*Internet Mathematics*, 11:555–571, 2015
Copyright © 2015 Sandia Corporation
ISSN: 1542-7951 print/1944-9488 online
DOI: 10.1080/15427951.2015.1037030

# WHY DO SIMPLE ALGORITHMS FOR TRIANGLE ENUMERATION WORK IN THE REAL WORLD?

**Jonathan W. Berry,[1] Luke A. Fostvedt,[2] Daniel J. Nordman,[1] Cynthia A. Phillips,[3] C. Seshadhri,[4] and Alyson G. Wilson[5]**

*[1]Sandia National Laboratories, Albuquerque, New Mexico, USA*
*[2]Iowa State University, Ames, Iowa, USA*
*[3]Sandia National Laboratories, Albuquerque, New Mexico, USA*
*[4]Sandia National Laboratories, Livermore, California, USA*
*[5]North Carolina State University, Raleigh, North Carolina, USA*

**Abstract** *Listing all triangles is a fundamental graph operation. Triangles can have important interpretations in real-world graphs, especially social and other interaction networks. Despite the lack of provably efficient (linear, or slightly super linear) worst-case algorithms for this problem, practitioners run simple, efficient heuristics to find all triangles in graphs with millions of vertices. How are these heuristics exploiting the structure of these special graphs to provide major speedups in running time?*

*We study one of the most prevalent algorithms used by practitioners. A trivial algorithm enumerates all paths of length* 2*, and checks if each such path is incident to a triangle. A good heuristic is to enumerate only those paths of length* 2 *in which the middle vertex has the lowest degree. It is easily implemented and is empirically known to give remarkable speedups over the trivial algorithm.*

*We study the behavior of this algorithm over graphs with heavy-tailed degree distributions, a defining feature of real-world graphs. The erased configuration model (ECM) efficiently generates a graph with asymptotically (almost) any desired degree sequence. We show that the expected running time of this algorithm over the distribution of graphs created by the ECM is controlled by the $\ell_{4/3}$-norm of the degree sequence. Norms of the degree sequence are a measure of the heaviness of the tail, and it is precisely this feature that allows non trivial speedups of simple triangle enumeration algorithms. As a corollary of our main theorem, we prove expected linear-time performance for degree sequences following a power law with exponent $\alpha \geq 7/3$, and non trivial speedup whenever $\alpha \in (2, 3)$.*

## 1. INTRODUCTION

Finding triangles in graphs is a classic theoretical problem with numerous practical applications. The recent explosion of work on social networks has led to a great interest in

An extended abstract of this paper appeared in *Innovations in Theoretical Computer Science* (ITCS), 1407:1116 (2014), 225–234.

This manuscript has been authored by Sandia Corporation under Contract No. DE-AC04-94AL85000 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

Address correspondence to C. Seshadhri, University of California, Santa Cruz, 1156 High Street, Baskin School of Engineering, Engineering 2, Mail Stop: 310, Santa Cruz, CA 95064, USA. Email: csesha@gmail.com

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uinm.

using algorithms that can find triangles in graphs quickly. The social sciences and physics communities often study triangles in real networks and use them to reason about underlying social processes [18, 31, 37, 11, 12, 20]. Much of the information about triangles in these four studies is determined by a complete enumeration of all triangles in a (small) graph. Triangle enumeration is also a fundamental subroutine for other, more complex, algorithmic tasks [7, 21].

From a theoretical perspective, Itai and Rodeh [22] gave algorithms for triangle finding in $O(n^{\omega})$ time (where $n$ is the number of vertices and $\omega$ is the matrix multiplication constant) using fast matrix multiplication. [36] Deep connections between matrix multiplication and (edge-weighted) triangle enumeration are shown in [36]. But much of this work is focused on dense graphs. Practitioners usually deal with massive sparse graphs with large variance in degrees, from which subquadratic time algorithms can be trivially obtained but are still too slow to run.

Practioners enumerate triangles on massive graphs (with millions of vertices) using fairly simple heuristics, which are often easily parallelizable. This work is motivated by the following question: *Can we theoretically explain why simple algorithms for triangle enumeration work in the real world?*

Consider a trivial algorithm. Take an undirected graph with $n$ vertices, $m$ edges, and degree sequence $d_1, d_2, \dots, d_n$ (so the degree of vertex $v$ is $d_v$). Call a path of length 2 ($P_2$) *closed* if it participates in a triangle and *open* otherwise. Simply enumerate all $P_2$s and output the closed ones. The total running time is $\Theta(\sum_v d_v^2)$ (assume that checking if a $P_2$ is closed can be done in constant time), because every $P_2$ involves a pair of neighbors for the middle vertex. We will henceforth refer to this as the trivial algorithm. A simple heuristic is to enumerate only those paths in which the middle vertex has the lowest degree of the 3 vertices in the path. We denote this algorithm by MINBUCKET .

1. Create $n$ empty buckets $B_1, B_2, \dots, B_n$.
2. For each edge $(u, v)$: if $d_u \leq d_v$, place it in $B_u$, otherwise place it in $B_v$. Break ties consistently.
3. For each bucket $B_v$: iterate over all $P_2$s formed by edges in $B_v$ and output those that are closed.

MINBUCKET is quite common in practice (sometimes taking the somewhat strange name *nodeIterator++*) and has clean parallel implementations with no load balancing issues [34, 17, 33]. For such simple algorithms, the total work pretty much determines the parallel runtime. For example, it would take $n$ processors with perfect speed up running a $\Theta(n^2)$-work algorithm to compete with a single processor running a $\Theta(n)$-work algorithm.

MINBUCKET is often the algorithm of choice for triangle enumeration because of its simplicity and because it beats the trivial algorithm by orders of magnitude, as shown in the previous citations. (A quick check shows at least 60 citations to [17], most cite involving papers that deal with massive scale graph algorithms.) The algorithm itself has been discovered and rediscovered in various forms over the past decades. The earliest reference the authors could find was from the mid-80s when a sequential version of the algorithm was devised [13]. We provide a more detailed history in later pages.

Nonetheless, MINBUCKET has a poor worst-case behavior. It would perform terribly on a high-degree regular bipartite graph. If the input sparse graph (with high variance in degree) simply consisted of many such bipartite graphs of varying sizes, MinBucket would perform no better than its trivial cousin. Then why is it good in practice?

### 1.1. Results

Since the seminal results of Barabási and Albert [3, 19, 10], researchers have assumed that massive graphs obtained from the real world have *heavy-tailed degree distributions* (often approximated as a power law). The average degree is thought to be a constant (or very slowly growing), but the variance is quite large. The usual approximation is to think of the number of vertices of degree $d$ as decaying roughly as $1/d^{\alpha}$ for some small constant $\alpha$.

This seems to have connections with MinBucket . If edges tend to connect vertices of fairly disparate degrees (quite likely in a graph with large variance in degrees), MinBucket might provably give good running times. This is exactly what we set out to prove for a natural distribution on heavy-tailed graphs.

Consider any list of positive integers $\mathbf{d} = (d_1, d_2, \dots, d_n)$, which we think of as a "desired" degree sequence. In other words, we wish to construct a graph on $n$ vertices where vertex $v \in [n]$ has degree $d_v$. The *configuration model* (CM) [4, 8, 26, 28] aims to create a random (multi)graph for this purpose. Imagine vertex $v$ being incident to $d_v$ "stubs," which can be thought of as half-edges. We take a random perfect matching between the stubs, so pairs of stubs are matched to each other. Each such pair creates an edge, and the result is a multigraph with the desired degree sequence. Usually, this is converted to a simple graph by removing parallel edges and self-loops [9]. We refer to this graph distribution as $ECM(\mathbf{d})$, for input degree sequence $\mathbf{d}$. This model has a fairly long history (which we relegate to a later section) and is a standard method to construct a graph with a desired degree sequence. It is closely connected to other models [15, 16, 24], in the context of eigenvalues of graphs with a given degree sequence. These models simply connect vertices $u$ and $v$ independently with probability proportional to the degree product $d_u d_v$, similarly to the Erdős–Rényi construction.

Our main theorem gives a bound on the expected running time of MinBucket for $ECM(\mathbf{d})$. We set $m = (\sum_v d_v)/2$. We will, henceforth, assume that $0 < d_1 \leq d_2 \cdots \leq d_n$ and that $d_n < \sqrt{m}/2$. This "truncation" is a standard assumption for analysis of the configuration model [26, 15, 24, 16, 28, 9]. We use $\sum_v$ as a shorthand for $\sum_{i=1}^{n}$, since it is a sum over all vertices. The runtime bottleneck for MinBucket is in $P_2$ enumeration, and checking whether a $P_2$ is closed is often assumed to be a constant time operation. Henceforth, when we say "running time," we mean the number of $P_2$s enumerated.

**Theorem 1.1.** *Consider a degree sequence* $\mathbf{d} = (d_1, d_2, \dots, d_n)$, *where* $m = (\sum_v d_v)/2$ *and* $d_n < \sqrt{m}/2$. *The expected (over* $ECM(\mathbf{d})$*) number of* $P_2$*s enumerated by* MinBucket *is* $O(n + m^{-2}(\sum_v d_v^{4/3})^3)$.

**Remark:** Although we focus our exposition on the ECM model, our main theorem applies to other random graph models, such as Chung–Lu graphs in particular; details are given in [6].

Before we actually make sense of this theorem, let us look at a corollary of this theorem. It has been repeatedly observed that degree sequences in real graphs have heavy tails, often approximated as *power laws* [3]. Power laws say something about the moments of the degree distribution (equivalently, norms of the degree sequence). Because it does not affect our main theorem or corollary, we choose a fairly loose definition of power law. This is a binned version of the usual definition, which states that the number of vertices of degree $d$ is proportional to $n/d^{\alpha}$. (Even up to constants, this is never precisely true because there are many gaps in real degree sequences.)

**Definition 1.2.** A degree sequence $\mathbf{d}$ satisfies a power law of exponent $\alpha > 1$ if the following holds for all $k \leq \log_2 d_n - 1$: for $d = 2^k$, the number of sequence terms in $[d, 2d]$ is $\Theta(n/d^{\alpha-1})$.

The following shows an application of our theorem for common values of $\alpha$. This bound is tight, as we show in Section 5. (When $\alpha > 3$, the trivial algorithm runs in linear time because $\sum_v d_v^2 = O(n)$.)

**Corollary 1.3.** *Suppose a degree sequence* $\mathbf{d}$ *(with largest term* $< \sqrt{m}/2$*) satisfies a power law with exponent* $\alpha \in (2, 3)$. *Then the expected running time of* MinBucket *of* $ECM(\mathbf{d})$ *is asymptotically better than the trivial algorithm, and is linear when* $\alpha > 7/3$.

### 1.2. Making sense of Theorem 1.1

First, as a sanity check, let us actually show that Theorem 1.1 beats the trivial bound, $\sum_v d_v^2$. This is a direct application of Hölder's inequality for conjugates $p = 3$ and $q = 3/2$.

$$\left(\sum_v d_v^{4/3}\right)^3 = \left(\sum_v d_v^{2/3} \cdot d_v^{2/3}\right)^3 \leq \left(\sum_v d_v^{\frac{2}{3}\cdot 3}\right)^{3\cdot\frac{1}{3}} \left(\sum_v d_v^{\frac{2}{3}\cdot\frac{3}{2}}\right)^{3\cdot\frac{2}{3}} = (2m)^2 \left(\sum_v d_v^2\right)$$

Rearranging, we get $m^{-2}(\sum_v d_v^{4/3})^3 = O(\sum_v d_v^2)$, showing that our bound at least holds the promise of being nontrivial.

Consider the uniform distribution on the vertices. As $m \geq n/2$ by assumption, we can write our running time bound as $n(\mathbf{E}[d_v^{4/3}])^3$, opposed to the trivial bound of $\sum_v d_v^2 = n\mathbf{E}[d_v^2]$. If the degree "distribution" (think of the random variable given by the degree of a uniform random vertex) has a small 4/3-moment, the running time is small. This can happen even though the second moment is large, and this is where MinBucket beats the trivial algorithm. In other words, if the tail of the degree sequence is heavy, but not too heavy, MinBucket will perform well.

And this is exactly what happens when $\alpha > 2$ for power-law degree sequences. When $\alpha > 7/3$, the 4/3-moment becomes constant and the running time is linear. (It is known that for ECM graphs over power-law degree sequences with $\alpha > 7/3$, the clustering coefficient [ratio of triangles to $P_2$s] converges to zero [28].) We show in Section 5 that the running time bound achieved in the following corollary, for power laws with $\alpha > 2$, is tight. When $\alpha \leq 2$, we know that MinBucket must be at least as fast as the trivial algorithm Section 5, but no theoretical guarantees exist on the basis of Theorem 1.1 that MinBucket can provide asymptotic improvements over the trivial algorithm. Intuitively, in the case $\alpha \leq 2$ of extreme heavy-tailed degree distributions, a significant proportion of nodes are expected to have very large degrees, which reduces the effectiveness of the MinBucket strategy of enumerating triangles through vertices of low degree in a 3-vertex path. For convenience, we use notation $A \lessdot B$ to denote $A = O(B)$ and use $A \doteqdot B$ to denote same order $A = \Theta(B)$.

*Proof of Corollary 1.3*. First, let us understand the trivial bound. Remember that $d_n$ is the maximum degree.

$$\sum_v d_v^2 \div \sum_{k=1}^{\log_2 d_n - 1} (n/2^{k(\alpha-1)})2^{2k} = n \sum_{k=1}^{\log_2 d_n - 1} 2^{k(3-\alpha)} \div n + nd_n^{3-\alpha},$$

if $\alpha \neq 3$ (for $\alpha = 3$, the bound is $n + n\log_2 d_n$). We can argue (Claim 3.3) that the expected number of wedges enumerated by the trivial algorithm is $\Omega(\sum_v d_v^2)$. Now, for the bound of Theorem 1.1.

$$m^{-2}\left(\sum_v d_v^{4/3}\right)^3 \lessdot n^{-2}\left(\sum_{k=1}^{\log_2 d_n - 1} (n/2^{k(\alpha-1)})2^{4k/3}\right)^3 = n\left(\sum_{k=1}^{\log_2 d_n - 1} 2^{k(7/3-\alpha)}\right)^3 \lessdot n + nd_n^{7-3\alpha}$$

when $d_n \neq 7/3$ (for $\alpha = 7/3$, the bound is $n + n\log_2 d_n$). Regardless of $d_n$, if $\alpha > 7/3$, then the running time of MinBucket is linear. Whenever $\alpha \in (2, 3)$, $7 - 3\alpha < 3 - \alpha$, and MinBucket is asymptotically faster than a trivial enumeration. □

### 1.3. Significance of Theorem 1.1

Theorem 1.1 connects the running time of a commonly used algorithm to the norms of the degree sequences, a well-studied property of real graphs. So this important property of heavy-tails in real graphs allows for the algorithmic benefit of MinBucket . We have discovered that for a fairly standard graph model inspired by real degree distributions, MinBucket is very efficient.

We think of this theorem as a proof of concept: theoretically showing that a common property of real-world inputs allows for the efficient performance of a simple heuristic. Because of our distributional assumptions as well as bounds on $\alpha$, we agree with the (skeptical) reader that this does not fully explain why MinBucket works in the real world.[1] Nonetheless, we believe that we are making progress toward that, especially for a question that is quite difficult to formalize. After all, there is hardly any consensus, in the social networks community, on what real graphs look like.

But the notion that distinctive properties of real-world graphs can be used to prove efficiency of simple algorithms is a useful way of thinking. This is one direction to follow for going beyond worst-case analysis. Our aim, here, is not to design better algorithms for triangle enumeration, but to give a theoretical argument for why current algorithms do well.

The proof is obtained (as expected) through various probabilistic arguments bounding the sizes of the different buckets. The ECM, although easy to implement and clean to define, creates some niggling problems for analysis of algorithms. The edges are not truly independent of each other, and we have to take care of these weak dependencies.

Why the 4/3-norm? Indeed, that is one of the most surprising features of this result (especially since the bound is tight for power laws of $\alpha > 2$). As we bound the buckets sizes and make sense of the various expressions, the total running time is expressed as a sum of various degree terms. Using appropriate approximations, it tends to rearrange into norms of the degree sequence. Our proof goes over two sections. In Section 3, we give various probabilistic calculations for the degree behavior in [32, (Table 1)], which set the

[1] As the astute reader would have noticed, our title is a question, not a statement.

stage for the run time accounting. In Section 4, we start with bounding bucket sizes and finally get to the 4/3-moment. In Section 5, we show that bounds achieved in the proof of Corollary 1.3 are tight. This is mostly a matter of using the tools of the previous sections. In Section 6.1, we give a tighter analysis, which gives an explicit expression for strong upper bounds on running time, and in Section 6 we experimentally show that these more careful bounds closely approximate the expected runtime of ECM graphs, with runtime constants under 1 for graphs up to 80 M nodes.

There is a related issue of how many triangles to expect in our random graph model as a function of the power-law slope. An answer can be provided that resembles those found in other studies with alternative random graph models having power-law degree distributions. For the ECM model, here, with a power-law slope $\alpha$, the expected number of triangles is $O(nd_n^{7-3\alpha})$ if $\alpha < 2$, $O(nd_n^{3-\alpha})$ if $2 < \alpha < 3$ and $O(1)$ if $\alpha > 3$. This is similar, for example, to findings of [32, (Table 1)] using a different model. That is, the expected number of triangles diverges in the range $2 < \alpha < 3$ commonly observed in real-world networks (e.g., at a rate involving $d_n^{3-\alpha}$) and converges for lighter-tail power laws with $\alpha > 3$.

## 2. RELATED WORK

The idea of using some sort of degree binning, orienting edges, or thresholding for finding and enumerating triangles has been used in many studies. Bounds for a sequential version of MinBucket using the degeneracy of a graph have been given by [13]. This does not give bounds for MinBucket, although their algorithm is similar in spirit. Using degree thresholding and matrix multiplication ideas from [22], [2] find triangles in $O(m^{1.41})$ Acyclic orientations for linear time triangle enumeration in planar graphs are used by [14]. As shown by [36], fast algorithms for weighted triangle enumeration lead to remarkable consequences, such as faster all-pairs shortest paths. In the work most closely related to this paper, [23] discusses various triangle finding algorithms, and also focuses on power-law graphs. He shows the trivial bound of $O(mn^{1/\alpha})$ when the power law exponent is $\alpha$. Essentially, the maximum degree is $n^{1/\alpha}$ and that directly gives a bound on the number of $P_2$s.

MinBucket has received attention from various experimental studies: [34] perform an experimental study of many algorithms, including a sequential version of MinBucket, which they show to be quite efficient; [17] specifically describes MinBucket in the context of Map-Reduce; [33] do many experiments on real graphs in Map-Reduce and show major speedups (a few orders of magnitude) for MinBucket over the trivial enumeration; and [35] gives a good survey of various methods used in practice for triangle counting and estimation.

Explicit triangle enumerations have been used for various applications on large graphs; [21] use triangle enumeration for a graph-based approach for solving systems of geometric constraints, and [7] touch every triangle as part of their community detection algorithm for large graphs.

Configuration models for generating random graphs with given degree sequences have a long history. Using this model to count graphs with a given degree sequence has been studied [4], as well as the connectivity of these graphs [38], and [25, 26] have studied various properties such as the largest connected component of this graph distribution. Physicists studying complex networks have also paid attention to this model [29], and [9] show that the simple graph generated by the ECM asymptotically matches the desired degree sequence. A model for power-law graphs, where edge $(u, v)$ is independently inserted with probability $d_u d_v/2m$ [1], was studied for more general degree sequences in subsequent work by [15, 16]; [24] independently discuss this model. Most of this work focuses on eigenvalues and average distances in these graphs. An excellent survey of these models, their similarities, and applications is given by [28].

## 3. DEGREE BEHAVIOR OF ECM(D)

We fix a degree sequence **d** and focus on the distribution ECM(**d**). All expectations and probabilities are over this distribution. Because of dependencies in the ECM we will need to formalize our arguments carefully. We first state a general lemma giving a one-sided tail bound for dependent random variables with special conditional properties. The proof is in Appendix A.

**Lemma 3.1.** *Let $Y_1, Y_2, \ldots, Y_k$ be independent random variables, and $X_i = f_i(Y_1, Y_2, \ldots, Y_i)$ be* 0–1 *random variables. Let $\alpha \in [0, 1]$. Suppose* $\Pr[X_1 = 1] \geq \alpha$ *and* $\Pr[X_i = 1|Y_1, Y_2, \ldots, Y_{i-1}] \geq \alpha$ *for all i. Then,* $\Pr[\sum_{i=1}^{k} X_i < \alpha k \delta] < \exp(-\alpha k(1-\delta)^2/2)$ *for any $\delta \in (0, 1)$.*

We now prove a tail bound on degrees of vertices; the probability that the degree of vertex $v$ deviates by a constant factor of $d_v$ is $\exp(-\Omega(d_v))$. Let $\beta, \beta', \delta, \delta'$ denote sufficiently small constants.

Before we proceed with our tail bounds, we describe a process to construct the random matching of stubs. We are interested in a particular vertex $v$. Order the stubs such that the $d_v$ $v$-stubs are in the beginning; the remaining stubs are ordered arbitrarily. We start with the first stub and match to a uniform random stub (other than itself). We then take the next unmatched stub, according to the order, and match to a uniform random unmatched stub. And so on and so forth. The final connections are clearly dependent, though the choice among unmatched stubs is done independently. This is formalized as follows. Let $Y_i$ be an independent uniform random integer in $[1, 2m - 2(i-1) - 1]$. This represents the choice at the $i$th step, since in the $i$th step, we have exactly $2m - 2(i-1) - 1$ choices. Imagine that we first draw these independent $Y_i$'s. Then we deterministically construct the matching on the basis of these numbers. (So the first stub is connected to the $Y_1$st stub, the second unmatched stub is connected to the $Y_2$nd unmatched stub, etc.)

**Lemma 3.2.** *Assume $d_n < \sqrt{m}/2$. Let $D_v$ be the random variable denoting the degree of $v$ in the resulting graph. There exist sufficiently small constants $\beta, \beta' \in (0, 1)$, such that* $\Pr[D_v < \beta' d_v] < \exp(-\beta d_v)$*, holding true even when allowing for any integer $d_v \geq 0$.*

**Proof.** Suppose $d_v > 1$. We again order the stubs so that the $d_v$ $v$-stubs are in the beginning. Let $X_j$ be the indicator random variable for the $j$th matching forming a new edge with $v$. Note that $\sum_{j=1}^{\lfloor d_v/2 \rfloor} X_j \leq D_v$. Observe that $X_j$ is a function of $Y_1, Y_2, \ldots, Y_j$. Consider any $Y_1, Y_2, \ldots, Y_{j-1}$ and suppose the matchings created by these variables link to vertices $v_0 = v, v_1, v_2, \ldots, v_{j-1}$ (distinct) such that there are $n_j$ links to vertex $v_j$ such that $\sum_{i=0}^{j-1} n_i = (j-1)$. Then, for $j = 1, \dots, \lfloor d_v/2 \rfloor$,

$$\begin{aligned}\mathbf{E}[X_j | Y_1, Y_2, \dots, Y_{j-1}] &\geq 1 - \frac{(d_v - j - n_0) + \sum_{1 \leq i \leq j-1; n_i \neq 0}(d_{v_i} - n_i)}{2m - 2(j-1) - 1} \\ &\geq 1 - \frac{-2(j-1) - 1 + \sum_{i=0}^{j-1} d_{v_i}}{2m - 2(j-1) - 1} \\ &\geq 1 - \frac{\sum_{i=0}^{j-1} d_{v_i}}{2m}.\end{aligned}$$

Note that $\sum_{i=0}^{j-1} d_{v_i} \leq (\sqrt{m}/2)^2 = m/4$ by the bound on the maximum degree, so we may bound $\mathbf{E}[X_j | Y_1, Y_2, \dots, Y_{j-1}] \geq 3/4$. By Lemma 3.1 (setting $\delta = 2/3$ and bounding $\alpha k > d_v/4$),

$$\begin{aligned}\Pr[D_v < d_v/8] \leq \Pr\left[\sum_{j=1}^{\lfloor d_v/2 \rfloor} X_j < d_v/8\right] &\leq \Pr\left[\sum_{j=1}^{\lfloor d_v/2 \rfloor} X_j < \lfloor d_v/2 \rfloor/2\right] \\ &< \exp(-d_v(1/3)^2/8).\end{aligned}$$

□

This suffices to prove the trivial bound for the trivial algorithm.

**Claim 3.3.** *The expected number of wedges enumerated by the trivial algorithm is* $\Omega(\sum_v d_v^2)$.

**Proof.** The expected number of wedges enumerated is $\Omega(\sum_v D_v^2)$, where $D_v$ is the actual degree of $v$. Using Lemma 3.2, $\mathbf{E}[D_v^2] = \Omega(d_v^2)$. □

We will need the following basic claim about the joint probability of two edges.

**Claim 3.4.** *Let $v$, $w$, $w'$ be three distinct vertices. The probability that edges $(v, w)$ and $(v, w')$ are present in the final graph is, at most,* $d_v^2 d_w d_{w'}/m^2$.

**Proof.** Assume $d_v > 1$. Let $C_{v,w}$ be the indicator random variable for edge $(v, w)$ being present (likewise, define $C_{v,w'}$). Label the stubs of each vertex as $s_1^v, \dots, s_{d_v}^v$; $s_1^w, \dots, s_{d_w}^w$; and $s_1^{w'}, \dots, s_{d_{w'}}^{w'}$. Let $C_{s_i^v, s_j^w}$ be the indicator random variable for the edge being present between stubs $s_i^v$ and $s_j^w$ (likewise, define $C_{s_i^v, s_j^{w'}}$). Then, the event $\{C_{v,w} C_{v,w'} = 1\}$ that edges $(v, w)$ and $(v, w')$ are present is a subset of the event $\cup_{1 \leq i \neq j \leq d_v} \cup_{k=1}^{d_w} \cup_{\ell=1}^{d_{w'}} \{C_{s_i^v, s_k^w} C_{s_j^v, s_\ell^{w'}} = 1\}$. Hence,

$$\Pr[C_{v,w} C_{v,w'} = 1] \leq \sum_{1 \leq i \neq j \leq d_v} \sum_{k=1}^{d_w} \sum_{\ell=1}^{d_{w'}} \Pr[C_{s_i^v, s_k^w} C_{s_j^v, s_\ell^{w'}} = 1].$$

Fix $1 \leq i \neq j \leq d_v$, $1 \leq k \leq d_w$ and $1 \leq \ell \leq d_{w'}$ and order stubs $s_i^v$, $s_j^v$ first in the ECM wiring. Then, $\Pr[C_{s_i^v, s_k^w} C_{s_j^v, s_\ell^{w'}} = 1] = \Pr[C_{s_i^v, s_k^w} = 1] \Pr[C_{s_j^v, s_\ell^{w'}} = 1 | C_{s_i^v, s_k^w} = 1]$, where $\Pr[C_{s_i^v,s_k^w} = 1] = [2m-1]^{-1}$ and $\Pr[C_{s_j^v,s_\ell^{w'}} = 1 | C_{s_i^v,s_k^w} = 1] = [2m-3]^{-1}$. Hence,

$$\Pr[C_{v,w}C_{v,w'} = 1] \leq d_v(d_v - 1)d_w d_{w'}/m^2,$$

using $(2m-1)(2m-3) \geq m^2$ when $m \geq 3$. □

## 4. GETTING THE 4/3 MOMENT

We will use a series of claims to express the running time of MinBucket in a convenient form. For vertex $v$, let $X_v$ be the random variable denoting the number of edges in $v$'s bin. The expected running time is, at most, $\mathbf{E}[\sum_v X_v(X_v - 1)]$. This is because the number of wedges in each bin is $\binom{X_v}{2} \leq X_v^2 - X_v$.

We further break $X_v$ into the sum $\sum_w Y_{v,w}$, where $Y_{v,w}$ is the indicator for edge $(v, w)$, being in $v$'s bin. As mentioned earlier, $C_{v,w}$ is the indicator for edge $(v, w)$ being present. Note that $Y_{v,w} \leq C_{v,w}$, since $(v, w)$ can be in $v$'s bin only if it actually appears as an edge.

We list out some bounds on expectations. Only the second really uses the binning of MinBucket .

**Claim 4.1.** *Consider vertices $v, w, w'$ $(w \neq w')$.*

- *$\mathbf{E}[Y_{v,w}Y_{v,w'}] \leq d_v^2 d_w d_{w'}/m^2$.*
- *There exist sufficient small constants $\delta, \delta' \in (0, 1)$ such that: if $d_w < \delta d_v$ then $\mathbf{E}[Y_{v,w}Y_{v,w'}] \leq 2\exp(-\delta' d_v)d_v^2 d_w d_{w'}/m^2$.*

**Proof.** We use the trivial bound of $Y_{v,w}Y_{v,w'} \leq C_{v,w}C_{v,w'}$. By Claim 3.4, $\mathbf{E}[Y_{v,w}Y_{v,w'}] \leq \mathbf{E}[C_{v,w}C_{v,w'}] \leq d_v^2 d_w d_{w'}/m^2$.

Now, for the interesting bound: The quantity $\mathbf{E}[Y_{v,w}Y_{v,w'}]$ is the probability that both $Y_{v,w}$ and $Y_{v,w'}$ are 1. For this to happen, we definitely require both $(v, w)$ and $(v, w')$ to be present as edges. Call this event $\mathcal{E}$. We also require (at the very least) the degree of $v$ to be at most the degree of $w$ (otherwise the edge $(v, w)$ will not be put in $v$'s bin.) Call this event $\mathcal{F}$. If $D_v$, $D_w$ denote the degrees of $v$ and $w$, note that $D_w \leq d_w < \delta d_v$, implying event $\mathcal{F}$ is contained in the event $\{D_v < \delta d_v\}$ when $d_w < \delta d_v$. Hence, the event $Y_{v,w}Y_{v,w'} = 1$ is contained in $\mathcal{E} \cap \{D_v < \delta d_v\}$. Assume $d_v > 2$, $d_w > 0$, $d_{w'} > 0$ or else $\mathbf{E}[Y_{v,w}Y_{v,w'}] = 0$ holds when $\delta < 1/2$ by the assumption $d_w < \delta d_v$.

As in the proof of Claim 3.4, let $C_{s_i^v,s_j^w}$ be the indicator random variable for the edge being present between stubs $s_i^v$ and $s_j^w$ of vertices $v, w$ (and analogously define $C_{s_i^v,s_j^{w'}}$). Then, $\mathcal{E}$ is contained in $\cup_{1\leq i\neq j\leq d_v} \cup_{k=1}^{d_w} \cup_{\ell=1}^{d_{w'}} \{C_{s_i^v,s_k^w} C_{s_j^v,s_\ell^{w'}} = 1\}$ so that

$$\begin{aligned}\Pr[Y_{v,w}Y_{v,w'} = 1] &\leq \Pr[\mathcal{E}, D_v < \delta d_v] \\ &\leq \sum_{1\leq i\neq j\leq d_v} \sum_{k=1}^{d_w} \sum_{\ell=1}^{d_{w'}} \Pr[C_{s_i^v,s_k^w} C_{s_j^v,s_\ell^{w'}} = 1, D_v < \delta d_v] \\ &= \sum_{1\leq i\neq j\leq d_v} \sum_{k=1}^{d_w} \sum_{\ell=1}^{d_{w'}} \Pr[C_{s_i^v,s_k^w} C_{s_j^v,s_\ell^{w'}} = 1] \Pr[D_v < \delta d_v | C_{s_i^v,s_k^w} C_{s_j^v,s_\ell^{w'}} = 1].\end{aligned}$$

Given fixed values of $i, j, k, \ell$, order stubs $s_i^v, s_j^v$ first in the ECM wiring. Then, $\Pr[C_{s_i^v,s_k^w} C_{s_j^v,s_\ell^{w'}} = 1] \leq m^{-2}$ as in the proof of Claim 3.4. Additionally, conditioned on $C_{s_i^v,s_k^w} C_{s_j^v,s_\ell^{w'}} = 1$, the remaining stubs form an ECM with respect to a new degree sequence formed by replacing $2m, d_v, d_w, d_{w'}$ in the original degree sequence by $2\tilde{m} = 2m - 4, d_v - 2, d_w - 1, d_{w'} - 1$. Let $\tilde{D}_v$ denote the degree of $v$ in the final graph from the new degree sequence. Then, conditioned on $C_{s_i^v,s_k^w} C_{s_j^v,s_\ell^{w'}} = 1$, $D_v = 2 + \tilde{D}_v$ so that conditional probability is bounded by

$$\begin{aligned}\Pr[D_v < \delta d_v | C_{s_i^v,s_k^w} C_{s_j^v,s_\ell^{w'}} = 1] &= \Pr[\tilde{D}_v < \delta d_v - 2 | C_{s_i^v,s_k^w} C_{s_j^v,s_\ell^{w'}} = 1] \\ &\leq \Pr[\tilde{D}_v < \delta(d_v - 2) | C_{s_i^v,s_k^w} C_{s_j^v,s_\ell^{w'}} = 1] \\ &\leq 2\exp(-\delta' d_v),\end{aligned}$$

since $\delta < 1$. That is, Lemma 3.2 applies to $\tilde{D}_v$ with respect to the new degree sequence, where $v$ has degree $d_v - 2$ and each degree in this new sequence is less than $\sqrt{\tilde{m}}/2$ by assumption. The bound $\Pr[Y_{v,w}Y_{v,w'} = 1] \leq 2\exp(-\delta' d_v) d_v^2 d_w d_{w'}/m^2$ then follows. □

Armed with these facts, we can bound the expected number of $P_2$s contained in a single bucket.

**Lemma 4.2.** *There exists a sufficiently small* $\delta \in (0, 1)$ *such that*

$$\boldsymbol{E}[X_v(X_v - 1)] = O\left(\exp(-\delta d_v) d_v^2 + m^{-2} d_v^2 \left(\sum_{w:d_w \geq \delta d_v} d_w\right)^2\right).$$

**Proof.** We will write out

$$X_v^2 = \left(\sum_w Y_{v,w}\right)^2 = \sum_w Y_{v,w}^2 + \sum_w \sum_{w' \neq w} Y_{v,w} Y_{v,w'},$$

where $\sum_w Y_{v,w}^2 = \sum_w Y_{v,w} = X_v$ as each $Y_{v,w}$ is a 0–1 variable. Hence,

$$\begin{aligned}\mathbf{E}[X_v(X_v - 1)] &= \sum_w \sum_{w' \neq w} \mathbf{E}[Y_{v,w}Y_{v,w'}] \leq \sum_{\substack{w: \\ d_w \geq \delta d_v}} \sum_{\substack{w \neq w': \\ d_{w'} \geq \delta d_v}} \mathbf{E}[Y_{v,w}Y_{v,w'}] \\ &\quad + \sum_{\substack{w: \\ d_w < \delta d_v}} \sum_{w' \neq w} \mathbf{E}[Y_{v,w}Y_{v,w'}] + \sum_{\substack{w': \\ d_{w'} < \delta d_v}} \sum_{w \neq w'} \mathbf{E}[Y_{v,w}Y_{v,w'}] \\ &= \sum_{\substack{w: \\ d_w \geq \delta d_v}} \sum_{\substack{w \neq w': \\ d_{w'} \geq \delta d_v}} \mathbf{E}[Y_{v,w}Y_{v,w'}] + 2 \sum_{\substack{w: \\ d_w < \delta d_v}} \sum_{w' \neq w} \mathbf{E}[Y_{v,w}Y_{v,w'}] \\ &\leq \frac{d_v^2}{m^2} \left(\sum_{w:d_w \geq \delta d_v} d_w\right)^2 + 2 \sum_{\substack{w: \\ d_w < \delta d_v}} \sum_{w' \neq w} \mathbf{E}[Y_{v,w}Y_{v,w'}],\end{aligned}$$

by splitting the sums (for a given $\delta \in (0, 1)$ to be specified next) into cases, $d_w \geq \delta d_v$ and $d_w < \delta d_v$, and using the trivial bound of Claim 4.1 for the first quantity. We satisfy the conditions to use the second part of Claim 4.1 as

$$\sum_{\substack{w: \\ d_w < \delta d_v}} \sum_{w' \neq w} \mathbf{E}[Y_{v,w} Y_{v,w'}] \leq 2 \sum_{\substack{w: \\ d_w < \delta d_v}} \sum_{w' \neq w} \exp(-\delta d_v) d_v^2 d_w d_{w'} / m^2$$
$$\leq 8 \exp(-\delta d_v) d_v^2,$$

where $\sum_{i=1}^{n} d_i = 2m$. □

With this bound for $\mathbf{E}[X_v(X_v - 1)]$, we are ready to prove Theorem 1.1.

**Theorem 4.3.** $E[\sum_v X_v(X_v - 1)] = O(n + m^{-2}(\sum_{i=1}^{n} d_i^{4/3})^3)$.

**Proof.** We use linearity of expectation and sum the bound in Lemma 4.2. Note that $\exp(-\delta d_v) d_v^2$ is a decreasing function of $d_v$ and is, hence, $O(1)$. Hence,

$$\mathbf{E}\left[\sum_v X_v(X_v - 1)\right] \lessdot n + m^{-2} \sum_v d_v^2 \left(\sum_{w: d_w \geq \delta d_v} d_w\right)^2$$
$$= n + m^{-2} \sum_v \sum_{w: d_w \geq \delta d_v} \sum_{w': d_{w'} \geq \delta d_v} d_v^2 d_w d_{w'}.$$

This is the moment where the 4/3 moment will appear. Since $d_w \geq \delta d_v$ and $d_{w'} \geq \delta d_v$, $d_v^{2/3} \leq \delta^{-2/3} d_w^{1/3} d_{w'}^{1/3}$. Therefore, $d_v^2 d_w d_{w'} = d_v^{4/3} d_v^{2/3} d_w d_{w'} \leq \delta^{-2/3}(d_v d_w d_{w'})^{4/3}$. Wrapping it up,

$$m^{-2} \sum_v \sum_{w: d_w \geq \delta d_v} \sum_{w': d_{w'} \geq \delta d_v} d_v^2 d_w d_{w'} \lessdot m^{-2} \sum_v \sum_{w: d_w \geq \delta d_v} \sum_{w': d_{w'} \geq \delta d_v} (d_v d_w d_{w'})^{4/3}$$
$$\lessdot m^{-2} \left(\sum_v d_v^{4/3}\right)^3.$$

□

## 5. PROVING TIGHTNESS

We show that the bound achieved by Theorem 1.1 is tight for power laws with $\alpha > 2$. This shows that the bounds given in the proof of Corollary 1.3 are tight. The proof, as expected, goes by reversing most of the inequalities given earlier. For convenience, we will assume for the lower bound that $d_n < \sqrt{m}/4$, instead of the $\sqrt{m}/2$ used for the upper bound. This makes for cleaner technical arguments.

**Claim 5.1.** *Let* $\mathbf{d}$ *be a power law-degree sequence with* $\alpha \in (2, 7/3)$ *with* $d_n < \sqrt{m}/4$. *Then, the expected number of* $P_2$*s enumerated by* MinBucket *over* $ECM(\mathbf{d})$ *is* $\Omega(n d_n^{7-3\alpha})$.

We first need a technical claim to give a lower bound for probabilities of edges falling in a bucket.

**Claim 5.2.** *Let $d_v > 3$. Consider vertices $v, w, w'$ ($w \neq w'$) and let $c$ be a sufficiently large constant. If $\min(d_w, d_{w'}) > cd_v$, then $\mathbf{E}[Y_{v,w}Y_{v,w'}] = \Omega(d_v^2 d_w d_{w'}/m^2)$.*

**Proof.** The random variable $Y_{v,w}Y_{v,w'}$ is 1 if $(v, w)$, $(v, w')$ are edges and the degrees of $w$ and $w'$ are less than that of $v$. As before, we will start the matching process by matching stubs of $v$. We partition the stubs into two groups denoted by $B_w$ and $B_{w'}$ and start by matching stubs in $B_w$. We set $|B_w| = \lfloor d_v/3 \rfloor$. What is the probability that a stub in $B_w$ connects with a $w$-stub? This is at least $1 - (1 - d_w/2m)^{\lfloor d_v/3 \rfloor} = \Omega(d_v d_w/m)$.

Condition on any matching of the stubs in $B_w$. What is the probability that a stub in $B_{w'}$ matches with a $w'$-stub? Because $\min(|B_{w'}|, d_{w'}) \geq 2|B_w|$, this probability is at least $1 - (1 - d_{w'}/4m)^{\lfloor d_v/3 \rfloor} = \Omega(d_v d_{w'}/m)$.

Now condition on any matching of the $v$-stubs. The number of unmatched stubs connected to $w$ is at least $d_w/2$ (similarly for $w'$). The remaining stubs connect according to a standard configuration model. For the remaining degree sequence, the total number of stubs is $2\tilde{m} = 2m - 2d_v$. For sufficiently large $m$, $d_n \leq \sqrt{m}/4 \leq \sqrt{\tilde{m}}/2$. Hence, we can use Lemma 3.2 (and a union bound) to argue that the probability that the final degrees of $w$ and $w'$ are at least $d_v$ is $\Omega(1)$. Multiplying all the bounds together, the probability $Y_{v,w}Y_{v,w'} = 1$ is $\Omega(d_v^2 d_w d_{w'}/m^2)$. □

We prove Claim 5.1.

**Proof.** (of Claim 5.1) Note that when $\alpha > 2$, then $m = O(n)$. We start with the arguments in the proof of Lemma 4.2 and we use notation $A \gg B$ to denote $A = \Omega(B)$. Applying Claim 5.2 for vertex $v$ such that $d_v > 3$,

$$\begin{aligned}\mathbf{E}[X_v(X_v - 1)] = \sum_{w}\sum_{w' \neq w} \mathbf{E}[Y_{v,w}Y_{v,w'}] &\geq \sum_{\substack{w:\\ d_w \geq cd_v}} \sum_{\substack{w \neq w':\\ d_{w'} \geq cd_v}} \mathbf{E}[Y_{v,w}Y_{v,w'}] \\ &\gg m^{-2} d_v^2 \sum_{\substack{w:\\ d_w \geq cd_v}} \sum_{\substack{w \neq w':\\ d_{w'} \geq cd_v}} d_w d_{w'} \\ &\geq m^{-2} d_v^2 \left( \sum_{\substack{w:\\ d_w \geq cd_v}} d_w \right)^2 - m^{-2} d_v^2 \sum_{w} d_w^2.\end{aligned}$$

The latter part, summed over all $v$ is, at most,

$$m^{-2} \left( \sum_{v} d_v^2 \right)^2 \leq m^{-2} \left( \max_{v} d_v \sum_{v} d_v \right)^2 \lessdot m.$$

Now we focus on the former part. Choose $v$ so that $cd_v \leq d_n/2$, and let $2^r$ be the largest power of 2 greater than $cd_v$. (Note that $r \leq \log_2 d_n - 1$.) We bound $\sum_{w:d_w \geq cd_v} d_w \geq \sum_{w:d_w \geq 2^r} d_w \gg \sum_{k=r}^{\log_2 d_n - 1} 2^k n/2^{k(\alpha-1)}$. This is $\sum_{k=r}^{\log_2 d_n - 1} n/2^{k(\alpha-2)}$, which is convergent when $\alpha > 2$. Hence, it is at least $\Omega(n 2^{-r(\alpha-2)}) = \Omega(n d_v^{-(\alpha-2)})$.

We sum over all (appropriate $v$).

$$\begin{aligned}\sum_{v:3<d_v\le d_n/2c} m^{-2}d_v^2\left(\sum_{\substack{w:\\ d_w\ge cd_v}} d_w\right)^2 &\gg (n/m)^2 \sum_{v:3<d_v\le d_n/2c} d_v^2 d_v^{-(2\alpha-4)}\\ &= (n/m)^2 \sum_{v:3<d_v\le d_n/2c} d_v^{6-2\alpha}\\ &\gg (n/m)^2 \sum_{k=2}^{\lfloor \log_2 n-\log_2(2c)\rfloor} n2^{k(7-3\alpha)}.\end{aligned}$$

When $\alpha < 7/3$, the sum is divergent. Noting that $m = \Theta(n)$, we bound by $\Omega(nd_n^{7-3\alpha})$. Overall, we lower bound the running time MinBucket by $\sum_{v:3<d_v\le d_n/2c} \mathbf{E}[X_v(X_v-1)]$, which is $\Omega(nd_n^{7-3\alpha} - m)$. For $\alpha < 7/3$, this is $\Omega(nd_n^{7-3\alpha})$, matching the upper bound in Corollary 1.3.

□

## 6. EMPIRICAL ANALYSIS

We experimentally show that our theoretical analysis does a reasonable job of capturing the expected performance of MinBucket on ECM graphs. As is, Theorem 1.1 is asymptotic and cannot be used for predicting empirical performance. For this reason, we choose a specific class of degree distributions and get a tighter theoretical analysis.

### 6.1. Tighter Bounds on the Running Time

Under a specific choice of degrees, we can pin down the running time of MinBucket up to lower order terms. Rather than starting with an arbitrary degree sequence, we draw the degree for each vertex independently at random from a reference degree distribution $\mathcal{D}$, given by probability density function (pdf) $f$. Specifically, $f(d)$ is the probability that a node draws degree value $d$ for a given integer $d \in [0, \infty)$. After nodes draw degree values, the rest of the ECM construction proceeds as described in Section 1.1.

Formally, let $\mathcal{D}_n$ be the distribution with support $\{1, 2, \dots, \lfloor\sqrt{n}/\log^2 n\rfloor\}$, where the probability of $d$ is proportional to $f(d)$. Note that we do not allow a degree of 0 and cap the max degree at $\sqrt{n}/\log^2 n$ (instead of $\sqrt{n}$). These are mostly conveniences for a cleaner proof. We pick the degree sequence by taking $n$ i.i.d. draws from $\mathcal{D}_n$. So, the degree sequence $\mathbf{d}$ is distributed according to the product $\mathcal{D}_n^n$. Then, we generate an ECM with $\mathbf{d}$. For convenience, we denote $1 - 1/\sum_{d\le n} f(d)$ by $\gamma_n$. Note that $\gamma_n \to 0$, as $n \to \infty$. The probability of $d$ under $\mathcal{D}_n$ is $f(d)(1-\gamma_n)$. We use $m = \sum_v d_v/2$ to denote the number of edges in the *multigraph* and heavily use $m \ge n/2$.

Our analysis assumes that when an edge joins two vertices of the same degree, the edge is placed in the bucket for both vertices. Thus, we slightly overcount the work for MinBucket . Let $X_{i,n}$ be the size of the bucket for an arbitrary node $i$ in a graph generated by ECM with $n$ nodes. We wish to bound the expected triangle-searching work $\mathbf{E}[\sum_{i=1}^n \binom{X_{i,n}}{2}]$ in an ECM graph, as the number of nodes $n \to \infty$. We denote the $r$th moment, $r > 0$, of the reference degree distribution $f$ as $\mathbf{E}[d^r] = \sum_{t=1}^{\infty} t^r \cdot f(t)$.

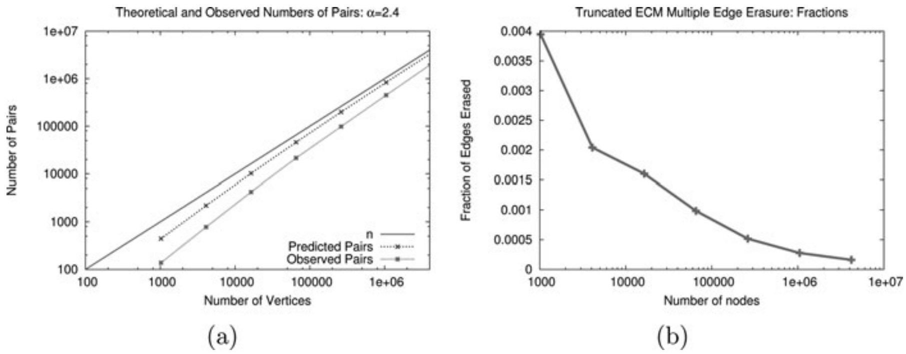

**Figure 1** Experimental results with $n$-node ECM graphs drawn from a truncated power-law distribution with exponent $\alpha = 2.4$. In (a) we see the work predicted by Theorem 6.1 and the average work observed in ten Monte Carlo trials. In (b) we show that the fraction of edges erased in the ECM process is tiny and shrinking.

**Theorem 6.1.** *Fix any n and a degree distribution $\mathcal{D}$ such that $\boldsymbol{E}[d]$ and $\boldsymbol{E}[d^{4/3}]$ are finite. Then*

$$\lim_{n\to\infty} \frac{1}{n}\boldsymbol{E}\left[\sum_{i=1}^{n}\binom{X_{i,n}}{2}\right] = \frac{1}{2(\boldsymbol{E}[d])^2}\sum_{t_1=1}^{\infty}\sum_{t_2=t_1}^{\infty}\sum_{t_3=t_1}^{\infty} t_1(t_1-1)t_2t_3 f(t_1)f(t_2)f(t_3) \in (0,\infty).$$

The triple sum in Theorem 6.1 represents an expectation $\mathbf{E}[IZ_1(Z_1-1)Z_2Z_3]$, where $Z_1, Z_2, Z_3$ are three independent random variables with distribution $f$ and $I$ is the indicator function of the event $\min\{Z_2, Z_3\} \geq Z_1$. The proof is basically analogous to that of Theorem 1.1, but extremely technical because of the additional randomness in the degree sequence. We do not present the proof here because of space (and readability) issues, but we refer to the tech report version of this article [6].

### 6.2. Experimental Results

Using the distribution described in Section 6.1, we construct ECM graphs with power-law degree distributions, and truncate degrees at $\sqrt{n}$. We choose $\alpha = 2.4$ (where the $^4/_3$ moment is finite). Figure 1(a) shows that the theoretical work predicted by Theorem 6.1 bounds the observed work for increasing graph sizes. Additionally, as shown in Figure 1(b), the effect of erasure on the underlying power-law distribution is small. For example, when $n = 4$ million vertices, Pearson's method of moments [30] estimates the power-law exponent as 2.396 from data after erasure.

## ACKNOWLEDGMENTS

We thank Ali Pinar for suggestions on improving the presentation.

## FUNDING

This work was funded under the Sandia National Laboratories Laboratory Directed Research and Development (LDRD) program.

## REFERENCES

[1] W. Aiello, F. Chung, and L. Lu. “A Random Graph Model for Power Law Graphs.” *Experimental Mathematics* 10 (2001), 53–66.

[2] N. Alon, R. Yuster, and U. Zwick. “Finding and Counting Given Length Cycles.” *Algorithmica* 17 (1997), 354–364.

[3] A.-L. Barabási and R. Albert. “Emergence of Scaling in Random Networks.” *Science* 286 (1999), 509–512.

[4] E. A. Bender and E. R. Canfield. “The Asymptotic Number of Labeled Graphs with Given Degree Sequences.” *Journal of Combinatorial Theory A* 24 (1978), 296–307.

[5] J. Berry, L. Fostvedt, D. Nordman, C. Phillips, C. Seshadhri, and A. Wilson. “Why Do Simple Algorithms for Triangle Enumeration Work in the Real World?” In *Innovations in Theoretical Computer Science (ITCS)* 1407:1116 (2014), 225–234.

[6] J. Berry, L. Fostvedt, D. Nordman, C. Phillips, C. Seshadhri, and A. Wilson. “Why do simple algorithms for triangle enumeration work in the real world?” Technical Report 1407.1116, arXiv, 2014. `http://arxiv.org/pdf/1407.1116v1.pdf`.

[7] J. W. Berry, B. Hendrickson, R. A. LaViolette, and C. A. Phillips. “Tolerating the Community Detection Resolution Limit with Edge Weighting”. *Physical Review E* 83:5, (2011), 056119.

[8] B. Bollobás. “A Probabilistic Proof of an Asymptotic Formula for the Number of Labelled Regular Graphs.” *European Journal on Combinatorics* 1 (1980), 311–316.

[9] T. Britton, M. Deijfen, and A. Martin-Löf. “Generating Simple Random Graphs with Prescribed Degree Distribution.” *Journal of Statistical Physics* 124:6 (2006), 1377–1397.

[10] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. “Graph Structure in the Web.” *Computer Networks* 33 (2000), 309–320.

[11] R. S. Burt. “Structural Holes and Good Ideas.” *American Journal of Sociology* 110:2 (2004), 349–399.

[12] R. S. Burt. “Secondhand Brokerage: Evidence on the Importance of Local Structure for Managers, Bankers, and Analysts.” *Academy of Management Journal* 50, (2007), 119–148.

[13] N. Chiba and T.Takao Nishizeki. “Arboricity and Subgraph Listing Algorithms.” *SIAM J. Comput.* 14 (1985), 210–223.

[14] M. Chrobak and D. Eppstein. “Planar Orientations with Low Out-Degree and Compaction of Adjacency Matrices.” *Theoretical Computer Science* 86 (1991), 243–266.

[15] F. Chung and L. Lu. “The Average Distances in Random Graphs with Given Expected Degrees.” *PNAS* 99 (2002), 15879–15882.

[16] F. Chung, L. Lu, and V. Vu. “Eigenvalues of Random Power Law Graphs.” *Annals of Combinatorics* 7 (2003), 21–33.

[17] J. Cohen. “Graph Twiddling in a MapReduce World.” *Computing in Science & Engineering* 11 (2009), 29–41.

[18] J. S. Coleman. “Social Capital in the Creation of Human Capital.” *American Journal of Sociology* 94 (1988), S95–S120.

[19] M. Faloutsos, P. Faloutsos, and C. Faloutsos. “On Power-Law Relationships of the Internet Topology.” In *Proceedings of SIGCOMM*, pp. 251–262. New York: ACM, 1999.

[20] B. Foucault Welles, A. Van Devender, and N.Noshir Contractor. “Is a Friend a Friend?: Investigating the Structure of Friendship Networks in Virtual Worlds.” In *CHI-EA’10*, pp. 4027–4032. New York: ACM, 2010.

[21] I. Fudos and C. M. Hoffmann. “A Graph-Constructive Approach to Solving Systems of Geometric Constraints.” *ACM Transactions on Graphics* 16:2 (1997), 179–216.

[22] A. Ital and M. Rodeh. “Finding a Minimum Circuit in a Graph.” *SIAM Journal on Computing* 7 (1978), 413–423.

[23] M. Latapy. “Main-Memory Triangle Computations for very Large (Sparse (Power-Law)) Graphs.” *Theoretical Computer Science* 407 (2008), 458–473.

[24] M. Mihail and C. Papadimitriou. "On the Eigenvalue Power Law." In *RANDOM*, LNCS, 254–262. Cambridge, MA: Springer, 2002.

[25] M. Molloy and B. Reed. "A Critical Point for Random Graphs with a Given Degree Sequence." *Random Structures and Algorithms* 6 (1995), 161–179.

[26] M. Molloy and B. Reed. "The Size of the Giant Component of a Random Graph with a Given Degree Sequence." *Combinatorics, Probability and Computing* 7 (1998), 295–305.

[27] R. Motwani and P. Raghavan. *Randomized Algorithms*. New York: Cambridge University Press, 1995.

[28] M. E. J. Newman. "The Structure and Function of Complex Networks." *SIAM Review* 45 (2003), 167–256.

[29] M. E. J. Newman, S. Strogatz, and D. Watts. "Random Graphs with Arbitrary Degree Distributions and Their Applications." *Physical Review E* 64 (2001), 026118.

[30] K. Pearson. "Method of Moments and Method of Maximum Likelihood." *Biometrika* 28 (1936), 34–59.

[31] A. Portes. "Social Capital: Its Origins and Applications in Modern sociology." *Annual Review of Sociology* 24:1 (1998), 1–24.

[32] D. Sergi. "Random Graph Model with Power-Law Distributed Triangle Subgraphs." *Phys Rev E* 72 (2005), 025103.

[33] S. Suri and S. Vassilvitskii. "Counting Triangles and the Curse of the Last Reducer." In *Proceedings of WWW'11*, pp. 607–614. New York: ACM, 2011.

[34] T. Schank and D. Wagner. "Finding, Counting and Listing all Triangles in Large Graphs, an Experimental Study." In *Experimental and Efficient Algorithms*, pp. 606–609. Berlin Heidelberg; Springer, 2005.

[35] C. E. Tsourakakis. "Fast Counting of Triangles in Large Real Networks Without Counting: Algorithms and Laws." In *ICDM*, pp. 608–617. IEEE, 2008.

[36] V. Vassilevska Williams and R. Williams. "Subcubic Equivalences Between Path, Matrix and Triangle Problems." In *Foundations of Computer Science (FOCS)*, pp. 645–654. IEEE Computer Society, 2010.

[37] D. Watts and S. Strogatz. "Collective Dynamics of 'Small-World' Networks." *Nature* 393 (1998), 440–442.

[38] N. C. Wormald. "The Asymptotic Connectivity of Labelled Regular Graphs." *Journal of Combinatorial Theory B* 31 (1981), 156–167.

## APPENDIX A PROOF OF LEMMA 3.1

**Proof.** Consider the sequence $X'_1, X'_2, \ldots, X'_k$ of i.i.d. Bernoulli random variables with $\mathbf{E}[X'_i] = \alpha$. It suffices to show that, for any $t \in \mathbb{R}$, $\Pr[\sum_{i=1}^k X_i < t] \leq \Pr[\sum_{i=1}^k X'_i < t]$. Given this, we apply a multiplicative Chernoff bound (Theorem 4.2 of [27]) for $\sum_{i=1}^k X'_i$ with $\mu = \alpha k$ to obtain $\Pr[\sum_{i=1}^k X_i < \alpha k \delta] < \exp(-\alpha(1-\delta)^2/2)$. Assume for any $t$ and some index $j$, $p_t \equiv \Pr[\sum_{i=1}^j X_i < t] \leq \Pr[\sum_{i=1}^j X'_i < t] \equiv p'_t$ holds; by assumption, this is true for $j = 1$. We prove this statement holds for $j+1$ and given $t$. Let $\mathbb{I}(A)$ denote the indicator function of event $A \equiv \sum_{i=1}^j X_i \in [t-1, t)$. Because $X_i$ is a 0–1 variable, we get $\Pr[\sum_{i=1}^{j+1} X_i < t] = p_{t-1} + \Pr[A, X_{j+1} = 0]$ where

$$\begin{aligned}
\Pr[A, X_{j+1} = 0] &= \mathbf{E}\{\mathbf{E}[\mathbb{I}(A)\mathbb{I}(X_{j+1} = 0)|Y_1, \ldots, Y_j]\} \\
&= \mathbf{E}\{\mathbb{I}(A)\mathbf{E}[\mathbb{I}(X_{j+1} = 0)|Y_1, \ldots, Y_j]\} \\
&\leq \mathbf{E}\{\mathbb{I}(A)\}(1-\alpha) = (p_t - p_{t-1})(1-\alpha)
\end{aligned}$$

using $\mathbb{I}(A)$ as a constant in the conditional expectation, that $\mathbf{E}\{\mathbb{I}(A)\} = \Pr(A) = p_t - p_{t-1}$, and that $\Pr[X_{j+1} = 0 | Y_1, \ldots, Y_j] \leq 1 - \alpha$ by assumption. The above gives

$$\begin{aligned}
\Pr\left[\sum_{i=1}^{j+1} X_i < t\right] &\leq p_{t-1} + (p_t - p_{t-1})(1-\alpha) = p_{t-1}\alpha + p_t(1-\alpha) \\
&\leq p'_{t-1}\alpha + p'_t(1-\alpha) \quad \text{(using induction hypothesis and } \alpha \in [0,1]) \\
&= p'_{t-1} + (p'_t - p'_{t-1})(1-\alpha) = \Pr\left[\sum_{i=1}^{j+1} X'_i < t\right].
\end{aligned}$$

□