

We make matrix $M$ in accordance with graph $G$, which is defined before as follows. Matrix consists of $n$ columns and $n$ rows. We put in $m_{ij}$ 1 if three $v_i$ is connected with three $v_j$ in first way (Figure 3.) and 0 if connected in second way (Figure 4.).

![Graph G and graph $G_M$](image)

We make graph $G_M$, which corresponds to matrix $M$. Each vertex of graph $G_M$ is related to the three of graph $G$ (Figure 5.).

Lemma L12 If in graph $G_M$ Chromatic number problem ($K=2$) is solvable, then Monochromatic triangle problem is solvable in corresponding graph $G$ also.

Proof: Each vertex of graph $G_M$ is related to the three of graph $G$. Since Chromatic number problem is solvable, we can color $G_M$ vertices in two colors so that one-color vertices are not connected. We put in one subset those threes in graph $G$, which correspond to the vertex in graph $G_M$ colored in one color. Each three in this subset is connected with another in second way (Figure 3.) and subset does not contain triangle.

Lemma L13 If in graph $G_M$ Chromatic number problem ($K=2$) is not solvable, then Monochromatic triangle problem is not solvable in corresponding graph $G$ also.
**Proof:** Let us assume, that in graph $G$ **Monochromatic triangle** problem is solvable if threes are combined differently. By Lemma L11 the problem is solvable also in this distribution of threes, which corresponds to graph $G_M$. But then we can solve **Chromatic number** problem in graph $G_M$. This is in conflict with given. □

**Theorem T11** **Monochromatic triangle** problem requires $\Omega(n^{1.5})$ quantum queries.

**Proof:** By Lemma L12 and Lemma L13, complexity of **Monochromatic triangle** problem is equivalent with complexity of **Chromatic number** problem. So quantum query complexity is $\Omega(n^{1.5})$. □

**References**


