TRIÈST: Counting Local and Global Triangles in Fully-dynamic Streams with Fixed Memory Size

LORENZO DE STEFANI, Brown University
ALESSANDRO EPASTO, Google, Inc
MATTEO RIONDATO, Two Sigma Investments, LP
ELI UPFAL, Brown University

"Ogni lassada xe persa." – Proverb from Trieste, Italy.

We present TRIÈST, a suite of one-pass streaming algorithms to compute unbiased, low-variance, high-quality approximations of the global and local (i.e., incident to each vertex) number of triangles in a fully-dynamic graph represented as an adversarial stream of edge insertions and deletions.

Our algorithms use reservoir sampling and its variants to exploit the user-specified memory space at all times. This is in contrast with previous approaches, which require hard-to-choose parameters (e.g., a fixed sampling probability) and offer no guarantees on the amount of memory they use. We analyze the variance of the estimations and show novel concentration bounds for these quantities.

Our experimental results on very large graphs demonstrate that TRIÈST outperforms state-of-the-art approaches in accuracy and exhibits a small update time.

CCS Concepts: • Mathematics of computing → Graph enumeration; Probabilistic algorithms; • Information systems → Data stream mining; • Human-centered computing → Social networks; • Theory of computation → Dynamic graph algorithms; Sketching and sampling;

Additional Key Words and Phrases: cycle counting, reservoir sampling, subgraph counting

ACM Reference format:

1 INTRODUCTION

Exact computation of characteristic quantities of Web-scale networks is often impractical or even infeasible due to the humongous size of these graphs. It is natural in these cases to resort to

1Any missed chance is lost forever.
efficient-to-compute approximations of these quantities that, when of sufficiently high quality, can be used as proxies for the exact values.

In addition to being huge, many interesting networks are fully-dynamic and can be represented as a stream whose elements are edges/nodes insertions and deletions which occur in an arbitrary (even adversarial) order. Characteristic quantities in these graphs are intrinsically volatile, hence there is limited added value in maintaining them exactly. The goal is rather to keep track, at all times, of a high-quality approximation of these quantities. For efficiency, the algorithms should aim at exploiting the available memory space as much as possible and they should require only one pass over the stream.

We introduce TRIEST, a suite of sampling-based, one-pass algorithms for adversarial fully-dynamic streams to approximate the global number of triangles and the local number of triangles incident to each vertex. Mining local and global triangles is a fundamental primitive with many applications (e.g., community detection [4], topic mining [13], spam/anomaly detection [3, 28], ego-networks mining [14] and protein interaction networks analysis [30].)

Many previous works on triangle estimation in streams also employ sampling (see Sect. 3), but they usually require the user to specify in advance an edge sampling probability \( p \) that is fixed for the entire stream. This approach presents several significant drawbacks. First, choosing a \( p \) that allows to obtain the desired approximation quality requires to know or guess a number of properties of the input (e.g., the size of the stream). Second, a fixed \( p \) implies that the sample size grows with the size of the stream, which is problematic when the stream size is not known in advance: if the user specifies a large \( p \), the algorithm may run out of memory, while for a smaller \( p \) it will provide a suboptimal estimation. Third, even assuming to be able to compute a \( p \) that ensures (in expectation) full use of the available space, the memory would be fully utilized only at the end of the stream, and the estimations computed throughout the execution would be suboptimal.

**Contributions.** We address all the above issues by taking a significant departure from the fixed-probability, independent edge sampling approach taken even by state-of-the-art methods [28]. Specifically:

- We introduce TRIEST (TRIangle Estimation from STreams), a suite of one-pass streaming algorithms to approximate, at each time instant, the global and local number of triangles in a fully-dynamic graph stream (i.e., a sequence of edges additions and deletions in arbitrary order) using a fixed amount of memory. This is the first contribution that enjoys all these properties. TRIEST only requires the user to specify the amount of available memory, an interpretable parameter that is definitively known to the user.

- Our algorithms maintain a sample of edges: they use the reservoir sampling [42] and random pairing [16] sampling schemes to exploit the available memory as much as possible. To the best of our knowledge, ours is the first application of these techniques to subgraph counting in fully-dynamic, arbitrarily long, adversarially ordered streams. We present an analysis of the unbiasedness and of the variance of our estimators, and establish strong concentration results for them. The use of reservoir sampling and random pairing requires additional sophistication in the analysis, as the presence of an edge in the sample is not independent from the concurrent presence of another edge. Hence, in our proofs we must consider the complex dependencies in events involving sets of edges. The gain is worth the effort: we prove that the variance of our algorithms is smaller than that of state-of-the-art methods [28], and this is confirmed by our experiments.

- We conduct an extensive experimental evaluation of TRIEST on very large graphs, some with billions of edges, comparing the performances of our algorithms to those of existing state-of-the-art contributions [20, 28, 36]. Our algorithms significantly and consistently
reduce the average estimation error by up to 90% w.r.t. the state of the art, both in the global and local estimation problems, while using the same amount of memory. Our algorithms are also extremely scalable, showing update times in the order of hundreds of microseconds for graphs with billions of edges.

**Paper organization.** We formally introduce the settings and the problem in Sect. 2. In Sect. 3 we discuss related works. We present and analyze TRIÉST and discuss our design choices in Sect. 4. The results of our experimental evaluation are presented in Sect. 5. We draw our conclusions in Sect. 6. Some of the proofs of our theoretical results are deferred to Appendix A.

2 PRELIMINARIES

We study the problem of counting global and local triangles in a fully-dynamic undirected graph as an arbitrary (adversarial) stream of edge insertions and deletions.

Formally, for any (discrete) time instant \( t \geq 0 \), let \( G^{(t)} = (V^{(t)}, E^{(t)}) \) be the graph observed up to and including time \( t \). At time \( t = 0 \) we have \( V^{(0)} = E^{(0)} = \emptyset \). For any \( t > 0 \), at time \( t + 1 \) we receive an element \( e_{t+1} = ((u, v), \bullet) \) from a stream, where \( \bullet \in \{+,-\} \) and \( u, v \) are two distinct vertices. The graph \( G^{(t+1)} = (V^{(t+1)}, E^{(t+1)}) \) is obtained by inserting a new edge or deleting an existing edge as follows:

\[
E^{(t+1)} = \begin{cases} 
E^{(t)} \cup \{(u, v)\} & \text{if } \bullet = " + " \\
E^{(t)} \setminus \{(u, v)\} & \text{if } \bullet = " - " 
\end{cases}.
\]  

If \( u \) or \( v \) do not belong to \( V^{(t)} \), they are added to \( V^{(t+1)} \). Nodes are deleted from \( V^{(t)} \) when they have degree zero.

Edges can be added and deleted in the graph in an arbitrary adversarial order, i.e., as to cause the worst outcome for the algorithm, but we assume that the adversary has no access to the random bits used by the algorithm. We assume that all operations have effect: if \( e \in E^{(t)} \) (resp. \( e \notin E^{(t)} \), \((+, e)\) (resp. \((-,-)\)) can not be on the stream at time \( t + 1 \).

Given a graph \( G^{(t)} = (V^{(t)}, E^{(t)}) \), a triangle in \( G^{(t)} \) is a set of three edges \( \{(u, v), (v, w), (w, u)\} \subseteq E^{(t)} \), with \( u, v, \) and \( w \) being three distinct vertices. We refer to \( \{u, v, w\} \subseteq V^{(t)} \) as the corners of the triangle. We denote with \( \Delta^{(t)} \) the set of all triangles in \( G^{(t)} \), and, for any vertex \( u \in V^{(t)} \), with \( \Delta_{u}^{(t)} \) the subset of \( \Delta^{(t)} \) containing all and only the triangles that have \( u \) as a corner.

**Problem definition.** We study the Global (resp. Local) Triangle Counting Problem in Fully-dynamic Streams, which requires to compute, at each time \( t \geq 0 \) an estimation of \( |\Delta^{(t)}| \) (resp. for each \( u \in V \) an estimation of \( |\Delta_{u}^{(t)}| \)).

**Multigraphs.** Our approach can be further extended to count the number of global and local triangles on a multigraph represented as a stream of edges. Using a formalization analogous to that discussed for graphs, for any (discrete) time instant \( t \geq 0 \), let \( G^{(t)} = (V^{(t)}, E^{(t)}) \) be the multigraph observed up to and including time \( t \), where \( E^{(t)} \) is now a bag of edges between vertices of \( V^{(t)} \). The multigraph evolves through a series of edge additions and deletions according to almost the same process described for graphs, with the exception that now all operations must have effect on the bag of edges \( E^{(t)} \). Thus, for example, we may have \((+, e)\) on the stream at time \( t \) and again \((+, e)\) at time \( t + 1 \). Some additional modifications to the model are needed to handle deletions appropriately, and we outline them in Sect. 4.4.3. The definition of triangle in a multigraph is the same as in a graph. As before we denote with \( \Delta^{(t)} \) the set of all triangles in \( G^{(t)} \), but now this set may contain multiple triangles with the same set of vertices, although each of these triangles will be a different set of three edges among those vertices, i.e., a different subset of the bag \( E^{(t)} \). For any vertex \( u \in V^{(t)} \), we still denote with \( \Delta_{u}^{(t)} \) the subset of \( \Delta^{(t)} \) containing all and only the triangles that
have \( u \) as a corner, with a similar caveat as \( \Delta^{(f)} \). The problems of global and local triangle counting in multigraph edge streams are defined exactly in the same way as for graph edge streams.

3 RELATED WORK

The literature on exact and approximate triangle counting is extremely rich, including exact algorithms, graph sparsifiers [40, 41], complex-valued sketches [22, 29], and MapReduce algorithms [32–35, 38]. Here we restrict the discussion to the works most related to ours, i.e., to those presenting algorithms for counting or approximating the number of triangles from data streams. We refer to the survey by Latapy [26] for an in-depth discussion of other works. Table 1 presents a summary of the comparison, in terms of desirable properties, between this work and relevant previous contributions.

Table 1. Comparison with previous contributions

<table>
<thead>
<tr>
<th>Work</th>
<th>Single pass</th>
<th>Fixed space</th>
<th>Local counts</th>
<th>Global counts</th>
<th>Fully-dynamic streams</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3]</td>
<td>✓</td>
<td>✓/✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[23]</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>[36]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>[20]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>[1]</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>[28]</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗/✓</td>
<td>✗</td>
</tr>
<tr>
<td>This work</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

\*\*\*The required space is \( O(|V^{(f)}|) \), which, although not dependent on the number of triangles or on the number of edges, is not fixed, in the sense that it cannot be fixed a-priori.\*\*\*

\*\*\*Global triangle counting is not mentioned in the article, but the extension is straightforward.\*\*\*

Many authors presented algorithms for more restricted (i.e., less generic) settings than ours, or for which the constraints on the computation are more lax [2, 7, 21, 24]. For example, Becchetti et al. [3] and Kolountzakis et al. [23] present algorithms for approximate triangle counting from static graphs by performing multiple passes over the data. Pavan et al. [36] and Jha et al. [20] propose algorithms for approximating only the global number of triangles from edge-insertion-only streams. Bulteau et al. [6] present a one-pass algorithm for fully-dynamic graphs, but the triangle count estimation is (expensively) computed only at the end of the stream and the algorithm requires, in the worst case, more memory than what is needed to store the entire graph. Ahmed et al. [1] apply the sampling-and-hold approach to insertion-only graph stream mining to obtain, only at the end of the stream and using non-constant space, an estimation of many network measures including triangles.

None of these works has all the features offered by TRÉST: performs a single pass over the data, handles fully-dynamic streams, uses a fixed amount of memory space, requires a single interpretable parameter, and returns an estimation at each time instant. Furthermore, our experimental results show that we outperform the algorithms from [20, 36] on insertion-only streams.

Lim and Kang [28] present an algorithm for insertion-only streams that is based on independent edge sampling with a fixed probability: for each edge on the stream, a coin with a user-specified fixed tails probability \( p \) is flipped, and, if the outcome is tails, the edge is added to the stored sample and the estimation of local triangles is updated. Since the memory is not fully utilized during most of the stream, the variance of the estimate is large. Our approach handles fully-dynamic streams...
and makes better use of the available memory space at each time instant, resulting in a better estimation, as shown by our analytical and experimental results.

Vitter [42] presents a detailed analysis of the reservoir sampling scheme and discusses methods to speed up the algorithm by reducing the number of calls to the random number generator. Random Pairing [16] is an extension of reservoir sampling to handle fully-dynamic streams with insertions and deletions. Cohen et al. [9] generalize and extend the Random Pairing approach to the case where the elements on the stream are key-value pairs, where the value may be negative (and less than −1). In our settings, where the value is not less than −1 (for an edge deletion), these generalizations do not apply and the algorithm presented by Cohen et al. [9] reduces essentially to Random Pairing.

4 ALGORITHMS
We present TRIÉST, a suite of three novel algorithms for approximate global and local triangle counting from edge streams. The first two work on insertion-only streams, while the third can handle fully-dynamic streams where edge deletions are allowed. We defer the discussion of the multigraph case to Sect. 4.4.

Parameters. Our algorithms keep an edge sample \( S \) containing up to \( M \) edges from the stream, where \( M \) is a positive integer parameter. For ease of presentation, we realistically assume \( M \geq 6 \). In Sect. 1 we motivated the design choice of only requiring \( M \) as a parameter and remarked on its advantages over using a fixed sampling probability \( p \). Our algorithms are designed to use the available space as much as possible.

Counters. TRIÉST algorithms keep counters to compute the estimations of the global and local number of triangles. They always keep one global counter \( \tau \) for the estimation of the global number of triangles. Only the global counter is needed to estimate the total triangle count. To estimate the local triangle counts, the algorithms keep a set of local counters \( \tau_u \) for a subset of the nodes \( u \in V(t) \). The local counters are created on the fly as needed, and always destroyed as soon as they have a value of 0. Hence our algorithms use \( O(M) \) space (with one exception, see Sect. 4.2).

Notation. For any \( t \geq 0 \), let \( G^S = (V^S, E^S) \) be the subgraph of \( G(t) \) containing all and only the edges in the current sample \( S \). We denote with \( N_u^S \) the neighborhood of \( u \) in \( G^S \): \( N_u^S = \{ v \in V(t) : (u,v) \in S \} \) and with \( N_{u,v}^S = N_u^S \cap N_v^S \) the shared neighborhood of \( u \) and \( v \) in \( G^S \).

Presentation. We only present the analysis of our algorithms for the problem of global triangle counting. For each presented result involving the estimation of the global triangle count (e.g., unbiasedness, bound on variance, concentration bound) and potentially using other global quantities (e.g., the number of pairs of triangles in \( \Delta(t) \) sharing an edge), it is straightforward to derive the correspondent variant for the estimation of the local triangle count, using similarly defined local quantities (e.g., the number of pairs of triangles in \( \Delta(t) \) sharing an edge.)

4.1 A first algorithm – TRIÉST-BASE
We first present TRIÉST-BASE, which works on insertion-only streams and uses standard reservoir sampling [42] to maintain the edge sample \( S \):

- If \( t \leq M \), then the edge \( e_t = (u,v) \) on the stream at time \( t \) is deterministically inserted in \( S \).
- If \( t > M \), TRIÉST-BASE flips a biased coin with heads probability \( M/t \). If the outcome is heads, it chooses an edge \( (w,z) \in S \) uniformly at random, removes \( (w,z) \) from \( S \), and inserts \( (u,v) \) in \( S \). Otherwise, \( S \) is not modified.
After each insertion (resp. removal) of an edge \((u, v)\) from \(S\), \textsc{triést-base} calls the procedure \textsc{UpdateCounters} that increments (resp. decrements) \(\tau\), \(\tau_u\) and \(\tau_v\) by \(|N_{u,v}^S|\), and \(\tau_c\) by one, for each \(c \in N_{u,v}^S\).

The pseudocode for \textsc{triést-base} is presented in Alg. 1.

\begin{algorithm}[h]
\caption{\textsc{triést-base}}
\begin{algorithmic}[1]
\Require Insertion-only edge stream \(\Sigma\), integer \(M \geq 6\)
\State \(S \leftarrow \emptyset\), \(t \leftarrow 0\), \(\tau \leftarrow 0\)
\For{each element \((+, (u, v))\) from \(\Sigma\)}
\State \(t \leftarrow t + 1\)
\If{\textsc{SampleEdge}((u, v), t)}
\State \(S \leftarrow S \cup \{(u, v)\}\)
\State \textsc{UpdateCounters}(+, (u, v))
\EndIf
\EndFor
\Function{SampleEdge}((u, v), t)
\If{\(t \leq M\)}
\State \Return True
\ElseIf{\textsc{FlipBiasedCoin}(\(\frac{M}{T}\)) = heads}
\State \((u', v')\) \leftarrow \text{random edge from} \(S\)
\State \(S \leftarrow S \setminus \{(u', v')\}\)
\State \textsc{UpdateCounters}(-, (u', v'))
\State \Return True
\Else
\State \Return False
\EndIf
\EndFunction
\State \textsc{UpdateCounters}(\(-, (u, v)\))
\State \(N_{u,v}^S \leftarrow N_u^S \cap N_v^S\)
\For{all \(c \in N_{u,v}^S\)}
\State \(\tau \leftarrow \tau \bullet 1\)
\State \(\tau_c \leftarrow \tau_c \bullet 1\)
\State \(\tau_u \leftarrow \tau_u \bullet 1\)
\State \(\tau_v \leftarrow \tau_v \bullet 1\)
\EndFor
\EndAlgorithm
\end{algorithm}

\subsection{Estimation}

For any pair of positive integers \(a\) and \(b\) such that \(a \leq \min\{M, b\}\) let
\[
\xi_{a,b} = \begin{cases}
\binom{b}{M}\frac{b-a}{M-a} = \prod_{i=0}^{a-1} \frac{b-i}{M-i} & \text{if } b \leq M \\
1 & \text{otherwise}
\end{cases}
\]

As shown in the following lemma, \(\xi_{k, t}^{-1}\) is the probability that \(k\) edges of \(G(t)\) are all in \(S\) at time \(t\), i.e., the \(k\)-th order inclusion probability of the reservoir sampling scheme. The proof can be found in App. A.1.

\begin{lemma}
For any time step \(t\) and any positive integer \(k \leq t\), let \(B\) be any subset of \(E(t)\) of size \(|B| = k \leq t\). Then, at the end of time step \(t\),
\[
\Pr(B \subseteq S) = \begin{cases}
0 & \text{if } k > M \\
\xi_{k, t}^{-1} & \text{otherwise}
\end{cases}
\]
\end{lemma}

We make use of this lemma in the analysis of \textsc{triést-base}.

Let, for any \(t \geq 0\), \(\xi(t) = \xi_{k, t}\) and let \(\tau(t)\) (resp. \(\tau_u(t)\)) be the value of the counter \(\tau\) at the end of time step \(t\) (i.e., after the edge on the stream at time \(t\) has been processed by \textsc{triést-base} (resp. the
value of the counter \( \tau_u \) at the end of time step \( t \) if there is such a counter, 0 otherwise. When queried at the end of time \( t \), TRIÊST-BASE returns \( \xi^{(t)}\tau^{(t)} \) (resp. \( \xi^{(t)}\tau_u^{(t)} \)) as the estimation for the global (resp. local for \( u \in V^{(t)} \)) triangle count.

4.1.2 Analysis. We now present the analysis of the estimations computed by TRIÊST-BASE. Specifically, we prove their unbiasedness (and their exactness for \( t \leq M \)) and then show an exact derivation of their variance and a concentration result. We show the results for the global counts, but results analogous to those in Thms. 4.2, 4.4, and 4.5 hold for the local triangle count for any \( u \in V^{(t)} \), replacing the global quantities with the corresponding local ones. We also compare, theoretically, the variance of TRIÊST-BASE with that of a fixed-probability edge sampling approach [28], showing that TRIÊST-BASE has smaller variance for the vast majority of the stream.

4.1.3 Expectation. We have the following result about the estimations computed by TRIÊST-BASE.

**Theorem 4.2.** We have

\[
\xi^{(t)}\tau^{(t)} = \tau^{(t)} = |\Delta^{(t)}| \text{ if } t \leq M
\]

\[
\mathbb{E}\left[\xi^{(t)}\tau^{(t)}\right] = |\Delta^{(t)}| \text{ if } t > M.
\]

The TRIÊST-BASE estimations are not only unbiased in all cases, but actually exact for \( t \leq M \), i.e., for \( t \leq M \), they are the true global/local number of triangles in \( G^{(t)} \).

To prove Thm. 4.2, we need to introduce a technical lemma. Its proof can be found in Appendix A.1. We denote with \( \Delta^S \) the set of triangles in \( G^S \).

**Lemma 4.3.** After each call to UpdateCounters, we have \( \tau = |\Delta^S| \) and \( \tau_u = |\Delta^S_v| \) for any \( v \in V^S \) s.t. \( |\Delta^S_v| \geq 1 \).

From here, the proof of Thm. 4.2 is a straightforward application of Lemma 4.3 for the case \( t \leq M \) and of that lemma, the definition of expectation, and Lemma 4.1 otherwise. The complete proof can be found in App. A.1.

4.1.4 Variance. We now analyze the variance of the estimation returned by TRIÊST-BASE for \( t > M \) (the variance is 0 for \( t \leq M \)).

Let \( r^{(t)} \) be the total number of unordered pairs of distinct triangles from \( \Delta^{(t)} \) sharing an edge,\(^2\) and \( w^{(t)} = \left( \frac{|\Delta^{(t)}|}{2} - r^{(t)} \right) \) be the number of unordered pairs of distinct triangles that do not share any edge.

**Theorem 4.4.** For any \( t > M \), let \( f(t) = \xi^{(t)} - 1 \),

\[
g(t) = \xi^{(t)} \frac{(M-3)(M-4)}{(t-3)(t-4)} - 1
\]

and

\[
h(t) = \xi^{(t)} \frac{(M-3)(M-4)(M-5)}{(t-3)(t-4)(t-5)} - 1 \text{ (} \leq 0 \text{).}
\]

We have:

\[
\text{Var}\left[\xi^{(t)}\tau^{(t)}\right] = |\Delta^{(t)}| f(t) + r^{(t)} g(t) + w^{(t)} h(t).
\] \hspace{1cm} \text{(1)}

In our proofs, we carefully account for the fact that, as we use reservoir sampling [42], the presence of an edge \( a \) in \( S \) is not independent from the concurrent presence of another edge \( b \) in \( S \). This is not the case for samples built using fixed-probability independent edge sampling, such

\(^2\)Two distinct triangles can share at most one edge.
as **mascot** [28]. When computing the variance, we must consider not only pairs of triangles that share an edge, as in the case for independent edge sampling approaches, but also pairs of triangles sharing no edge, since their respective presences in the sample are not independent events. The gain is worth the additional sophistication needed in the analysis, as the contribution to the variance by triangles no sharing edges is *non-positive* \((h(t) \leq 0)\), i.e., it reduces the variance. A comparison of the variance of our estimator with that obtained with a fixed-probability independent edge sampling approach, is discussed in Sect. 4.1.6.

**Proof of Thm. 4.4.** Assume \(|\Delta(t)| > 0\), otherwise the estimation is deterministically correct and has variance 0 and the thesis holds. Let \(\lambda \in \Delta(t)\) and \(\delta^{(t)}_\lambda\) be as in the proof of Thm. 4.2. We have \(\text{Var}[\delta^{(t)}_\lambda] = \xi^{(t)} - 1\) and from this and the definition of variance and covariance we obtain

\[
\text{Var}\left[\xi^{(t)} r^{(t)}\right] = \text{Var}\left[\sum_{\lambda \in \Delta(t)} \delta^{(t)}_\lambda\right] = \sum_{\lambda \in \Delta(t)} \sum_{\gamma \in \Delta(t)} \text{Cov}\left[\delta^{(t)}_\lambda, \delta^{(t)}_\gamma\right] = \sum_{\lambda \in \Delta(t)} \text{Var}\left[\delta^{(t)}_\lambda\right] + \sum_{\lambda, \gamma \in \Delta(t)} \text{Cov}\left[\delta^{(t)}_\lambda, \delta^{(t)}_\gamma\right] = |\Delta(t)| (\xi^{(t)} - 1) + \sum_{\lambda, \gamma \in \Delta(t)} \left(\mathbb{E}\left[\delta^{(t)}_\lambda \delta^{(t)}_\gamma\right] - \mathbb{E}\left[\delta^{(t)}_\lambda\right] \mathbb{E}\left[\delta^{(t)}_\gamma\right]\right) = |\Delta(t)| (\xi^{(t)} - 1) + \sum_{\lambda, \gamma \in \Delta(t), \lambda \neq \gamma} \left(\mathbb{E}\left[\delta^{(t)}_\lambda \delta^{(t)}_\gamma\right] - \mathbb{E}\left[\delta^{(t)}_\lambda\right] - \mathbb{E}\left[\delta^{(t)}_\gamma\right]\right).
\]

Assume now \(|\Delta(t)| \geq 2\), otherwise we have \(r^{(t)} = w^{(t)} = 0\) and the thesis holds as the second term on the r.h.s. of (2) is 0. Let \(\lambda\) and \(\gamma\) be two distinct triangles in \(\Delta(t)\). If \(\lambda\) and \(\gamma\) do not share an edge, we have \(\delta^{(t)}_\lambda \delta^{(t)}_\gamma = \xi^{(t)} \bar{\xi}^{(t)} = \xi^2\), if all six edges composing \(\lambda\) and \(\gamma\) are in \(S\) at the end of time step \(t\), and \(\delta^{(t)}_\lambda \delta^{(t)}_\gamma = 0\) otherwise. From Lemma 4.1 we then have that

\[
\mathbb{E}\left[\delta^{(t)}_\lambda \delta^{(t)}_\gamma\right] = \xi^2 \Pr\left(\delta^{(t)}_\lambda \delta^{(t)}_\gamma = \xi^2\right) = \xi^2 \frac{1}{\bar{\xi}} = \xi \prod_{j=3}^{5} \frac{M - j}{t - j} = \xi (M - 3)(M - 4)(M - 5)(t - 3)(t - 4)(t - 5).
\]

If instead \(\lambda\) and \(\gamma\) share exactly an edge we have \(\delta^{(t)}_\lambda \delta^{(t)}_\gamma = \xi^2\), if all five edges composing \(\lambda\) and \(\gamma\) are in \(S\) at the end of time step \(t\), and \(\delta^{(t)}_\lambda \delta^{(t)}_\gamma = 0\) otherwise. From Lemma 4.1 we then have that

\[
\mathbb{E}\left[\delta^{(t)}_\lambda \delta^{(t)}_\gamma\right] = \xi^2 \Pr\left(\delta^{(t)}_\lambda \delta^{(t)}_\gamma = \xi^2\right) = \xi^2 \frac{1}{\bar{\xi}} = \xi \prod_{j=3}^{4} \frac{M - j}{t - j} = \xi (M - 3)(M - 4)(t - 3)(t - 4).
\]
The thesis follows by combining (2), (3), (4), recalling the definitions of \( r^{(t)} \) and \( w^{(t)} \), and slightly reorganizing the terms.

4.1.5 Concentration. We have the following concentration result on the estimation returned by \textsc{triest}

\[ \Phi = \sqrt{8e^{-2} \frac{3h^{(t)} + 1}{|\Delta^{(t)}|} \ln \left( \frac{(3h^{(t)} + 1)e}{\delta} \right)}. \]

If
\[ M \geq \max \left\{ t\Phi \left( 1 + \frac{1}{2} \ln^{2/3} (t\Phi) \right), 12e^{-1} + e^2, 25 \right\}, \]
then \( |\hat{\xi}^{(t)} - |\Delta^{(t)}|| < e|\Delta^{(t)}| \) with probability \( > 1 - \delta \).

The roadmap to proving Thm. 4.5 is the following:

1. we first define two simpler algorithms, named \textsc{indep} and \textsc{mix}. The algorithms use, respectively, fixed-probability independent sampling of edges and reservoir sampling (but with a different estimator than the one used by \textsc{triest-base});
2. we then prove concentration results on the estimators of \textsc{indep} and \textsc{mix}. Specifically, the concentration result for \textsc{indep} uses a result by Hajnal and Szemerédi [17] on graph coloring, while the one for \textsc{mix} will depend on the concentration result for \textsc{indep} and on a Poisson-approximation-like technical result stating that probabilities of events when using reservoir sampling are close to the probabilities of those events when using fixed-probability independent sampling;
3. we then show that the estimates returned by \textsc{triest-base} are close to the estimates returned by \textsc{mix};
4. finally, we combine the above results and show that, if \( M \) is large enough, then the estimation provided by \textsc{mix} is likely to be close to \( |\Delta^{(t)}| \) and since the estimation computed by \textsc{triest-base} is close to that of \textsc{mix}, then it must also be close to \( |\Delta^{(t)}| \).

Note: for ease of presentation, in the following we use \( \phi^{(t)} \) to denote the estimation returned by \textsc{triest-base}, i.e., \( \phi^{(t)} = \hat{\xi}^{(t)}r^{(t)} \).

The \textsc{indep} algorithm. The \textsc{indep} algorithm works as follows: it creates a sample \( S_{\text{IN}} \) by sampling edges in \( E^{(t)} \) independently with a fixed probability \( p \). It estimates the global number of triangles in \( G^{(t)} \) as
\[ \phi_{\text{IN}}^{(t)} = \frac{\tau_{\text{IN}}^{(t)}}{p^3}, \]
where \( \tau_{\text{IN}}^{(t)} \) is the number of triangles in \( S_{\text{IN}} \). This is for example the approach taken by \textsc{mascot-c} [28].

The \textsc{mix} algorithm. The \textsc{mix} algorithm works as follows: it uses reservoir sampling (like \textsc{triest-base}) to create a sample \( S_{\text{MIX}} \) of \( M \) edges from \( E^{(t)} \), but uses a different estimator than the one used by \textsc{triest-base}. Specifically, \textsc{mix} uses
\[ \phi_{\text{MIX}}^{(t)} = \left( \frac{t}{M} \right)^3 r^{(t)} \]

\(^3\)If \( |\Delta^{(t)}| = 0 \), our algorithms correctly estimate 0 triangles.
as an estimator for $|\Delta(t)|$, where $\tau(t)$ is, as in TRIEST-BASE, the number of triangles in $G_S$ (TRIEST-BASE uses $\phi(t) = \frac{t(t-1)(t-2)}{3(M-1)(M-2)} \tau(t)$ as an estimator.)

We call this algorithm MIX because it uses reservoir sampling to create the sample, but computes the estimate as if it used fixed-probability independent sampling, hence in some sense it “mixes” the two approaches.

Concentration results for INDEP and MIX. We now show a concentration result for INDEP. Then we show a technical lemma (Lemma 4.7) relating the probabilities of events when using reservoir sampling to the probabilities of those events when using fixed-probability independent sampling. Finally, we use these results to show that the estimator used by MIX is also concentrated (Lemma 4.9).

**Lemma 4.6.** Let $t \geq 0$ and assume $|\Delta(t)| > 0.4$ For any $\epsilon, \delta \in (0, 1)$, if

$$p \geq 2^{-\epsilon^2} \ln \left( \frac{3h(t) + 1}{\delta} \right) \frac{3h(t) + 1}{|\Delta(t)|}$$

then

$$\Pr\left( |\phi_{in}^{(t)} - \Delta(t)| < \epsilon |\Delta(t)| \right) > 1 - \delta .$$

**Proof.** Let $H$ be a graph built as follows: $H$ has one node for each triangle in $G(t)$ and there is an edge between two nodes in $H$ if the corresponding triangles in $G(t)$ share an edge. By this construction, the maximum degree in $H$ is $3h(t)$. Hence by the Hajnal-Szemerédi’s theorem [17] there is a proper coloring of $H$ with at most $3h(t) + 1$ colors such that for each color there are at least $L = \frac{|\Delta(t)|}{3h(t) + 1}$ nodes with that color.

Assign an arbitrary numbering to the triangles of $G(t)$ (and, therefore, to the nodes of $H$) and let $X_i$ be a Bernoulli random variable, indicating whether the triangle $i$ in $G(t)$ is in the sample at time $t$. From the properties of independent sampling of edges we have $\Pr(X_i = 1) = p^3$ for any triangle $i$. For any color $c$ of the coloring of $H$, let $X_c$ be the set of r.v.’s $X_i$ such that the node $i$ in $H$ has color $c$. Since the coloring of $H$ which we are considering is proper, the r.v.’s in $X_c$ are independent, as they correspond to triangles which do not share any edge and edges are sampled independent of each other. Let $Y_c$ be the sum of the r.v.’s in $X_c$. The r.v. $Y_c$ has a binomial distribution with parameters $|X_c|$ and $p^3$. By the Chernoff bound for binomial r.v.’s, we have that

$$\Pr \left( |p^{-3}Y_c - |X_c|| > \epsilon |X_c| \right) < 2 \exp \left( -\epsilon^2 p^3 |X_c| / 2 \right)$$

$$< 2 \exp \left( -\epsilon^2 p^3 L / 2 \right)$$

$$\leq \frac{\delta}{3h(t) + 1} ,$$

where the last step comes from the requirement in (5). Then by applying the union bound over all the (at most) $3h(t) + 1$ colors we get

$$\Pr(\exists \text{ color } c \text{ s.t. } |p^{-3}Y_c - |X_c|| > \epsilon |X_c|) < \delta .$$

---

1.10 Lorenzo De Stefani, Alessandro Epasto, Matteo Riondato, and Eli Upfal

ACM Transactions on Knowledge Discovery from Data, Vol. 1, No. 1, Article 1. Publication date: January 2017.

For $|\Delta(t)| = 0$, INDEP correctly and deterministically returns 0 as the estimation.
Since $\phi_{IN}(t) = p^{-3} \sum_{c} Y_c$, from the above equation we have that, with probability at least $1 - \delta$,

$$|\phi_{IN}^{(t)} - |\Delta^{(t)}|| \leq \left| \sum_{c} p^{-3} Y_c - \sum_{c} |X_c| \right| \leq \sum_{c} |p^{-3} Y_c - |X_c|| \leq \sum_{c} \epsilon |X_c| \leq \epsilon |\Delta^{(t)}| .$$

□

The above result is of independent interest and can be used, for example, to give concentration bounds to the estimation computed by MASCOT-\textsc{c} [28].

We remark that we can not use the same approach from Lemma 4.6 to show a concentration result for TRIEST-\textsc{base} because it uses reservoir sampling, hence the event of having a triangle $a$ in $S$ and the event of having another triangle $b$ in $S$ are not independent.

We can however show the following general result, similar in spirit to the well-know Poisson approximation of balls-and-bins processes [31]. Its proof can be found in App. A.1.

Fix the parameter $M$ and a time $t > M$. Let $S_{\text{mix}}$ be a sample of $M$ edges from $E^{(t)}$ obtained through reservoir sampling (as mix would do), and let $S_{\text{in}}$ be a sample of the edges in $E^{(t)}$ obtained by sampling edges independently with probability $M/t$ (as INDEP would do). We remark that the size of $S_{\text{in}}$ is in $[0, t]$ but not necessarily $M$.

**Lemma 4.7.** Let $f : 2^{E^{(t)}} \to \{0, 1\}$ be an arbitrary binary function from the powerset of $E^{(t)}$ to $\{0, 1\}$. We have

$$\Pr(f(S_{\text{mix}}) = 1) \leq e\sqrt{M} \Pr(f(S_{\text{in}}) = 1) .$$

We now use the above two lemmas to show that the estimator $\phi_{\text{mix}}^{(t)}$ computed by mix is concentrated. We will first need the following technical fact.

**Fact 4.8.** For any $x \geq 5$, we have

$$\ln \left( x(1 + \ln^{2/3} x) \right) \leq \ln^2 x .$$

**Lemma 4.9.** Let $t \geq 0$ and assume $|\Delta^{(t)}| < 0$. For any $\epsilon, \delta \in (0, 1)$, let

$$\Psi = 2\epsilon^2 3h(t) + 1 \ln \left( e \frac{3h(t) + 1}{\delta} \right) .$$

If

$$M \geq \max \left\{ t\sqrt{\Psi} \left( 1 + \frac{1}{2} \ln^{2/3} \left( t\sqrt{\Psi} \right) \right), 25 \right\}$$

then

$$\Pr \left( |\phi_{\text{mix}}^{(t)} - |\Delta^{(t)}|| < \epsilon |\Delta^{(t)}| \right) \geq 1 - \delta .$$

**Proof.** For any $S \subseteq E^{(t)}$ let $\tau(S)$ be the number of triangles in $S$, i.e., the number of triplets of edges in $S$ that compose a triangle in $G^{(t)}$. Define the function $g : 2^{E^{(t)}} \to \mathbb{R}$ as

$$g(S) = \left( \frac{t}{M} \right)^3 \tau(S) .$$

Assume that we run INDEP with $p = M/t$, and let $S_{\text{in}} \subseteq E^{(t)}$ be the sample built by INDEP (through independent sampling with fixed probability $p$). Assume also that we run MIX with parameter $M$,
and let $S_{\text{mix}}$ be the sample built by $\text{mix}$ (through reservoir sampling with a reservoir of size $M$). We have that $\phi^{(t)}_{\text{in}} = g(S_{\text{in}})$ and $\phi^{(t)}_{\text{mix}} = g(S_{\text{mix}})$. Define now the binary function $f : 2^{E^{(t)}} \rightarrow \{0, 1\}$ as

$$f(S) = \begin{cases} 1 & \text{if } |g(S) - |\Delta^{(t)}|| > \epsilon|\Delta^{(t)}| \\ 0 & \text{otherwise} \end{cases}.$$  

We now show that, for $M$ as in the hypothesis, we have

$$p \geq \sqrt{2e^{-2} \frac{3h^{(t)} + 1}{|\Delta^{(t)}|}} \ln \left(\sqrt{M} \frac{3h^{(t)} + 1}{\delta} \right).$$  

(6)

Assume for now that the above is true. From this, using Lemma 4.6 and the above fact about $g$ we get that

$$\Pr \left( |\phi^{(t)}_{\text{in}} - |\Delta^{(t)}|| > \epsilon|\Delta^{(t)}| \right) = \Pr (f(S_{\text{in}}) = 1) < \frac{\delta}{e \sqrt{M}}.$$  

From this and Lemma 4.7, we get that

$$\Pr (f(S_{\text{mix}}) = 1) \leq \delta$$

which, from the definition of $f$ and the properties of $g$, is equivalent to

$$\Pr \left( |\phi^{(t)}_{\text{mix}} - |\Delta^{(t)}|| > \epsilon|\Delta^{(t)}| \right) \leq \delta$$

and the proof is complete. All that is left is to show that (6) holds for $M$ as in the hypothesis.

Since $p = M/t$, we have that (6) holds for

$$M^3 \geq t^32e^{-2} \frac{3h^{(t)} + 1}{|\Delta^{(t)}|} \ln \left(\sqrt{M} \frac{3h^{(t)} + 1}{\delta} \right)$$

$$= t^32e^{-2} \frac{3h^{(t)} + 1}{|\Delta^{(t)}|} \left( \ln \left( \frac{3h^{(t)} + 1}{\delta} \right) + \frac{1}{2} \ln M \right).$$  

(7)

We now show that (7) holds.

Let $A = t \sqrt{\Psi}$ and let $B = t \sqrt{\Psi} \ln^{2/3} \left(t \sqrt{\Psi}\right)$. We now show that $A^3 + B^3$ is greater or equal to the r.h.s. of (7), hence $M^3 = (A + B)^3 > A^3 + B^3$ must also be greater or equal to the r.h.s. of (7), i.e., (7) holds. This really reduces to show that

$$B^3 \geq t^32e^{-2} \frac{3h^{(t)} + 1}{|\Delta^{(t)}|} \frac{1}{2} \ln M$$

as the r.h.s. of (7) can be written as

$$A^3 + t^32e^{-2} \frac{3h^{(t)} + 1}{|\Delta^{(t)}|} \frac{1}{2} \ln M.$$  

We actually show that

$$B^3 \geq t^3\frac{1}{2} \ln M$$

(9)

which implies (8) which, as discussed, in turn implies (7). Consider the ratio

$$\frac{B^3}{t^3\Psi \frac{1}{2} \ln M} \geq \frac{1}{2} \ln \left( t \sqrt{\Psi} \left( 1 + \ln^{2/3} \left(t \sqrt{\Psi}\right) \right) \right).$$  

(10)
We now show that \( t \sqrt{\Psi} \geq 5 \). By the assumptions \( t > M \geq 25 \) and by

\[
t \sqrt{\Psi} \geq \frac{t}{\sqrt{|\Delta(t)|}} \geq \sqrt{t}
\]

which holds because \( |\Delta(t)| \leq t^{3/2} \) (in a graph with \( t \) edges there can not be more than \( t^{3/2} \) triangles) we have that \( t \sqrt{\Psi} \geq 5 \). Hence Fact 4.8 holds and we can write, from (10):

\[
\frac{\ln^2(t \sqrt{\Psi})}{\ln \left( t \sqrt{\Psi} \left( 1 + \ln^{2/3} \left( t \sqrt{\Psi} \right) \right) \right)} \geq \frac{\ln^2(t \sqrt{\Psi})}{\ln \left( t \sqrt{\Psi} \right)} \geq 1,
\]

which proves (9), and in cascade (8), (7), (6), and the thesis. \( \square \)

**Relationship between TRIÊST-BASE and MIX.** When both TRIÊST-BASE and MIX use a sample of size \( M \), their respective estimators \( \phi(t) \) and \( \phi_{\text{mix}}(t) \) are related as discussed in the following result, whose straightforward proof is deferred to App. A.1.

**Lemma 4.10.** For any \( t > M \) we have

\[
\left| \phi(t) - \phi_{\text{mix}}(t) \right| \leq \phi_{\text{mix}}(t) \frac{4}{M - 2}.
\]

**Tying everything together.** Finally we can use the previous lemmas to prove our concentration result for TRIÊST-BASE.

**Proof of Thm. 4.5.** For \( M \) as in the hypothesis we have, from Lemma 4.9, that

\[
\Pr \left( \phi_{\text{mix}}(t) \leq (1 + \epsilon/2)|\Delta(t)| \right) \geq 1 - \delta.
\]

Suppose the event \( \phi_{\text{mix}}(t) \leq (1 + \epsilon/2)|\Delta(t)| \) (i.e., \( \phi_{\text{mix}}(t) - |\Delta(t)| \leq \epsilon|\Delta(t)|/2 \)) is indeed verified. From this and Lemma 4.10 we have

\[
|\phi(t) - \phi_{\text{mix}}(t)| \leq \left( 1 + \frac{\epsilon}{2} \right) |\Delta(t)| \frac{4}{M - 2} \leq |\Delta(t)| \frac{6}{M - 2},
\]

where the last inequality follows from the fact that \( \epsilon < 1 \). Hence, given that \( M \geq 12 \epsilon^{-1} + \epsilon^2 \geq 12 \epsilon^{-1} + 2 \), we have

\[
|\phi(t) - \phi_{\text{mix}}(t)| \leq |\Delta(t)| \frac{\epsilon}{2}.
\]

Using the above, we can then write:

\[
|\phi(t) - |\Delta(t)|| = |\phi(t) - \phi_{\text{mix}}(t) + \phi_{\text{mix}}(t) - |\Delta(t)||
\]

\[
\leq |\phi(t) - \phi_{\text{mix}}(t)| + |\phi_{\text{mix}}(t) - |\Delta(t)||
\]

\[
\leq \frac{\epsilon}{2} |\Delta(t)| + \frac{\epsilon}{2} |\Delta(t)| = \epsilon |\Delta(t)|
\]

which completes the proof. \( \square \)

**4.1.6 Comparison with fixed-probability approaches.** We now compare the variance of TRIÊST-BASE to the variance of the fixed probability sampling approach MASCOT-C [28], which samples edges independently with a fixed probability \( p \) and uses \( p^{-3}|\Delta_S| \) as the estimate for the global number of triangles at time \( t \). As shown by Lim and Kang [28, Lemma 2], the variance of this estimator is

\[
\text{Var}[p^{-3}|\Delta_S|] = |\Delta(t)| f(p) + r(t) g(p),
\]

where \( f(p) = p^{-3} - 1 \) and \( g(p) = p^{-1} - 1 \).
Assume that we give MASCOT-C the additional information that the stream has finite length $T$, and assume we run MASCOT-C with $p = M/T$ so that the expected sample size at the end of the stream is $M$.\footnote{We are giving MASCOT-C a significant advantage: if only space $M$ were available, we should run MASCOT-C with a sufficiently smaller $p' < p$, otherwise there would be a constant probability that MASCOT-C would run out of memory.} Let $\mathcal{V}_{\text{fix}}^{(t)}$ be the resulting variance of the MASCOT-C estimator at time $t$, and let $\mathcal{V}^{(t)}$ be the variance of our estimator at time $t$ (see (1)). For $t \leq M$, $\mathcal{V}^{(t)} = 0$, hence $\mathcal{V}^{(t)} \leq \mathcal{V}_{\text{fix}}^{(t)}$.

For $M < t < T$, we can show the following result. Its proof is more tedious than interesting so we defer it to App. A.1.

**Lemma 4.11.** Let $0 < \alpha < 1$ be a constant. For any constant $M > \max(\frac{8a}{1-\alpha}, 42)$ and any $t \leq \alpha T$ we have $\mathcal{V}^{(t)} < \mathcal{V}_{\text{fix}}^{(t)}$.

For example, if we set $\alpha = 0.99$ and run TRIEST-BASE with $M \geq 400$ and MASCOT-C with $p = M/T$, we have that TRIEST-BASE has strictly smaller variance than MASCOT-C for 99% of the stream.

What about $t = T$? The difference between the definitions of $\mathcal{V}_{\text{fix}}^{(t)}$ and $\mathcal{V}^{(t)}$ is in the presence of $\bar{f}(M/T)$ instead of $f(t)$ (resp. $\bar{g}(M/T)$ instead of $g(t)$) as well as the additional term $w^{(t)}h(M, t) \leq 0$ in our $\mathcal{V}^{(t)}$. Let $M(T)$ be an arbitrary slowly increasing function of $T$. For $T \to \infty$ we can show that $\lim_{T \to \infty} \frac{\bar{f}(M(T)/T)}{f(T)} = \lim_{T \to \infty} \frac{\bar{g}(M(T)/T)}{g(T)} = 1$, hence, informally, $\mathcal{V}^{(T)} \to \mathcal{V}_{\text{fix}}^{(T)}$, for $T \to \infty$.

A similar discussion also holds for the method we present in Sect. 4.2, and explains the results of our experimental evaluations, which shows that our algorithms have strictly lower (empirical) variance than fixed probability approaches for most of the stream.

### 4.1.7 Update time

The time to process an element of the stream is dominated by the computation of the shared neighborhood $N_{u,v}$ in UPDATECOUNTERS. A MERGESORT-based algorithm for the intersection requires $O(\deg(u) + \deg(v))$ time, where the degrees are w.r.t. the graph $G_S$. By storing the neighborhood of each vertex in a Hash Table (resp. an AVL tree), the update time can be reduced to $O(\min\{\deg(u), \deg(v)\})$ (resp. amortized time $O(\min\{\deg(v), \deg(u)\} + \log \max\{\deg(v), \deg(u)\})$).

### 4.2 Improved insertion algorithm – TRIEST-IMPR

TRIEST-IMPR is a variant of TRIEST-BASE with small modifications that result in higher-quality (i.e., lower variance) estimations. The changes are:

1. **UPDATECOUNTERS** is called *unconditionally for each element on the stream*, before the algorithm decides whether or not to insert the edge into $S$. W.r.t. the pseudocode in Alg. 1, this change corresponds to moving the call to UPDATECOUNTERS on line 6 to before the **if** block. MASCOT [28] uses a similar idea, but TRIEST-IMPR is significantly different as TRIEST-IMPR allows edges to be removed from the sample, since it uses reservoir sampling.

2. TRIEST-IMPR never decrements the counters when an edge is removed from $S$. W.r.t. the pseudocode in Alg. 1, we remove the call to UPDATECOUNTERS on line 13.

3. UPDATECOUNTERS performs a weighted increase of the counters using $\eta^{(t)} = \max\{1, (t - 1)(t - 2)/(M(M - 1))\}$ as weight. W.r.t. the pseudocode in Alg. 1, we replace “1” with $\eta^{(t)}$ on lines 19–22 (given change 2 above, all the calls to UPDATECOUNTERS have $\bullet = +$).

The resulting pseudocode for TRIEST-IMPR is presented in Alg. 2.

**Counters.** If we are interested only in estimating the global number of triangles in $G^{(t)}$, TRIEST-IMPR needs to maintain only the counter $\tau$ and the edge sample $S$ of size $M$, so it still requires space $O(M)$. If instead we are interested in estimating the local triangle counts, at any time $t$ TRIEST-IMPR maintains (non-zero) local counters only for the nodes $u$ such that at least one triangle with a corner
ALGORITHM 2 TRIÊST-IMPR

**Input:** Insertion-only edge stream \( \Sigma \), integer \( M \geq 6 \)

1: \( S \leftarrow \emptyset, t \leftarrow 0, \tau \leftarrow 0 \)
2: for each element \( (+, (u, v)) \) from \( \Sigma \) do
3: \( t \leftarrow t + 1 \)
4: UpdateCounters(\( u, v \))
5: if SampleEdge((\( u, v \), \( t \))) then
6: \( S \leftarrow S \cup \{(u, v)\} \)
7: function SampleEdge((\( u, v \), \( t \))
8: if \( t \leq M \) then
9: \( \text{return True} \)
10: else if FlipBiasedCoin(\( \frac{M}{2} \)) = heads then
11: \( (u', v') \leftarrow \text{random edge from } S \)
12: \( S \leftarrow S \setminus \{(u', v')\} \)
13: \( \text{return True} \)
14: return False
15: function UpdateCounters(\( u, v \))
16: \( N_{u,v}^S \leftarrow N_u^S \cap N_v^S \)
17: \( \eta = \max\{1, (t-1)(t-2)/(M(M-1))\} \)
18: for all \( c \in N_{u,v}^S \) do
19: \( \tau_c \leftarrow \tau_c + \eta \)
20: \( \tau_u \leftarrow \tau_u + \eta \)
21: \( \tau_v \leftarrow \tau_v + \eta \)

\( u \) has been detected by the algorithm up until time \( t \). The number of such nodes may be greater than \( O(M) \), but this is the price to pay to obtain estimations with lower variance (Thm. 4.13).

4.2.1 Estimation. When queried for an estimation, TRIÊST-IMPR returns the value of the corresponding counter, unmodified.

4.2.2 Analysis. We now present the analysis of the estimations computed by TRIÊST-IMPR, showing results involving their unbiasedness, their variance, and their concentration around their expectation. Results analogous to those in Thms. 4.12, 4.13, and 4.15 hold for the local triangle count for any \( u \in V(t) \), replacing the global quantities with the corresponding local ones.

4.2.3 Expectation. As in TRIÊST-BASE, the estimations by TRIÊST-IMPR are exact at time \( t \leq M \) and unbiased for \( t > M \). The proof of the following theorem follows the same steps as the one for Thm 4.2, and can be found in App. A.2.

**Theorem 4.12.** We have \( \tau(\tau) = |\Delta(t)| \) if \( t \leq M \) and \( E[\tau(t)] = |\Delta(t)| \) if \( t > M \).

4.2.4 Variance. We now show an upper bound to the variance of the TRIÊST-IMPR estimations for \( t > M \). The proof relies on a very careful analysis of the covariance of two triangles which depends on the order of arrival of the edges in the stream (which we assume to be adversarial). For any \( \lambda \in \Delta(t) \) we denote as \( t_\lambda \) the time at which the last edge of \( \lambda \) is observed on the stream. Let \( z(t) \) be the number of unordered pairs \( (\lambda, \gamma) \) of distinct triangles in \( G(t) \) that share an edge \( g \) and are such that:

1. \( g \) is neither the last edge of \( \lambda \) nor \( \gamma \) on the stream; and
\[ \min \{ t_\lambda, t_y \} > M + 1. \]

**Theorem 4.13.** Then, for any time \( t > M \), we have
\[ \text{Var} \left[ \tau^{(t)} \right] \leq |\Delta^{(t)}| (\eta^{(t)} - 1) + z^{(t)} \frac{t - 1 - M}{M}. \]  \( (11) \)

The bound to the variance presented in (11) is extremely pessimistic and loose. Specifically, it does not contain the negative contribution to the variance given by the \( (|\Delta^{(t)}| - z^{(t)}) \) triangles that do not satisfy the requirements in the definition of \( z^{(t)} \). Among these pairs there are, for example, all pairs of triangles not sharing any edge, but also many pairs of triangles that share an edge, as the definition of \( z^{(t)} \) consider only a subset of these. All these pairs would give a negative contribution to the variance, i.e., decrease the r.h.s. of (11), whose more correct form would be
\[ |\Delta^{(t)}| (\eta^{(t)} - 1) + z^{(t)} \frac{t - 1 - M}{M} + \left( \frac{|\Delta^{(t)}|}{2} - z^{(t)} \right) \omega_{M,t} \]
where \( \omega_{M,t} \) is (an upper bound to) the minimum negative contribution of a pair of triangles that do not satisfy the requirements in the definition of \( z^{(t)} \). Sadly, computing informative upper bounds to \( \omega_{M,t} \) is not straightforward, even in the restricted setting where only pairs of triangles not sharing any edge are considered.

To prove Thm. 4.13 we first need Lemma 4.14, whose proof is deferred to App. A.2.

For any time step \( t \) and any edge \( e \in E^{(t)} \), we denote with \( t_e \) the time step at which \( e \) is on the stream. For any \( \lambda \in \Delta^{(t)} \), let \( \lambda = (\ell_1, \ell_2, \ell_3) \), where the edges are numbered in order of appearance on the stream. We define the event \( D_\lambda \) as the event that \( \ell_1 \) and \( \ell_2 \) are both in the edge sample \( S \) at the end of time step \( t_\lambda - 1 \).

**Lemma 4.14.** Let \( \lambda = (\ell_1, \ell_2, \ell_3) \) and \( \gamma = (g_1, g_2, g_3) \) be two disjoint triangles, where the edges are numbered in order of appearance on the stream, and assume, w.l.o.g., that the last edge of \( \lambda \) is on the stream before the last edge of \( \gamma \). Then
\[ \Pr(D_\gamma \mid D_\lambda) \leq \Pr(D_\gamma). \]

We can now prove Thm. 4.13.

**Proof of Thm. 4.13.** Assume \( |\Delta^{(t)}| > 0 \), otherwise \textsc{triest-impr} estimation is deterministically correct and has variance 0 and the thesis holds. Let \( \lambda \in \Delta^{(t)} \) and let \( \delta_\lambda \) be a random variable that takes value \( \xi_{2,t-1} \) if both \( \ell_1 \) and \( \ell_2 \) are in \( S \) at the end of time step \( t_\lambda - 1 \), and 0 otherwise. Since
\[ \text{Var} \left[ \delta_\lambda \right] = \xi_{2,t-1} - 1 \leq \xi_{2,t-1}, \]
we have:
\[ \text{Var} \left[ \tau^{(t)} \right] = \text{Var} \left[ \sum_{\lambda \in \Delta^{(t)}} \delta_\lambda \right] = \sum_{\lambda \in \Delta^{(t)}} \text{Var} \left[ \delta_\lambda \right] + \sum_{\lambda, \gamma \in \Delta^{(t)}} \text{Cov} \left[ \delta_\lambda, \delta_\gamma \right] \]
\[ \leq |\Delta^{(t)}| (\xi_{2,t-1} - 1) + \sum_{\lambda, \gamma \in \Delta^{(t)}} \left( \mathbb{E}[\delta_\lambda \delta_\gamma] - \mathbb{E}[\delta_\lambda] \mathbb{E}[\delta_\gamma] \right) \]
\[ \leq |\Delta^{(t)}| (\xi_{2,t-1} - 1) + \sum_{\lambda, \gamma \in \Delta^{(t)}} \left( \mathbb{E}[\delta_\lambda \delta_\gamma] - 1 \right). \]
\( (12) \)
For any $\lambda \in \Delta^{(t)}$ define $q_{\lambda} = \xi_{2, t_{\lambda} - 1}$. Assume now $|\Delta^{(t)}| \geq 2$, otherwise we have $r^{(t)} = w^{(t)} = 0$ and the thesis holds as the second term on the r.h.s. of (12) is 0. Let now $\lambda$ and $\gamma$ be two distinct triangles in $\Delta^{(t)}$ (hence $t \geq 5$). We have

$$\mathbb{E}[\delta_1 \delta_\gamma] = q_{\lambda} q_{\gamma} \Pr(\delta_1 \delta_\gamma = q_{\lambda} q_{\gamma})$$

The event \( \delta_1 \delta_\gamma = q_{\lambda} q_{\gamma} \) is the intersection of events $D_\lambda \cap D_\gamma$, where $D_\lambda$ is the event that the first two edges of $\lambda$ are in $S$ at the end of time step $t_\lambda - 1$, and similarly for $D_\gamma$. We now look at $\Pr(D_\lambda \cap D_\gamma)$ in the various possible cases.

Assume that $\lambda$ and $\gamma$ do not share any edge, and, w.l.o.g., that the third (and last) edge of $\lambda$ appears on the stream before the third (and last) edge of $\gamma$, i.e., $t_\lambda < t_\gamma$. From Lemma 4.14 and Lemma 4.1 we then have

$$\Pr(D_\lambda \cap D_\gamma) = \Pr(D_\gamma | D_\lambda) \Pr(D_\lambda) \leq \Pr(D_\gamma) \Pr(D_\lambda) \leq \frac{1}{q_{\lambda} q_{\gamma}}.$$ 

Consider now the case where $\lambda$ and $\gamma$ share an edge, $g$. W.l.o.g., let us assume that $t_\lambda \leq t_\gamma$ (since the shared edge may be the last on the stream both for $\lambda$ and for $\gamma$, we may have $t_\lambda = t_\gamma$). There are the following possible sub-cases:

- **$g$ is the last on the stream among all the edges of $\lambda$ and $\gamma$** In this case we have $t_\lambda = t_\gamma$.
  The event \( D_\lambda \cap D_\gamma \) happens if and only if the four edges that, together with $g$, compose $\lambda$ and $\gamma$ are all in $S$ at the end of time step $t_\lambda - 1$. Then, from Lemma 4.1 we have
  $$\Pr(D_\lambda \cap D_\gamma) = \frac{1}{\xi_{4, t_{\lambda} - 1}} = \frac{1}{q_{\lambda}} \frac{M - 2}{(t_\lambda - 3)(t_\lambda - 4)} \leq \frac{1}{q_{\lambda}} \frac{M - 1}{(t_\lambda - 1)(t_\lambda - 2)} \leq \frac{1}{q_{\lambda} q_{\gamma}}.$$  

- **$g$ is the last on the stream among all the edges of $\lambda$ and the first among all the edges of $\gamma$**
  In this case, we have that $D_\lambda$ and $D_\gamma$ are independent. Indeed the fact that the first two edges of $\lambda$ (neither of which is $g$) are in $S$ when $g$ arrives on the stream has no influence on the probability that $g$ and the second edge of $\gamma$ are inserted in $S$ and are not evicted until the third edge of $\gamma$ is on the stream. Hence we have
  $$\Pr(D_\lambda \cap D_\gamma) = \Pr(D_\lambda) \Pr(D_\gamma) = \frac{1}{q_{\lambda} q_{\gamma}}.$$ 

- **$g$ is the last on the stream among all the edges of $\lambda$ and the second among all the edges of $\gamma$**
  In this case we can follow an approach similar to the one in the proof for Lemma 4.14 and have that
  $$\Pr(D_\lambda \cap D_\gamma) \leq \Pr(D_\gamma) \Pr(D_\lambda) \leq \frac{1}{q_{\lambda} q_{\gamma}}.$$ 

The intuition behind this is that if the first two edges of $\lambda$ are in $S$ when $g$ is on the stream, their presence lowers the probability that the first edge of $\gamma$ is in $S$ at the same time, and hence that the first edge of $\gamma$ and $g$ are in $S$ when the last edge of $\gamma$ is on the stream.

- **$g$ is not the last on the stream among all the edges of $\lambda$**
  There are two situations to consider, or better, one situation and all other possibilities. The situation we consider is that
  (1) $g$ is the first edge of $\gamma$ on the stream; and
  (2) the second edge of $\gamma$ to be on the stream is on the stream at time $t_2 > t_\lambda$.
  Suppose this is the case. Recall that if $D_\lambda$ is verified, then we know that $g$ is in $S$ at the beginning of time step $t_\lambda$. Define the following events:
  - $E_1$: “the set of edges evicted from $S$ between the beginning of time step $t_\lambda$ and the beginning of time step $t_2$ does not contain $g$.”
  - $E_2$: “the second edge of $\gamma$, which is on the stream at time $t_2$, is inserted in $S$ and the edge that is evicted is not $g$."

ACM Transactions on Knowledge Discovery from Data, Vol. 1, No. 1, Article 1. Publication date: January 2017.
• \( E_3 \): “the set of edges evicted from \( S \) between the beginning of time step \( t_2 + 1 \) and the beginning of time step \( t_y \) does not contain either \( g \) or the second edge of \( y \).”

We can then write

\[
\Pr(D_y \mid D_\lambda) = \Pr(E_1 \mid D_\lambda) \Pr(E_2 \mid E_1 \cap D_\lambda) \Pr(E_3 \mid E_2 \cap E_1 \cap D_\lambda).
\]

We now compute the probabilities on the r.h.s., where we denote with \( \mathbb{1}_{t_2 > M(1)} \) the function that has value 1 if \( t_2 > M \), and value 0 otherwise:

\[
\Pr(E_1 \mid D_\lambda) = \prod_{j = \max\{t_\lambda, M+1\}}^{t_2-1} \left( 1 - \frac{M}{j} \right) + \frac{M}{j} \left( \frac{M - 1}{M} \right)
\]

\[
= \prod_{j = \max\{t_\lambda, M+1\}}^{t_2-1} \frac{j - 1}{j} = \max\{t_\lambda - 1, M\} \over \max\{M, t_2 - 1\};
\]

\[
\Pr(E_2 \mid E_1 \cap D_\lambda) = \frac{M}{\max\{t_2, M\}} \frac{M - \mathbb{1}_{t_2 > M(1)} M}{M} = \frac{M - \mathbb{1}_{t_2 > M(1)} M}{\max\{t_2, M\}};
\]

\[
\Pr(E_3 \mid E_2 \cap E_1 \cap D_\lambda) = \prod_{j = \max\{t_2+1, M+1\}}^{t_y-1} \left( 1 - \frac{M}{j} \right) + \frac{M}{j} \left( \frac{M - 2}{M} \right)
\]

\[
= \prod_{j = \max\{t_2+1, M+1\}}^{t_y-1} \frac{j - 2}{j} = \frac{\max\{t_2, M\} \max\{t_2 - 1, M - 1\}}{\max\{t_y - 2, M - 1\} \max\{t_y - 1, M\}}.
\]

Hence, we have

\[
\Pr(D_y \mid D_\lambda) = \frac{\max\{t_\lambda - 1, M\}(M - \mathbb{1}_{t_2 > M(1)}) \max\{t_2 - 1, M - 1\}}{\max\{M, t_2 - 1\} \max\{(t_y - 2)(t_y - 1), M(M - 1)\}}.
\]

With a (somewhat tedious) case analysis we can verify that

\[
\Pr(D_y \mid D_\lambda) \leq \frac{1}{q_y} \max\{M, t_\lambda - 1\} \over M.
\]

Consider now the complement of the situation we just analyzed. In this case, two edges of \( y \), that is, \( g \) and another edge \( h \) are on the stream before time \( t_\lambda \), in some non-relevant order (i.e., \( g \) could be the first or the second edge of \( y \) on the stream). Define now the following events:

• \( E_1 \): “\( h \) and \( g \) are both in \( S \) at the beginning of time step \( t_\lambda \).”

• \( E_2 \): “the set of edges evicted from \( S \) between the beginning of time step \( t_\lambda \) and the beginning of time step \( t_y \) does not contain either \( g \) or \( h \).”

We can then write

\[
\Pr(D_y \mid D_\lambda) = \Pr(E_1 \mid D_\lambda) \Pr(E_2 \mid E_1 \cap D_\lambda).
\]

If \( t_\lambda \leq M + 1 \), we have that \( \Pr(E_1 \mid D_\lambda) = 1 \). Consider instead the case \( t_\lambda > M + 1 \). If \( D_\lambda \) is verified, then both \( g \) and the other edge of \( \lambda \) are in \( S \) at the beginning of time step \( t_\lambda \). At this time, all subsets of \( E^{(\lambda, -1)} \) of size \( M \) and containing both \( g \) and the other edge of \( \lambda \) have an equal probability of being \( S \), from Lemma A.1. There are \( \binom{t_\lambda - 3}{M - 3} \) such sets. Among these, there are \( \binom{t_\lambda - 4}{M - 3} \) sets that also contain \( h \). Therefore, if \( t_\lambda > M + 1 \), we have

\[
\Pr(E_1 \mid D_\lambda) = \frac{\binom{t_\lambda - 4}{M - 3}}{\binom{t_\lambda - 3}{M - 3}} = \frac{M - 2}{t_\lambda - 3}.
\]
Considering what we said before for the case $t_{\lambda} \leq M + 1$, we then have
\[
\Pr(E_1 \mid D_{\lambda}) = \min \left\{ 1, \frac{M - 2}{t_{\lambda} - 3} \right\}.
\]
We also have
\[
\Pr(E_2 \mid E_1 \cap D_{\lambda}) = \prod_{j=\max\{t_{\lambda},M+1\}}^{t_{\lambda}-1} \frac{j - 2}{j} = \max\{((t_{\lambda} - 2)(t_{\lambda} - 1), M(M - 1)) \mid (t_{\lambda} - 2)(t_{\lambda} - 1), M(M - 1) \}.
\]
Therefore,
\[
\Pr(D_\gamma \mid D_{\lambda}) = \min \left\{ 1, \frac{M - 2}{t_{\lambda} - 3} \right\} \max\{((t_{\lambda} - 2)(t_{\lambda} - 1), M(M - 1)) \mid (t_{\lambda} - 2)(t_{\lambda} - 1), M(M - 1) \}.
\]
With a case analysis, one can show that
\[
\Pr(D_\gamma \mid D_{\lambda}) \leq \frac{1}{q_\gamma} \max\{M, t_{\lambda} - 1\}.
\]

To recap we have the following two scenarios when considering two distinct triangles $\gamma$ and $\lambda$:

1. if $\lambda$ and $\gamma$ share an edge and, assuming w.l.o.g. that the third edge of $\lambda$ is on the stream no later than the third edge of $\gamma$, and the shared edge is neither the last among all edges of $\lambda$ to appear on the stream nor the last among all edges of $\gamma$ to appear on the stream, then we have
\[
\text{Cov}[\delta_{\lambda}, \delta_{\gamma}] \leq \mathbb{E}[\delta_{\lambda} \delta_{\gamma}] - 1 = q_{\lambda} q_\gamma \Pr(\delta_{\lambda} \delta_{\gamma} = q_{\lambda} q_\gamma) - 1
\]
\[
\leq q_{\lambda} q_\gamma \frac{1}{q_{\lambda} q_\gamma} \max\{M, t_{\lambda} - 1\} - 1 \leq \frac{\max\{M, t_{\lambda} - 1\}}{M} - 1 \leq \frac{t - 1 - M}{M};
\]
where the last inequality follows from the fact that $t_{\lambda} \leq t$ and $t - 1 \geq M$.

For the pairs $(\lambda, \gamma)$ such that $t_{\lambda} \leq M + 1$, we have $\max\{M, t_{\lambda} - 1\}/M = 1$ and therefore $\text{Cov}[\delta_{\lambda}, \delta_{\gamma}] \leq 0$. We should therefore only consider the pairs for which $t_{\lambda} > M + 1$. Their number is given by $z(t)$.

2. in all other cases, including when $\gamma$ and $\lambda$ do not share an edge, we have $\mathbb{E}[\delta_{\lambda} \delta_{\gamma}] \leq 1$, and since $\mathbb{E}[\delta_{\lambda}] \mathbb{E}[\delta_{\gamma}] = 1$, we have
\[
\text{Cov}[\delta_{\lambda}, \delta_{\gamma}] \leq 0.
\]
Hence, we can bound
\[
\sum_{\lambda, \gamma \in \Delta(t)} \text{Cov}[\delta_{\lambda}, \delta_{\gamma}] \leq z(t) \frac{t - 1 - M}{M}
\]
and the thesis follows by combining this into (12).

4.2.5 Concentration. We now show a concentration result on the estimation of TRIED-IMPR, which relies on Chebyshev’s inequality [31, Thm. 3.6] and Thm. 4.13.

**Theorem 4.15.** Let $t \geq 0$ and assume $|\Delta(t)| > 0$. For any $\epsilon, \delta \in (0, 1)$, if
\[
M > \max \left\{ \sqrt{\frac{2(t - 1)(t - 2)}{\delta \epsilon^2 |\Delta(t)|} + \frac{1}{4} + \frac{1}{\delta \epsilon^2 |\Delta(t)|^2} + \frac{2z(t)(t - 1)}{\delta \epsilon^2 |\Delta(t)|}} \right\}
\]
then $|\tau(t) - |\Delta(t)|| < \epsilon |\Delta(t)|$ with probability $> 1 - \delta$. 

ACM Transactions on Knowledge Discovery from Data, Vol. 1, No. 1, Article 1. Publication date: January 2017.
PROOF. By Chebyshev’s inequality it is sufficient to prove that
\[
\frac{\text{Var}[\tau^{(t)}]}{\epsilon^2|\Delta^{(t)}|^2} < \delta.
\]
We can write
\[
\frac{\text{Var}[\tau^{(t)}]}{\epsilon^2|\Delta^{(t)}|^2} \leq \frac{1}{\epsilon^2|\Delta^{(t)}|^2} \left((\eta(t) - 1) + z^{(t)} \frac{t - 1 - M}{M|\Delta^{(t)}|}\right).
\]
Hence it is sufficient to impose the following two conditions:

**Condition 1**
\[
\frac{\delta}{2} > \frac{\eta(t) - 1}{\epsilon^2|\Delta^{(t)}|} > \frac{1}{\epsilon^2|\Delta^{(t)}|} \frac{(t - 1)(t - 2) - M(M - 1)}{M(M - 1)},
\]
which is verified for:
\[
M(M - 1) > \frac{2(t - 1)(t - 2)}{\delta\epsilon^2|\Delta^{(t)}|} + 2.
\]
As \( t > M \), we have \( \frac{2(t - 1)(t - 2)}{\delta\epsilon^2|\Delta^{(t)}|} + 2 > 0 \). The condition in (13) is thus verified for:
\[
M > \frac{1}{2} \left(\frac{2(t - 1)(t - 2)}{\delta\epsilon^2|\Delta^{(t)}|} + 1 + 1\right).
\]

**Condition 2**
\[
\frac{\delta}{2} > \frac{z^{(t)}(t - 1 - M)}{M\epsilon^2|\Delta^{(t)}|^2},
\]
which is verified for:
\[
M > \frac{2z^{(t)}(t - 1)}{\delta\epsilon^2|\Delta^{(t)}|^2 + 2z^{(t)}}.
\]
The theorem follows. \( \square \)

In Thms. 4.13 and 4.15, it is possible to replace the value \( z^{(t)} \) with the more interpretable \( r^{(t)} \), which is agnostic to the order of the edges on the stream but gives a looser upper bound to the variance.

### 4.3 Fully-dynamic algorithm – TRIEST-FD

TRIEST-FD computes unbiased estimates of the global and local triangle counts in a fully-dynamic stream where edges are inserted/deleted in any arbitrary, adversarial order. It is based on random pairing (RP) [16], a sampling scheme that extends reservoir sampling and can handle deletions. The idea behind the RP scheme is that edge deletions seen on the stream will be “compensated” by future edge insertions. Following RP, TRIEST-FD keeps a counter \( d_i \) (resp. \( d_o \)) to keep track of the number of uncompensated edge deletions involving an edge \( e \) that was (resp. was not) in \( S \) at the time the deletion for \( e \) was on the stream.

When an edge deletion for an edge \( e \in E^{(t-1)} \) is on the stream at the beginning of time step \( t \), then, if \( e \in S \) at this time, TRIEST-FD removes \( e \) from \( S \) (effectively decreasing the number of edges stored in the sample by one) and increases \( d_i \) by one. Otherwise, it just increases \( d_o \) by one. When an edge insertion for an edge \( e \not\in E^{(t-1)} \) is on the stream at the beginning of time step \( t \), if \( d_i + d_o = 0 \), then TRIEST-FD follows the standard reservoir sampling scheme. If \( |S| < M \), then \( e \) is deterministically inserted in \( S \) without removing any edge from \( S \) already in \( S \), otherwise it is
ALGORITHM 3 TRIĘST-FD

Input: Fully-dynamic edge stream \( \Sigma \), integer \( M \geq 6 \)
1: \( S \leftarrow \emptyset \), \( d_i \leftarrow 0 \), \( d_o \leftarrow 0 \), \( t \leftarrow 0 \), \( s \leftarrow 0 \)
2: for each element \((\bullet, (u, v))\) from \( \Sigma \) do
3: \( t \leftarrow t + 1 \)
4: \( s \leftarrow s \cdot 1 \)
5: if \( \bullet = + \) then
6: if SAMPLEEdge \((u, v)\) then
7: \( \text{UpdateCounters}(+, (u, v)) \) \> UpdateCounters is defined as in Alg. 1.
8: else if \((u, v) \in S\) then
9: \( \text{UpdateCounters}(-, (u, v)) \)
10: \( S \leftarrow S \setminus \{(u, v)\} \)
11: \( d_i \leftarrow d_i + 1 \)
12: else \( d_o \leftarrow d_o + 1 \)

13: function SAMPLEEdge\((u, v)\)
14: if \( d_o + d_i = 0 \) then
15: if \(|S| < M\) then
16: \( S \leftarrow S \cup \{(u, v)\} \)
17: return True
18: else if \( \text{FlipBiasedCoin}(\frac{M}{t}) = \text{heads} \) then
19: Select \((z, w)\) uniformly at random from \( S \)
20: \( \text{UpdateCounters}(-, (z, w)) \)
21: \( S \leftarrow (S \setminus \{(z, w)\}) \cup \{(u, v)\} \)
22: return True
23: else if \( \text{FlipBiasedCoin}(\frac{d_i}{d_i + d_o}) = \text{heads} \) then
24: \( S \leftarrow S \cup \{(u, v)\} \)
25: \( d_i \leftarrow d_i - 1 \)
26: return True
27: else
28: \( d_o \leftarrow d_o - 1 \)
29: return False

inserted in \( S \) with probability \( M/t \), replacing an uniformly-chosen edge already in \( S \). If instead \( d_i + d_o > 0 \), then \( e \) is inserted in \( S \) with probability \( d_i/(d_i + d_o) \); since it must be \( d_i > 0 \), then it must be \( |S| < M \) and no edge already in \( S \) needs to be removed. In any case, after having handled the eventual insertion of \( e \) into \( S \), the algorithm decreases \( d_i \) by 1 if \( e \) was inserted in \( S \), otherwise it decreases \( d_o \) by 1. TRIĘST-FD also keeps track of \( s^{(t)} = |E^{(t)}| \) by appropriately incrementing or decrementing a counter by 1 depending on whether the element on the stream is an edge insertion or deletion. The pseudocode for TRIĘST-FD is presented in Alg. 3 where the UpdateCounters procedure is the one from Alg. 1.

4.3.1 Estimation. We denote as \( M^{(t)} \) the size of \( S \) at the end of time \( t \) (we always have \( M^{(t)} \leq M \)). For any time \( t \), let \( d_i^{(t)} \) and \( d_o^{(t)} \) be the value of the counters \( d_i \) and \( d_o \) at the end of time \( t \) respectively, and let \( \omega^{(t)} = \min\{M, s^{(t)} + d_i^{(t)} + d_o^{(t)}\} \). Define

\[
\kappa^{(t)} = 1 - \sum_{j=0}^{2} \binom{s^{(t)}}{j} \binom{d_i^{(t)} + d_o^{(t)}}{\omega^{(t)} - j} \binom{s^{(t)} + d_i^{(t)} + d_o^{(t)}}{\omega^{(t)}}. \tag{14}
\]
For any three positive integers \(a, b, c\) s.t. \(a \leq b \leq c\), define\(^6\)
\[
\psi_{a,b,c} = \frac{c}{b} / \frac{c-a}{b-a} = \prod_{i=0}^{a-1} \frac{c-i}{b-i}.
\]

When queried at the end of time \(t\), for an estimation of the global number of triangles, TRIÈST-FD returns
\[
\rho^{(t)} = \begin{cases} 0 & \text{if } M^{(t)} < 3 \\ \frac{s^{(t)}}{\psi_{3,M^{(t)},s^{(t)}}} & \text{otherwise} \end{cases}
\]

TRIÈST-FD can keep track of \(\kappa^{(t)}\) during the execution, each update of \(\kappa^{(t)}\) taking time \(O(1)\). Hence the time to return the estimations is still \(O(1)\).

4.3.2 Analysis. We now present the analysis of the estimations computed by TRIÈST-IMPR, showing results involving their unbiasedness, their variance, and their concentration around their expectation. Results analogous to those in Thms. 4.16, 4.17, and 4.18 hold for the local triangle count for any \(u \in V^{(t)}\), replacing the global quantities with the corresponding local ones.

4.3.3 Expectation. Let \(t^*\) be the first \(t \geq M + 1\) such that \(|E^{(t)}| = M + 1\), if such a time step exists (otherwise \(t^* = +\infty\)).

**Theorem 4.16.** We have \(\rho^{(t)} = |\Delta^{(t)}|\) for all \(t < t^*\), and \(E[\rho^{(t)}] = |\Delta^{(t)}|\) for \(t \geq t^*\).

The proof, deferred to App. A.3, relies on properties of RP and on the definitions of \(\kappa^{(t)}\) and \(\rho^{(t)}\). Specifically, it uses Lemma A.6, which is the correspondent of Lemma 4.1 but for RP, and some additional technical lemmas (including an equivalent of Lemma 4.3 but for RP) and combine them using the law of total expectation by conditioning on the value of \(M^{(t)}\).

4.3.4 Variance.

**Theorem 4.17.** Let \(t > t^*\) s.t. \(|\Delta^{(t)}| > 0\) and \(s^{(t)} \geq M\). Suppose we have \(d^{(t)} = d^{(t)}_o + d^{(t)}_i \leq \alpha s^{(t)}\) total unpaired deletions at time \(t\), with \(0 \leq \alpha < 1\). If \(M \geq \frac{1}{2 \sqrt{\alpha - \alpha}} 7 \ln s^{(t)}\) for some \(\alpha < \alpha' < 1\), we have:
\[
\text{Var} \left[ \rho^{(t)} \right] \leq (\kappa^{(t)})^{-2} |\Delta^{(t)}| \left( \psi_{3,M(1-\alpha'),s^{(t)}} - 1 \right) + (\kappa^{(t)})^{-2} 2
\]
\[
\left( \psi_{3,M(1-\alpha'),s^{(t)}}^{-1} \psi_{\psi_{5,M(1-\alpha'),s^{(t)}}^{-1} \psi_{5,M(1-\alpha'),s^{(t)}}} - 1 \right)
\]

The proof of Thm. 4.17 is deferred to App. A.3. It uses two results on the variance of \(\rho^{(t)}\) conditioned on a specific value of \(M^{(t)}\) (Lemmas A.9 and A.10), and an analysis of the probability distribution of \(M^{(t)}\) (Lemma A.11 and Corollary A.12). These results are then combined using the law of total variance.

4.3.5 Concentration. The following result relies on Chebyshev’s inequality and on Thm. 4.17, and the proof (reported in App. A.3) follows the steps similar to those in the proof for Thm. 4.13.

\(^6\)We follow the convention that \(\binom{0}{0} = 1\).
Theorem 4.18. Let $t \geq t^*$ s.t. $|\Delta(t)| > 0$ and $s(t) \geq M$. Let $d(t) = d_o(t) + d_i(t) \leq \alpha s(t)$ for some $0 \leq \alpha < 1$. For any $\varepsilon, \delta \in (0, 1)$, if for some $\alpha < \alpha' < 1$

$$M > \max \left\{ \frac{1}{\sqrt{a' - \alpha}} \ln s(t), \right\}$$

$$\left(1 - \alpha'\right)^{-1} \left( \frac{2s(t)(s(t) - 1)(s(t) - 2)}{\delta \varepsilon^2 |\Delta(t)| ((\kappa(t))^2 + 2 |\Delta(t)|^2}) + 2 \right),$$

$$\left(1 - \alpha'\right)^{-1} \left( \frac{r(t)s(t)}{\delta \varepsilon^2 |\Delta(t)|^2 (\kappa(t))^2 + 2 r(t)} \right)$$

then $|\rho(t) - |\Delta(t)|| < \varepsilon |\Delta(t)|$ with probability $> 1 - \delta$.

4.4 Counting global and local triangles in multigraphs

We now discuss how to extend TRIÈST to approximate the local and global triangle counts in multigraphs.

4.4.1 TRIÈST-BASE on multigraphs. TRIÈST-BASE can be adapted to work on multigraphs as follows. First of all, the sample $S$ should be considered a bag, i.e., it may contain multiple copies of the same edge. Secondly, the function UPDATECOUNTERS must be changed as presented in Alg. 4, to take into account the fact that inserting or removing an edge $(u, v)$ from $S$ respectively increases or decreases the global number of triangles in $G^S$ by a quantity that depends on the product of the number of edges $(c, u) \in S$ and $(c, v) \in S$, for $c$ in the shared neighborhood (in $G^S$) of $u$ and $v$ (and similarly for the local number of triangles incident to $c$).

Algorithm 4 UPDATECOUNTERS function for TRIÈST-BASE on multigraphs

1: function UPDATECOUNTERS((•, (u, v)))
2: $N_{u,v}^S \leftarrow N_u^S \cap N_v^S$
3: for all $c \in N_{u,v}^S$ do
4: $y_{c,u} \leftarrow$ number of edges between $c$ and $u$ in $S$
5: $y_{c,v} \leftarrow$ number of edges between $c$ and $v$ in $S$
6: $y_c \leftarrow y_{c,u} \cdot y_{c,v}$
7: $r \leftarrow r \cdot y_c$
8: $r_{c} \leftarrow r_{c} \cdot y_{c}$
9: $r_{u} \leftarrow r_{u} \cdot y_{c}$
10: $r_{v} \leftarrow r_{v} \cdot y_{c}$

For this modified version of TRIÈST-BASE, that we call TRIÈST-BASE-M, an equivalent version of Lemma 4.3 holds. Therefore, we can prove a result on the unbiasedness of TRIÈST-BASE-M equivalent (i.e., with the same statement) as Thm. 4.2. The proof of such result is also the same as the one for Thm. 4.2.

To analyze the variance of TRIÈST-BASE-M, we need to take into consideration the fact that, in a multigraph, a pair of triangles may share two edges, and the variance depends (also) on the number of such pairs. Let $r_1(t)$ be the number of unordered pairs of distinct triangles from $\Delta(t)$ sharing an edge and let $r_2(t)$ be the number of unordered pairs of distinct triangles from $\Delta(t)$ sharing two edges.
Additionally, the function $q$ (such pairs may exist in a multigraph, but not in a simple graph). Let $q^{(t)} = \frac{\lfloor \Delta^{(t)} \rfloor}{2} - r_1^{(t)} - r_2^{(t)}$ be the number of unordered pairs of distinct triangles that do not share any edge.

**Theorem 4.19.** For any $t > M$, let $f(t) = \xi^{(t)} - 1$,

$$g(t) = \xi^{(t)} \frac{(M-3)(M-4)}{(t-3)(t-4)} - 1$$

and

$$h(t) = \xi^{(t)} \frac{(M-3)(M-4)(M-5)}{(t-3)(t-4)(t-5)} - 1 \ (\leq 0),$$

and

$$j(t) = \xi^{(t)} \frac{M-3}{t-3} - 1 .$$

We have:

$$\text{Var} \left[ \xi(t) r_2^{(t)} \right] = |\Delta^{(t)}| f(t) + r_1^{(t)} g(t) + r_2^{(t)} j(t) + q^{(t)} h(t).$$

The proof follows the same lines as the one for Thm. 4.4, with the additional steps needed to take into account the contribution of the $r_2^{(t)}$ pairs of triangles in $G^{(t)}$ sharing two edges.

### 4.4.2 TRIEST-IMPR on multigraphs.

A variant TRIEST-IMPR-M of TRIEST-IMPR for multigraphs can be obtained by using the function UpdateCounters defined in Alg. 4, modified to increment\(^7\) the counters by $\eta^{(t)} g_c^{(t)}$, rather than $y_c^{(t)}$, where $\eta^{(t)} = \max\{1, (t-1)(t-2)/(M(M-1))\}$. The result stated in Thm. 4.12 holds also for the estimations computed by TRIEST-IMPR-M. An upper bound to the variance of the estimations, similar to the one presented in Thm. 4.13 for TRIEST-IMPR, could potentially be obtained, but its derivation would involve a high number of special cases, as we have to take into consideration the order of the edges in the stream.

### 4.4.3 TRIEST-FD on multigraphs.

TRIEST-FD can be modified in order to provide an approximation of the number of global and local triangles on multigraphs observed as a stream of edge deletions and deletions. It is however necessary to clearly state the data model. We assume that for all pairs of vertices $u, v \in V^{(t)}$, each edge connecting $u$ and $v$ is assigned a label that is unique among the edges connecting $u$ and $v$. An edge is therefore uniquely identified by its endpoints and its label as $((u, v), label)$. Elements of the stream are now in the form $(\bullet, (u, v), label)$, where $\bullet$ is either $+$ or $\cdot$. This assumption, somewhat strong, is necessary in order to apply the random pairing sampling scheme [16] to fully-dynamic multigraph edge streams.

Within this model, we can obtain an algorithm TRIEST-FD-M for multigraphs by adapting TRIEST-FD as follows. The sample $S$ is a set of elements $((u, v), label)$. When a deletion $(-, (u, v), label)$ is on the stream, the sample $S$ is modified if and only if $((u, v), label)$ belongs to $S$. This change can be implemented in the pseudocode from Alg. 3 by modifying line 8 to be

```
else if ((u, v), label) ∈ S then ”.
```

Additionally, the function UpdateCounters to be used is the one presented in Alg. 4.

We can prove a result on the unbiasedness of TRIEST-FD-M equivalent (i.e., with the same statement) as Thm. 4.16. The proof of such result is also the same as the one for Thm. 4.16. An upper bound to the variance of the estimations, similar to the one presented in Thm. 4.17 for TRIEST-FD, could be obtained by considering the fact that in a multigraph two triangles can share two edges, in a fashion similar to what we discussed in Thm. 4.19.

---

\(^7\)As in TRIEST-IMPR, all calls to UpdateCounters in TRIEST-IMPR-M have $\bullet = +$. See also Alg. 2.
4.5 Discussion

We now briefly discuss over the algorithms we just presented, the techniques they use, and the theoretical results we obtained for TRIÊST, in order to highlight advantages, disadvantages, and limitations of our approach.

**On reservoir sampling.** Our approach of using reservoir sampling to keep a random sample of edges can be extended to many other graph mining problems, including approximate counting of other subgraphs more or less complex than triangles (e.g., squares, trees with a specific structure, wedges, cliques, and so on). The estimations of such counts would still be unbiased, but as the number of edges composing the subgraph(s) of interest increases, the variance of the estimators also increases, because the probability that all edges composing a subgraph are in the sample (or all but the last one when the last one arrives, as in the case of TRIÊST-IMPR), decreases as their number increases. Other works in the triangle counting literature [20, 36] use samples of wedges, rather than edges. They perform worse than TRIÊST in both accuracy and runtime (see Sect. 5), but the idea of sampling and storing more complex structures rather than simple edges could be a potential direction for approximate counting of larger subgraphs.

**On the analysis of the variance.** We showed an exact analysis of the variance of TRIÊST-BASE but for the other algorithms we presented upper bounds to the variance of the estimates. These bounds can still be improved as they are not currently tight. For example, we already commented on the fact that the bound in (11) does not include a number of negative terms that would tighten it (i.e., decrease the bound), and that could potentially be no smaller than the term depending on $z(t)$. The absence of such terms is due to the fact that it seems very challenging to obtain non-trivial upper bounds to them that are valid for every $t > M$. Our proof for this bound uses a careful case-by-case analysis, considering the different situations for pair of triangles (e.g., sharing or not sharing an edge, and considering the order of edges on the stream). It may be possible to obtain tighter bounds to the variance by following a more holistic approach that takes into account the fact that the sizes of the different classes of triangle pairs are highly dependent on each other. Another issue with the bound to the variance from (11) is that the quantity $z(t)$ depends on the order of edges on the stream. As already discussed, the bound can be made independent of the order by loosening it even more. Very recent developments in the sampling theory literature [12] presented sampling schemes and estimators whose second-order sampling probabilities do not depend on the order of the stream, so it should be possible to obtain such bounds also for the triangle counting problem, but a sampling scheme different than reservoir sampling would have to be used, and a careful analysis is needed to establish its net advantages in terms of performances and scalability to billion-edges graphs.

**On the trade-off between speed and accuracy.** We concluded both previous paragraphs in this subsection by mentioning techniques different than reservoir sampling of edges as potential directions to improve and extend our results. In both cases these techniques are more complex not only in their analysis but also computationally. Given that the main goal of algorithms like TRIÊST is to make it possible to analyze graphs with billions (and possibly more) nodes, the gain in accuracy need to be weighted against expected slowdowns in execution. As we show in our experimental evaluation in the next section, TRIÊST, especially in the TRIÊST-IMPR variant, actually seems to strike the right balance between accuracy and tradeoff, when compared with existing contributions.
5 EXPERIMENTAL EVALUATION

We evaluated TRIÈST on several real-world graphs with up to a billion edges. The algorithms were implemented in C++, and ran on the Brown University CS department cluster. Each run employed a single core and used at most 4 GB of RAM. The code is available from http://bigdata.cs.brown.edu/triangles.html. Most of this section is related to experiments on graphs, while results for multigraphs are described in Sect 5.3.

Datasets. We created the streams from the following publicly available graphs (properties in Table 2).

- **Patent (Co-Aut.) and Patent (Cit.)** The Patent (Co-Aut.) and Patent (Cit.) graphs are obtained from a dataset of ≈ 2 million U.S. patents granted between ’75 and ’99 [18]. In Patent (Co-Aut.), the nodes represent inventors and there is an edge with timestamp t between two co-inventors of a patent if the patent was granted in year t. In Patent (Cit.), nodes are patents and there is an edge (a, b) with timestamp t if patent a cites b and a was granted in year t.

- **LastFm** The LastFm graph is based on a dataset [8, 39] of ≈ 20 million last.fm song listenings, ≈ 1 million songs and ≈ 1000 users. There is a node for each song and an edge between two songs if ≥ 3 users listened to both on day t.

- **Yahoo!-Answers** The Yahoo! Answers graph is obtained from a sample of ≈ 160 million answers to ≈ 25 millions questions posted on Yahoo! Answers [10]. An edge connects two users at time max(t1, t2) if they both answered the same question at times t1, t2 respectively. We removed 6 outliers questions with more than 5000 answers.

- **Twitter** This is a snapshot [5, 25] of the Twitter followers/following network with ≈ 41 million nodes and ≈ 1.5 billions edges. We do not have time information for the edges, hence we assign a random timestamp to the edges (of which we ignore the direction).

Ground truth. To evaluate the accuracy of our algorithms, we computed the ground truth for our smaller graphs (i.e., the exact number of global and local triangles for each time step), using an exact algorithm. The entire current graph is stored in memory and when an edge u, v is inserted (or deleted) we update the current count of local and global triangles by checking how many triangles are completed (or broken). As exact algorithms are not scalable, computing the exact triangle count is feasible only for small graphs such as Patent (Co-Aut.), Patent (Cit.) and LastFm. Table 2 reports the exact total number of triangles at the end of the stream for those graphs (and an estimate for the larger ones using TRIÈST-IMPR with $M = 1000000$).

5.1 Insertion-only case

We now evaluate TRIÈST on insertion-only streams and compare its performances with those of state-of-the-art approaches [20, 28, 36], showing that TRIÈST has an average estimation error significantly smaller than these methods both for the global and local estimation problems, while using the same amount of memory.

Estimation of the global number of triangles. Starting from an empty graph we add one edge at a time, in timestamp order. Figure 1 illustrates the evolution, over time, of the estimation computed by TRIÈST-IMPR with $M = 1,000,000$. For smaller graphs for which the ground truth can be computed exactly, the curve of the exact count is practically indistinguishable from TRIÈST-IMPR estimation, showing the precision of the method. The estimations have very small variance even on the very large Yahoo! Answers and Twitter graphs (point-wise max/min estimation over ten runs is almost low variance).
Table 2. Properties of the dynamic graph streams analyzed. \(|V|, |E|, |E_u|, |\Delta|\) refer respectively to the number of nodes in the graph, the number of edge addition events, the number of distinct edges additions, and the maximum number of triangles in the graph (for Yahoo! Answers and Twitter estimated with TRÍEST-IMPR with \(M = 1000000\), otherwise computed exactly with the naïve algorithm).

| Graph           | \(|V|\)   | \(|E|\)   | \(|E_u|\)  | \(|\Delta|\) |
|-----------------|-----------|-----------|------------|-------------|
| Patent (Co-Aut.)| 1,162,227 | 3,660,945 | 2,724,036  | 3.53 \times 10^6 |
| Patent (Cit.)   | 2,745,762 | 13,965,410| 13,965,132 | 6.91 \times 10^6 |
| LastFm          | 681,387   | 43,518,693| 30,311,117 | 1.13 \times 10^9 |
| Yahoo! Answers  | 2,432,573 | 1.21 \times 10^9 | 1.08 \times 10^9 | 7.86 \times 10^{10} |
| Twitter         | 41,652,230| 1.47 \times 10^9 | 1.20 \times 10^9 | 3.46 \times 10^{10} |

coincident with the average estimation). These results show that TRÍEST-IMPR is very accurate even when storing less than a 0.001 fraction of the total edges of the graph.

**Comparison with the state of the art.** We compare quantitatively with three state-of-the-art methods: mascoT [28], JHA ET AL. [20] and PAVAN ET AL. [36]. mascoT is a suite of local triangle counting methods (but provides also a global estimation). The other two are global triangle counting approaches. None of these can handle fully-dynamic streams, in contrast with TRÍEST-FD. We first compare the three methods to TRÍEST for the global triangle counting estimation. mascoT comes in two memory efficient variants: the basic mascoT-c variant and an improved mascoT-i variant.\(^9\) Both variants sample edges with fixed probability \(p\), so there is no guarantee on the amount of memory used during the execution. To ensure fairness of comparison, we devised the following experiment. First, we run both mascoT-c and mascoT-i for \(\ell = 10\) times with a fixed \(p\) using the same random bits for the two algorithms run-by-run (i.e. the same coin tosses used to select the edges) measuring each time the number of edges \(M_i\) stored in the sample at the end of the stream (by construction this the is same for the two variants run-by-run). Then, we run our algorithms using \(M = M_i\) (for \(i \in \{\ell\}\)). We do the same to fix the size of the edge memory for JHA ET AL. [20] and PAVAN ET AL. [36].\(^10\) This way, all algorithms use the same amount of memory for storing edges (run-by-run).

We use the MAPE (Mean Average Percentage Error) to assess the accuracy of the global triangle estimators over time. The MAPE measures the average percentage of the prediction error with respect to the ground truth, and is widely used in the prediction literature [19]. For \(t = 1, \ldots, T\), let \(\overline{\Delta}^{(t)}\) be the estimator of the number of triangles at time \(t\), the MAPE is defined as \(\frac{1}{T} \sum_{t=1}^{T} \left| \frac{\Delta^{(t)} - \overline{\Delta}^{(t)}}{\Delta^{(t)}} \right|\).\(^11\)

In Fig. 2(a), we compare the average MAPE of TRÍEST-BASE and TRÍEST-IMPR as well as the two mascoT variants and the other two streaming algorithms for the Patent (Co-Aut.) graph, fixing \(p = 0.01\). TRÍEST-IMPR has the smallest error of all the algorithms compared.

---

\(^9\)In the original work [28], this variant had no suffix and was simply called mascoT. We add the -i suffix to avoid confusion. Another variant mascoT-A can be forced to store the entire graph with probability 1 by appropriately selecting the edge order (which we assume to be adversarial) so we do not consider it here.

\(^10\)More precisely, we use \(M_i/2\) estimators in PAVAN ET AL. as each estimator stores two edges. For JHA ET AL. we set the two reservoirs in the algorithm to have each size \(M_i/2\). This way, all algorithms use \(M_i\) cells for storing (w)edges.

\(^11\)The MAPE is not defined for \(t\) s.t. \(\Delta^{(t)} = 0\) so we compute it only for \(t\) s.t. \(|\Delta^{(t)}| > 0\). All algorithms we consider are guaranteed to output the correct answer for \(t\) s.t. \(|\Delta^{(t)}| = 0\).
We now turn our attention to the efficiency of the methods. Whenever we refer to one operation, we mean handling one element on the stream, either one edge addition or one edge deletion. The average update time per operation is obtained by dividing the total time required to process the entire stream by the number of operations (i.e., elements on the streams).

Figure 2(b) shows the average update time per operation in Patent (Co-Aut.) graph, fixing $p = 0.01$. Both JHA ET AL. [20] and PAVAN ET AL. [36] are up to $\approx 3$ orders of magnitude slower than the Mascot variants and TRiest. This is expected as both algorithms have an update complexity of $\Omega(M)$ (they have to go through the entire reservoir graph at each step), while both Mascot algorithms and TRiest need only to access the neighborhood of the nodes involved in the edge addition.\footnote{We observe that PAVAN ET AL. [36] would be more efficient with batch updates. However, we want to estimate the triangles continuously at each update. In their experiments they use batch sizes of million of updates for efficiency.}

This allows both algorithms to efficiently exploit larger memory sizes. We can use efficiently $M$ up to 1 million edges in our experiments, which only requires few megabytes of...
RAM. MASCOT is one order of magnitude faster than TRIEST (which runs in \approx 28 \text{ micros/op}), because it does not have to handle edge removal from the sample, as it offers no guarantees on the used memory. As we will show, TRIEST has much higher precision and scales well on billion-edges graphs.

Table 3. Global triangle estimation MAPE for TRIEST and MASCOT. The rightmost column shows the reduction in terms of the avg. MAPE obtained by using TRIEST. Rows with Y in column “Impr.” refer to improved algorithms (TRIEST-IMPR and MASCOT-I) while those with N to basic algorithms (TRIEST-BASE and MASCOT-C).

<table>
<thead>
<tr>
<th>Graph</th>
<th>Impr.</th>
<th>p</th>
<th>Max. MAPE</th>
<th>Avg. MAPE</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MASCOT</td>
<td>TRIEST</td>
<td>MASCOT</td>
</tr>
<tr>
<td>Patent (Cit.)</td>
<td>N 0.01</td>
<td>0.01</td>
<td>0.9231</td>
<td>0.2583</td>
<td>0.6517</td>
</tr>
<tr>
<td></td>
<td>Y 0.01</td>
<td>0.01</td>
<td>0.1907</td>
<td>0.0363</td>
<td>0.1149</td>
</tr>
<tr>
<td></td>
<td>N 0.1</td>
<td>0.1</td>
<td>0.0839</td>
<td>0.0124</td>
<td>0.0605</td>
</tr>
<tr>
<td></td>
<td>Y 0.1</td>
<td>0.1</td>
<td>0.0317</td>
<td>0.0037</td>
<td>0.0245</td>
</tr>
<tr>
<td>Patent (Co-aut.)</td>
<td>N 0.01</td>
<td>0.01</td>
<td>2.3017</td>
<td>0.3029</td>
<td>0.8055</td>
</tr>
<tr>
<td></td>
<td>Y 0.01</td>
<td>0.01</td>
<td>0.1741</td>
<td>0.0261</td>
<td>0.1063</td>
</tr>
<tr>
<td></td>
<td>N 0.1</td>
<td>0.1</td>
<td>0.0648</td>
<td>0.0175</td>
<td>0.0390</td>
</tr>
<tr>
<td></td>
<td>Y 0.1</td>
<td>0.1</td>
<td>0.0225</td>
<td>0.0034</td>
<td>0.0174</td>
</tr>
<tr>
<td>LastFm</td>
<td>N 0.01</td>
<td>0.01</td>
<td>0.1525</td>
<td>0.0185</td>
<td>0.0627</td>
</tr>
<tr>
<td></td>
<td>Y 0.01</td>
<td>0.01</td>
<td>0.0273</td>
<td>0.0046</td>
<td>0.0141</td>
</tr>
<tr>
<td></td>
<td>N 0.1</td>
<td>0.1</td>
<td>0.0075</td>
<td>0.0028</td>
<td>0.0047</td>
</tr>
<tr>
<td></td>
<td>Y 0.1</td>
<td>0.1</td>
<td>0.0048</td>
<td>0.0013</td>
<td>0.0031</td>
</tr>
</tbody>
</table>

Given the slow execution of the other algorithms on the larger datasets we compare in details TRIEST only with MASCOT. Table 3 shows the average MAPE of the two approaches. The results confirm the pattern observed in Figure 2(a): TRIEST-BASE and TRIEST-IMPR both have an average error significantly smaller than that of the basic MASCOT-C and improved MASCOT variant respectively. We achieve up to a 91\% (i.e., 9-fold) reduction in the MAPE while using the same amount of memory. This experiment confirms the theory: reservoir sampling has overall lower or equal variance in all steps for the same expected total number of sampled edges.

To further validate this observation we run TRIEST-IMPR and the improved MASCOT-I variant using the same (expected memory) $M = 10000$. Figure 3 shows the max-min estimation over 10 runs and the standard deviation of the estimation over those runs. TRIEST-IMPR shows significantly lower standard deviation (hence variance) over the evolution of the stream, and the max and min lines are also closer to the ground truth. This confirms our theoretical observations in the previous sections. Even with very low $M$ (about 2/10000 of the size of the graph) TRIEST gives high-quality estimations.

Local triangle counting. We compare the precision in local triangle count estimation of TRIEST with that of MASCOT [28] using the same approach of the previous experiment. We can not compare with JHA ET AL. and PAVAN ET AL. algorithms as they provide only global estimation. As in [28], we

---

13The experiments by Jha et al. [20] use $M$ in the order of $10^3$, and in those by Pavan et al. [36], large $M$ values require large batches for efficiency.

14We attempted to run the other two algorithms but they did not complete after 12 hours for the larger datasets in Table 3 with the prescribed $p$ parameter setting.
Fig. 2. Average MAPE and average update time of the various methods on the Patent (Co-Aut.) graph with $p = 0.01$ (for MASCOT, see the main text for how we computed the space used by the other algorithms) – insertion only. TRIEST-IMPR has the lowest error. Both PAVAN ET AL. and JHA ET AL. have very high update times compared to our method and the two MASCOT variants.

Fig. 3. Accuracy and stability of the estimation of TRIEST-IMPR with $M = 10000$ and of MASCOT-I with same expected memory, on LastFM, over 10 runs. TRIEST-IMPR has a smaller standard deviation and moreover the max/min estimation lines are closer to the ground truth. Average estimations not shown as they are qualitatively similar.

Measure the Pearson coefficient and the average $\epsilon$ error (see [28] for definitions). In Table 4 we report the Pearson coefficient and average $\epsilon$ error over all timestamps for the smaller graphs.\textsuperscript{15} TRIEST (significantly) improves (i.e., has higher correlation and lower error) over the state-of-the-art MASCOT, using the same amount of memory.

Trade-offs between memory and accuracy. We study the trade-offs between the sample size $M$, the running time, and the accuracy of the estimators. Figure 4(a) shows the trade-offs between the accuracy of the estimation (as MAPE) and the size $M$ for the smaller graphs for which the ground

\textsuperscript{15}For efficiency, in this test we evaluate the local number of triangles of all nodes every 1000 edge updates.
Table 4. Comparison of the quality of the local triangle estimations between our algorithms and the state-of-the-art approach in [28]. Rows with Y in column “Impr.” refer to improved algorithms (TRIEST-IMPR and MASCOT-i) while those with N to basic algorithms (TRIEST-BASE and MASCOT-c). In virtually all cases we significantly outperform MASCOT using the same amount of memory.

<table>
<thead>
<tr>
<th>Graph</th>
<th>Impr.</th>
<th>$p$</th>
<th>Avg. Pearson</th>
<th>Avg. $\epsilon$ Err.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MASCOT</td>
<td>TRIEST</td>
</tr>
<tr>
<td>LastFm</td>
<td>0.1</td>
<td>0.09</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.97</td>
<td>1.00</td>
<td>+2.48%</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>0.85</td>
<td>0.98</td>
<td>+14.28%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.97</td>
<td>0.99</td>
<td>+2.04%</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.92</td>
<td>0.98</td>
<td>+6.61%</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>0.32</td>
<td>0.70</td>
<td>+117.74%</td>
</tr>
<tr>
<td>Patent (Cit.)</td>
<td>0.1</td>
<td>0.41</td>
<td>0.82</td>
<td>+99.09%</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.24</td>
<td>0.61</td>
<td>+156.30%</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>0.05</td>
<td>0.18</td>
<td>+233.05%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.16</td>
<td>0.48</td>
<td>+191.85%</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.06</td>
<td>0.24</td>
<td>+300.46%</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>0.00</td>
<td>0.003</td>
<td>+922.02%</td>
</tr>
<tr>
<td>Patent (Co-aut.)</td>
<td>0.1</td>
<td>0.55</td>
<td>0.87</td>
<td>+58.40%</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.34</td>
<td>0.71</td>
<td>+108.80%</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>0.08</td>
<td>0.26</td>
<td>+222.84%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.25</td>
<td>0.52</td>
<td>+112.40%</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.09</td>
<td>0.28</td>
<td>+204.98%</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>0.01</td>
<td>0.03</td>
<td>+191.46%</td>
</tr>
</tbody>
</table>

The truth number of triangles can be computed exactly using the naïve algorithm. Even with small $M$, TRIEST-IMPR achieves very low MAPE value. As expected, larger $M$ corresponds to higher accuracy and for the same $M$ TRIEST-IMPR outperforms TRIEST-BASE.

Figure 4(b) shows the average time per update in microseconds ($\mu$s) for TRIEST-IMPR as function of $M$. Some considerations on the running time are in order. First, a larger edge sample (larger $M$) generally requires longer average update times per operation. This is expected as a larger sample corresponds to a larger sample graph on which to count triangles. Second, on average a few hundreds microseconds are sufficient for handling any update even in very large graphs with billions of edges. Our algorithms can handle hundreds of thousands of edge updates (stream elements) per second, with very small error (Fig. 4(a)), and therefore TRIEST can be used efficiently and effectively in high-velocity contexts. The larger average time per update for Patent (Co-Auth.) can be explained by the fact that the graph is relatively dense and has a small size (compared to the larger Yahoo! and Twitter graphs). More precisely, the average time per update (for a fixed $M$) depends on two main factors: the average degree and the length of the stream. The denser the graph is, the higher the update time as more operations are needed to update the triangle count every time the sample is modified. On the other hand, the longer the stream, for a fixed $M$, the lower is the frequency of updates to the reservoir (it can be show that the expected number of updates to the reservoir is $O(M(1 + \log(\frac{t}{M})))$ which grows sub-linearly in the size of the stream $t$).
This explains why the average update time for the large and dense Yahoo! and Twitter graphs is so small, allowing the algorithm to scale to billions of updates.

Fig. 4. Trade-offs between $M$ and MAPE and average time per update in $\mu$s – edge insertion only. Larger $M$ implies lower errors but generally higher update times.

Alternative edge orders. In all previous experiments the edges are added in their natural order (i.e., in order of their appearance). While the natural order is the most important use case, we have assessed the impact of other ordering on the accuracy of the algorithms. We experiment with both the uniform-at-random (u.a.r.) order of the edges and the random BFS order: until all the graph is explored a BFS is started from a u.a.r. unvisited node and edges are added in order of their visit (neighbors are explored in u.a.r. order). The results for the random BFS order and u.a.r. order (Fig. 5) confirm that TRIEST has the lowest error and is very scalable in every tested ordering.

5.2 Fully-dynamic case

We evaluate TRIEST-FD on fully-dynamic streams. We cannot compare TRIEST-FD with the algorithms previously used [20, 28, 36] as they only handle insertion-only streams.

In the first set of experiments we model deletions using the widely used sliding window model, where a sliding window of the most recent edges defines the current graph. The sliding window model is of practical interest as it allows to observe recent trends in the stream. For Patent (Co-Aut.) & (Cit.) we keep in the sliding window the edges generated in the last 5 years, while for LastFm we keep the edges generated in the last 30 days. For Yahoo! Answers we keep the last 100 millions edges in the window.

Figure 6 shows the evolution of the global number of triangles in the sliding window model using TRIEST-FD using $M = 200,000$ ($M = 1,000,000$ for Yahoo! Answers). The sliding window scenario is significantly more challenging than the addition-only case (very often the entire sample of edges is flushed away) but TRIEST-FD maintains good variance and scalability even when, as for LastFm and Yahoo! Answers, the global number of triangles varies quickly.

16Excluding Twitter for which we used the random order, given the lack of timestamps.
17The sliding window model is not interesting for the Twitter dataset as edges have random timestamps. We omit the results for Twitter but TRIEST-FD is fast and has low variance.
Continuous monitoring of triangle counts with TRIEST-FD allows to detect patterns that would otherwise be difficult to notice. For LastFM (Fig. 6(c)) we observe a sudden spike of several order of magnitudes. The dataset is anonymized so we cannot establish which songs are responsible for this spike. In Yahoo! Answers (Fig. 6(d)) a popular topic can create a sudden (and shortly lived) increase in the number of triangles, while the evolution of the Patent co-authorship and co-citation networks is slower, as the creation of an edge requires filing a patent (Fig. 6(a) and (b)). The almost constant increase over time of the number of triangles in Patent graphs is consistent with previous observations of densification in collaboration networks as in the case of nodes’ degrees [27] and the observations on the density of the densest subgraph [15].

Table 5 shows the results for both the local and global triangle counting estimation provided by TRIEST-FD. In this case we can not compare with previous works, as they only handle insertions. It is evident that precision improves with M values, and even relatively small M values result in a low MAPE (global estimation), high Pearson correlation and low ε error (local estimation). Figure 7 shows the tradeoffs between memory (i.e., accuracy) and time. In all cases our algorithm is very fast and it presents update times in the order of hundreds of microseconds for datasets with billions of updates (Yahoo! Answers).

Alternative models for deletion. We evaluate TRIEST-FD using other models for deletions than the sliding window model. To assess the resilience of the algorithm to massive deletions we run the following experiments. We added edges in their natural order but each edge addition is followed with probability q by a mass deletion event where each edge currently in the the graph is deleted with probability d independently. We run experiments with $q = 3,000,000^{-1}$ (i.e., a mass deletion expected every 3 millions edges) and $d = 0.80$ (in expectation 80% of edges are deleted). The results are shown in Table 6.

We observe that TRIEST-FD maintains a good accuracy and scalability even in face of a massive (and unlikely) deletions of the vast majority of the edges: e.g., for LastFM with $M = 200000$ (resp. $M = 1,000,000$) we observe 0.04 (resp. 0.006) Avg. MAPE.

---

The decline at the end is due to the removal of the last edges from the sliding window after there are no more edge additions.
Fig. 6. Evolution of the global number of triangles in the fully-dynamic case (sliding window model for edge deletion). The curves are indistinguishable on purpose, to remark the fact that \textsc{triest-fd} estimations are extremely accurate and consistent. We comment on the observed patterns in the text.

Table 5. Estimation errors for \textsc{triest-fd}.

<table>
<thead>
<tr>
<th>Graph</th>
<th>Avg. Global</th>
<th>MAPE</th>
<th>Pearson</th>
<th>(\epsilon) Err.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastFM</td>
<td>200000</td>
<td>0.005</td>
<td>0.980</td>
<td>0.020</td>
</tr>
<tr>
<td></td>
<td>1000000</td>
<td>0.002</td>
<td>0.999</td>
<td>0.001</td>
</tr>
<tr>
<td>Patent (Co-Aut.)</td>
<td>200000</td>
<td>0.010</td>
<td>0.660</td>
<td>0.300</td>
</tr>
<tr>
<td></td>
<td>1000000</td>
<td>0.001</td>
<td>0.990</td>
<td>0.006</td>
</tr>
<tr>
<td>Patent (Cit.)</td>
<td>200000</td>
<td>0.170</td>
<td>0.090</td>
<td>0.160</td>
</tr>
<tr>
<td></td>
<td>1000000</td>
<td>0.040</td>
<td>0.600</td>
<td>0.130</td>
</tr>
</tbody>
</table>

5.3 Multigraphs

We now evaluate our algorithms designed for multigraphs. We obtained multigraph versions of Patent (Co-Auth.) (resp. LastFM) by allowing multiple edges to be placed between pairs of authors...
(resp. songs) at multiple time steps (i.e., edges with different timestamps) if the two authors co-author multiple papers (resp. the songs are co-listened on different dates). We ran our insertion-only algorithms on these multigraphs and report the results in the next paragraphs.

Figure 8 shows the evolution of the number of triangles in the two datasets as estimated by our TRIEST-FD algorithm using $M = 100,000$. For these smaller datasets we are able to compute the exact number of triangles. Our algorithm is very precise with average, min and max estimations close to the ground truth. The overall observations made for the simple graph case also hold for the multigraph case: our suite of algorithms allows precise and efficient estimation of the number of triangles with limited memory.

Figure 9 shows the average update time in microseconds using TRIEST-FD algorithm in our multigraph datasets: few microseconds are sufficient on average to update the triangle estimation, which is consistent with the results of the previous sections.

Finally we evaluate the accuracy of the estimation using our TRIEST-BASE-M and TRIEST-IMPR-M algorithms. The results are shown in Table 7. We observe that TRIEST-BASE-M and TRIEST-IMPR-M maintain a good accuracy with performance comparable to the one observed for the simple graph case.
Fig. 8. Evolution of the global number of triangles in the insertion-only case on multigraphs using \textsc{triest-impr-m} and \( M = 100,000 \). The algorithm estimations are consistently very accurate, and the curves are shown as almost indistinguishable on purpose to highlight this fact.

Fig. 9. Trade-offs between the avg. update time (\( \mu s \)) and \( M \) for \textsc{triest-impr-m} – multigraphs.

Table 7. Estimation errors for \textsc{triest-base-m} and \textsc{triest-impr-m} – multigraphs.

<table>
<thead>
<tr>
<th>Graph</th>
<th>( M )</th>
<th>Global Error \textsc{triest-base-m}</th>
<th>Global Error \textsc{triest-impr-m}</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastFM</td>
<td>100000</td>
<td>Avg. MAPE 0.015 Max MAPE 0.024</td>
<td>Avg. MAPE 0.008 Max MAPE 0.015</td>
</tr>
<tr>
<td></td>
<td>1000000</td>
<td>0.006 0.012</td>
<td>0.003 0.008</td>
</tr>
<tr>
<td>Paten (Co-Aut.)</td>
<td>100000</td>
<td>0.068 0.141</td>
<td>0.023 0.049</td>
</tr>
<tr>
<td></td>
<td>1000000</td>
<td>0.011 0.017</td>
<td>0.003 0.006</td>
</tr>
</tbody>
</table>
6 CONCLUSIONS

We presented TRIÈST, the first suite of algorithms that use reservoir sampling and its variants to continuously maintain unbiased, low-variance estimates of the local and global number of triangles in fully-dynamic graphs streams of arbitrary edge/vertex insertions and deletions using a fixed, user-specified amount of space. Our experimental evaluation shows that TRIÈST outperforms state-of-the-art approaches and achieves high accuracy on real-world datasets with more than one billion of edges, with update times of hundreds of microseconds.

APPENDIX

A ADDITIONAL THEORETICAL RESULTS

In this section we present the theoretical results (statements and proofs) not included in the main body.

A.1 Theoretical results for TRIÈST-BASE

Before proving Lemma 4.1, we need to introduce the following lemma, which states a well known property of the reservoir sampling scheme.

\[ \text{A.1 (42, Sect. 2).} \quad \text{For any } t > M, \text{ let } A \text{ be any subset of } E^{(t)} \text{ of size } |A| = M. \text{ Then, at the end of time step } t, \]

\[ \Pr(S = A) = \frac{1}{\binom{|E^{(t)}|}{M}} = \frac{1}{\binom{t}{M}}, \]

i.e., the set of edges in \( S \) at the end of time \( t \) is a subset of \( E^{(t)} \) of size \( M \) chosen uniformly at random from all subsets of \( E^{(t)} \) of the same size.

\[ \text{Proof of Lemma 4.1.} \quad \text{If } k > \min\{M, t\}, \text{ we have } \Pr(B \subseteq S) = 0 \text{ because it is impossible for } B \text{ to be equal to } S \text{ in these cases. From now on we then assume } k \leq \min\{M, t\}. \]

If \( t \leq M \), then \( E^{(t)} \subseteq S \) and \( \Pr(B \subseteq S) = 1 = \xi_{k,t}^{-1}. \)

Assume instead that \( t > M \), and let \( B \) be the family of subsets of \( E^{(t)} \) that 1. have size \( M \), and 2. contain \( B \):

\[ B = \{ C \subset E^{(t)} : |C| = M, B \subseteq C \}. \]

We have

\[ |B| = \binom{|E^{(t)}| - k}{M - k} = \binom{t - k}{M - k}. \]

From this and and Lemma A.1 we then have

\[ \Pr(B \subseteq S) = \Pr(S \in B) = \sum_{C \in B} \Pr(S = C) \]

\[ = \binom{t - k}{M - k} \cdot \binom{t - k}{M - k} = \binom{t - i}{M - i} \prod_{i=0}^{k-1} \frac{M - i}{t - i} = \xi_{k,t}^{-1}. \]

\[ \square \]

A.1.1 Expectation.

\[ \text{Proof of Lemma 4.3.} \quad \text{We only show the proof for } \tau, \text{ as the proof for the local counters follows the same steps.} \]

The proof proceeds by induction. The thesis is true after the first call to UPDATECOUNTERS at time \( t = 1 \). Since only one edge is in \( S \) at this point, we have \( \Delta^S = 0 \), and \( N_{u,v}^S = \emptyset \), so UPDATECOUNTERS does not modify \( \tau \), which was initialized to 0. Hence \( \tau = 0 = \Delta^S \).
Assume now that the thesis is true for any subsequent call to UpdateCounters up to some point in the execution of the algorithm where an edge \((u, v)\) is inserted or removed from \(S\). We now show that the thesis is still true after the call to UpdateCounters that follows this change in \(S\). Assume that \((u, v)\) was inserted in \(S\) (the proof for the case of an edge being removed from \(S\) follows the same steps). Let \(S^b = S \setminus \{(u, v)\}\) and \(\tau^b\) be the value of \(\tau\) before the call to UpdateCounters and, for any \(w \in V_{S^b}\), let \(\tau^b_w\) be the value of \(\tau_w\) before the call to UpdateCounters. Let \(\Delta^S_{u,v}\) be the set of triangles in \(G_{S}\) that have \(u\) and \(v\) as corners. We need to show that, after the call, \(\tau = |\Delta^S|\).

Clearly we have
\[
\Delta^S = \Delta^b \cup \Delta^S_{u,v} \quad \text{and} \quad \Delta^b \cap \Delta^S_{u,v} = \emptyset,
\]
so
\[
|\Delta^S| = |\Delta^b| + |\Delta^S_{u,v}|
\]

We have \(|\Delta^S_{u,v}| = |\Delta^S_{u,v}|\) and, by the inductive hypothesis, we have that \(\tau^b = |\Delta^b|\). Since UpdateCounters increments \(\tau\) by \(|\Delta^S_{u,v}|\), the value of \(\tau\) after UpdateCounters has completed is exactly \(|\Delta^S|\).

We can now prove Thm. 4.2 on the unbiasedness of the estimation computed by \(\text{Triest-base}\) (and on their exactness for \(t \leq M\)).

**Proof of Thm. 4.2.** We prove the statement for the estimation of global triangle count. The proof for the local triangle counts follows the same steps.

If \(t \leq M\), we have \(G_{S} = G^{(t)}\) and from Lemma 4.3 we have \(\tau^{(t)} = |\Delta^{(t)}| = |\Delta^{(t)}|\), hence the thesis holds.

Assume now that \(t > M\), and assume that \(|\Delta^{(t)}| > 0\), otherwise, from Lemma 4.3, we have \(\tau^{(t)} = |\Delta^{S}| = 0\) and \(\text{Triest-base}\) estimation is deterministically correct. Let \(\lambda = (a, b, c) \in \Delta^{(t)}\), (where \(a, b, c\) are edges in \(E^{(t)}\) and let \(\delta^{(t)}_{\lambda}\) be a random variable that takes value \(\xi^{(t)}\) if \(\lambda \in \Delta_{S}\) (i.e., \(\{a, b, c\} \subseteq S\)) at the end of the step instant \(t\), and 0 otherwise. From Lemma 4.1, we have that
\[
\mathbb{E}\left[\delta^{(t)}_{\lambda}\right] = \xi^{(t)} \Pr\{(a, b, c) \subseteq S\} = \xi^{(t)} \frac{1}{\xi^{(t)}} = \xi^{(t)} = 1.
\]

We can write
\[
\xi^{(t)} \tau^{(t)} = \sum_{\lambda \in \Delta^{(t)}} \delta^{(t)}_{\lambda}
\]
and from this, (16), and linearity of expectation, we have
\[
\mathbb{E}\left[\xi^{(t)} \tau^{(t)}\right] = \sum_{\lambda \in \Delta^{(t)}} \mathbb{E}\left[\delta^{(t)}_{\lambda}\right] = |\Delta^{(t)}|.
\]

\[\square\]

A.1.2 **Concentration.**

**Proof of Lemma 4.7.** Using the law of total probability, we have
\[
\Pr\left(f(S_{\text{IN}}) = 1\right) = \sum_{k=0}^{t} \Pr\left(f(S_{\text{IN}}) = 1 \mid |S_{\text{IN}}| = k\right) \Pr\left(|S_{\text{IN}}| = k\right) \\
\geq \Pr\left(f(S_{\text{IN}}) = 1 \mid |S_{\text{IN}}| = M\right) \Pr\left(|S_{\text{IN}}| = M\right) \\
\geq \Pr\left(f(S_{\text{mix}}) = 1\right) \Pr\left(|S_{\text{IN}}| = M\right),
\]
where the last inequality comes from Lemma A.1: the set of edges included in \(S_{\text{mix}}\) is a uniformly-at-random subset of \(M\) edges from \(E^{(t)}\), and the same holds for \(S_{\text{IN}}\) when conditioning its size being \(M\).
Using the Stirling approximation $\sqrt{2\pi n(n/e)^n} \leq n! \leq e\sqrt{n(n/e)^n}$ for any positive integer $n$, we have

$$\operatorname{Pr}(|S_n| = M) = \left(\frac{t}{M}\right)^M \left(\frac{t-M}{t}\right)^{t-M} \geq \frac{t^t \sqrt{2\pi e^{-t}}}{e^{2\sqrt{M(t-M)}}} \frac{M^M(t-M)^{t-M}}{t^t} \geq \frac{1}{e\sqrt{M}}.$$ 

Plugging this into (17) concludes the proof.

**FACT A.2.** For any $x > 2$, we have

$$\frac{x^2}{(x-1)(x-2)} \leq 1 + \frac{4}{x-2}.$$ 

**PROOF OF LEMMA 4.10.** We start by looking at the ratio between $\frac{t(t-1)(t-2)}{M(M-1)(M-2)}$ and $(t/M)^3$. We have:

$$1 \leq \frac{t(t-1)(t-2)}{M(M-1)(M-2)} \left(\frac{M}{t}\right)^3 = \frac{M^2}{(M-1)(M-2)} \frac{(t-1)(t-2)}{t^2} \leq \frac{M^2}{(M-1)(M-2)} \leq 1 + \frac{4}{M-2},$$ 

where the last step follows from Fact A.2. Using this, we obtain

$$\left|\phi^{(t)} - \bar{\phi}^{(t)}\right| = \tau^{(t)} \frac{t(t-1)(t-2)}{M(M-1)(M-2)} \left|\tau^{(t)} \left(\frac{t}{M}\right)^3 - 1\right| \leq \tau^{(t)} \left(\frac{t}{M}\right)^3 \frac{4}{M-2} \leq \phi^{(t)} \frac{4}{M-2}.$$ 

**A.1.3 Variance comparison.** We now prove Lemma 4.11, about the fact that the variance of the estimations computed by TRIEST-BASE is smaller, for most of the stream, than the variance of the estimations computed by MASCOT-C [28]. We first need the following technical fact.

**FACT A.3.** For any $x > 42$, we have

$$\frac{x^2}{(x-3)(x-4)} \leq 1 + \frac{8}{x}.$$ 

**PROOF OF LEMMA 4.11.** We focus on $t > M > 42$ otherwise the theorem is immediate. We show that for such conditions $f(M,t) < \tilde{f}(M/T)$ and $g(M,t) < \tilde{g}(M/T)$. Using the fact that $t \leq aT$ and
Fact A.2, we have
\[
f(M, t) - \bar{f}(M/T) = \frac{t(t-1)(t-2)}{M(M-1)(M-2)} - \frac{T^3}{M^3}
\]
\[
< \frac{\alpha^3 T^3}{M^3} \frac{M^2}{(M-1)(M-2)} - \frac{T^3}{M^3}
\]
\[
\leq \frac{\alpha^3 T^3}{M^3} \left(1 + \frac{4}{M-2}\right) - \frac{T^3}{M^3}
\]
\[
\leq \frac{T^3}{M^3} \left(\alpha^3 + \frac{4\alpha^3}{M-2} - 1\right).
\]  
(18)

Given that \(T\) and \(M\) are \(\geq 42\), the r.h.s. of (18) is non-positive iff
\[
\alpha^3 + \frac{4\alpha^3}{M-2} - 1 \leq 0.
\]
Solving for \(M\) we have that the above is verified when \(M \geq \frac{4\alpha^3}{1-\alpha} + 2\). This is always true given our assumption that \(M > \max(\frac{8\alpha}{1-\alpha}, 42)\): for any \(0 < \alpha < 0.6\), we have \(\frac{4\alpha^3}{1-\alpha} + 2 < 42 \leq M\) and for any \(0.6 \leq \alpha < 1\) we have \(\frac{4\alpha^3}{1-\alpha} + 2 < \frac{8\alpha}{1-\alpha} \leq M\). Hence the r.h.s. of (18) is \(\leq 0\) and \(f(M, t) < \bar{f}(M/T)\).

We also have:
\[
g(M, t) - \bar{g}(M/T) = \frac{t(t-1)(t-2)(M-3)(M-4)}{(t-3)(t-4)M(M-1)(M-2)} - \frac{T}{M}
\]
\[
< \frac{t}{M} \frac{t^2}{(t-3)(t-4)} - \frac{T}{M}
\]
\[
\leq \frac{t}{M} \left(1 + \frac{8}{t}\right) - \frac{T}{M},
\]  
(19)

where the last inequality follow from Fact A.3, since \(t > M > 42\). Now, from (19) since \(t \leq \alpha T\) and \(t > M\), we can write:
\[
g(M, t) - \bar{g}(M/T) < \frac{T}{M} \left(\alpha + \frac{8\alpha}{M} - 1\right).
\]
The r.h.s. of this equation is non-positive given the assumption \(M > \frac{8\alpha}{1-\alpha}\), hence \(g(M, t) < \bar{g}(M/T)\).

\[\square\]

**A.2 Theoretical results for TRIÊST-IMPR**

**A.2.1 Expectation.**

**Proof of Thm. 4.12.** If \(t \leq M\) TRIÊST-IMPR behaves exactly like TRIÊST-BASE, and the statement follows from Lemma 4.2.

Assume now \(t > M\) and assume that \(|\Delta(t)| > 0\), otherwise, the algorithm deterministically returns 0 as an estimation and the thesis follows. Let \(\lambda \in \Delta(t)\) and denote with \(a, b,\) and \(c\) the edges of \(\lambda\) and assume, w.l.o.g., that they appear in this order (not necessarily consecutively) on the stream. Let \(t_\lambda\) be the time step at which \(c\) is on the stream. Let \(\delta_\lambda\) be a random variable that takes value \(\xi_{2, t_\lambda-1}\) if \(a\) and \(b\) are in \(S\) at the end of time step \(t_\lambda - 1\), and 0 otherwise. Since it must be \(t_\lambda - 1 \geq 2\), from Lemma 4.1 we have that
\[
\Pr(\delta_\lambda = \xi_{2, t_\lambda-1}) = \frac{1}{\xi_{2, t_\lambda-1}}.
\]  
(20)
When \( c = (u, v) \) is on the stream, i.e., at time \( t_{\lambda} \), TRIEST-IMPR calls \textsc{UpdateCounters} and increments the counter \( \tau \) by \( |N_{u,v}^{S}|\xi_{s,t_{\lambda}-1}^{2} \), where \( |N_{u,v}^{S}| \) is the number of triangles with \((u, v)\) as an edge in \( \Delta^{S \cup \{c\}} \). All these triangles have the corresponding random variables taking the same value \( \xi_{s,t_{\lambda}-1}^{2} \).

This means that the random variable \( \tau^{(t)} \) can be expressed as

\[
\tau^{(t)} = \sum_{\lambda \in \Delta^{(t)}} \delta_{\lambda} .
\]

From this, linearity of expectation, and (20), we get

\[
\mathbb{E} [\tau^{(t)}] = \sum_{\lambda \in \Delta^{(t)}} \mathbb{E} [\delta_{\lambda}] = \sum_{\lambda \in \Delta^{(t)}} \xi_{s,t_{\lambda}-1} \Pr (\delta_{\lambda} = \xi_{s,t_{\lambda}-1}) = \sum_{\lambda \in \Delta^{(t)}} \xi_{s,t_{\lambda}-1} \frac{1}{\xi_{s,t_{\lambda}-1}} = |\Delta^{(t)}| .
\]

\[\square\]

A.2.2 Variance.

Proof of Lemma 4.14. Consider first the case where all edges of \( \lambda \) appear on the stream before any edge of \( \gamma \), i.e.,

\[
t_{\ell_{1}} < t_{\ell_{2}} < t_{\gamma_{1}} < t_{g_{2}} < t_{g_{1}} .
\]

The presence or absence of either or both \( \ell_{1} \) and \( \ell_{2} \) in \( S \) at the beginning of time step \( t_{\ell_{1}} \) (i.e., whether \( D_{\lambda} \) happens or not) has no effect whatsoever on the probability that \( g_{1} \) and \( g_{2} \) are in the sample \( S \) at the beginning of time step \( t_{g_{1}} \). Hence in this case,

\[
\Pr (D_{\gamma} \mid D_{\lambda}) = \Pr (D_{\gamma}) .
\]

Consider now the case where, for any \( i \in \{1, 2\} \), the edges \( g_{1}, \ldots, g_{i} \) appear on the stream before \( \ell_{3} \) does. Define now the events

- \( A_{i} \): “the edges \( g_{1}, \ldots, g_{i} \) are in the sample \( S \) at the beginning of time step \( t_{\ell_{3}} \).”
- \( B_{i} \): if \( i = 1 \), this is the event “the edge \( g_{2} \) is inserted in the sample \( S \) during time step \( t_{g_{2}} \).” If \( i = 2 \), this is the whole event space, i.e., the event that happens with probability 1.
- \( C \): “neither \( g_{1} \) nor \( g_{2} \) were among the edges removed from \( S \) between the beginning of time step \( t_{\ell_{3}} \) and the beginning of time step \( t_{g_{1}} \).”

We can rewrite \( D_{\gamma} \) as

\[
D_{\gamma} = A_{i} \cap B_{i} \cap C .
\]

Hence

\[
\Pr (D_{\gamma} \mid D_{\lambda}) = \Pr (A_{i} \cap B_{i} \cap C \mid D_{\lambda}) = \Pr (A_{i} \mid D_{\lambda}) \Pr (B_{i} \cap C \mid A_{i} \cap D_{\lambda}) .
\]

We now show that

\[
\Pr (A_{i} \mid D_{\lambda}) \leq \Pr (A_{i}) .
\]

If we assume that \( t_{\ell_{3}} \leq M + 1 \), then all the edges that appeared on the stream up until the beginning of \( t_{\ell_{3}} \) are in \( S \). Therefore,

\[
\Pr (A_{i} \mid D_{\lambda}) = \Pr (A_{i}) = 1 .
\]

Assume instead that \( t_{\ell_{3}} > M + 1 \). Among the \( \binom{t_{\ell_{3}}-1}{M} \) subsets of \( E(t_{\ell_{3}}-1) \) of size \( M \), there are \( \binom{t_{\ell_{3}}-3}{M-2} \) that contain \( \ell_{1} \) and \( \ell_{2} \). From Lemma A.1, we have that at the beginning of time \( t_{\ell_{3}} \), \( S \) is a subset of size \( M \) of \( E(t_{\ell_{3}}-1) \) chosen uniformly at random. Hence, if we condition on the fact that \( \{\ell_{1}, \ell_{2}\} \subset S \),
we have that $S$ is chosen uniformly at random from the $(t_{\ell_2} - 3)\choose M - 2 - i$ subsets of $E^{(t_{\ell_2} - 1)}$ of size $M$ that contain $\ell_1$ and $\ell_2$. Among these, there are $(t_{\ell_2} - 3)\choose M - 2 - i$ that also contain $g_1, \ldots, g_i$. Therefore,

$$\Pr(A_i \mid D_\lambda) = \frac{\binom{t_{\ell_2} - 3}{M - 2 - i}}{\binom{t_{\ell_2} - 3}{M - 2}} = \prod_{j=0}^{i-1} \frac{M - 2 - j}{t_{\ell_2} - 3 - j}.$$ 

From Lemma 4.1 we have

$$\Pr(A_i) = \frac{1}{\mathcal{E}_{t_{\ell_2} - 1}} = \prod_{j=0}^{i-1} \frac{M - j}{t_{\ell_2} - 1 - j},$$

where the last equality comes from the assumption $t_{\ell_2} > M + 1$. From the same assumption and from the fact that for any $j \geq 0$ and any $y \geq x > j$ it holds $\frac{x - j}{y - j} \leq \frac{x}{y}$, then we have

$$\Pr(A_i \mid D_\lambda) \leq \Pr(A_i).$$

This implies, from (21), that

$$\Pr(D_y \mid D_\lambda) \leq \Pr(A_i) \Pr(B_i \cap C \mid A_i \cap D_\lambda). \tag{22}$$

Consider now the events $B_i$ and $C$. When conditioned on $A_i$, these event are both independent from $D_\lambda$: if the edges $g_1, \ldots, g_i$ are in $S$ at the beginning of time $t_{\ell_2}$, the fact that the edges $\ell_1$ and $\ell_2$ were also in $S$ at the beginning of time $t_{\ell_2}$ has no influence whatsoever on the actions of the algorithm (i.e., whether an edge is inserted in or removed from $S$). Thus,

$$\Pr(A_i) \Pr(B_i \cap C \mid A_i \cap D_\lambda) = \Pr(A_i) \Pr(B_i \cap C \mid A_i).$$

Putting this together with (22), we obtain

$$\Pr(D_y \mid D_\lambda) \leq \Pr(A_i) \Pr(B_i \cap C \mid A_i) \leq \Pr(A_i \cap B_i \cap C) \leq \Pr(D_y),$$

where the last inequality follows from the fact that $D_y = A_i \cap B_i \cap C$ by definition. \hfill \Box

### A.3 Theoretical results for Triest-FD

#### A.3.1 Expectation.
Before proving Thm. 4.16 we need the following technical lemmas.

The following is a corollary of [16, Thm. 1].

**Lemma A.4.** For any $t > 0$, and any $j$, $0 \leq j \leq s^{(t)}$, let $\mathcal{B}^{(t)}$ be the collection of subsets of $E^{(t)}$ of size $j$. For any $B \in \mathcal{B}^{(t)}$ it holds

$$\Pr \left( S = B \mid M^{(t)} = j \right) = \frac{1}{\binom{|E^{(t)}|}{j}}.$$ 

That is, conditioned on its size at the end of time step $t$, $S$ is equally likely to be, at the end of time step $t$, any of the subsets of $E^{(t)}$ of that size.

The next lemma is an immediate corollary of [16, Thm. 2].

**Lemma A.5.** Recall the definition of $\kappa^{(t)}$ from (14). We have

$$\kappa^{(t)} = \Pr(M^{(t)} \geq 3).$$

The next lemma follows from Lemma A.4 in the same way as Lemma 4.1 follows from Lemma A.1.

**Lemma A.6.** For any time step $t$ and any $j$, $0 \leq j \leq s^{(t)}$, let $B$ be any subset of $E^{(t)}$ of size $|B| = k \leq s^{(t)}$. Then, at the end of time step $t$,

$$\Pr \left( B \subseteq S \mid M^{(t)} = j \right) = \begin{cases} 0 & \text{if } k > j \\ 1 & \text{otherwise} \end{cases}.$$
The next two lemmas discuss properties of TRIÊST-FD for \( t < t^* \), where \( t^* \) is the first time that \( |E(t)| \) had size \( M + 1 \) (\( t^* \geq M + 1 \)).

**Lemma A.7.** For all \( t < t^* \), we have:

1. \( d_o(t) = 0 \); and
2. \( S = E(t) \); and
3. \( M(t) = s(t) \).

**Proof.** Since the third point in the thesis follows immediately from the second, we focus on the first two points.

The proof is by induction on \( t \). In the base base for \( t = 1 \): the element on the stream must be an insertion, and the algorithm deterministically inserts the edge in \( S \). Assume now that it is true for all time steps up to (but excluding) some \( t \leq t^* - 1 \). We now show that it is also true for \( t \).

Assume the element on the stream at time \( t \) is a deletion. The corresponding edge must be in \( S \), from the inductive hypothesis. Hence TRIÊST-FD removes it from \( S \) and increments the counter \( d_i \) by 1. Thus it is still true that \( S = E(t) \) and \( d_o(t) = 0 \), and the thesis holds.

Assume now that the element on the stream at time \( t \) is an insertion. From the inductive hypothesis we have that the current value of the counter \( d_o \) is 0.

If the counter \( d_i \) has currently value 0 as well, then, because of the hypothesis that \( t < t^* \), it must be that \( |S| = M(t-1) = s(t-1) < M \). Therefore TRIÊST-FD always inserts the edge in \( S \). Thus it is still true that \( S = E(t) \) and \( d_o(t) = 0 \), and the thesis holds.

If otherwise \( d_i > 0 \), then TRIÊST-FD flips a biased coin with probability of heads equal to

\[
\frac{d_i}{d_i + d_o} = \frac{d_i}{d_i} = 1,
\]

therefore TRIÊST-FD always inserts the edge in \( S \) and decrements \( d_i \) by one. Thus it is still true that \( S = E(t) \) and \( d_o(t) = 0 \), and the thesis holds. \( \square \)

The following result is an immediate consequence of Lemma A.5 and Lemma A.7.

**Lemma A.8.** For all \( t < t^* \) such that \( s(t) \geq 3 \), we have \( \kappa(t) = 1 \).

We can now prove Thm. 4.16.

**Proof of Thm. 4.16.** Assume for now that \( t < t^* \). From Lemma A.7, we have that \( s(t) = M(t) \). If \( M(t) < 3 \), then it must be \( s(t) < 3 \), hence \( |\Delta^{(t)}| = 0 \) and indeed the algorithm returns \( \rho^{(t)} = 0 \) in this case. If instead \( M(t) = s(t) \geq 3 \), then we have

\[
\rho^{(t)} = \frac{\tau^{(t)}}{\kappa^{(t)}}.
\]

From Lemma A.8 we have that \( \kappa^{(t)} = 1 \) for all \( t < t^* \), hence \( \rho^{(t)} = \tau^{(t)} \) in these cases. Since (an identical version of) Lemma 4.3 also holds for TRIÊST-FD, we have \( \tau^{(t)} = |\Delta^S| = |\Delta^{(t)}| \), where the last equality comes from the fact that \( S = E(t) \) (Lemma A.7). Hence \( \rho^{(t)} = |\Delta^{(t)}| \) for any \( t \leq t^* \), as in the thesis.

Assume now that \( t \geq t^* \). Using the law of total expectation, we can write

\[
\mathbb{E} \left[ \rho^{(t)} \right] = \sum_{j=0}^{\min \{ s(t), M \}} \mathbb{E} \left[ \rho^{(t)} | M^{(t)} = j \right] \Pr \left( M^{(t)} = j \right). \tag{23}
\]
Assume that $|\Delta(t)| > 0$, otherwise, the algorithm deterministically returns 0 as an estimation and the thesis follows. Let $\lambda$ be a triangle in $\Delta(t)$, and let $\delta_\lambda^{(t)}$ be a random variable that takes value
\[
\frac{\psi_{3,M(t),s(t)}}{\kappa(t)} = \frac{s(t)(s(t)-2)(s(t)-2)}{M(t)(M(t)-1)(M(t)-2)\kappa(t)}
\]
if all edges of $\lambda$ are in $S$ at the end of the time instant $t$, and 0 otherwise. Since (an identical version of) Lemma 4.3 also holds for TRIËST-FD, we can write
\[
\rho(t) = \sum_{\lambda \in \Delta(t)} \delta_\lambda^{(t)}.
\]
Then, using Lemma A.5 and Lemma A.6, we have, for $3 \leq j \leq \min\{M, s(t)\}$,
\[
\mathbb{E}\left[\rho(t) \mid M(t) = j\right] = \sum_{\lambda \in \Delta(t)} \frac{\psi_{3,j,s(t)}}{\kappa(t)} \Pr\left(\delta_\lambda^{(t)} = \frac{\psi_{3,j,s(t)}}{\kappa(t)} \mid M(t) = j\right) = |\Delta(t)| \frac{\psi_{3,j,s(t)}}{\kappa(t)} \frac{1}{\psi_{3,j,s(t)}} = \frac{1}{\kappa(t)} |\Delta(t)|,
\]
and
\[
\mathbb{E}\left[\rho(t) \mid M(t) = j\right] = 0, \text{ if } 0 \leq j \leq 2.
\]
Plugging this into (23), and using Lemma A.5, we have
\[
\mathbb{E}\left[\rho(t)\right] = |\Delta(t)| \frac{1}{\kappa(t)} \sum_{j=3}^{\min\{s(t),M\}} \Pr(M(t) = j) = |\Delta(t)| .
\]

A.3.2 Variance. We now move to prove Thm. 4.17 about the variance of the TRIËST-FD estimator. We first need some technical lemmas.

**Lemma A.9.** For any time $t \geq t^*$, and any $j$, $3 \leq j \leq \min\{s(t), M\}$, we have:
\[
\text{Var}\left[\rho(t) \mid M(t) = j\right] = (\kappa(t))^{-2} \left(|\Delta(t)| \left(\frac{\psi_{3,j,s(t)}}{\kappa(t)} - 1\right) + r(t) \left(\frac{\psi_{3,j,s(t)}}{\kappa(t)} - 1\right)\right).
\]
An analogous result holds for any $u \in V(t)$, replacing the global quantities with the corresponding local ones.

**Proof.** The proof is analogous to that of Theorem 4.4, using $j$ in place of $M$, $s(t)$ in place of $t$, $\hat{\psi}_{a,i,s(t)}$ in place of $\hat{\xi}_{a,t}$, and using Lemma A.6 instead of Lemma 4.1. The additional $(\kappa(t))^{-2}$ multiplicative term comes from the $(\kappa(t))^{-1}$ term used in the definition of $\rho(t)$. \qed

The term $w(t) \left(\frac{\psi_{3,j,s(t)}}{\kappa(t)} - 1\right)$ is non-positive.

**Lemma A.10.** For any time $t \geq t^*$, and any $j$, $6 \leq j \leq \min\{s(t), M\}$, if $s(t) \geq M$ we have:
\[
\text{Var}\left[\rho(t) \mid M(t) = i\right] \leq (\kappa(t))^{-2} \left(|\Delta(t)| \left(\frac{\psi_{3,j,s(t)}}{\kappa(t)} - 1\right) + r(t) \left(\frac{\psi_{3,j,s(t)}}{\kappa(t)} - 1\right)\right), \text{ for } i \geq j
\]
\[
\text{Var}\left[\rho(t) \mid M(t) = i\right] \leq (\kappa(t))^{-2} \left(|\Delta(t)| \left(\frac{\psi_{3,j,s(t)}}{\kappa(t)} - 1\right) + r(t) \left(\frac{\psi_{3,j,s(t)}}{\kappa(t)} - 1\right)\right), \text{ for } i < j
\]
An analogous result holds for any $u \in V(t)$, replacing the global quantities with the corresponding local ones.
The proof follows by observing that the term \( w^{(t)} \left( \psi^2_{S_i^t, s^{(t)}_i} \psi^{-1}_{S_i^t, s^{(t)}_i} - 1 \right) \) is non-positive, and that (26) is a non-increasing function of the sample size.

The following lemma deals with properties of the r.v. \( M^{(t)} \).

**Lemma A.11.** Let \( t > t^* \), with \( s^{(t)} \geq M \). Let \( d^{(t)} = d^{(t)}_0 + d^{(t)}_1 \) denote the total number of unpaired deletions at time \( t \).\(^{19}\) The sample size \( M^{(t)} \) follows the hypergeometric distribution:\(^{20}\)

\[
\Pr \left( M^{(t)} = j \right) = \begin{cases} \frac{\binom{s^{(t)}}{j} \binom{d^{(t)}}{M-j}}{\binom{s^{(t)}+d^{(t)}}{M}} & \text{for } \max\{ M - d^{(t)}, 0 \} \leq j \leq M, \\ \text{otherwise} & \end{cases}
\]

(27)

We have

\[
\mathbb{E} \left[ M^{(t)} \right] = M \frac{s^{(t)}}{s^{(t)} + d^{(t)}},
\]

(28)

and for any \( 0 < c < 1 \)

\[
\Pr \left( M^{(t)} > \mathbb{E} \left[ M^{(t)} \right] - cM \right) \geq 1 - \frac{1}{e^{c^2 M}}.
\]

(29)

**Proof.** Since \( t > t^* \), from the definition of \( t^* \) we have that the \( M^{(t)} \) has reached size \( M \) at least once (at \( t^* \)). From this and the definition of \( d^{(t)} \) (number of uncompensated deletion), we have that \( M^{(t)} \) can not be less than \( M - d^{(t)} \). The rest of the proof for (27) and for (28) follows from [16, Thm. 2].

The concentration bound in (29) follows from the properties of the hypergeometric distribution discussed by Skala [37].

The following is an immediate corollary from Lemma A.11.

**Corollary A.12.** Consider the execution of \( \text{TRIÊST-FD} \) at time \( t > t^* \). Suppose we have \( d^{(t)} \leq \alpha s^{(t)} \), with \( 0 \leq \alpha < 1 \) and \( s^{(t)} \geq M \). If \( M \geq \frac{1}{2 \sqrt{\alpha' - \alpha}} c' \ln s^{(t)} \) for \( \alpha < \alpha' < 1 \), we have:

\[
\Pr \left( M^{(t)} \geq M(1 - \alpha') \right) > 1 - \frac{1}{(s^{(t)})^{c'\alpha'}}.
\]

We can now prove Thm. 4.17.

**Proof of Thm. 4.17.** From the law of total variance we have:

\[
\text{Var} \left[ \rho^{(t)} \right] = \sum_{j=0}^{M} \text{Var} \left[ \rho^{(t)} | M^{(t)} = j \right] \Pr \left( M^{(t)} = j \right)
\]

\[
+ \sum_{j=0}^{M} \mathbb{E} \left[ \rho^{(t)} | M^{(t)} = j \right]^2 \left( 1 - \Pr \left( M^{(t)} = j \right) \right) \Pr \left( M^{(t)} = j \right)
\]

\[
- 2 \sum_{j=1}^{M} \sum_{i=0}^{j-1} \mathbb{E} \left[ \rho^{(t)} | M^{(t)} = j \right] \Pr \left( M^{(t)} = j \right) \mathbb{E} \left[ \rho^{(t)} | M^{(t)} = i \right] \Pr \left( M^{(t)} = i \right).
\]

\(^{19}\)While both \( d^{(t)}_0 \) and \( d^{(t)}_1 \) are r.v.s, their sum is not.

\(^{20}\)We use here the convention that \( \binom{0}{0} = 1 \), and \( \binom{0}{1} = 1 \).
As shown in (24) and (25), for any $j = 0, 1, \ldots, M$ we have $\mathbb{E}[\rho(t)|M(t) = j] \geq 0$. This in turn implies:

$$\text{Var} \left[ \rho(t) \right] \leq \sum_{j=0}^{M} \text{Var} \left[ \rho(t)|M(t) = j \right] \text{Pr} \left( M(t) = j \right)$$

$$+ \sum_{j=0}^{M} \mathbb{E} \left[ \rho(t)|M(t) = j \right]^2 (1 - \text{Pr} \left( M(t) = j \right)) \text{Pr} \left( M(t) = j \right).$$ (30)

Let us consider separately the two main components of (30). From Lemma A.10 we have:

$$\sum_{j=0}^{M} \text{Var} \left[ \rho(t)|M(t) = j \right] \text{Pr} \left( M(t) = j \right) =$$

$$\sum_{j \geq M(1-\alpha')}^{M} \text{Var} \left[ \rho(t)|M(t) = j \right] \text{Pr} \left( M(t) = j \right) + \sum_{j=0}^{M(1-\alpha')} \text{Var} \left[ \rho(t)|M(t) = j \right] \text{Pr} \left( M(t) = j \right)$$

$$\leq (\kappa^{(t)})^{-2} \left( |\Delta^{(t)}| \left( \psi_{3,s(t)} - 1 \right) + r^{(t)} \left( \psi_{3,s(t)}^{-1}(\psi_{5,s(t)} - 1) \right) \right) \times \text{Pr} \left( M(t) > M(1-\alpha') \right)$$

$$\leq (\kappa^{(t)})^{-2} \left( |\Delta^{(t)}| \left( \frac{(s^{(t)})^3}{6} + r^{(t)} \frac{s^{(t)}}{6} \right) \right) \text{Pr} \left( M(t) \leq M(1-\alpha') \right)$$ (32)

According to our hypothesis $M \geq \frac{1}{2\sqrt{\alpha'-\alpha}} 7 \ln s^{(t)}$, thus we have, from Corollary A.12:

$$\text{Pr} \left( M(t) \leq M(1-\alpha') \right) \leq \frac{1}{(s^{(t)})^7}.$$

As $r^{(t)} < |\Delta^{(t)}|^2$ and $|\Delta^{(t)}| \leq (s^{(t)})^3$ we have:

$$(\kappa^{(t)})^{-2} \left( |\Delta^{(t)}| \left( \frac{(s^{(t)})^3}{6} + r^{(t)} \frac{s^{(t)}}{6} \right) \right) \text{Pr} \left( M(t) \leq M(1-\alpha') \right) \leq (\kappa^{(t)})^{-2}$$

We can therefore rewrite (32) as:

$$\sum_{j=0}^{M} \text{Var} \left[ \rho(t)|M(t) = j \right] \text{Pr} \left( M(t) = j \right) \leq (\kappa^{(t)})^{-2} \left( |\Delta^{(t)}| \left( \psi_{3,M(1-\alpha'),s(t)} - 1 \right) \right)$$

$$+ (\kappa^{(t)})^{-2} \left( r^{(t)} \left( \psi_{3,M(1-\alpha'),s(t)}^{-1}\psi_{5,M(1-\alpha'),s(t)} - 1 \right) + 1 \right).$$ (33)

Let us now consider the term $\sum_{j=0}^{M} \mathbb{E}[\rho(t)|M(t) = j]^2 (1 - \text{Pr} \left( M(t) = j \right)) \text{Pr} \left( M(t) = j \right)$. Recall that, from (24) and (25), we have $\mathbb{E}[\rho(t)|M(t) = j] = |\Delta^{(t)}|(\kappa^{(t)})^{-1}$ for $j = 3, \ldots, M$, and $\mathbb{E}[\rho(t)|M(t) = j] = 0$ for $j = 0, 1, 2$. From Corollary A.12 we have that for $j \leq (1-\alpha')M$ and $M \geq \frac{1}{2\sqrt{\alpha'-\alpha}} 7 \ln s^{(t)}$

$$\text{Pr} \left( M(t) = j \right) \leq \text{Pr} \left( M(t) \leq (1-\alpha')M \right) \leq \frac{1}{(s^{(t)})^7},$$

and thus:

$$\sum_{j=0}^{(1-\alpha')M} \mathbb{E}[\rho(t)|M(t) = j]^2 (1 - \text{Pr} \left( M(t) = j \right)) \text{Pr} \left( M(t) = j \right) \leq \frac{(1-\alpha')M|\Delta^{(t)}|^2(\kappa^{(t)})^{-2}}{(s^{(t)})^7}$$

$$\leq (1-\alpha')(\kappa^{(t)})^{-2},$$ (34)
where the last passage follows since, by hypothesis, $M \leq s(t)$.

Let us now consider the values $j > (1 - \alpha') M$, we have:

$$
\sum_{j > (1 - \alpha') M}^M \mathbb{E} \left[ \rho^{(t)}|M^{(t)} = j \right] (1 - \Pr(M^{(t)} = j)) \Pr(M^{(t)} = j) \\
\leq \alpha' M |\Delta^{(t)}|^2 (\kappa^{(t)})^{-2} \left( 1 - \sum_{j > (1 - \alpha') M}^M \Pr(M^{(t)} = j) \right) \\
\leq \alpha' M |\Delta^{(t)}|^2 (\kappa^{(t)})^{-2} \left( 1 - \Pr(M^{(t)} > (1 - \alpha') M) \right) \\
\leq \frac{\alpha' M |\Delta^{(t)}|^2 (\kappa^{(t)})^{-2}}{(s(t))^2} \\
\leq \alpha' (\kappa^{(t)})^{-2},
$$

(35)

where the last passage follows since, by hypothesis, $M \leq s(t)$.

The theorem follows from composing the upper bounds obtained in (33), (34) and (35) according to (30).

\[\square\]

A.3.3 Concentration. We now prove Thm. 4.18 about TRIÆST-FD.

PROOF OF THM. 4.18. By Chebyshev’s inequality it is sufficient to prove that

$$
\frac{\text{Var}[\rho^{(t)}]}{\epsilon^2 |\Delta^{(t)}|^2} < \delta.
$$

From Lemma 4.17, for $M \geq \frac{1}{\sqrt{\alpha' - \alpha}} \ln s(t)$ we have:

$$
\text{Var} \left[ \rho^{(t)} \right] \leq (\kappa^{(t)})^{-2} |\Delta^{(t)}| \left( \psi_{3, M(1-\alpha'), s(t)} - 1 \right) \\
+ (\kappa^{(t)})^{-2} \epsilon(t) \left( \psi_{3, M(1-\alpha'), s(t)}^{-1}, \psi_{5, M(\alpha - \alpha'), s(t)} - 1 \right) \\
+ (\kappa^{(t)})^{-2} \epsilon^2
$$

Let $M' = (1 - \alpha') M$. In order to verify that the lemma holds, it is sufficient to impose the following two conditions:

**Condition (1)**

$$
\frac{\delta}{2} > \frac{(\kappa^{(t)})^{-2} \left( |\Delta^{(t)}| \left( \psi_{3, M(1-\alpha'), s(t)} - 1 \right) + 2 \right)}{\epsilon^2 |\Delta^{(t)}|^2}.
$$

As by hypothesis $|\Delta^{(t)}| > 0$, we can rewrite this condition as:

$$
\frac{\delta}{2} > \frac{(\kappa^{(t)})^{-2} \left( \psi_{3, M(1-\alpha'), s(t)} - \frac{|\Delta^{(t)}| - 2}{|\Delta^{(t)}|} \right)}{\epsilon^2 |\Delta^{(t)}|}.
$$
which is verified for:

\[ M'(M' - 1)(M' - 2) > \frac{2s^{(t)}(s^{(t)} - 1)(s^{(t)} - 2)}{\delta \varepsilon^2 |\Delta^{(t)}| (\kappa^{(t)})^2 + 2 \frac{|\Delta^{(t)}|^2 - 2}{|\Delta^{(t)}|}}. \]

\[ M' > \sqrt{\frac{2s^{(t)}(s^{(t)} - 1)(s^{(t)} - 2)}{\delta \varepsilon^2 |\Delta^{(t)}| (\kappa^{(t)})^2 + 2 \frac{|\Delta^{(t)}|^2 - 2}{|\Delta^{(t)}|}}} + 2, \]

\[ M > (1 - \alpha')^{-1} \left( \sqrt{\frac{2s^{(t)}(s^{(t)} - 1)(s^{(t)} - 2)}{\delta \varepsilon^2 |\Delta^{(t)}| (\kappa^{(t)})^2 + 2 \frac{|\Delta^{(t)}|^2 - 2}{|\Delta^{(t)}|}}} + 2 \right). \]

**Condition (2)**

\[ \frac{\delta}{2} > \frac{(\kappa^{(t)})^{-2} r^{(t)} (\psi_{s,M(1-\alpha'),s^{(t)}}^{-1})^2}{\varepsilon^2 |\Delta^{(t)}| (\kappa^{(t)})^2} \left( \psi_{s,M(1-\alpha'),s^{(t)}}^{-1} - 1 \right). \tag{36} \]

As we have:

\[ (\kappa^{(t)})^{-2} r^{(t)} (\psi_{s,M(1-\alpha'),s^{(t)}}^{-1})^2 \leq (\kappa^{(t)})^{-2} r^{(t)} \left( \frac{s^{(t)}}{6M(1 - \alpha')} - 1 \right) \]

The condition (36) is verified for:

\[ M > \frac{(1 - \alpha')^{-1}}{3} \left( \frac{r^{(t)} s^{(t)}}{\delta \varepsilon^2 |\Delta^{(t)}| (\kappa^{(t)})^2 + 2 r^{(t)}} \right). \]

The theorem follows. \( \square \)

**ACKNOWLEDGMENTS**

This work was supported in part by the National Science Foundation grant IIS-1247581 (https://www.nsf.gov/awardsearch/showAward?AWD_ID=1247581) and the National Institutes of Health grant R01-CA180776 (https://projectreporter.nih.gov/project_info_details.cfm?icde=0&aid=8685211).

This work is supported, in part, by funding from Two Sigma Investments, LP. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Two Sigma Investments, LP or the National Science Foundation.

**REFERENCES**


