1 Quantum Streaming Algorithms

1.1 Classical Streaming for Triangle Counting and Distinguishing

Problem 1.1. Triangle Counting TC

INSTANCE: Graph \( G = (V, E) \)

QUESTION: Approximate the number of triangles in \( G \).

A related problem that is usually considered in the literature is that of Triangle Distinguishing, which is defined as follows.

Problem 1.2. Triangle Distinguishing TD

INSTANCE: Graph \( G = (V, E) \), a number \( T \), and the promise that \( G \) has either 0 triangles or \( T \) triangles.

QUESTION: Does \( G \) have 0 triangles or \( T \) triangles?

Clearly \( TD \leq TC \). Hence, a lower bound on \( TD \) implies a lower bound on \( TC \).

We will state lower bounds on \( TD \) (and hence \( TC \)). We now state the problems that are used for these lower bounds.

Definition 1. Let \( n \in \mathbb{N} \).

1. A perfect matching \( M \) over \([2n]\) is a set of \( n \) ordered pairs \((i, j)\), where \( i \) and \( j \) are distinct elements of \([2n]\), such that every \( \ell \in [2n] \) is in exactly 1 ordered pair.

2. Let \( M \) be a perfect matching \( M \) over \([2n]\). We identify \( M \) with the following \( n \times 2n \) matrix. For every ordered pair \((i, j)\) in the matching there is a row with 1’s in the \( i \)th and \( j \)th spot, and 0’s everywhere else. Note that a perfect matching can be associated to many different matrices. We will turn this around: we will give Bob a perfect matching by giving him a matrix.

Problem 1.3. Boolean Hidden Matching BHM

INSTANCE: Alice gets a string \( x \in \{0, 1\}^{2n} \). Bob gets (a) a perfect matching \( M \) over \([2n]\) via a matrix as described in Definition 1, and (b) a string \( w \in \{0, 1\}^n \) where \( w \) is promised to satisfy either \( Mx = w \) or \( Mx = \overline{w} \) (where \( \overline{w} \) is \( w \) with every bit flipped).

QUESTION: Determine which is the case: \( Mx = w \) or \( Mx = \overline{w} \).

Theorem 1. (Gavinsky et al. [4]) The randomized 1-way communication complexity, with Alice sending, is \( \Omega(\sqrt{n}) \).

Notation 1. Let \( n \) denote the number of vertices, \( m \) denote the number of edges, and \( T \) is as in the problem statement. \( \Delta_V \) (respectively \( \Delta_E \)) is the maximum number of triangles in \( G \) that share a vertex (respectively an edge).

The following are known.

Theorem 2.

1. (Jayaram & Kallaugher [5]) There is a single-pass streaming algorithm for TC that uses space \( \tilde{O}\left(\frac{m\Delta_E}{T} + \frac{m\sqrt{\Delta_V}}{T}\right) \).

2. (Braverman et al. [3]) Any single-pass streaming algorithm for TD (and hence for TC) uses space \( \Omega\left(\frac{m\Delta_E}{T}\right) \). This proof uses a reduction of INDEX to TD.
3. (Kallaugher and Price [7]) Any single-pass streaming algorithm for TD (and hence for TC) uses space $\Omega\left(\frac{m\Delta E}{T}\right)$. This proof uses a reduction of BHM to TD.

4. Any single-pass streaming algorithm for TD (and hence for TC) requires space $\Omega\left(\frac{m\Delta V}{\Delta E} + \frac{m\sqrt{\Delta V}}{T}\right)$. This follows from Parts 2 and 3. Note that we now have matching bounds for one-pass streaming algorithms for TC.

1.2 Quantum Streaming for Triangle Counting and Distinguishing

Quantum streaming algorithms were first defined by Khadiev et al. [8] (see also Ablayev et al. [1]). We will discuss modifying the proofs of the lower bounds for streaming on TD and TC from Theorem 2 to obtain lower bounds for quantum streaming for these problems.

Theorem 2 used that INDEX has communication complexity $\Omega(n)$. Fortunately, Ambainis et al. [2] showed that INDEX also has quantum communication complexity $\Omega(n)$. Hence we have the following analog to Theorem 2.2 by the same proof:

**Theorem 3.** Any single-pass quantum streaming algorithm for TD (and hence for TC) requires space $\Omega\left(\frac{m\Delta E}{T}\right)$. This proof uses a reduction of INDEX to TD. This follows from Theorem 2.2 and the work of Ambainis et al. [2].

Can we do the same for Theorem 2.3? No. Gavinsky et al. [4] showed that the quantum communication complexity of BHM is $O(\log n)$. Hence we do not have a non-trivial lower bound for TC or TD in the region where $\Delta E = O(1)$ and $T = \Omega(n)$. Indeed, there is a quantum streaming algorithm that works well in that region. Kallaugher [6] showed the following.

**Theorem 4.** Restrict TC to the graphs where $\Delta E = O(1)$, $\Delta V = \Omega(T)$, and $T = \Omega(m)$. There is a single-pass quantum streaming algorithm for TC that uses space $\tilde{O}(m^{2/5})$.

Open 1. Find a lower bound of the form $\Omega(m^c)$ for TC in the case where $\Delta E = O(1)$, $\Delta V = \Omega(T)$, and $T = \Omega(m)$.

1.3 Classical Streaming for k-Clique Counting and Distinguishing

In this section, we define two problems for k-clique finding which are analogous to Triangle Counting and Triangle Distinguishing.

**Problem 1.4.** k-Clique Counting (kCC)

INSTANCE: Graph $G = (V, E)$ and $k \in \mathbb{N}$.

QUESTION: Approximate the number of cliques of size $k$ in $G$.

**Problem 1.5.** k-Clique Distinguishing (kCD)

INSTANCE: Graph $G = (V, E)$, $C \in \mathbb{N}$, and the promise that $G$ has either 0 k-cliques or $\geq C$ k-cliques.

QUESTION: Determine if $G$ has 0 k-cliques or $\geq C$ k-cliques.

Clearly kCD $\leq$ kCC. Hence a lower bound on kCD implies a lower bound on kCC.

Theorem 2.2 stated a $\Omega\left(\frac{m\Delta E}{T}\right)$ space lower bound for single-pass streaming algorithms for Triangle Distinguishing. A similar proof gives the same lower bound for k-Clique Distinguishing (with $T$ being the number of k-cliques); however this gives a trivial lower bound on
most graphs, since $\Delta_E$ is usually small. We want a stronger lower bound for more general graphs. Additionally, since the quantum streaming complexity of triangle counting in the parameter setting $\Delta_E = O(1)$ and $T = \Omega(m)$ is an open problem it might be instructive to look for lower bounds on $k$-clique counting for $k \geq 4$ in this parameter setting to understand if the difficulty of this problem is unique for triangle counting.

For the next exercise you need the following definition and theorem.

**Definition 2.** Let $k, n \in \mathbb{N}$.

1. A **perfect hypermatching** $M$ over $[kn]$ is a set of $n$ ordered $k$-tuples $(i_1, \ldots, i_k)$, where $i_1, \ldots, i_k$ are distinct elements of $[kn]$, such that every $\ell \in [kn]$ is in exactly 1 ordered $k$-tuple.

2. Let $M$ be a perfect hypermatching $M$ over $[kn]$. We identify $M$ with the following $n \times kn$ matrix. For every ordered $k$-tuple $(i_1, \ldots, i_k)$ in the hypermatching there is a row with 1’s in the $i_1$th, $i_2$th, $\ldots$, $i_k$th spot, and 0’s everywhere else. Note that a perfect hypermatching can be associated to many different matrices. We will turn this around: we will give Bob a perfect hypermatching by giving him a matrix.

**Problem 1.6.** **Boolean Hidden Hypermatching BHHM**

**INSTANCE:** Alice gets a string $x \in \{0, 1\}^{kn}$. Bob gets (a) a perfect hypermatching $M$ over $[kn]$ via a matrix as described in Definition 2, and (b) a string $w \in \{0, 1\}^n$ where $w$ is promised to satisfy either $Mx = w$ or $Mx = \overline{w}$ (where $\overline{w}$ is $w$ with every bit flipped).

**QUESTION:** Determine which is the case: $Mx = w$ or $Mx = \overline{w}$.

**Theorem 5.**

1. (Verbin & Yu [10]) The randomized one-way communication complexity for BHHM, with Alice sending, is $\Omega(n^{1-(1/k)})$.

2. (Shi et al. [9]) the quantum one-way communication complexity for BHHM, with Alice sending, is $\Omega(n^{1-(2/k)})$.

**Exercise 1.**

1. Any classical single-pass streaming algorithm for $k$CD requires $\Omega(m^{1-1/k})$ bits of space. (Hint: Use the lower bound on BHHM from Theorem 5.1).

2. Any quantum single-pass streaming algorithm for $k$CD requires $\Omega(m^{1-2/k})$ qubits of space. (Hint: Use the lower bound on BHHM from Theorem 5.2).

### 1.4 Future Directions

**Open 2.**

1. We have looked at counting and detecting triangles and $k$-cliques. Look at the problems of counting and detecting other subgraphs such as $k$-cycles.

2. Obtain classical and quantum upper and lower bounds on $p$-pass streaming algorithms.
References


https://doi.org/10.1137/1.9781611973082.2.