COLORFUL TRIANGLE COUNTING
AND A MAPREDUCE IMPLEMENTATION

RASMUS PAGH AND CHARALAMPOS E. TSOURAKAKIS

ABSTRACT. In this note we introduce a new randomized algorithm for counting triangles in graphs. We show that under mild conditions, the estimate of our algorithm is strongly concentrated around the true number of triangles. Specifically, let G be a graph with n vertices, t triangles and let Δ be the maximum number of triangles an edge of G is contained in. Our randomized algorithm colors the vertices of G with $N = 1/p$ colors uniformly at random, counts monochromatic triangles, i.e., triangles whose vertices have the same color, and scales that count appropriately. We show that if $p \geq \max\left(\frac{\Delta \log n}{t}, \sqrt{\frac{\log n}{t}}\right)$ then for any constant $\epsilon > 0$ our unbiased estimate T is concentrated around its expectation, i.e., $\mathbf{Pr}\left[|T - \mathbb{E}[T]| \geq \epsilon \mathbb{E}[T]\right] = o(1)$. Finally, our algorithm is amenable to being parallelized. We present a simple MAPREDUCE implementation of our algorithm.

1. INTRODUCTION

Triangle counting is a fundamental algorithmic problem with many applications. The interested reader is urged to see [18] and references therein. The fastest exact triangle counting algorithm to date (in terms of number of edges in the graph) is due to Alon, Yuster and Zwick [4] and runs in time $O(m^{\frac{2\omega}{\omega+1}})$, where currently the matrix multiplication exponent ω is 2.371 [10]. For planar graphs linear time algorithms are known, e.g., [20]. Practical methods for exact triangle counting use enumeration techniques, see e.g., [19, 21] and references therein. For many applications, especially in the context of large social networks, an exact count is not crucial but rather a fast, high quality estimate. Most of the work on approximate triangle counting is sampling-based and has considered a (semi-)streaming setting [6, 7, 8, 15, 26]. A different line of research is based on a linear algebraic approach [5, 23]. There is a close connection between these two lines of research, see [2, 24, 23, 26, 27]. Currently to the best of our knowledge, the state-of-the-art approximate counting method relies on a hybrid algorithm that first sparsifies the graph and then samples triples according to a degree based partitioning trick [18].

In this short note, we present a new sampling approach to approximating the number of triangles in a graph $G(V, E)$, that significantly improves existing sampling approaches. Furthermore, it is easily implemented in parallel. The key idea of our algorithm is to correlate the sampling of edges such that if two edges of a triangle are sampled, the third edge is always sampled. This decreases the degree of the multivariate polynomial that expresses the number of sampled triangles. We analyze our method using a powerful theorem due to Hajnal and Szemerédi [12]. This note is organized as follows: in Section 2 we discuss the theoretical preliminaries for our analysis and in Section 1.1 we present our randomized algorithm. In Section 3

Algorithm 1 Colorful Triangle Sampling

Require: Unweighted graph $G([n], E)$
Require: Number of colors $N = 1/p$
  Let $f : V \to [N]$ have uniformly random values
  $E' \leftarrow \{\{u, v\} \in E \mid f(u) = f(v)\}$
  $T \leftarrow$ number of triangles in the graph (V, E')
  return T/p^2

---

we present our main theoretical results, we analyze our algorithm and we discuss some of its important properties. In Section 4 we present an implementation of our algorithm in the popular MapReduce framework. Finally, in Section 5 we conclude with future research directions.

1.1. **Algorithm.** Our algorithm, summarized as Algorithm 1, samples each edge with probability p, where $N = 1/p$ is integer, as follows. Let $f : [n] \to [N]$ be a random coloring of the vertices of $G([n], E)$, such that for all $v \in [n]$ and $i \in [N]$, $\mathbf{Pr}\,[f(v) = i] = p$. We call an edge *monochromatic* if both its endpoints have the same color. Our algorithm samples exactly the set E' of monochromatic edges, counts the number T of triangles in $([n], E')$ (using any exact or approximate triangle counting algorithm), and multiplies this count by p^{-2}.

Previous work [25, 26] has used a related sampling idea, the difference being that edges were sampled *independently* with probability p. Some intuition why this sampling procedure is less efficient than what we propose can be obtained by considering the case where a graph has t edge-disjoint triangles. With independent edge sampling there will be no triangles left (with probability $1 - o(1)$) if $p^3 t = o(1)$. Using our colorful sampling idea there will be $\omega(1)$ triangles in the sample with probability $1 - o(1)$ as long as $p^2 t = \omega(1)$. This means that we can choose a smaller sample, and still get a accurate estimates from it.

2. Theoretical Preliminaries

In Section 3.2 we use the following version of the Chernoff bound [9].

Lemma 1 (Chernoff Inequality). *Let $X_1, X_2, \ldots, X_k$ be independently distributed $\{0, 1\}$ variables with $E[X_i] = p$. Then for any $\epsilon > 0$, we have*

$$\mathbf{Pr}\left[|\frac{1}{k}\sum_{i=1}^{k} X_i - p| > \epsilon p\right] \leq 2e^{-\epsilon^2 pk/2}$$

Hajnal and Szemerédi [12] proved in 1970 the following conjecture of Paul Erdös:

Theorem 1 (Hajnal-Szemerédi Theorem). *Every graph with n vertices and maximum vertex degree at most k is $k + 1$ colorable with all color classes of size at least $\frac{n}{k+1}$.*

3. Analysis

We wish to pick p as small as possible but at the same time have a strong concentration of the estimate around its expected value. How small can p be? In Section 3.1 we present a second moment argument which gives a sufficient condition for picking

p. Our main theoretical result, stated as Theorem 3 in Section 3.2, provides a sufficient condition to this question. In Section 3.3 we analyze the complexity of our method. Finally, in Section 3.4 we discuss several aspects of our work.

3.1. **Second Moment Method.** Using the second moment method we are able to obtain the following strong theoretical guarantee:

Theorem 2. *Let n, t, Δ, T denote the number of vertices in G, the number of triangles in G, the maximum number of triangles an edge of G is contained in and the number of monochromatic triangles in the randomly colored graph respectively. Also let $N = \frac{1}{p}$ the number of colors used. If $p \geq \max\left(\frac{\Delta \log n}{t}, \sqrt{\frac{\log n}{t}}\right)$, then $T \sim \mathbb{E}\left[T\right]$ with probability $1 - \frac{1}{\log n}$.*

Proof. By Chebyshev's inequality, if $\mathbb{V}ar\left[T\right] = o(\mathbb{E}\left[T\right]^2)$ then $T \sim \mathbb{E}\left[T\right]$ with probability $1 - o(1)$ [3]. Let X_i be a random variable for the i-th triangle, $i = 1, \ldots, t$, such that $X_i = 1$ if the i-th triangle is monochromatic. The number of monochromatic triangles T is equal to the sum of these indicator variables, i.e., $T = \sum_{i=1}^{t} X_i$. By the linearity of expectation and by the fact that $\mathbf{Pr}\left[X_i = 1\right] = p^2$ we obtain that $\mathbb{E}\left[T\right] = p^2 t$. It is easy to check that the only case where two indicator variables are dependent is when they share an edge. In this case the covariance is non-zero and for any $p > 0$, $\mathbb{C}ov\left[X_i, X_j\right] = p^3 - p^4 < p^3$. We write $i \sim j$ if and only if X_i, X_j are dependent.

We obtain the following upper bound on the variance of T, where δ_e is the number of triangles edge e is contained in and $\Delta = \max_{e \in E(G)} \delta_e$:

$$\mathbb{V}ar\left[T\right] \leq \mathbb{E}\left[T\right] + \sum_{i \sim j} \mathbb{C}ov\left[X_i \wedge X_j\right] < p^2 t + p^3 \sum_e \delta_e^2 \leq p^2 t + 3p^3 t \Delta$$

We pick p large enough to obtain $\mathbb{V}ar\left[T\right] = o(\mathbb{E}\left[T\right]^2)$. It suffices:

$$p^4 t^2 \gg p^2 t + 3p^3 t \Delta \Rightarrow p^2 t \gg 1 + 3p\Delta \tag{1}$$

We consider two cases, determined by which of the two terms of the right hand side is larger:

• CASE 1 $(p\Delta < 1/3)$:

Since the right hand side of Inequality (1) is constant, it suffices that $p^2 t = \omega(n)$ where $\omega(n)$ is some slowly growing function. We pick $\omega(n) = \log n$ and hence $p \geq \sqrt{\frac{\log n}{t}}$.

• CASE 2 $(p\Delta \geq 1/3)$:

In this case the right hand side of Inequality (1) is $\Theta(p\Delta)$ and therefore it suffices to pick $\frac{pt}{\Delta} = \log n$.

Combining the above two cases we get that if

$$p \geq \max\left(\frac{\Delta \log n}{t}, \sqrt{\frac{\log n}{t}}\right)$$

inequality (1) is satisfied and hence by Chebyshev's inequality $T \sim \mathbb{E}\left[T\right]$ with probability $1 - \frac{1}{\log n}$.
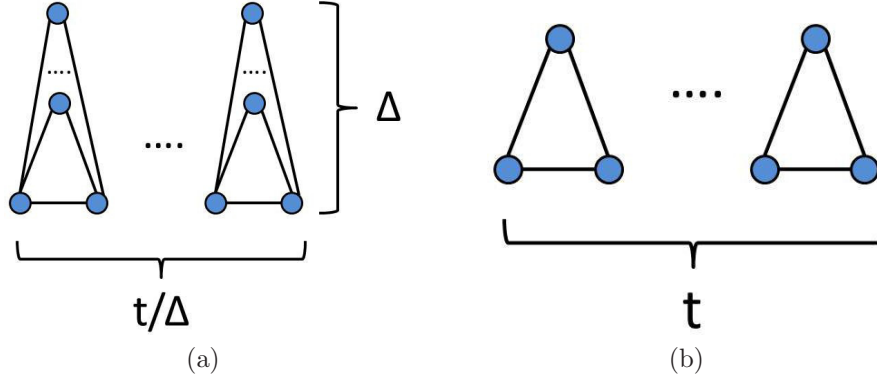
$\square$

Figure 1: Conditions of Theorem 2 are tight. In order to hope for concentration p has to be greater than (a) $\frac{\Delta}{t}$ and (b) $t^{-1/2}$.

3.1.1. *Extremal Cases and Tightness of Theorem 2.* Given the assumptions of Theorem 2, is the condition on p tight? The answer is affirmative as shown in Figure 1. Specifically, in Figure 1(a) G consists of t/Δ "books" of triangles, each of size Δ. This shows that p has to be at least $\omega(n)\frac{\Delta}{t}$ to hope for concentration, where $\omega(n)$ is some growing function of n. Similarly, when G consists of t disjoint triangles as shown in Figure 1(b) p has to be at least $\omega(n)t^{-1/2}$. Therefore, unless we know more about G, we cannot hope for milder conditions on p, i.e., Theorem 2 provides an optimal condition on p.

3.2. **Concentration via the Hajnal-Szemerédi Theorem.** Here, we present a different approach to obtaining concentration, based on partitioning the set of triangles/indicator variables in sets containing many independent random indicator variables and then taking a union bound. Our theoretical result is the following theorem:

Theorem 3. *Let $t_{\max}$ be the maximum number of triangles a vertex v is contained in. Also, let n, t, p, T be defined as above and ϵ a small positive constant. If $p^2 \geq \frac{16t_{\max}\log n}{\epsilon^2 t}$, then $\mathbf{Pr}\left[|T - \mathbb{E}\left[T\right]| > \epsilon\mathbb{E}\left[T\right]\right] \leq n^{-1}$.*

Proof. Let X_i be defined as above, $i = 1, \ldots, t$. Construct an auxiliary graph H as follows: add a vertex in H for every triangle in G and connect two vertices representing triangles t_1 and t_2 if and only if they have a common vertex. The maximum degree of H is $3t_{\max} = O(\delta^2)$, where $\delta = O(n)$ is the maximum degree in the graph. Invoke the Hajnal-Szemerédi Theorem on H: we can partition the vertices of H (triangles of G) into sets $S_1, \ldots, S_q$ such that $|S_i| > \Omega(\frac{t}{t_{\max}})$ and $q = \Theta(t_{\max})$. Let $k = \frac{t}{t_{\max}}$. Note that the set of indicator variables X_i corresponding to any set S_j is independent. Applying the Chernoff bound for each set $S_i, i = 1, \ldots, q$ we obtain

$$Pr\left[|\frac{1}{k}\sum_{i=1}^{k}X_i - p^2| > \epsilon p^2\right] \leq 2e^{-\epsilon^2 p^2 k/2}$$

.

If $p^2 k\epsilon^2 \geq 4d'\log n$, then $2e^{-\epsilon^2 p^2 k/2}$ is upper bounded by $n^{-d'}$, where $d' > 0$ is a constant. Since $q = O(n^3)$ by taking a union bound over all sets S_i we see that the

triangle count is approximated within a factor of ϵ with probability at least $1 - n^{3-d'}$. Setting $d' = 4$ completes the proof. □

It's worth noting that for any constant $K > 0$ the above proof gives that if $p^2 \geq \frac{4(K+3)t_{\max} \log n}{\epsilon^2 t}$ then $\mathbf{Pr}\left[|T - \mathbb{E}[T]| > \epsilon \mathbb{E}[T]\right] \leq n^{-K}$.

3.3. **Complexity.** The running time of our procedure of course depends on the subroutine we use on the second step, i.e., to count triangles in the edge set E'. Let $\deg(i)$ denote the degree of vertex i. Assuming we use an exact method that examines each vertex independently and counts the number of edges among its neighbors (a.k.a. Node Iterator method [21]) our algorithm runs in $O(n + m + p^2 \sum_{i \in [n]} \deg^2(i))$ expected time [1] by efficiently storing the graph and retrieving the neighbors of v colored with the same color as v in $O(1 + p \deg(v))$ expected time. Note that this implies that the speedup with respect to the counting task is $1/p^2$.

3.4. **Discussion.** The use of Hajnal-Szemerédi Theorem in the context of proving concentration is not new, e.g., [13, 18]. Despite the fact that the second moment argument gave us strong conditions on p, the use of Hajnal-Szemerédi has the potential of improving the Δ factor. The condition we provide on p is *sufficient* to obtain concentration. Note –see Figure 2– that it was necessary to partition the triangles into vertex disjoint rather than edge disjoint triangles since we need mutually independent variables per chromatic class in order to apply the Chernoff bound. If we were able to remove the dependencies in the chromatic classes defined by edge disjoint triangles, then the overall result could probably be improved. It's worth noting that for $p = 1$ we obtain that $t \geq n\omega(n)$, where $\omega(n)$ is any slowly growing function of n. This is –to the best of our knowledge– the mildest condition on the triangle density needed for a randomized algorithm to obtain concentration. Finally, notice that when $t \leq \frac{\Delta^2 \log n}{t_{\max}}$ and $t_{\max} \geq 1$ Theorem 3 yields a better bound than Theorem 2. The same holds when $t > \Delta^2 \log n$ and $t_{\max} \leq 1$. The latter scenario is far more restrictive and both Theorem 2 and Theorem 3 give for instance the same bound $p \geq \sqrt{\frac{\log n}{t}}$ for the graph of Figure 1b.

Furthermore, the powerful theorem of Kim and Vu [16, 28] that was used in previous work [25] is not immediately applicable here: let Y_e be an indicator variable for each edge e such that $Y_e = 1$ *if and only if e is monochromatic*, i.e., both its endpoints receive the same color. Note that the number of triangles is a boolean polynomial $T = \frac{1}{3} \sum_{\Delta(e,f,g)} \left(Y_e Y_f + Y_f Y_g + Y_e Y_g\right)$ but the boolean variables are not independent as the Kim-Vu [16] theorem requires. It's worth noting that the degree of the polynomial is two. Essentially, this is the reason for which our method obtains better results than existing work [25] where the degree of the multivariate polynomial is three [17, 25]. It's worth noting that previous work [18, 25] sampled edges independently whereas our new method samples subsets of vertices but in a careful manner in order to decrease the degree of the multivariate polynomial. Finally, it's worth noting that using a simple doubling procedure [25] and the median boosting trick of Jerrum, Valiant and Vazirani [14] we can pick p effectively in practice despite the fact that it depends on the quantity t which we want to estimate by introducing an extra logarithm in the running time.

---

[1]We assume that uniform sampling of a color takes constant time. If not, then we obtain the term $O(n \log\left(\frac{1}{p}\right)$ for the vertex coloring procedure.
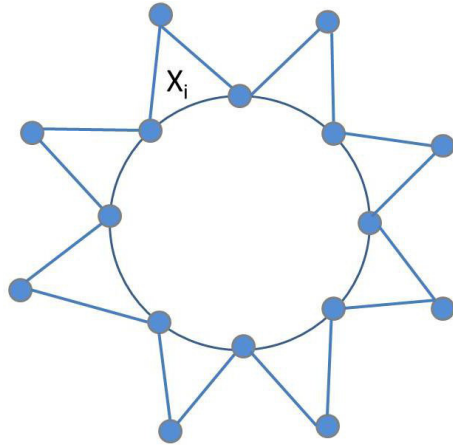
Figure 2: Consider the indicator variable X_i corresponding to the i-th triangle. Note that $\mathbf{Pr}\left[X_i|\text{rest are monochromatic}\right] = p \neq \mathbf{Pr}\left[X_i\right] = p^2$. The indicator variables are *pairwise* but not mutually independent.

Our main contribution in this work is the construction of a *triangle sparsifier* [25], i.e., a sparse graph H whose number of triangles approximates (upon appropriate scaling) the number of triangles in the input graph G. This construction –due to the mildest condition on the triangle density it requires– allows us to count triangles reliably and fast in large graphs where other methods fail. For instance, naive sampling [21] fails when $t = o(n^2)$ which is typically the case, see also Table 1. Our method improves significantly previous work [25] via the correlated sampling we introduce. It's important to point out that one may use an exact counting method such as the node iterator to count the number of triangles in H but our method can also be combined with other sampling methods, see [18] in order to obtain better performance both theoretically and practically.

Finally, from an experimentation point of view, it is interesting to see how well the upper bound $3\Delta t$ matches the sum $\sum_{e \in E(G)} \delta_e^2$, where δ_e is the number of triangles edge e is contained in, and the typical values for Δ and $t_{\max}$ in real-world graphs. The following table shows these numbers for five graphs [2] taken from the SNAP library [1]. We see that Δ and $t_{\max}$ are significantly less than their upperbounds and that typically $3\Delta t$ is significantly larger than $\sum_{e \in E(G)} \delta_e^2$ except for the collaboration network of Arxiv Astro Physics. The results are shown in Table 1.

4. A MapReduce Implementation

MapReduce [11] has become the *de facto* standard in academia and industry for analyzing large scale networks. Recent work by Suri and Vassilvitskii [22] proposes two algorithms for counting triangles. The first is an efficient MapReduce implementation of the Node Iterator algorithm, see also [21] and the second is based on partitioning the graph into overlapping subsets so that each triangle is present in at least one of the subsets.

---

[2]AS:Autonomous Systems, Oregon: Oregon route views, Enron: Email communication network, ca-HepPh and AstroPh:Collaboration networks. Self-edges were removed.

Name	Nodes(n)	Edges(m)	Triangle Count(t)	Δ	$t_{\max}$	$\sum_{e\in E(G)} \delta_e^2$	$3\Delta t$
AS	7,716	12,572	6,584	344	2,047	595,632	6,794,688
Oregon	11,492	23,409	19,894	537	3,638	2,347,560	32,049,234
Enron	36,692	183,831	727,044	420	17,744	75,237,684	916,075,440
ca-HepPh	12,008	118,489	3,358,499	450	39,633	1.8839×10^9	4.534×10^9
AstroPh	18,772	198,050	1,351,441	350	11,269	148,765,753	1.419×10^9

Table 1: Values for the variables involved in our formulae for five real-world graphs. Typically, Δ and $t_{\max}$ are significantly less than the obvious upper bounds $n-2$ and $\binom{n-1}{2}$ respectively. Furthermore, $3\Delta t$ is significantly larger than $\sum_{e\in E(G)} \delta_e^2$.

Algorithm 2 MAPREDUCE Colorful Triangle Counting $G(V, E), p = 1/N$

Map: Input $\langle e = (u, f(u), v, f(v)); 1 \rangle$ {Let f be a uniformly at random coloring of the vertices with N colors}
if $f(u) = f(v)$ then emit $\langle f(u); (u, v) \rangle$
Reduce: Input $\langle c; E_c = \{(u, v)\} \subseteq E \rangle$ { Every edge $(u, v) \in E_c$ has color c, i.e., $f(u) = f(v)$}
Scale each triangle by $\frac{1}{p^2}$.

Our method is amenable to being implemented in MAPREDUCE and the skeleton of such an implementation is shown in Algorithm 2[3]. We implicitly assume that in a first round vertices have received a color uniformly at random from the N available colors and that we have the coloring information for the endpoints of each edge. Each mapper receives an edge together with the colors of its edgepoints. If the edge is monochromatic, then it's emitted with the color as the key and the edge as the value. Edges with the same color are shipped to the same reducer where locally a triangle counting algorithm is applied. The total count is scaled appropriately. Trivially, the following lemma holds by the linearity of expectation and the fact that the endpoints of any edge receive a *given* color c with probability p^2.

Lemma 2. *The expected size to any reduce instance is $O(p^2 m)$ and the expected total space used at the end of the map phase is $O(pm)$.*

5. CONCLUSIONS

In this note we introduced a new randomized algorithm for approximate triangle counting, which is implemented easily in parallel. We showed such an implementation in the popular MAPREDUCE programming framework. The key idea that improves the existing work is that by our new sampling method the degree of the multivariate polynomial expressing the number of triangles decreases by one, compared to previous work, e.g., [17, 25]. We used the Second Moment Method and a powerful result of Hajnal-Szemerédi Theorem to obtain strong concentration results.

---

[3]It's worth pointing out for completeness reasons that in practice one would not scale the triangles after the first reduce. It would emit the count of monochromatic triangles which would be summed up in a second round and scaled by $1/p^2$.

We observe that our result extends to any subset of triangles satisfying some predicate (e.g., containing a certain vertex), in the sense that counting such triangles in the sample leads to a concentrated estimate of the number in the original graph.

In future work we plan to investigate sampling methods for counting triangles in weighted graphs, other types of subgraphs and several systems-oriented aspects of our work.

6. Acknowledgments

References

[1] SNAP Stanford Library http://snap.stanford.edu/

[2] Achlioptas, D., Mcsherry, F.: *Fast computation of low-rank matrix approximations.* Journal of the ACM, Vol. 54 (2007)

[3] Alon, N., Spencer, J.L *The Probabilistic Method* Wiley-Interscience, 3rd Edition, 2008.

[4] Alon, N., Yuster, R., Zwick, U.: *Finding and Counting Given Length Cycles.* Algorithmica, Volume 17, Number 3, 209–223, 1997.

[5] Avron, H.: *Counting triangles in large graphs using randomized matrix trace estimation* Proceedings of KDD-LDMTA'10, 2010.

[6] Bar-Yosseff, Z., Kumar, R., Sivakumar, D.: *Reductions in streaming algorithms, with an application to counting triangles in graphs.* ACM-SIAM Symposium on Discrete Algorithms (SODA'02), 2002.

[7] Becchetti, L., Boldi, P., Castillo, C., Gionis, A.: *Efficient Semi-Streaming Algorithms for Local Triangle Counting in Massive Graphs.* The 14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '08), 2008.

[8] Buriol, L., Frahling, G., Leonardi, S., Marchetti-Spaccamela, A., Sohler, C.: *Counting Triangles in Data Streams.* Symposium on Principles of Database Systems (PODS'06), 2006.

[9] Chernoff, H.: *A Note on an Inequality Involving the Normal Distribution* Annals of Probability, Volume 9, Number 3, pp. 533-535, 1981.

[10] Coppersmith D., Winograd S.: *Matrix multiplication via arithmetic progressions.* In ACM Symposium on Theory of Computing (STOC '87), 1987.

[11] Dean, J., Ghemawat, S.: *Simplified data processing on large clusters.* In Proceedings of OSDI, pp. 137-150, 2004.

[12] Hajnal, A. and Szemerédi, E.: *Proof of a Conjecture of Erdös.* In Combinatorial Theory and Its Applications, Vol. 2 (Ed. P. Erdös, A. Rényi, and V. T. Sós), pp. 601-623, 1970.

[13] Janson, S., Ruciński, A.: *The Infamous Upper Tail* Random Structures and Algorithms, **20**, pp. 317-342, 2002.

[14] Jerrum, M., Valiant, L., Vazirani, V.: *Random generation of cominatorial structures from a uniform distribution* Theoretical Computer Science, **43**(2-3), pp.169-188, 1986.

[15] Jowhari, H., Ghodsi, M.: *New Streaming Algorithms for Counting Triangles in Graphs.* Computing and Combinatorics (COCOON '05), 2005.

[16] Kim, J. H. and Vu, V. H. *Concentration of multivariate polynomials and its applications.* Combinatorica, **20**(3), pp. 417-434, 2000.

[17] Kim, J. H. and Vu, V. H.: *Divide and conquer martingales and the number of triangles in a random graph.* Journal of Random Structures and Algorithms, **24**(2), pp. 166-174, 2004.

[18] Kolountzakis, M.N., Miller, G.L., Peng, R., Tsourakakis, C.E.: *Efficient Triangle Counting in Large Graphs via Degree-based Vertex Partitioning* Internet Mathematics (to appear)

[19] Latapy, M.: *Main-memory Triangle Computations for Very Large (Sparse (Power-Law)) Graphs* Theoretical Computer Science (TCS) 407 (1-3), pages 458-473, 2008

[20] Papadimitriou, C., Yannakakis, M.: *The clique problem for planar graphs.* Information Processing Letters, 13, pp. 131–133, 1981.

[21] Schank, T., Wagner, D.: *Approximating Clustering Coefficient and Transitivity.* Journal of Graph Algorithms and Applications, 9, 265–275, 2005.
[22] Suri, S., Vassilvitskii, V.: *Counting Triangles and the Curse of the Last Reducer.* To appear in the 20th World Wide Web Conference (WWW '11), 2011.
[23] Tsourakakis, C.E.: *Fast Counting of Triangles in Large Real Networks, without counting: Algorithms and Laws.* IEEE International Conference on Data Mining (ICDM'08), 2008.
[24] Tsourakakis, C.E.: *MACH: Fast Randomized Tensor Decompositions* SIAM International Conference on Data Mining (2010)
[25] Tsourakakis, Kolountzakis, M.N., Miller, G.L.: *Triangle Sparsifiers* Journal of Graph Theory and Applications, to appear (available upon request).
[26] Tsourakakis, C.E., Kang, U, Miller, G.L., Faloutsos, C.: *Doulion: Counting Triangles in Massive Graphs with a Coin.* The 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '09), 2009.
[27] Tsourakakis, C.E., Drineas, P., Michelakis, E., Koutis, I., Faloutsos, C.: *Spectral Counting of Triangles via Element-Wise Sparsification.* Advances in Social Networks Analysis and Mining (2010)
[28] Vu, V.H.: *On the concentration of multivariate polynomials with small expectation.* Random Structures and Algorithms **16**, 4, 344–363, 2000.

IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 København S, Denmark
   E-mail address: pagh@itu.dk

Department of Mathematical Sciences, Carnegie Mellon University, 5000 Forbes Av., 15213, Pittsburgh, PA, U.S.A
   E-mail address: ctsourak@math.cmu.edu