Multiparty Quantum Communication Complexity of Triangle Finding[*]

François Le Gall[1] and Shogo Nakajima[2]

1 Department of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University, Kyoto, Japan
legall@i.kyoto-u.ac.jp

2 Department of Computer Science, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan
nakajimashogo@is.s.u-tokyo.ac.jp

Abstract

Triangle finding (deciding if a graph contains a triangle or not) is a central problem in quantum query complexity. The quantum communication complexity of this problem, where the edges of the graph are distributed among the players, was considered recently by Ivanyos et al. in the two-party setting. In this paper we consider its k-party quantum communication complexity with $k \geq 3$. Our main result is a $\tilde{O}(m^{7/12})$-qubit protocol, for any constant number of players k, deciding with high probability if a graph with m edges contains a triangle or not. Our approach makes connections between the multiparty quantum communication complexity of triangle finding and the quantum query complexity of graph collision, a well-studied problem in quantum query complexity.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Quantum communication complexity, triangle finding, graph collision

Digital Object Identifier 10.4230/LIPIcs.TQC.2017.6

1 Introduction

1.1 Triangle finding

A triangle in an undirected graph $G = (V, E)$ is a set of three vertices v_1, v_2, and v_3 such that $\{v_1, v_2\}$, $\{v_1, v_3\}$, and $\{v_2, v_3\}$ are edges. The problem of deciding whether a given graph contains a triangle or not is called triangle finding, and has been the subject of thorough investigations in the past years in both the classical and quantum settings.

In the classical setting, several new applications of this problem have been discovered recently. In particular, Vassilevska Williams and Williams [20] showed in 2010 a surprising reduction from Boolean matrix multiplication to triangle finding. Several works followed (e.g., [17, 21]), which have now placed triangle finding as a central problem in the recent theory of fine-grained complexity.

In the quantum setting, triangle finding has played a prominent role in the development of quantum query algorithms. For query algorithms solving graph-theoretic problems like triangle finding, information about the set of edges E can be obtained only by queries

[*] This work is supported by the Grant-in-Aid for Young Scientists (A) No. 16H05853, the Grant-in-Aids for Scientific Research (A) No. 15H01677 and 16H01705, and the Grant-in-Aid for Scientific Research on Innovative Areas No. 24106009 of the Japan Society for the Promotion of Science and the Ministry of Education, Culture, Sports, Science and Technology in Japan.

12th Conference on the Theory of Quantum Computation, Communication, and Cryptography (TQC 2017).
Editor: Mark M. Wilde; Article No. 6; pp. 6:1–6:11
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to an oracle representing the adjacency matrix of the input graph: given two vertices u and v of G, the oracle outputs one if $\{u, v\} \in E$ and zero if $\{u, v\} \notin E$ (in the quantum setting the queries can naturally be done in superposition). The trivial upper bound on the quantum query complexity of triangle finding is $O(n^{3/2})$, where n denotes the number of vertices of the graph, by Grover search. A series of works spreading over more than a decade [4, 7, 10, 12, 14, 16] successively improved this bound to $O(n^{5/4})$ by using more advanced techniques like quantum walks, learning graphs, variable costs quantum search and quantum nested walks. On the other hand, the best known lower bound on the quantum query complexity of triangle finding is the trivial $\Omega(n)$. Understanding whether the $O(n^{5/4})$ upper bound is tight or not is now the main open problem concerning the quantum query complexity of triangle finding in dense graphs. Several quantum query algorithms for triangle finding over sparse graphs have been constructed as well [6, 7, 8, 13].

1.2 Communication complexity of triangle finding

In this paper we consider triangle finding not in the quantum query complexity model, but in the quantum communication complexity model. As usual when considering graph-theoretic problems in the communication complexity setting, we assume that the edges of the graphs are distributed among the players (in this paper we consider the most general case where the subsets of edges owned by the players can overlap). In the two-party case, for instance, the first player Alice receives a set of edges $E_A \subseteq E$ and the second player Bob receives a set of edges $E_B \subseteq E$ such that $E_A \cup E_B = E$ (the intersection of these two sets is not necessarily empty). The players must decide if the whole graph contains a triangle or not. We will use $\mathsf{TF}^k_{n,m}$ to denote this distributed version of triangle finding, where k represents the number of players, $n = |V|$ and m is an upper bound on $|E|$.

The problem TF^2_{n,n^2} has been studied by Ivanyos et al. [9] and is well understood: its bounded-error quantum communication complexity is $\Theta(n)$. Indeed, it is easy to see that in the two-party setting triangle finding reduces to the computation of the disjointness[1] function $\mathsf{DISJ}_{n'}$ with $n' = n^2$. The upper bound then follows from the $O(\sqrt{n'})$-qubit protocol by Aaronson and Ambainis for disjointness [1]. The lower bound follows by combining the observation that conversely disjointness can be reduced to triangle finding with the $\Omega(\sqrt{n'})$-qubit lower bound on the quantum communication complexity of disjointness [19]. More generally, for possibly sparse graphs, the bounded-error quantum communication complexity of $\mathsf{TF}^2_{n,m}$ is $\Theta(\sqrt{m})$. Note that the classical bounded-error communication complexity of this problem is $\Theta(m)$: the upper bound follows from the trivial protocol where Alice sends all her input to Bob and the lower bound follows from lower bounds on the classical communication complexity of disjointness [11, 18].

1.3 Our contributions

In this paper, we consider the three-party quantum communication complexity of triangle finding, i.e., the problem $\mathsf{TF}^3_{n,m}$ where the edges of the graph are distributed among three players (Alice, Bob and Charlie). In the classical bounded-error communication complexity setting, the communication complexity of this problem is again $\Theta(m)$, since it is not easier than the two-party case (we can consider that one player has no edge as input). To our knowledge the quantum communication complexity of this problem has never been studied before the present work.

[1] The disjointness function $\mathsf{DISJ}_{n'}$ in the two-party setting is the following problem: Alice has a subset $x \subseteq \{1, \ldots, n'\}$, Bob has a subset $y \subseteq \{1, \ldots, n'\}$, and they want to decide if $x \cap y \neq \emptyset$.

Note that the communication complexity of $\mathsf{TF}^k_{n,m}$ for any constant $k > 3$ is equal (up to possible constant factors) to the communication complexity of $\mathsf{TF}^3_{n,m}$, which further motivates the study of the latter problem. Indeed, the former problem is again obviously not easier than the latter problem and, conversely, since a triangle consists of three edges, in the k-party case we can apply a protocol for the three-party case independently for each triple of players (the number of such triples is constant if k is constant) in order to decide whether the whole graph has a triangle or not.

Our main result is the following upper bound.[2]

▶ **Theorem 1.** *The bounded-error quantum communication complexity of* $\mathsf{TF}^3_{n,m}$ *is* $\tilde{O}(m^{7/12})$.

Let us briefly explain the main ideas that lead to the construction of our quantum protocol showing Theorem 1. The main part of the protocol consists of procedures simulating the quantum query algorithm for graph collision by Magniez, Santha, and Szegedy [16]. Indeed, for the dense case (i.e., $m \approx n^2$), it is fairly easy to see that a simple combination of a procedure implementing Grover search and another procedure simulating (in the communication complexity setting) the $\tilde{O}(n^{2/3})$-query algorithm for graph collision by Magniez, Santha, and Szegedy [16] gives the claimed $\tilde{O}(n^{7/6})$ upper bound. For sparse graphs, a first observation is that a quantum query algorithm for graph collision exploiting the sparsity of the given graph would help us to design an efficient quantum communication protocol for three-party triangle finding. However, whether graph collision can be solved with $O(n^{2/3-c})$ queries for some constant $c > 0$ even for $m = n^{4/3}$ (i.e., even when the graph is significantly sparse) is a long-standing open problem. To overcome this difficulty we consider a variant of graph collision, design a quantum algorithm for it based on quantum walks, and then show how to implement this algorithm efficiently in our setting of communication complexity (exploiting the property that each player has complete knowledge of part of the edges). We also divide the set of vertices of the graph into two sets: the set of vertices with degree smaller than n^s and the set of vertices with degree larger than n^s, where s is a parameter. This classification helps us, via Ambainis' variable costs quantum search technique [3], to reduce the communication cost needed to simulate the quantum algorithm for the variant of graph collision.

Next, we investigate whether the upper bound of Theorem 1 is tight. The trivial lower bound on the bounded-error quantum communication complexity of $\mathsf{TF}^3_{n,m}$ is $\Omega(\sqrt{m})$, since the three-party case is not easier than the two-party case. We first consider the dense case and observe that proving any better lower bound would require a breakthrough:

▶ **Proposition 2.** *If the bounded-error quantum communication complexity of* TF^3_{n,n^2} *is* $\Omega(n^{1+\epsilon})$ *for some constant* $\epsilon > 0$*, then the quantum query complexity of graph collision is* $\Omega(n^{1/2+\epsilon})$.

Proposition 2 indeed shows that proving any nontrivial lower bound on the quantum communication complexity of triangle finding would give a nontrivial lower bound on the quantum query complexity of graph collision (proving such a lower bound is a long-standing open problem in quantum query complexity). We then consider the sparse case. Theorem 1 implies that, for any value of m, any improvement over $\tilde{O}(m^{7/12})$ for the quantum communication complexity of $\mathsf{TF}^3_{n,m}$ would imply an improvement over $\tilde{O}(n^{7/6})$ for TF^3_{n,n^2} (since we can apply Theorem 1 with $n = \sqrt{m}$). We also show the following sparse version of Proposition 2:

[2] In Theorem 1 and through the paper, the notation $\tilde{O}(\cdot)$ removes the polylog(n) factors.

▶ **Proposition 3.** *If the bounded-error quantum communication complexity of* $\mathsf{TF}^3_{n,m}$ *is* $\Omega(m^{4/7+\epsilon})$ *for some* m *(seen as a function of* n*) and some constant* $\epsilon > 0$*, then the quantum query complexity of graph collision is* $\Omega(n^{1/2+\delta})$ *for some* $\delta > 0$*.*

Proposition 3 shows that giving a lower bound of the form $\Omega(m^{4/7+\epsilon})$ for some value $m < n^2$, and in particular showing that the bounds of Theorem 1 are optimal for some value of m, would also lead to a significant breakthrough. Note nevertheless that there is a gap between the best lower bound $\Omega(\sqrt{m})$ on the bounded-error quantum communication complexity of $\mathsf{TF}^3_{n,m}$ and the quantity $\Omega(m^{4/7})$ from Proposition 3. It thus still remains possible that in the sparse regime the trivial lower bound $\Omega(\sqrt{m})$ can be improved without any impact on the quantum query complexity of graph collision.

2 Preliminaries

2.1 Quantum communication complexity

Let $A_1, \ldots, A_k$ be k finite sets. Consider k players and assume that for each $i \in \{1, \ldots, k\}$ the i-th player receives as input an element $a_i \in A_i$. In the model of communication complexity, first introduced in the classical two-party setting by Yao [22], the players want to compute a function $f\colon A_1 \times \cdots A_k \to \{0,1\}$ by running a protocol such that, at the end of the protocol, each player outputs $f(a_1, \ldots, a_k)$, and they want to minimize the communication they need to compute the function f. In the quantum communication model, introduced by Yao [23], the players are allowed to communicate with qubits. More precisely, the quantum communication complexity of a quantum protocol $\mathcal{P}$ is the maximum (over all inputs) number of qubits that $\mathcal{P}$ sends. The bounded-error quantum communication complexity of f is the minimum communication complexity of any quantum protocol that computes f with probability (over the random coins used by the protocol) at least $2/3$.

2.2 Quantum query complexity of graph problems

For any finite set S and any $r \in \{1, \ldots, |S|\}$ we denote $\mathcal{X}(S, r)$ the set of all subsets of r elements of S.

Let $G = (V, E)$ be an undirected and unweighted graph, where V denotes the set of vertices and E denotes the set of edges. In the quantum query complexity setting, we only access the set of edges E through a quantum unitary operation $\mathcal{O}_G$ defined as follows. For any pair $\{u, v\} \in \mathcal{X}(V, 2)$, any bit $b \in \{0,1\}$, and any binary string $z \in \{0,1\}^*$, the operation $\mathcal{O}_G$ maps the basis state $|\{u,v\}\rangle|b\rangle|z\rangle$ to the state

$$\mathcal{O}_G|\{u,v\}\rangle|b\rangle|z\rangle = \left\{ \begin{array}{ll} |\{u,v\}\rangle|b \oplus 1\rangle|z\rangle & \text{if } \{u,v\} \in E, \\ |\{u,v\}\rangle|b\rangle|z\rangle & \text{if } \{u,v\} \notin E, \end{array} \right.$$

where $\oplus$ denotes the bit parity. Consider a quantum algorithm that computes some property of G. We say that the algorithm uses k queries if the operation $\mathcal{O}_G$, which is given as an oracle, is called k times by the algorithm.

We describe below two quantum query algorithms that we will use to construct our quantum protocol for $\mathsf{TF}^3_{n,m}$ in the communication complexity setting.

2.2.1 Quantum search with variable costs

Let X be a finite set of size N. Let $f_G\colon X \to \{0,1\}$ be a Boolean function depending on the input graph G. Assume that, for each $x \in X$, there exists a checking procedure $\mathcal{P}^x$ that computes $f_G(x)$ using t_x queries to $\mathcal{O}_G$ with high probability. The goal is to find an element $x \in X$ such that $f_G(x) = 1$ if such an element exists. When we use Grover search, this task can be solved with $O(\sqrt{N} \times t_{max})$ queries with high probability, where $t_{max} = \max_{x \in X} t_x$. Ambainis [3] proposed a more general quantum algorithm, which solves with high probability this task using

$$\tilde{O}\left(\sqrt{\sum_{x \in X} t_x^2}\right)$$

queries. In this paper, we call this algorithm Ambainis' variable costs search.

2.2.2 Quantum walk over Johnson graphs

Let S be a finite set and r be an integer such that $1 \leq r \leq |S|$. Let $f_G \colon \mathcal{X}(S, r) \to \{0, 1\}$ be a Boolean function depending on a graph G. We say that a set $A \in \mathcal{X}(S, r)$ is marked if $f_G(A) = 1$. Consider the task whose goal is to find a marked set, if such a set exists, or report that there is no marked set. Ambainis [2] developed the quantum walk search approach, which solves this task using a quantum walk over a Johnson graph.

Let us first define Johnson graphs.

▶ **Definition 4.** Let X be a finite set and $k \in \{1, \ldots, |X|\}$. A Johnson graph $J(X, k)$ is an undirected graph with vertex set $\mathcal{X}(X, k)$ where two vertices $R, R^{'} \in \mathcal{X}(X, k)$ are adjacent if and only if $|R \cap R^{'}| = k - 1$.

The state of a quantum walk over a Johnson graph $J(S, r)$ corresponds to a vertex of the Johnson graph (i.e., to a set in $\mathcal{X}(S, r)$). The key idea of the quantum walk search approach is that each state A of the walk has a data structure $D(A)$, which in general depends on G. There are three costs of the walk to consider:

- Set up cost S: The worst case number of queries to $\mathcal{O}_G$ needed to construct $D(A)$ for $A \in \mathcal{X}(S, r)$.
- Update cost U: The worst case number of queries to $\mathcal{O}_G$ needed to update $D(A)$ to $D(A')$ when one step of the quantum walk is performed (i.e., a state A of the walk moves to A' for some $A' \in \mathcal{X}(S, r)$ such that $|A \cap A'| = r - 1$).
- Checking cost C: The worst case number of queries to $\mathcal{O}_G$ needed to check if the current set A is marked by using $D(A)$ (i.e., checking whether $f_G(A) = 1$).

Let $\varepsilon > 0$ be the fraction of marked sets. The quantum walk search approach finds a marked set if such a set exists with quantum query complexity

$$\tilde{O}\left(\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\sqrt{r} \times \mathsf{U} + \mathsf{C}\right)\right),$$

with high probability (see [2, 15]).

2.3 Graph collision

Graph collision is a variant of collision problems such as element distinctness or two-to-one collision. In the quantum query complexity setting this problem is defined as follows. Given a known graph $G = (V, E)$ with $|V| = n$ and an oracle $f \colon V \to \{0, 1\}$, the graph collision problem asks whether there exists an edge $\{a, b\} \in E$ such that $f(a) = f(b) = 1$. The best known upper bound on the quantum query complexity of graph collision, obtained in [16] using quantum walks, is $\tilde{O}(n^{2/3})$. No lower bound better than the trivial $\Omega(\sqrt{n})$ is known.

In this paper, we consider the following three-party distributed version of graph collision, which is parametrized by two disjoint vertex sets $\mathcal{V}_A$, $\mathcal{V}_B$ such that $|\mathcal{V}_A| = |\mathcal{V}_B| = n$:

Three-Party Graph Collision, $\mathsf{GC}^3_{\mathcal{V}_A,\mathcal{V}_B}$

Alice's input: Boolean function $f_A : \mathcal{V}_A \to \{0,1\}$
Bob's input: Boolean function $f_B : \mathcal{V}_B \to \{0,1\}$
Charlie's input: set of edges $\mathcal{E}$ between $\mathcal{V}_A$ and $\mathcal{V}_B$
Output: $\mathsf{GC}^3_{\mathcal{V}_A,\mathcal{V}_B}(f_A, f_B, \mathcal{E}) = \bigvee_{\{i,j\}\in\mathcal{E}} f_A(i)f_B(j)$

This problem can be solved using $\tilde{O}(n^{2/3})$ qubits of communication by implementing, using standard techniques (see, e.g., [5]) to convert a query algorithm into a quantum protocol, the quantum query algorithm mentioned above since Charlie knows completely the set of edges $\mathcal{E}$ of the corresponding graph.

3 Upper Bound

In this section we show a quantum protocol for $\mathsf{TF}^3_{n,m}$ that has $\tilde{O}(m^{7/12})$-qubit communication complexity, which proves Theorem 1.

Let $G = (V, E)$, with E distributed among Alice, Bob and Charlie, be the input of $\mathsf{TF}^3_{n,m}$. Let E_A be the edges owned by Alice, E_B be the edges owned by Bob and E_C be the edges owned by Charlie. We will write $V = \{v_1, \ldots, v_n\}$. Let s be a parameter, to be chosen later, such that $0 \leq s \leq 1$.

3.1 Reduction to finding triangles in tripartite graphs

Observe that triangles with three edges in E_A (or three edges in E_B, or three edges in E_C) can be found without communication. Detecting if G contains a triangle with two edges in the same set (e.g., two edges in E_A and one edge in E_B) can be done easily with $O(\sqrt{m})$-qubit of communication, by a straightforward reduction to the two-party case and then using the two-party protocol from [9] described in the introduction. The hard case is detecting the existence of a triangle with one edge in E_A, one edge in E_B and one edge in E_C. We show below how to reduce this problem to triangle finding in some tripartite graph.

Consider the following tripartite graph $G^{'}$. The set of vertices of $G^{'}$ is the union of the three sets $I = \{v^1_1, \ldots, v^1_n\}$, $J = \{v^2_1, \ldots, v^2_n\}$, and $K = \{v^3_1, \ldots, v^3_n\}$. The set of edges of $G^{'}$ is $\mathcal{E}_A \cup \mathcal{E}_B \cup \mathcal{E}_C$, where $\mathcal{E}_A$, $\mathcal{E}_B$ and $\mathcal{E}_C$ are constructed from E as follows:

- Put edges $\{v^1_s, v^2_t\}$ and $\{v^1_t, v^2_s\}$ to $\mathcal{E}_A$ if and only if $\{v_s, v_t\} \in E_A$.
- Put edges $\{v^1_s, v^3_t\}$ and $\{v^1_t, v^3_s\}$ to $\mathcal{E}_B$ if and only if $\{v_s, v_t\} \in E_B$.
- Put edges $\{v^2_s, v^3_t\}$ and $\{v^2_t, v^3_s\}$ to $\mathcal{E}_C$ if and only if $\{v_s, v_t\} \in E_C$.

Observe that, without communicating with each other, Alice, Bob and Charlie can construct the tripartite graph $G^{'}$ in the following sense: Alice can create $\mathcal{E}_A$, Bob can create $\mathcal{E}_B$, and Charlie can create $\mathcal{E}_C$.

Note that $G^{'}$ contains a triangle if and only if G contains a triangle with one edge in E_A, one edge in E_B and one edge in E_C. For instance, if the graph G contains a triangle consisting of three vertices v_a, v_b, v_c in V such that Alice has the edge $\{v_a, v_b\} \in E_A$, Bob has the edge $\{v_a, v_c\} \in E_B$, and Charlie has the edge $\{v_b, v_c\} \in E_C$, then the tripartite graph $G^{'}$ contains the triangle with three edges $\{v^1_a, v^2_b\} \in \mathcal{E}_A$, $\{v^1_a, v^3_c\} \in \mathcal{E}_B$ and $\{v^2_b, v^3_c\} \in \mathcal{E}_C$.

3.2 Protocol for dense graphs

The dense case is easy to deal with: we can simply combine Grover search (implemented in a distributed setting) with the protocol for graph collision mentioned in Section 2.3. This gives a quantum protocol with communication complexity $\tilde{O}(\sqrt{n} \times n^{2/3}) = \tilde{O}(n^{7/6})$. For later reference we state this upper bound as follows.

▶ **Proposition 5.** *The bounded-error quantum communication complexity of* TF^3_{n,n^2} *is* $\tilde{O}(n^{7/6})$.

3.3 Classifying the vertices of G'

For any vertex v in $G^{'}$, let us denote the degree of v by d_v. For any $v \in I$, let us denote the set of neighbors in J of v by $N^I_J(v)$, and denote the set of neighbors in K of v by $N^I_K(v)$. For any $v \in J$, let us denote the set of neighbors in I of v by $N^J_I(v)$, and denote the set of neighbors in K of v by $N^J_K(v)$. For any $v \in K$, let us denote the set of neighbors in I of v by $N^K_I(v)$, and denote the set of neighbors in J of v by $N^K_J(v)$. Alice, Bob, and Charlie classify all vertices in I into two sets:

$$I^s_h = \{v \in I \mid |N^I_J(v)| \geq n^s \text{ or } |N^I_K(v)| \geq n^s\},$$
$$I^s_l = I \setminus I^s_h,$$

all vertices in J into two sets:

$$J^s_h = \{v \in J \mid |N^J_I(v)| \geq n^s \text{ or } |N^J_K(v)| \geq n^s\},$$
$$J^s_l = J \setminus J^s_h,$$

all vertices in K into two sets:

$$K^s_h = \{v \in K \mid |N^K_I(v)| \geq n^s \text{ or } |N^K_J(v)| \geq n^s\},$$
$$K^s_l = K \setminus K^s_h.$$

We will say that a vertex v of $G^{'}$ is s-high if $v \in I^s_h \cup J^s_h \cup K^s_h$, and say it is s-low if $v \in I^s_l \cup J^s_l \cup K^s_l$.

The classification of I can be done with $\tilde{O}(\frac{m}{n^s})$ bits of communication as follows. Since Alice holds the set of edges $\mathcal{E}_A$ between I and J, Alice knows, with no communication, the set $\{v \in I \mid |N^I_J(v)| \geq n^s\}$. Then Alice sends this set to both Bob and Charlie with $\tilde{O}(\frac{|\mathcal{E}_A|}{n^s}) = \tilde{O}(\frac{m}{n^s})$ bits of communication. Since Bob holds the set of edges $\mathcal{E}_B$ between I and K, Bob knows, with no communication, the set $\{v \in I \mid |N^I_K(v)| \geq n^s\}$, and then sends this set to both Alice and Charlie with $\tilde{O}(\frac{|\mathcal{E}_B|}{n^s}) = \tilde{O}(\frac{m}{n^s})$ bits of communication. Thus they obtain the sets I^s_h and I^s_l with $\tilde{O}(\frac{m}{n^s})$-bit communication. Similarly, they can obtain the classifications of J and K using $\tilde{O}(\frac{m}{n^s})$ bits of communication.

3.4 Finding a triangle with a low vertex

The following proposition is the main technical contribution of this paper.

▶ **Proposition 6.** *The existence of a triangle of* $G^{'}$ *containing at least one* s*-low vertex can be checked in* $\tilde{O}(\sqrt{m}n^{s/6})$ *qubits of communication.*

Proof. Let us consider, without loss of generality, the case where Alice, Bob, and Charlie check if $G^{'}$ has a triangle with an s-low vertex in I^s_l. In this case, Alice simulates Ambainis' variable costs search over I_l^s. The goal is to find one vertex (in I_l^s) of a triangle of $G^{'}$. For each $i \in I_l^s$ the checking procedure $\mathcal{P}^i$ of the search decides if there exists an edge $\{j,k\} \in \mathcal{E}_C$ such that $\{i,j,k\}$ is a triangle of $G^{'}$. The checking procedure $\mathcal{P}^i$ can be simulated as follows.

Let us fix $i \in I_l^s$. Let q be a parameter to be chosen later such that $0 \leq q \leq 1$. Alice and Bob define two bijective functions: $g_A^i \colon \{1,\ldots,|N_J^I(i)|\} \to N_J^I(i)$, and $g_B^i \colon \{|N_J^I(i)|+1,\ldots,|N_J^I(i)|+|N_K^I(i)|\} \to N_K^I(i)$, respectively. Then Alice and Bob send $|N_J^I(i)|$ and $|N_K^I(i)|$ to Charlie. After receiving the two values $|N_J^I(i)|$ and $|N_K^I(i)|$, Charlie simulates the following quantum walk search $\mathcal{A}_{\mathcal{W}}^i$ in order to check if there exists an edge in $\mathcal{E}_C$ that forms a triangle of $G^{'}$ with i. The walk $\mathcal{A}_{\mathcal{W}}^i$ searches for a set $R \in \mathcal{X}(\{1,\ldots,|N_J^I(i)|+|N_K^I(i)|\}, \lceil(|N_j(i)|+|N_K^I(i)|)^q\rceil) = \mathcal{X}(\{1,\ldots,d_i\}, \lceil d_i^q\rceil)$ which contains two indices $x \in \{1,\ldots,|N_J^I(i)|\}$ and $y \in \{|N_J^I(i)+1,\ldots,|N_J^I(i)+|N_K^I(i)|\}$ such that $\{i, g_A^i(x), g_B^i(y)\}$ is a triangle of $G^{'}$. When the set of marked sets is not empty, the fraction of marked sets is

$$\varepsilon = \Omega\left(\left(|N_J^I(i)|+|N_K^I(i)|\right)^{2(q-1)}\right) = \Omega\left(d_i^{2(q-1)}\right).$$

The data structure $D(R)$ stores $\{(x, g_A^i(x)) \mid x \in R \cap \{1,\ldots,|N_J^I(i)|\}\}$ and $\{(y, g_B^i(y)) \mid y \in R \cap \{|N_J^I(i)|+1,\ldots,|N_J^I(i)|+|N_K^I(i)|\}\}$. In order to construct this data structure $D(R)$ of the initial state of the walk, Charlie asks Alice to send the vertex $g_A^i(r)$ to him if $r \leq |N_J^I(i)|$, and asks Bob to send the vertex $g_B^i(r)$ to him if $r > |N_J^I(i)|$, for each $r \in R$. More precisely, for any $r \in R$, Alice and Bob perform the following unitary operators $\mathcal{O}_{g_A^i}$, $\mathcal{O}_{g_B^i}$ to the basis state $|r\rangle|0\rangle$, respectively, where $|0\rangle$ consisting of $\lceil \log n \rceil$ qubits. For any $r \in R$, the unitary operator $\mathcal{O}_{g_A^i}$ maps the basis state $|r\rangle|0\rangle$ to the state

$$\mathcal{O}_{g_A^i}|r\rangle|0\rangle = \begin{cases} |r\rangle|g_A^i(r)\rangle & \text{if } r \leq |N_J^I(i)|, \\ |r\rangle|0\rangle & \text{if } r > |N_J^I(i)|. \end{cases}$$

For any $r \in R$, the unitary operator $\mathcal{O}_{g_B^i}$ maps the basis state $|r\rangle|0\rangle$ to the state

$$\mathcal{O}_{g_B^i}|r\rangle|0\rangle = \begin{cases} |r\rangle|g_B^i(r)\rangle & \text{if } r > |N_J^I(i)|, \\ |r\rangle|0\rangle & \text{if } r \leq |N_J^I(i)|. \end{cases}$$

Thus the setup communication cost of this walk is $\mathsf{S}_{\mathcal{C}} = \tilde{O}(|R|) = \tilde{O}(d_i^q)$ qubits. The update communication cost is $\mathsf{U}_{\mathcal{C}} = \tilde{O}(1)$ qubits, and the checking communication cost is $\mathsf{C}_{\mathcal{C}} = 0$. Thus Charlie can simulate, with high probability, the quantum walk search $\mathcal{A}_{\mathcal{W}}^i$ with

$$\tilde{O}\left(\mathsf{S}_{\mathcal{C}} + \sqrt{1/\varepsilon}\left(|R|^{1/2} \times \mathsf{U}_{\mathcal{C}} + \mathsf{C}_{\mathcal{C}}\right)\right) = \tilde{O}(d_i^q + d_i^{1-q/2}), \tag{1}$$

qubits of communication. Setting $q = \frac{2}{3}$ gives the upper bound $\tilde{O}(d_i^{2/3})$.

For each $i \in I_l^s$, Alice, Bob and Charlie can thus implement $\mathcal{P}^i$ with $\tilde{O}(d_i^{2/3})$ qubits of communication. Alice can therefore simulate Ambainis' variable costs search with

$$\tilde{O}\left(\sqrt{\sum_{i \in I_l^s}\left(d_i^{2/3}\right)^2}\right).$$

qubits of communication. To analyze this upper bound, we divide the set of s-low vertices I_l^s into subsets $I_{l,p}^s = \{i \in I_l^s \mid 2^{p-1} \leq d_i \leq 2^p\}$, for $p = 1,\ldots,\lceil \log n^s \rceil$. Note that $|I_{l,p}^s| = O(\frac{m}{2^{p-1}})$, for each $p = 1,\ldots,\lceil \log n^s \rceil$. The quantum communication complexity of the quantum protocol is thus

$$
\begin{aligned}
\tilde{O}\left(\sqrt{\sum_{i\in I_l^s} d_i^{4/3}}\right) &= \tilde{O}\left(\sqrt{\sum_{p=1}^{\lceil s\log n\rceil} |I_{l,p}^s|(2^p)^{4/3}}\right)\\
&= \tilde{O}\left(\sqrt{\sum_{p=1}^{\lceil s\log n\rceil} \frac{m}{2^{p-1}}(2^p)^{4/3}}\right)\\
&= \tilde{O}\left(\sqrt{\lceil s\log n\rceil \times m(2^{s\log n})^{1/3}}\right)\\
&= \tilde{O}\left(\sqrt{m(n^s)^{1/3}}\right)\\
&= \tilde{O}\left(\sqrt{m}n^{s/6}\right),
\end{aligned}
$$

as claimed. ◀

3.5 Putting everything together

Checking if $G^{'}$ contains a triangle can be divided into four problems:

1. Checking if $G^{'}$ contains a triangle with one vertex in I_l^s, another vertex in J, and the other vertex in K.
2. Checking if $G^{'}$ contains a triangle with one vertex in I, another vertex in J_l^s, and the other vertex in K.
3. Checking if $G^{'}$ contains a triangle with one vertex in I, another vertex in J, and the other vertex in K_l^s.
4. Checking if $G^{'}$ contains a triangle with one vertex in I_h^s, another vertex in J_h^s, the other vertex in K_h^s.

Cases 1, 2 and 3 can be solved with $\tilde{O}(\sqrt{m}n^{s/6})$ qubits of communication, from Proposition 6. For case 4 (checking if $G^{'}$ contains a triangle with three s-high vertices), Alice, Bob, and Charlie directly use Proposition 5. Since $I_h^s = O(\frac{m}{n^s})$, $J_h^s = O(\frac{m}{n^s})$, and $K_h^s = O(\frac{m}{n^s})$, Case 4 can be solved with $\tilde{O}\left(\left(\frac{m}{n^s}\right)^{7/6}\right)$ qubits of communication.

Thus the total communication cost of the quantum protocol for $\mathsf{TF}^3_{n,m}$ is

$$
\tilde{O}\left(\frac{m}{n^s} + m^{1/2}n^{s/6} + \frac{m^{7/6}}{n^{7s/6}}\right),
$$

which is optimized by taking s such that $n^s = m^{1/2}$, giving the final quantum communication complexity of $\tilde{O}(m^{7/12})$.

4 Lower Bounds

In this section we give the proofs of Propositions 2 and 3. Let us denote by $\mathcal{Q}_{\mathsf{GC}}(n)$ the quantum query complexity of graph collision, when parametrized by graphs with n vertices.

Proof of Proposition 2. From the construction of the protocol giving the bound of Proposition 5, it follows that there exists a quantum protocol which computes, with high probability, TF^3_{n,n^2} with $\tilde{O}(\sqrt{n}\times\mathcal{Q}_{\mathsf{GC}}(n))$ qubits of communication. Thus, an $\Omega(n^{1+\epsilon})$ lower bound on the bounded quantum communication complexity of TF^3_{n,n^2} for some constant $\epsilon > 0$ implies an $\Omega(n^{1/2+\epsilon})$ lower bound on the quantum query complexity of graph collision. ◀

Proof of Proposition 3. Let s be a parameter such that $0 \leq s \leq 1$. From Section 3.5 and the construction of the protocol giving the bound of Proposition 5, it follows that there exists a quantum communication protocol which computes $\mathsf{TF}^3_{n,m}$ with bounded-error quantum communication complexity

$$\tilde{O}\left(\frac{m}{n^s} + m^{1/2}n^{s/6} + \sqrt{\frac{m}{n^s}} \times \mathcal{Q}_{\mathsf{GC}}(m/n^s)\right).$$

Suppose an $\Omega(m^{4/7+\epsilon})$ lower bound on the bounded-error quantum communication complexity of $\mathsf{TF}^3_{n,m}$ for some constant $\epsilon > 0$. Setting $n^s = m^{3/7+6\epsilon}$ gives the upper bound

$$\tilde{O}\left(m^{4/7+\epsilon} + m^{2/7-3\epsilon} \times \mathcal{Q}_{\mathsf{GC}}(m^{4/7-6\epsilon})\right).$$

This implies the claimed lower bound $\Omega(n^{\frac{2/7+4\epsilon}{4/7-6\epsilon}})$ on the quantum query complexity of graph collision. ◀

References

1 Scott Aaronson and Andris Ambainis. Quantum search of spatial regions. In *Proceedings of the 51st Symposium on Foundations of Computer Science*, pages 200–209, 2003.

2 Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007.

3 Andris Ambainis. Quantum search with variable times. *Theory of Computing Systems*, 47(3):786–807, 2010.

4 Aleksandrs Belovs. Span programs for functions with constant-sized 1-certificates. In *Proceedings of the 44th Symposium on Theory of Computing*, pages 77–84, 2012.

5 Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the 30th Symposium on Theory of Computing*, pages 63–68, 1998.

6 Harry Buhrman, Christoph Dürr, Mark Heiligman, and Peter Høyer. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34(6):1324–1330, 2005.

7 Titouan Carette, Mathieu Laurière, and Frédéric Magniez. Extended learning graphs for triangle finding. In *Proceedings of the 34th International Symposium on Theoretical Aspects of Computer Science*, pages 20:1–20:14, 2017.

8 Andrew M. Childs and Robin Kothari. Quantum query complexity of minor-closed graph properties. *SIAM Journal on Computing*, 41(6):1426–1450, 2012.

9 Gábor Ivanyos, Hartmut Klauck, Troy Lee, Miklos Santha, and Ronald de Wolf. New bounds on the classical and quantum communication complexity of some graph properties. In *Proceedings of the 32nd International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 148–159, 2012.

10 Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Nested quantum walks with quantum data structures. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1474–1485, 2013.

11 Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.

12 François Le Gall. Improved quantum algorithm for triangle finding via combinatorial arguments. In *Proceedings of the 28th Symposium on the Theory of Computing*, pages 216–225, 2014.

13 François Le Gall and Shogo Nakajima. Quantum algorithm for triangle finding in sparse graphs. In *Proceedings of the 26th International Symposium on Algorithms and Computation*, pages 590–600, 2015.

14 Troy Lee, Frédéric Magniez, and Miklos Santha. Improved quantum query algorithms for triangle finding and associativity testing. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1486–1502, 2013.

15 Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011.

16 Frédéric Magniez, Miklos Santha, and Mario Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007.

17 Mihai Patrascu. Towards polynomial lower bounds for dynamic problems. In *Proceedings of the 42nd Symposium on Theory of Computing*, pages 603–610, 2010.

18 Alexander A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992.

19 Alexander A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya Mathematics*, 67(1):145–159, 2003.

20 Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *Proceedings of the 51st Symposium on Foundations of Computer Science*, pages 645–654, 2010.

21 Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM Journal on Computing*, 42(3):831–854, 2013.

22 Andrew C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11th Symposium on Theory of Computing*, pages 209–213, 1979.

23 Andrew C. Yao. Quantum circuit complexity. In *Proceedings of the 34st Symposium on Foundations of Computer Science*, pages 352–361, 1993.