A Mathematical Analysis of the R-MAT Random Graph Generator

Chris Groër * Blair D. Sullivan
Steve Poole
Oak Ridge National Laboratory, Oak Ridge, TN 37830

January 5, 2011

Abstract

The R-MAT graph generator introduced by Chakrabarti, Faloutsos, and Zhan [6] offers a simple, fast method for generating very large directed graphs. These properties have made it a popular choice as a method of generating graphs for objects of study in a variety of disciplines, from social network analysis to high performance computing. We analyze the graphs generated by R-MAT and model the generator in terms of occupancy problems in order to prove results about the degree distributions of these graphs. We prove that the limiting degree distributions can be expressed as a mixture of normal distributions with means and variances that can be easily calculated from the R-MAT parameters. Additionally, this paper offers an efficient computational technique for computing the exact degree distribution and concise expressions for a number of properties of R-MAT graphs.

1 Introduction

The R-MAT model for graph generation was introduced by Chakrabarti, Faloutsos, and Zhan [6]. The generator has an elegant, parsimonious design that is also very easy to implement. Additionally, R-MAT is easily parallelized and it is capable of quickly generating very large graphs. In the initial description of this generator, the authors state that R-MAT "naturally generates power-law (or 'DGX' [4]) degree distributions." The authors demonstrate that they

*Notice: This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC05-00OR22725 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.

are able to find parameters such that graphs generated by R-MAT provide a reasonable match to the degree distributions derived from empirical data.

Due to its speed, simplicity, and the availability of open source implementations [2], R-MAT is a widely used graph generator. Graphs generated by R-MAT have been used in a variety of research disciplines including graph theoretic benchmarks [1, 15], social network analysis [9], computational biology [3], and network monitoring [14]. Despite the wide use of this generator, there has been only minimal mathematical analysis of the graphs that it produces.

In this paper, we begin to fill this gap by providing a rigorous analysis of the degree distributions of graphs generated by R-MAT. We begin by developing an exact formula for the probability of observing a given edge in an R-MAT graph where this probability is defined in terms of the binary representation of the edge's endpoints. We then analyze the degree distribution of graphs generated by R-MAT by modeling the edge generation procedure in terms of the classical occupancy problem from probability theory. Our main result is that the in-degree, out-degree, and total degree distributions tend to a limiting distribution that can be expressed as a mixture of normal distributions with means and variances easily calculated in terms of the initial parameters.

The paper is organized as follows. In Section 2, we present a description of the R-MAT random graph generator, describe how the generation of each random edge can be viewed in terms of binary digits, and prove some elementary properties related to these probabilities. We also contrast the R-MAT generator with a larger set of generators based on the matrix Kronecker product [12]. Section 3 examines the degree distribution of vertices in the graphs before duplicate edge removal in the R-MAT algorithm. Section 4 contains our main results related to the degree distributions for vertices in R-MAT graphs after duplicate removal, and Section 5 describes some computational techniques that can be used to speed up the calculation of the exact degree distributions.

2 The R-MAT Graph Generator

To describe the way that the R-MAT generator produces a random graph, we first need a bit of notation. Let $G = (V, E)$ be a directed multigraph on $n = 2^k$ vertices with M edges. Letting $V = \{0, 1, \ldots, n-1\}$, we write the adjacency matrix for G as $A = \{a_{ij}\}$ with entry a_{ij} corresponding to the edge(s) from vertex i to vertex j. Duplicate edges are recorded by permitting the entries in A to be non-negative integers, with $a_{ij} = k$ if the multigraph has k edges from i to j.

2.1 Original Model

The R-MAT model for graph generation operates by recursively subdividing the adjacency matrix of a directed graph into four equally-sized partitions and distributing M edges within these partitions with unequal probabilities. The distribution is determined by four non-negative pa-

rameters $\alpha, \beta, \gamma, \delta$ such that $\alpha+\beta+\gamma+\delta = 1$. Starting with $a_{ij} = 0$ for all $0 \leq i, j \leq n-1$, the algorithm places an edge in the matrix by choosing one of the four partitions with probability α, β, γ or δ, respectively. The chosen quadrant is then subdivided into four smaller partitions, and the procedure repeated until we have selected a 1×1 partition, where we increment that entry of the adjacency matrix by one. For example, in Figure 1, we recursively partition the matrix five times before arriving at the shaded 1×1 partition. In general, since $|V| = 2^k$, exactly k subdivisions are required. The algorithm repeats the edge generation process M times to produce a matrix with $\sum_i \sum_j a_{ij} = M$. Since R-MAT creates digraphs without duplicate edges, the final step of the algorithm is to replace each nonzero matrix entry by one, creating a 0/1-adjacency matrix A' for a digraph G' on 2^k vertices with $M' \leq M$ edges. Finally, we note that the initial description in [6] suggests that one should "add some noise" to the $\alpha, \beta, \gamma, \delta$ parameters at each stage of the recursion. However, because no specifics are provided and since our main results deal with the limiting distributions, we do not address this issue.

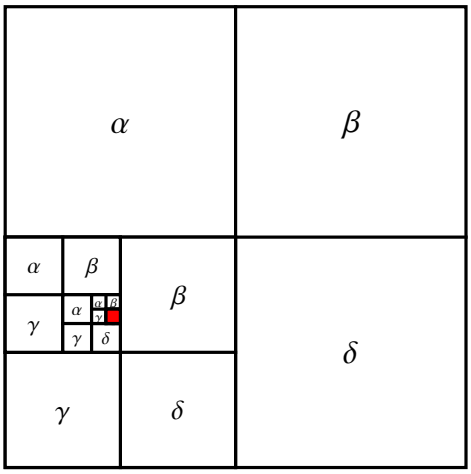

Figure 1: Generating an Edge with R-MAT

2.2 A Bitwise Interpretation

R-MAT's generation of the nonzero elements in the adjacency matrix has a bitwise interpretation that is particularly convenient for computer implementation. Note that when generating each edge ij in a digraph on 2^k vertices, we choose a total of k quadrants at random based on the value of the parameters α, β, γ and δ. For each of the k steps, we generate a random $r \sim U(0, 1)$ and select one of the four quadrants based on the value of r. For $1 \leq t \leq k$, we associate the t-th quadrant selected with the t-th bit of i and j (counting from the left). If the upper left quadrant is chosen at step t, then we set bit t of i and j to both be zero. If the upper right quadrant is selected, then we set bit t of i to 0 and bit t of j to 1. Similarly,

selecting the lower left quadrant at step t corresponds to setting bit t of i to 1 and bit t of j to 0, while the bottom right quadrant corresponds to setting the t-th bit in both i and j to 1. An example of this interpretation of the R-MAT generator is given in Example 1, and Algorithm 1 provides pseudocode for this algorithm.

Step	0	1	2	3	4	5
Quadrant Selected		Bottom Left	Upper Left	Bottom Right	Upper Right	Bottom Right
Bits of i	*****	1****	10***	101**	1010*	10101
Bits of j	*****	0****	00***	001**	0011*	00111

Example 1: The generation of the edge ij depicted in Figure 1 requires five steps. We begin in Step 0 with 5 empty bit positions for both i and j (these are denoted with a $*$) and then set each bit to 0 or 1 moving from left to right based on the quadrant selected at each step.

Algorithm 1 Given parameters $\alpha, \beta, \gamma, \delta$ with $\alpha + \beta + \gamma + \delta = 1$, generate a 0/1-adjacency matrix $A = \{a_{ij}\}$ for a graph on 2^k vertices containing at most M edges.

Set $a_{ij} = 0$ for $0 \leq i, j \leq 2^k - 1$
for $m = 1$ to M **do**
  Set $i = 0$, $j = 0$ // Initialize all bits to 0
  for $t = 0$ to $k - 1$ **do**
    Generate $r \sim U(0, 1)$
    if $r \in [\alpha, \alpha + \beta)$ **then**
      $j = j + 2^{k-1-t}$ // Set bit to 1 in j
    else if $r \in [\alpha + \beta, \alpha + \beta + \gamma)$ **then**
      $i = i + 2^{k-1-t}$ // Set bit to 1 in i
    else if $r \in [\alpha + \beta + \gamma, 1)$ **then**
      $i = i + 2^{k-1-t}$ and $j = j + 2^{k-1-t}$ // Set bit to 1 in i and j
    end if
  end for
  $a_{ij} = a_{ij} + 1$
end for
Replace all nonzero entries in A with ones

2.3 Preliminaries

We now give a number of definitions and basic lemmas necessary for our analysis of graphs generated with the R-MAT algorithm. For the remainder of this paper, unless otherwise noted, G will denote a directed multigraph on $n = 2^k$ vertices and M edges, generated by R-MAT

without duplicate removal (lines 1-15 of Algorithm 1). The duplicate-free graph recovered by replacing each positive entry of A with a one (line 16 of Algorithm 1) will be denoted G', and we write M' for the number of edges in G'.

Definition 2.1. Let $G = (V, E)$ be a directed graph (which may have multiple edges), and let $u \in V$ be a vertex of G. We define the out-degree of u, notated $d_G^+(u)$, to be the number of edges $e \in E$ so that e is of the form (u, v) for some $v \in V$. Similarly, the in-degree of u, denoted $d_G^-(u)$, is the number of edges $e \in E$ of the form (v, u) for some $v \in V$. The total degree of u, written $d_G(u)$, is the number of edges $e \in E$ so that $e = (u, v)$ and/or $e = (v, u)$ for some $v \in V$.

Definition 2.2. Given some vertex u with $0 \leq u \leq 2^k - 1$, let u_z denote the number of zeros in u's binary representation.

Definition 2.3. Given some edge $e = (u, v)$ in G where $0 \leq u, v \leq 2^k - 1$, write $u = \sum_{i=0}^{k-1} u_i 2^i$ and $v = \sum_{i=0}^{k-1} v_i 2^i$ in binary so that $u_i, v_i \in \{0, 1\}$ for $0 \leq i \leq k - 1$. Define e_α to be the number of (u_i, v_i) pairs that are $(0, 0)$, e_β to be the number of $(0, 1)$ pairs, e_γ to be the number of $(1, 0)$ pairs, and e_δ to be the number of (u_i, v_i) pairs equal to $(1, 1)$. Note that $e_\alpha + e_\beta + e_\gamma + e_\delta = k$.

Lemma 2.4. *The probability of generating an edge $e = (u, v)$ at some iteration of the R-MAT algorithm is equal to*

$$p(e) = p(u, v) = \alpha^{e_\alpha} \beta^{e_\beta} \gamma^{e_\gamma} \delta^{e_\delta}.$$

Proof. For the edge $e = (u, v)$, we must choose the upper left quadrant (corresponding to α) exactly e_α times, the upper right quadrant (corresponding to β) exactly e_β times, and so on. The result follows since the selection of subsequent quadrants is independent. □

Observation 2.5. *If $p(u, v) = \alpha^{e_\alpha} \beta^{e_\beta} \gamma^{e_\gamma} \delta^{e_\delta}$, then $p(v, u) = \alpha^{e_\alpha} \gamma^{e_\beta} \beta^{e_\gamma} \delta^{e_\delta}$.*

Lemma 2.6. *For a vertex u, the sum of the i^{th} powers of the probabilities of edges starting from u is given by*

$$\sum_{v=0}^{2^k-1} p(u, v)^i = (\alpha^i + \beta^i)^{u_z} (\gamma^i + \delta^i)^{k-u_z}.$$

Similarly, for edges ending at u, we have

$$\sum_{v=0}^{2^k-1} p(v, u)^i = (\alpha^i + \gamma^i)^{u_z} (\beta^i + \delta^i)^{k-u_z}.$$

Proof. Since $e_\alpha + e_\beta = u_z$ and $e_\gamma + e_\delta = k - u_z$, we can apply Lemma 2.4 to obtain

$$\begin{aligned}
\sum_{v=0}^{2^k-1} p(u,v)^i &= \sum_{e_\alpha=0}^{u_z} \sum_{e_\gamma=0}^{k-u_z} \binom{u_z}{e_\alpha} \binom{k-u_z}{e_\gamma} (\alpha^{e_\alpha} \beta^{u_z-e_\alpha} \gamma^{e_\gamma} \delta^{k-u_z-e_\gamma})^i \\
&= \sum_{e_\alpha=0}^{u_z} \binom{u_z}{e_\alpha} \alpha^{ie_\alpha} \beta^{i(u_z-e_\alpha)} \sum_{e_\gamma=0}^{k-u_z} \binom{k-u_z}{e_\gamma} \gamma^{ie_\gamma} \delta^{i(k-u_z-e_\gamma)} \\
&= (\alpha^i + \beta^i)^{u_z} (\gamma^i + \delta^i)^{k-u_z},
\end{aligned}$$

where the final equality follows from the binomial theorem. Using Observation 2.5, the proof for edges ending at u is analogous. □

Definition 2.7. Let $\alpha, \beta, \gamma, \delta > 0$ satisfying $\alpha + \beta + \gamma + \delta = 1$ denote the probabilities of assigning an edge to each of the four quadrants of a matrix (as in Figure 1). Define $\lambda = \alpha + \beta$, which can be interpreted as the probability of choosing "up" in a step of the recursion algorithm. Similarly, let $\mu = \alpha + \gamma$ be the probability of moving "left".

Definition 2.8. For $0 \leq i \leq k$, let $P_i = \lambda^i (1-\lambda)^{k-i}$ and $Q_i = \mu^i (1-\mu)^{k-i}$.

Corollary 2.9. *Given a vertex u, the probability of an edge being of the form (u, v) for some v is*

$$\sum_{v=0}^{2^k-1} p(u,v) = \lambda^{u_z} (1-\lambda)^{k-u_z} = P_{u_z},$$

and the probability of an edge of the form vu is

$$\sum_{v=0}^{2^k-1} p(v,u) = \mu^{u_z} (1-\mu)^{k-u_z} = Q_{u_z}.$$

Proof. This is a special case of Lemma 2.6 with $i = 1$. □

Definition 2.10. For a vertex u in G, define $\mathbf{p}_\mathbf{u}^+$ to be the vector of probabilities $\mathbf{p}_\mathbf{u}^+ = \{p(u,v)\}_{v=0}^{2^k-1}$, let $\mathbf{p}_\mathbf{u}^-$ be the vector of probabilities $\mathbf{p}_\mathbf{u}^- = \{p(v,u)\}_{v=0}^{2^k-1}$, and let $\mathbf{p}_\mathbf{u}$ be the vector of $2^{k+1} - 1$ probabilities obtained by appending $\mathbf{p}_\mathbf{u}^-$ to $\mathbf{p}_\mathbf{u}^+$ where we keep only the first copy of $p(u,u)$. Additionally, let $\hat{\mathbf{p}}_\mathbf{u}^+$ denote the probability vector (the entries sum to 1) obtained by appending the value $1 - \sum_v p(u,v)$ to $\mathbf{p}_\mathbf{u}^+$. We similarly define $\hat{\mathbf{p}}_\mathbf{u}^-$ and $\hat{\mathbf{p}}_\mathbf{u}$.

Lemma 2.11. *For a vertex u, there are at most $(u_z+1)(k-u_z+1)$ distinct values of $p(u,v)$ in the vectors $\mathbf{p}_\mathbf{u}^+$ and of $p(v,u)$ in $\mathbf{p}_\mathbf{u}^-$.*

Proof. From Lemma 2.4, it follows that $p(u,v) = \alpha^{e_\alpha} \beta^{u_z-e_\alpha} \gamma^{e_\gamma} \delta^{k-u_z-e_\gamma}$. As v runs from 0 to $2^k - 1$, there are $u_z + 1$ possibilities for e_α and $k + 1 - u_z$ possibilities for e_γ, implying that $\alpha^{e_\alpha} \beta^{u_z-e_\alpha} \gamma^{e_\gamma} \delta^{k-u_z-e_\gamma}$ assumes at most $(u_z+1)(k-u_z+1)$ distinct values. The proof is analogous for $\mathbf{p}_\mathbf{u}^-$. □

2.4 R-MAT & Kronecker Generators

The R-MAT generator is similar to a larger class of graph generators based on the matrix Kronecker product [17]. The stochastic Kronecker generator defined in [12] begins with an $N_1 \times N_1$ probability matrix P_1 and is expanded to an $N_1^k \times N_1^k$ probability matrix P_k via Kronecker exponentiation. If one begins with an initial 2×2 matrix P_1, then it is not difficult to see that the entry in row i, column j of this probability matrix, $P_k[i, j]$, is equal to the edge probability defined in Lemma 2.4. In [12], the authors state that

> Stochastic Kronecker graphs include several other generators, as special cases: For $\alpha = \beta$, we obtain an Erdös-Rényi random graph; for $\alpha = 1$ and $\beta = 0$, we obtain a deterministic Kronecker graph; setting the G_1 matrix to a 2×2 matrix, we obtain the R-MAT generator.

However, there is an important distinction in how the edges in the random graph are generated given these probabilities. Given an initial 2×2 matrix, the stochastic Kronecker model described in [12] requires $2^k \cdot 2^k$ iterations where the generation of each edge ij is the result of an independent Bernoulli trial with "success" probability $P_k[i, j]$. On the other hand, in the R-MAT generator the user supplies a maximum number of edges $M = c \cdot 2^k$, and the algorithm then runs for M iterations. Each R-MAT iteration is independent and it is possible for any of the 2^{2k} edges to be added to G at any stage. The probability that the edge ij is added at any particular iteration is equal to $P_k[i, j]$. This procedure can lead to generating the same edge more than once, and these duplicates are discarded during the final step of the R-MAT generator when G is transformed into the graph G' (step 16 in Algorithm 1).

We note that a more recent paper [13] addresses the relationship between R-MAT and stochastic Kronecker graphs in more detail. Additionally, they propose a way of speeding up the generation of stochastic Kronecker graphs by using the recursive partitioning procedure used in R-MAT. Given an $N_1 \times N_1$ initial probability matrix and letting E be the expected number of edges in a stochastic Kronecker graph K produced via these parameters, a random graph is produced by running E iterations of R-MAT's recursive partitioning. While R-MAT requires a 2×2 probability matrix for the partitioning, this interpretation of the Kronecker model allows an arbitrary $N_1 \times N_1$ probability matrix for the partitioning. Finally, while R-MAT removes duplicate edges from G to form the simple graph G', this version of the stochastic Kronecker generator keeps these multi-edges in the graph. This implies that if the initial Kronecker probability matrix is 2×2, then the resulting random graphs are produced in the same manner as those produced by R-MAT if one ignores the final duplicate removal step in Algorithm 1.

3 The R-MAT multigraph G

Having shown some simple facts related to the edge probabilities for graphs generated by R-MAT, we now explore the degree distributions. Our ultimate goal is to determine the degree distributions in the graph G' which is obtained from G by removing duplicates.

Lemma 3.1. *The probability that a vertex u has out-degree d in G is*

$$\mathbf{Pr}[d_G^+(u) = d] = \binom{M}{d} (P_{u_z})^d (1 - P_{u_z})^{M-d},$$

and the probability that a vertex u has in-degree d is

$$\mathbf{Pr}[d_G^-(u) = d] = \binom{M}{d} (Q_{u_z})^d (1 - Q_{u_z})^{M-d}.$$

Proof. We will only prove the result for out-degree, as the proof for in-degree is analogous. In terms of the adjacency matrix, the probability that u has out-degree d is the probability that the sum of the entries in row u is equal to d after all M edges have been added. Noting that P_{u_z} is the probability of incrementing an entry in row u when adding an edge to the graph, the probability of incrementing entries in row u exactly d times out of M is then $\binom{M}{d}(P_{u_z})^d(1 - P_{u_z})^{M-d}$, where the binomial coefficient corresponds to choosing which d steps generate an out-neighbor for u. This completes the proof. □

Corollary 3.2. *The probability distributions of the out-degree and in-degree of a vertex u in G are determined by the parameters $\alpha, \beta, \gamma, \delta$ and the number of zeros in the binary representation of u. In particular, they are given by the binomial distributions $B(M, P_{u_z})$ and $B(M, Q_{u_z})$, respectively.*

The following result was proven by Chakrabarti & Faloutsos in [5]. We include it here for completeness, with a slightly different proof.

Lemma 3.3. *The expected number of vertices in G with out-degree d is*

$$\sum_{i=0}^{k} \binom{k}{i} \binom{M}{d} (P_i)^d (1 - P_i)^{M-d},$$

and the expected number of vertices in G with in-degree d is

$$\sum_{i=0}^{k} \binom{k}{i} \binom{M}{d} (Q_i)^d (1 - Q_i)^{M-d}.$$

Proof. This follows directly from the fact that each vertex u has a k-bit binary representation, and the probability of out-degree (or in-degree) d is completely determined by u_z. For each $i \in [0, k]$, there are $\binom{k}{i}$ vertices with $u_z = i$, and the probability such a vertex has out-degree d is $\binom{M}{d} (P_i)^d (1 - P_i)^{M-d}$ from Lemma 3.1 (likewise, in terms of Q_i for in-degree). □

4 The R-MAT simple directed graph G'

Recall that G' is the graph generated by running the R-MAT algorithm to place M edges among $n = 2^k$ vertices, then removing any duplicate edges (the edge (u, v) is in G' if and only if G has at least one (u, v) edge), and we write M' for the number of edges in G'. In this section, we are able to use a number of results from the rich theory of occupancy problems in order to derive both exact and limiting degree distributions for the graph G'.

The classical occupancy problem is often described in terms of tossing r indistinguishable balls into m distinguishable urns and finding the probability that exactly n of these urns are non-empty (see [8, 11]). The R-MAT generator can be modeled as such a problem by envisioning the 4^k positions in the adjacency matrix as the set of urns, and the M randomly generated edges as the set of balls tossed into these urns. The number of edges in the graph G' then corresponds to the number of non-empty urns.

4.1 Occupancy Problems - Notation and Background

In the simplest ball and urn model, a ball falls into each of m urns with the same probability (namely, $1/m$). In the case of R-MAT, however, the edges are generated with different probabilities (see Lemma 2.4), and so we must use a more general model where each urn potentially has a different probability of receiving a ball. In this model, a ball falls into urn i with probability q_i (we assume $\sum_{i=1}^{m} q_i = 1$), and the probability vector $\mathbf{q} = \{q_1, q_2, \ldots, q_m\}$ denotes the set of these probabilities for each of the m urns. The following definition clarifies the specific quantity of interest.

Definition 4.1. Given m urns with probabilities $\mathbf{q} = \{q_1, q_2, \ldots, q_m\}$, let $U(r, l, m, \mathbf{q}, t)$ denote the probability that exactly t of the first $l \leq m$ urns are empty after tossing r balls into the set of m urns.

In what follows, for a random variable X, we denote the expected value of X as $\mathbf{E}[X]$, its variance by $\mathbf{Var}[X]$, and we use $\mathbf{Cov}[X, Y]$ to denote the covariance of two random variables X and Y. Using this notation, we now give the mean and variance of the number of empty urns as well as an exact formula for the probability distribution of the number of empty urns.

Theorem 4.2 (Johnson and Kotz [11], p. 107–113)**.** *Given a set of m urns with probabilities represented by the probability vector $\mathbf{q} = \{q_1, q_2, \ldots, q_m\}$, let X be the random variable corresponding to the number of empty urns after tossing r balls into these urns. Then the mean and variance of X are given by*

$$\mathbf{E}[X] = \mu(\mathbf{q}, r) = \sum_{i=1}^{m} (1-q_i)^r, \text{ and} \tag{1}$$

$$\mathbf{Var}[X] = \sigma(\mathbf{q}, r) = \sum_{i=1}^{m} \left[(1-q_i)^r \left(1-(1-q_i)^r\right)\right] + 2\sum_{i=1}^{m}\sum_{j=i+1}^{m} (1-q_i-q_j)^r - (1-q_i)^r(1-q_j)^r, \tag{2}$$

and the probability that exactly t of the m urns are empty is given by

$$U(r, m, m, \mathbf{q}, t) = \mathbf{Pr}[X = t] = \sum_{i=0}^{m-t} \binom{t+i}{t} (-1)^i \sum_{\substack{\mathcal{A}\subseteq\{1,2,\ldots,m\} \\ |\mathcal{A}|=t+i}} \left(1 - \sum_{j\in\mathcal{A}} q_j\right)^r.$$

Proof. Letting $X_j = 0$ if urn j is occupied and $X_j = 1$ if urn j is empty, $\mathbf{Pr}[X_j = 1] = (1-q_j)^r$. By linearity of expectation,

$$\mathbf{E}[X] = \sum_{j=1}^{m} \mathbf{E}[X_j] = \sum_{j=1}^{m} (1-q_j)^r.$$

For the variance, we use the formula

$$\mathbf{Var}[X] = \sum_{j=1}^{m} \mathbf{Var}[X_j] + 2\sum_{i<j}^{m} \mathbf{Cov}[X_i, X_j]$$

and the result follows by calculating

$$\begin{aligned} \mathbf{Var}[X_j] &= \mathbf{E}\left[X_j^2\right] - (\mathbf{E}[X_j])^2 = \mathbf{Pr}[X_j = 1] - \mathbf{Pr}[X_j = 1]^2; \\ \mathbf{Cov}[X_i, X_j] &= \mathbf{E}[X_i X_j] - \mathbf{E}[X_i]\,\mathbf{E}[X_j] \end{aligned}$$

where $\mathbf{E}[X_i X_j] = \mathbf{Pr}[X_i X_j = 1] = (1-q_i-q_j)^r$. Turning to the probability distribution, for a subset $\mathcal{A} \subseteq \{1, 2, \ldots, m\}$ with $|\mathcal{A}| = j \leq m$, let $P_{\mathcal{A}}$ denote the probability that all of the j urns represented by the set $\mathcal{A}$ are empty and that the remaining remaining $m-j$ urns are non-empty. Then $\mathbf{Pr}[X = t]$ can be calculated by summing over all possible sets of t urns:

$$\mathbf{Pr}[X = t] = \sum_{\substack{\mathcal{A}\subseteq\{1,2,\ldots,l\} \\ |\mathcal{A}|=t}} P_{\mathcal{A}}. \tag{3}$$

The result follows by noting that the probability that every urn in $\mathcal{A}$ is empty is equal to $(1 - \sum_{j\in\mathcal{A}} q_j)^r$ and then using inclusion-exclusion to rewrite the sum. □

The following corollary considers the case when one is concerned with number of empty urns contained in some subset of the m urns.

Corollary 4.3. *Given a set of m urns with probabilities $\mathbf{q} = \{q_1, q_2, \ldots, q_m\}$, for $1 \leq l \leq m$, let Y_l be the random variable representing the number of empty urns among urns $1, 2, \ldots, l$ after tossing r balls into the set of all m urns. Then*

$$U(r, l, m, \mathbf{q}, t) = \mathbf{Pr}\left[Y_l = t\right] = \sum_{i=0}^{l-t} \binom{t+i}{t} (-1)^i \sum_{\substack{\mathcal{B} \subseteq \{1,2,\ldots,l\} \\ |\mathcal{B}| = t+i}} \big(1 - \sum_{j \in \mathcal{B}} q_j\big)^r. \tag{4}$$

Proof. The proof is nearly identical to the proof of Theorem 4.2, except that we replace the quantity $P_{\mathcal{A}}$ with $P_{\mathcal{B}}$ which is the probability that each of the $j \leq l \leq m$ urns represented by the set $\mathcal{B}$ is empty and that the remaining $l - j$ urns are full. □

4.2 Exact Degree Distributions in G'

Having stated the required results related to occupancy problems, we return to the analysis of the out-degree distribution for some vertex u in G'. Perhaps the most obvious way of modeling the out-degree of u in terms of balls and urns is to envision all possible edges leaving u as a set of 2^k urns. In this model, the probability of a ball falling in each urn can be calculated via Lemma 2.4. The generation of the M edges in G corresponds to tossing M balls, but as these M balls are scattered over the entire adjacency matrix and not just the row for vertex u, we must condition on the number of balls that end up in this row. Using the vectors defined in Definition 2.10, we have the following results which provide an exact formula for the out-degree distribution of u in G'.

Theorem 4.4. *Given a vertex u, let $\bar{\mathbf{p}}_{\mathbf{u}}^{+} = \{p(u,v)/\sum_{w=0}^{2^k-1} p(u,w)\}_{v=0}^{2^k-1}$. The probability that a vertex u has out-degree d in G' is*

$$\mathbf{Pr}[d^{+}_{G'}(u) = d] = \sum_{j=0}^{M-d} \mathbf{Pr}[d^{+}_{G}(u) = d + j] U(d + j, 2^k, 2^k, \bar{\mathbf{p}}_{\mathbf{u}}^{+}, 2^k - d).$$

Proof. The vertex u has degree d in G' if it had degree $d + j$ in G *and* those $d + j$ edges went to precisely d distinct vertices. Since we are conditioning on the event that these $d + j$ edges are all of the form (u, w) for some $w \in \{0, 1, \ldots, 2^k - 1\}$, the probability that a particular one of these edges is of the form (u, v) is $p(u,v)/\sum_{w=0}^{2^k-1} p(u,w)$. Thus, $U(d + j, 2^k, 2^k, \bar{\mathbf{p}}_{\mathbf{u}}^{+}, 2^k - d)$ is the probability of getting d distinct ends from these $d + j$ edges. The result follows by considering separately each possible value of j from 0 to $M - d$. □

The analogous results for in-degree and total degree can be obtained using a nearly identical argument.

4.3 Limiting Degree Distributions in G'

Theorem 4.4 allows us to compute the probability that a given vertex u has out-degree d. However, for a graph with M edges, this computation involves a sum of $M-d$ terms involving very large binomial coefficients and summation over a large set of subsets. Thus, it is clear that we must turn to limiting distributions if we wish to obtain computationally tractable expressions for the degree distributions of very large graphs.

Given an occupancy problem with unequal urn probabilities, limiting distributions are known for the number of empty urns under a variety of conditions (see Chapter 6 of [11] for a survey of these kinds of results). In this subsection, we state a particular result of this kind and then prove that R-MAT satisfies the necessary conditions. Chistyakov [7] shows that if the urn probabilities are bounded in a particular fashion, and if the ratio of balls to urns approaches a constant as they tend to infinity together, then the probability distribution of the number of empty urns is asymptotically normal.

Theorem 4.5 (Chistyakov [7])**.** *Given a set of m urns with probabilities $\mathbf{q} = \{q_1, q_2, \ldots, q_m\}$ with $\sum_i^m q_i = 1$, let X be the random variable corresponding to the number of empty urns after tossing r balls into these urns. Then if $r, m \to \infty$ with $r/m \to C_1$ where $0 < C_1 < \infty$ and $m \cdot q_i \leq C_2 < \infty$ for each i, then the probability distribution of X is asymptotically normal.*

To apply Theorem 4.5, the quantity $m \cdot q_i$ must be uniformly bounded for all m urns as $m \to \infty$. However, in the event that only $m-1$ of the urns satisfy this bound, a straightforward modification to Chistyakov's proof demonstrates that the distribution for the number of empty urns among these $m-1$ urns remains asymptotically normal.

Corollary 4.6. *Given a set of m urns with probabilities $\mathbf{q} = \{q_1, q_2, \ldots, q_m\}$ where $\sum_i^m q_i = 1$, let Y be the random variable corresponding to the number of empty urns among the first $m-1$ of the m urns after tossing r balls into the set of all m urns. If $r, m \to \infty$ with $r/m \to C_1$ where $0 < C_1 < \infty$ and $m \cdot q_i \leq C_2 < \infty$ for $i = 1, 2, \ldots, m-1$, then Y is asymptotically normally distributed.*

We now prove that the limiting approximations given in Theorem 4.5 and Corollary 4.6 apply to graphs generated with R-MAT for nearly all choices of parameters. The proof is divided into two cases. The first case has $\alpha, \beta, \gamma, \delta \leq 1/2$ and the second case has $\max(\alpha, \beta, \gamma, \delta) > 1/2$.

Lemma 4.7. *Let G' be an R-MAT graph with $n = 2^k$ vertices and $p(e)$ denote the probability of edge e being generated in an iteration of the R-MAT algorithm. If $0 < \alpha, \beta, \gamma, \delta \leq 1/2$, then for any edge e, the quantity $n \cdot p(e)$ is uniformly bounded above by the constant 1.*

Proof. Without loss of generality, we assume $\alpha \geq \beta, \gamma, \delta$. By Lemma 2.4, we have $p(e) = \alpha^{e_\alpha} \beta^{e_\beta} \gamma^{e_\gamma} \delta^{e_\delta}$, with $e_\alpha + e_\beta + e_\gamma + e_\delta = k$. Then $p(e) \leq \alpha^k \leq 2^{-k}$, so $n \cdot p(e) \leq 2^k \cdot 2^{-k} = 1$, proving our result. □

To handle the case where the largest R-MAT parameter is greater than $1/2$, we require a result regarding the limiting behavior of sums of binomial coefficients. The following lemma can be proved by applying Chebyshev's Inequality (see [16], page 47).

Lemma 4.8. *For any $\epsilon > 0$ and $n \in \mathbb{N}$,*

$$\sum_{\{k:|k/n-1/2|\geq\epsilon\}} \binom{n}{k} 2^{-n} \leq \frac{1}{4n\epsilon^2}.$$

We require the following corollary to this result.

Corollary 4.9. *If $c > 1/2$ and $n \in \mathbb{N}$, then*

$$\lim_{n\to\infty} \sum_{k=0}^{cn} \binom{n}{k} 2^{-n} = 1.$$

In order to apply Corollary 4.6, we must now show that as $n \to \infty$, there is a uniform bound C so that for each vertex u the quantity $n \cdot p(u,v) \leq C < \infty$ for all v whenever one of $\alpha, \beta, \gamma, \delta$ is greater than $1/2$. The next result uses Corollary 4.9 to show that the proportion of vertices satisfying this bound with $C = 1$ tends to one as $n \to \infty$.

Lemma 4.10. *Let $p(u,v)$ denote the probability of edge (u,v) being generated in an iteration of the R-MAT algorithm where $n = 2^k$ vertices, $\min(\alpha,\beta,\gamma,\delta) > 0$, and $\max(\alpha,\beta,\gamma,\delta) > 1/2$. Let $\psi_n(\alpha,\beta,\gamma,\delta)$ be the number of vertices u so that for all v,*

$$n \cdot p(u,v) \leq 1.$$

Then

$$\lim_{n\to\infty} \frac{\psi_n(\alpha,\beta,\gamma,\delta)}{n} = 1.$$

Proof. Without loss of generality, let $0 < \beta, \gamma, \delta < \frac{1}{2} < \alpha < 1$ and let $\epsilon = \min(\beta,\gamma,\delta)$.
Claim: There exists a $\theta > 1/2$ such that

$$\alpha^x (1-\alpha-2\epsilon)^{(1-x)} \leq 1/2 \text{ for all } 0 \leq x \leq \theta. \tag{5}$$

Since $\alpha > 1/2$, we have $\alpha/(1-\alpha-2\epsilon) > 1$, and so $\alpha^x(1-\alpha-2\epsilon)^{(1-x)}$ is strictly increasing for $x \geq 0$. For those u with $u_z \leq \theta k$, note that $\beta, \gamma, \delta \leq 1-\alpha-2\epsilon$. We can then bound $p(u,v)$ as follows:

$$p(u,v) = \alpha^{e_\alpha}\beta^{e_\beta}\gamma^{e_\gamma}\delta^{e_\delta} \leq \alpha^{e_\alpha}(1-\alpha-2\epsilon)^{k-e_\alpha} \leq \alpha^{u_z}(1-\alpha-2\epsilon)^{k-u_z} \leq (\alpha^\theta(1-\alpha-2\epsilon)^{1-\theta})^k.$$

Assuming that the claim holds, it follows that

$$\{u : u_z \leq \theta k\} \subseteq \left\{u : p(u,v) \leq 2^{-k} \text{ for all } v\right\} = \{u : n \cdot p(u,v) \leq 1 \text{ for all } v\}. \tag{6}$$

Since the number of vertices with $u_z = i$ is $\binom{k}{i}$, we have

$$|\{u : u_z \leq \theta k\}| = \sum_{i=0}^{\lfloor \theta k \rfloor} \binom{k}{i}.$$

Together with (6), we see that

$$\sum_{i=0}^{\lfloor \theta k \rfloor} \binom{k}{i} 2^{-k} \leq \frac{\psi_n(\alpha, \beta, \gamma, \delta)}{n} \leq 1.$$

Since $\theta > 1/2$, Corollary 4.9 applies and it follows that $\frac{\psi_n(\alpha,\beta,\gamma,\delta)}{n}$ tends to 1 as n tends to infinity. We now need only to prove the claim.

Proof of Claim: Equality is achieved in (5) when $\theta = \frac{\log(2(1-\alpha-2\epsilon))}{\log((1-\alpha-2\epsilon)/\alpha)}$. As we have already seen that $\alpha^x(1-\alpha-2\epsilon)^{(1-x)}$ is uniformly increasing for $x \geq 0$, we must only show that this choice of θ is greater than $1/2$ for valid choices of α and ϵ:

$$\begin{aligned}
\tfrac{\log(2(1-\alpha-2\epsilon))}{\log((1-\alpha-2\epsilon)/\alpha)} > 1/2 \quad &\Leftrightarrow \\
\log\left(2(1-\alpha-2\epsilon)\right) < \log\left(\sqrt{\tfrac{1-\alpha-2\epsilon}{\alpha}}\right) \quad &\Leftrightarrow \\
\log\left(\tfrac{2(1-\alpha-2\epsilon)\sqrt{\alpha}}{\sqrt{1-\alpha-2\epsilon}}\right) < 0 \quad &\Leftrightarrow \\
2\sqrt{\alpha(1-\alpha-2\epsilon)} < 1 \quad &\Leftrightarrow \\
4\alpha(1-\alpha-2\epsilon) < 1 \quad &\Leftrightarrow \\
1/2 - \alpha/2 - 1/(8\alpha) < \epsilon.
\end{aligned}$$

The final inequality holds since the left hand side is negative for $1/2 < \alpha < 1$. □

Lemmas 4.7 and 4.10 imply that as long as the R-MAT parameters α, β, γ, and δ are all strictly positive, then for almost all vertices u, the limit $\lim_{n\to\infty} n \cdot p(u,v)$ is uniformly bounded above for all v. This allows us to apply Corollary 4.6 which leads to a proof of our main result, namely that the limiting distributions for the out-degree, in-degree, and total degree of a vertex u in G' are asymptotically normal.

Theorem 4.11. *Let u be a vertex in a graph G' generated by R-MAT with $n = 2^k$ vertices and $M = O(n)$ edges before removing duplicates. For all but a vanishing proportion of vertices u, as $n, M \to \infty$, the quantities $d^+_{G'}(u)$, $d^+_{G'}(u)$, and $d_{G'}(u)$ are asymptotically normally distributed.*

Proof. For out-degree, we treat each of the n possible edges (u, v) as an urn with probability $p(u, v)$ and have an additional urn representing all edges that do not begin at u. We envision tossing M balls into these $n + 1$ urns. Depending on the values of α, β, γ, and δ, either Lemma 4.7 or 4.10 implies that for all but a vanishing proportion of vertices u, the quantity $(n + 1) \cdot p(u, v)$ is uniformly bounded for every v. Since $M = O(n)$, we can apply Corollary 4.6 so that the the distribution of the number of empty urns in this model is asymptotically normal. We compute $\mathbf{E}[X]$ and $\mathbf{Var}[X]$ by accounting for all $n+1$ urns when using equations (1) and (2) and so we also use the probability vector $\hat{\mathbf{p}}^+_{\mathbf{u}}$ which accounts for the probability of the final urn. The number of edges originating at u that are present in G' is thus the random

variable $n+1-X$, which is normally distributed with mean $n+1-\mathbf{E}[X]$ and variance $\mathbf{Var}[X]$. The proof for in-degree is analogous. For total degree, we have a total of $2n-1$ possible edges as we count the uu edge only once. Envisioning now a total of $2n$ urns (with the last urn representing all edges of the form (v,w) for $v,w \neq u$), we again toss M balls into these urns and invoke Corollary 4.6 to show that the distribution of the number of empty urns is asymptotically normal and the result follows. □

We note that exact expressions for the mean and the variance for these limiting distributions can be computed more easily than one might expect. In the case of out-degree, for a particular value of $n = 2^k$, there are $k+1$ underlying distributions corresponding to the $k+1$ possible values of u_z. By also taking advantage of the fact that there are many duplicate probabilities one can simplify these calculations even further, and we find that the overall complexity of this calculation is only $O(k^5)$. Finally, by modifying a result given in [7], we are able to deduce approximations which require only $O(k^3)$ work to obtain the values required for the entire out-degree distribution. The contribution of the final urn to the mean and variance turns out to be negligible (since it is essentially always full), and by applying standard asymptotic expressions given in [7], we find that for fixed k and a particular value of u_z with $\theta^+_{i,j} = \alpha^i \beta^{u_z - i} \gamma^j \delta^{k-u_z-j}$, we can express the required mean and variance to compute the out-degree distribution as follows:

$$\mu(\mathbf{p_u}^+, M) = \sum_{i=0}^{u_z} \sum_{j=0}^{k-u_z} \binom{u_z}{i} \binom{k-u_z}{j} e^{-M\theta^+_{i,j}} + O(1), \text{ and} \tag{7}$$

$$\sigma(\mathbf{p_u}^+, M) = \sum_{i=0}^{u_z} \sum_{j=0}^{k-u_z} \binom{u_z}{i} \binom{k-u_z}{j} \left(e^{-M\theta^+_{i,j}} - e^{-2M\theta^+_{i,j}}\right) - M \left(\sum_{i=0}^{u_z} \sum_{j=0}^{k-u_z} \binom{u_z}{i} \binom{k-u_z}{j} \theta^+_{i,j} e^{-M\theta^+_{i,j}} \right)^2 + O(1). \tag{8}$$

An approximation for the in-degree mean and variance is obtained by replacing $\theta^+_{i,j}$ in the above expressions with $\theta^-_{i,j} = \alpha^i \gamma^{u_z - i} \beta^j \delta^{k-u_z-j}$ (note that if $\beta = \gamma$, then the in- and out-degree distributions are identical).

In order to obtain the limiting in-, out-, or total degree distribution of a randomly chosen vertex in G' with 2^k vertices, we construct a mixture of $k+1$ normal distributions, one for each possible value of $u_z = 0, 1, \ldots, k$. In this mixture, the distribution for $u_z = j$ is weighted by the probability that a randomly chosen vertex has j zeros in its binary representation, namely $\binom{k}{j}/2^k$. The resulting distribution typically exhibits "waves" created by summing up these weighted $k+1$ distributions that (usually) have different means and variances. Figure 2 shows the predicted and observed out- and in-degree distributions for 2048 randomly generated graphs with $n = 2^{12}$ nodes, $M = 2^{17}$ edges before removing duplicates, and $\alpha = .55, \beta = .15, \gamma = .1, \delta = .2$.

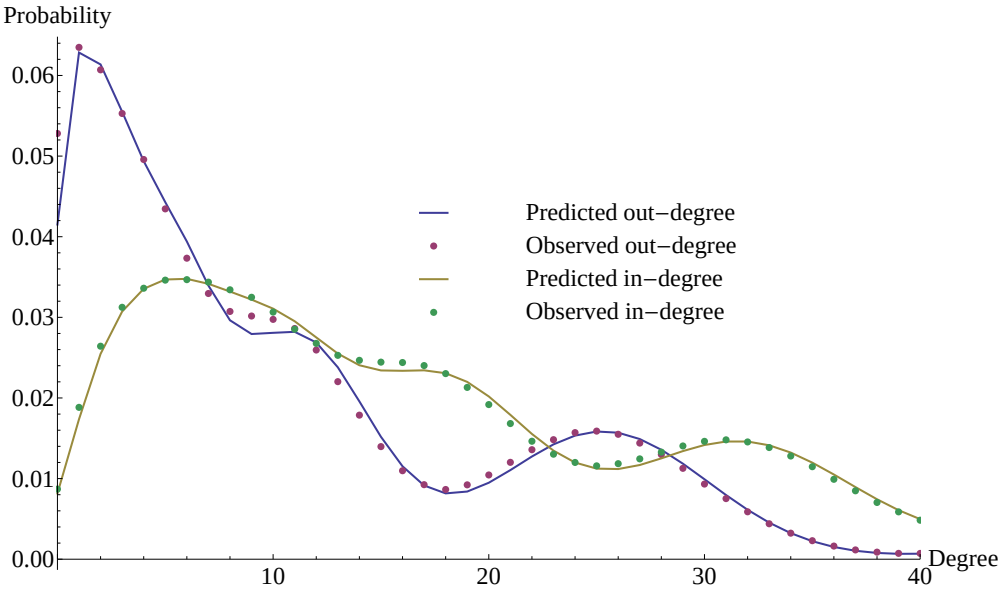

Figure 2: The observed and predicted out- and in-degree distributions for a set of 2048 random graphs generated with the same set of R-MAT parameters $\alpha = .55, \beta = .15, \gamma = .1, \delta = .2$

4.4 The number of edges in G'

A potential drawback of the R-MAT generator is that the number of edges in the final graph, M', is itself a random variable whose value is not realized until the generation is complete. We can treat this random variable by again using the ball and urn model of the occupancy problem. We now have one urn for each position in the adjacency matrix and we are tossing M balls into these urns. Using arguments similar to those used earlier in the paper, we can derive expressions for the mean and variance of M', allowing one to have an estimate of M' in terms of the R-MAT parameters prior to generating a random graph.

Theorem 4.12. *The expected number of edges in the graph G' containing $n = 2^k$ vertices generated with parameters $\alpha, \beta, \gamma, \delta$ and M edges before removing duplicates is given by*

$$\mathbf{E}[M'] = 4^k - \sum_{u_z=0}^{k} \binom{k}{u_z} \sum_{i=0}^{u_z} \sum_{j=0}^{k-u_z} \binom{u_z}{i} \binom{k-u_z}{j} (1 - \alpha^i \beta^{u_z - i} \gamma^j \delta^{k-u_z-j})^M. \tag{9}$$

Proof. First, note that the number of edges in G' can be obtained by summing up the out-degrees of all the vertices so that $M' = \sum_u d^+_{G'}(u)$. By linearity of expectation,

$$\mathbf{E}[M'] = \mathbf{E}\left[\sum_u d^+_{G'}(u)\right] = \sum_u \mathbf{E}\left[d^+_{G'}(u)\right].$$

For a given vertex u, let X_v be a random variable where $X_v = 1$ if the edge (u, v) exists in G and $X_v = 0$ otherwise for $v = 0, 1, \ldots, 2^k - 1$ so that

$$\mathbf{E}\left[d^+_{G'}(u)\right] = \mathbf{E}\left[X_0 + X_1 + \cdots + X_{2^k-1}\right] = \mathbf{E}\left[X_0\right] + \mathbf{E}\left[X_1\right] + \cdots + \mathbf{E}\left[X_{2^k-1}\right].$$

Noting that $\mathbf{E}\left[X_v\right]$ is one minus the probability that the edge (u, v) is never generated in M independent trials, we have $\mathbf{E}\left[X_v\right] = 1 - (1 - p(u, v))^M$ so that

$$\mathbf{E}\left[d^+_{G'}(u)\right] = 2^k - \sum_{v=0}^{2^k-1} (1 - p(u, v))^M. \tag{10}$$

However, the value of (10) is completely determined by u_z, so that we must only consider the $k + 1$ different values of u_z to compute $E[M']$. Since $p(u, v) = \alpha^{e_\alpha}\beta^{e_\beta}\gamma^{e_\gamma}\delta^{e_\delta}$ by Lemma 2.4 where $e_\alpha + e_\beta = u_z$ and $e_\gamma + e_\delta = k - u_z$, we can rewrite the sum in (10) by counting the occurrences of each possible value of $p(u, v)$, obtaining

$$\sum_{v=0}^{2^k-1} (1 - p(u, v))^M = \sum_{e_\alpha=0}^{u_z} \binom{u_z}{i} \sum_{e_\gamma=0}^{k-u_z} \binom{k - u_z}{j} (1 - \alpha^{e_\alpha}\beta^{u_z-e_\alpha}\gamma^{e_\gamma}\delta^{k-u_z-e_\gamma})^M.$$

The result follows since there are $\binom{k}{u_z}$ choices of u for each value of u_z. □

Theorem 4.12 allows one to compute the expected number of edges for an R-MAT graph on 2^k vertices by summing up $(k+1)(k+2)(k+3)/6$ values, providing an efficient method of computing the number of duplicate edges given the initial α, β, γ parameters.

An exact formula for the variance of M' is significantly more cumbersome, but we can apply the asymptotic formulas given in [7] to provide an efficient approximation for the variance similar to (8). For a fixed value of k and letting $\phi_{i,j,u_z} = \alpha^i\beta^{u_z-i}\gamma^j\delta^{k-u_z-j}$, an approximation for the variance of M' is given by

$$\begin{aligned}\mathbf{Var}\left[M'\right] \;=\; & \sum_{u_z=0}^{k} \sum_{i=0}^{u_z} \sum_{j=0}^{k-u_z} \binom{k}{u_z} \binom{u_z}{i} \binom{k - u_z}{j} e^{-M\phi_{i,j,u_z}} - e^{-2M\phi_{i,j,u_z}} \\ & -M \left(\sum_{u_z=0}^{k} \sum_{i=0}^{u_z} \sum_{j=0}^{k-u_z} \binom{k}{u_z} \binom{u_z}{i} \binom{k - u_z}{j} \phi_{i,j,u_z} e^{-M\phi_{i,j,u_z}} \right)^2 + O(1).\end{aligned} \tag{11}$$

In the general case when r balls are tossed into m urns, if $r/m \to 0$ and $r^2/m \to \infty$, then the limiting distribution of the number of empty (or full) urns is asymptotically normal. However, in our case with $r = c2^k$ balls and $m = 4^k$ urns, the limit of r^2/m tends to a constant so that the above result does not apply. A recent result of Hwang and Janson [10] implies that if we could show that $\mathbf{Var}\left[M'\right]$ tends to infinity with M, then M' is normally distributed. Whether or not this hypothesis holds for all choices of R-MAT parameters is still an open

problem. However, computational experiments suggest that M' is indeed normally distributed for graphs containing up to 2^{20} vertices. To empirically study the distribution of M', we generated 2^{16} random graphs with $n = 2^{20}$ vertices, $M = 8{*}2^{20}$ edges, and R-MAT parameters $\alpha = .55, \beta = .1, \gamma = .1, \delta = .25$. Using the approximation $(1-x)^M = e^{-Mx} + O(Mx^2)$ in (9), we predict that $\mathbf{E}[M'] = 8,266,452$ and (12) predicts a variance of $\mathbf{Var}[M'] = 139,619$. These compare quite favorably with the observed sample mean and variance of 8,266,453 and 139,266. Figure 3 shows a histogram of the observed values of M' versus a normal distribution with the predicted mean and variance.

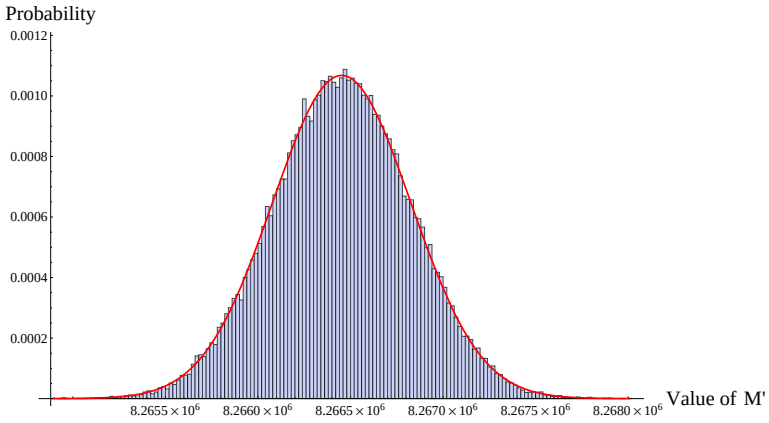

Figure 3: A histogram of the observed values of M' for 2^{16} random graphs generated with the same R-MAT parameters and $n = 2^{20}$ vertices. The solid line shows a normal distribution with the predicted parameters of $\mu = 8,266,452$ and $\sigma^2 = 139,619$ calculated via (9) and (12).

We performed a chi-square goodness-of-fit test to evaluate the null hypothesis that the observed values of M' are drawn from a normal distribution with the predicted mean and variance. We created bins of size 112 (chosen by using a common statistical rule of thumb) and calculated the test statistic to be $X^2 = 33.44$. Under the assumption that the null hypothesis is true, we expect this test statistic to be drawn from a chi-square distribution with 27 degrees of freedom. Comparing our computed value of X^2 with the upper critical values of this distribution, we fail to reject the null hypothesis at a significance level of .1, providing empirical confirmation of the hypothesis that M' is a normally distributed random variable for these parameters.

5 Computational Considerations

In this section, we discuss computational techniques we developed to compute the exact degree distributions, which may be of independent interest. These methods allowed us to calculate

the exact degree distribution of graphs with significantly less computational work than a naïve approach.

Theorem 4.4 provides a means to compute the exact degree distribution of a vertex in G'. However, as we noted earlier it is cumbersome and not very useful from a computational perspective. We observe that by using Corollary 4.3, we can derive an alternative expression for the out-degree distribution.

Theorem 5.1. *For a vertex u in a graph G' generated by R-MAT with $n = 2^k$ vertices and M edges before removing duplicates, parameters $\alpha, \beta, \gamma, \delta$ and probability vector $\hat{\mathbf{p}}_{\mathbf{u}}^{+} = \{p_1, p_2, \ldots, p_{2^k+1}\}$, the probability that u has out-degree d is*

$$\begin{aligned}\mathbf{Pr}[d^{+}_{G'}(u) = d] &= U(M, 2^k, 2^k + 1, \hat{\mathbf{p}}_{\mathbf{u}}^{+}, 2^k - d) \\ &= \sum_{i=0}^{2^k-d} \binom{d+i}{d} (-1)^i \sum_{\substack{\mathcal{A} \subseteq \{1,2,\ldots,2^k\} \\ |\mathcal{A}| = d+i}} \big(1 - \sum_{j \in \mathcal{A}} p_j\big)^M. \end{aligned} \tag{12}$$

Proof. Consider the set of $2^k + 1$ urns where the first 2^k correspond to all the possible edges leaving u, and the final urn corresponds to the set of $4^k - 2^k$ edges that are not of the form (u, v) for some v. This final urn receives a ball with probability $1 - \sum_v p(u, v) = 1 - (\alpha + \beta)^{u_z}(\gamma + \delta)^{k-u_z}$ (Lemma 2.6). Each of the M edges generated falls into exactly one of these urns, and by definition, $U(M, 2^k, 2^k + 1, \hat{\mathbf{p}}_{\mathbf{u}}^{+}, 2^k - d)$ gives the probability that exactly d of the first 2^k urns are non-empty. □

The formula given in Theorem 5.1 provides a more concise expression for the out-degree distribution for vertex u in G'. However, it is still a formidable computational task to calculate the entire probability distribution for a vertex u, even for small graphs. For example, suppose we have a graph with $k = 6$ (64 vertices), and we wish to compute $\mathbf{Pr}\left[d^{+}_{G'}(u) = 1\right]$ for some vertex u. To illustrate the magnitude of this calculation, when $i = 32$ for the sum in equation (12), we have to consider $\binom{64}{32} > 10^{18}$ subsets of $\{1, 2, \ldots, 64\}$ to obtain just this contribution to the sum.

However, since $p(u, v)$ assumes relatively few values as v ranges over all 2^k possible values (Lemma 2.11), many elements in the vector $\mathbf{p}_{\mathbf{u}}^{+}$ will be identical. Thus, when we consider all the subsets $\mathcal{A} \subseteq \{1, 2, \ldots, 2^k\}$ of a particular size s in equation (12), we expect that the quantity $\sum_{j \in \mathcal{A}} (1 - p_j)^M$ will assume only a small number of the $\binom{2^k}{s}$ possibilities. We can use this to our advantage when computing the exact distribution.

Definition 5.2. Given a vector of real numbers $\mathbf{v} = \{v_1, v_2, \ldots, v_n\}$, for $0 < i, j \leq n$, define $V(i, j, \mathbf{v})$ to be the set of ordered pairs $\{(w_1, n_1), (w_2, n_2), \ldots\}$ where the values of w_k are all distinct and represent all possible values obtained when summing up j values taken from $v_1, v_2, \ldots, v_i$ and n_k counts the number of subsets that sum to w_k.

Definition 5.3. Given some set of ordered pairs $V(i, j, \mathbf{v}) = \{(w_1, n_1), (w_2, n_2), \ldots, (w_m, n_m)\}$ and some real number z, we define $z + V(i, j, \mathbf{v})$ to be the set of ordered pairs $\{(w_1 + z, n_1), (w_2 + z, n_2), \ldots, (w_m + z, n_m)\}$.

For example, if $\mathbf{v} = \{0.2, 0.4, 0.2, 0.1\}$, then $V(4, 1, \mathbf{v}) = \{(0.2, 2), (0.4, 1), (0.1, 1)\}$ as these represent the distinct singleton elements of $\mathbf{v}$ and their multiplicities, and $V(3, 2, \mathbf{v}) = \{(0.6, 2), (0.4, 1)\}$ as we can form $0.6 = 0.4 + 0.2$ from both $v_1 + v_2$ and $v_2 + v_3$, and we can form $0.4 = 0.2 + 0.2$ from $v_1 + v_3$. Furthermore, $0.1 + V(3, 2, \mathbf{v})$ is defined to be the set $\{(0.7, 2), (0.5, 1)\}$.

Lemma 5.4. *Given a vector $\mathbf{v} = \{v_1, v_2, \ldots, v_n\}$, let $\{w_1, w_2, \ldots, w_m\}$ be the distinct values of $v_i \in \mathbf{v}$ and let $\{n_1, n_2, \ldots, n_m\}$ be their multiplicities. Then the sets $V(i, j, \mathbf{v})$ satisfy the recursion*

- $V(i, 0, \mathbf{v}) = \emptyset$ *and* $V(i, j, \mathbf{v}) = \emptyset$ *if* $j > i$.
- $V(i, 1, \mathbf{v}) = \{(w_1, n_1), (w_2, n_2), \ldots, (w_m, n_m)\}$,
- $V(n, n, \mathbf{v}) = \{(\sum_{i=1}^{n} v_i, 1)\}$,
- *For* $0 < i, j < n$, $V(i, j, \mathbf{v}) = V(i-1, j, \mathbf{v}) \bigcup v_i + V(i-1, j-1, \mathbf{v})$.

Proof. The first three claims follow directly from Definition 5.3. For the final claim, note that $V(i, j, \mathbf{v})$ can be divided into two disjoint sets: the set of all partial sums of j terms taken from $v_1, v_2, \ldots, v_{i-1}$ and the set of all partial sums of j terms taken from $v_1, v_2, \ldots, v_i$ where one of the terms is v_i. The first set is $V(i-1, j, \mathbf{v})$ and the second is $V(i-1, j-1, \mathbf{v}) + v_i$, proving the claim. □

In practice, we wish to minimize the number of ordered pairs required to represent $V(i, j, \mathbf{v})$. If we encounter pairs (w_s, n_s) and (w_t, n_t) with $w_s = w_t$ when we take the union of $V(i-1, j, \mathbf{v})$ and $v_i + V(i-1, j-1, \mathbf{v})$, then we "merge" these two pairs by replacing them with a single set $(w_s, n_s + n_t)$.

In order to obtain the entire exact out-degree probability distribution for a given u via equation (12), we must compute $\mathbf{Pr}\left[d^+_{G'}(u) = d\right]$ for $d = 0, 1, \ldots, 2^k$ which requires the computation of $\sum_{j \in \mathcal{A}} (1-p_j)^M$ for $\mathcal{A}$ ranging over all 2^{2^k} subsets of $\{1, 2, \ldots, 2^k\}$. In other words, for $\mathbf{p}^+_{\mathbf{u}}$ as given in Definition 2.10, we require the values of $V(2^k, 0, \mathbf{p}^+_{\mathbf{u}}), V(2^k, 1, \mathbf{p}^+_{\mathbf{u}}), \ldots, V(2^k, 2^k, \mathbf{p}^+_{\mathbf{u}})$. Algorithm 2 demonstrates how to use our recursion for V to minimize the amount of work required in computing this exact distribution.

In lines 3-4, we use the recursion to compute and store the contribution of each subset to the sum in Theorem 5.1. The individual probabilities are then computed in line 7 by utilizing the precomputed values.

Although the recursive computations required in Algorithm 2 can require quite a bit of memory, it has allowed us to compute the exact out-degree distribution for graphs where a direct approach is infeasible. For example, we used this procedure to calculate the entire out-degree distribution for a 64-node graph with $M = 128$ edges (before removing duplicates). As the out-degree distribution of a vertex u is completely determined by the value of u_z, we implemented the algorithm in *Mathematica* and ran it for the seven possible values of u_z, obtaining the entire distribution after about four hours and using about 3.5 GB of memory

Algorithm 2 Given a vertex u in an R-MAT graph with 2^k vertices and M edges before removing duplicates, calculate the exact out-degree distribution for u in the graph G' .

1: Compute the probability vector $\hat{\mathbf{p}}_{\mathbf{u}}^{+} = \{p_1, p_2, \ldots, p_{2^k+1}\}$ via Lemma 2.4.
2: **for** $j = 0$ to 2^k **do**
3: Construct the set $V(2^k, j, \hat{\mathbf{p}}_{\mathbf{u}}^{+}) = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$.
4: Set $S_j = \sum_{i=1}^{m} y_i(1 - x_i)^M$.
5: **end for**
6: **for** $d = 0$ to 2^k **do**
7: Compute $\mathbf{Pr}\left[d_{G'}^{+}(u) = d\right] = \sum_{j=0}^{d} (-1)^j \binom{2^k - d + j}{j} S_{2^k - d + j}$
8: **end for**

on a desktop machine. The naïve approach to computing this distribution requires one to perform roughly $O(2^{67})$ operations, a task that is certainly infeasible without substantial computational resources. These seven different distributions are shown in Figure 4 where the distribution for each value of u_z is weighted by $\binom{6}{u_z}$, the number of vertices that fall into this category. We also show the overall distribution for the out-degree of u.

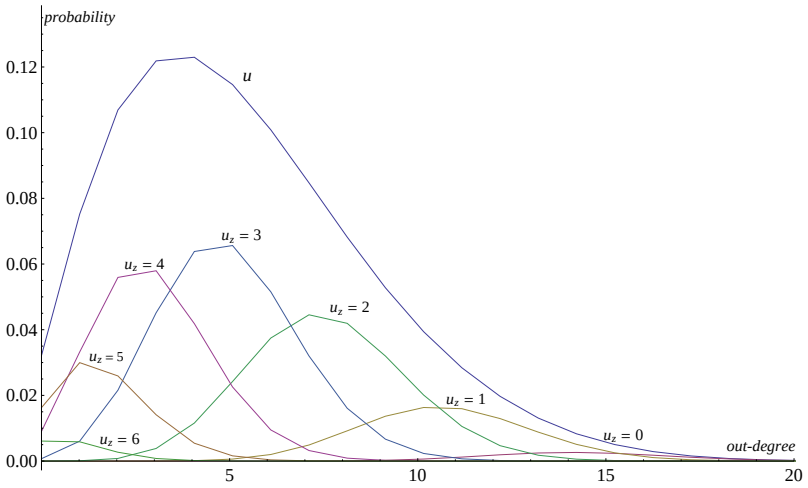

Figure 4: The exact out-degree distribution for the seven different values of u_z and the overall out-degree distribution for a 64-node graph with $M = 8{\cdot}64$ and $\alpha = .55, \beta = .1, \gamma = .1, \delta = .25$.

6 Conclusions

The R-MAT graph generator is widely used due to its simplicity and ease of implementation. By modeling the creation of edges as an occupancy problem, we have obtained exact and asymptotic formulas for the degree distributions of these graphs, as well as the mean and variance of the number of edges in the final graph. The asymptotic formulas can be calculated

quickly and allow a practitioner to determine the effect of initial parameter choices on the resulting graph. Finally, we have provided a computational technique that allows one to accelerate the computation of some exact probability distributions arising from our analysis.

7 Acknowledgements

This work was supported by the Department of Defense and used resources in the Extreme Scale System Center at Oak Ridge National Laboratory.

The investigation of the degree-distribution of the R-MAT generator grew out of a broader discussion of the SSCA#2 benchmark [1]. We would especially like to thank Richard Barrett and Jeff Kuehn at Oak Ridge National Laboratory for their work on testing the benchmark, and helpful discussions regarding the graphs arising from the R-MAT generator implemented in the benchmark. Additionally, conversations with Sheila Vaidya, Andy Yoo, and Yang Liu at Lawrence Livermore National Laboratory, Blair Perot from the University of Massachusetts-Amherst, and Michael Merrill were important in motivating and providing context for this work.

Finally, the authors thank Henry Cohn of Microsoft Research for pointing them in the right direction for proving the limiting behavior of the sum of binomial coefficients using Chebyshev's Inequality and Ed D'Azevedo of Oak Ridge National Laboratory and several anonymous referees for careful readings of the paper.

References

[1] D. Bader, J. Feo, J. Gilbert, J. Kepner, D. Koester, E. Loh, K. Madduri, B. Mann, and T. Meuse. The SSCA2 Benchmark, 2007.

[2] D. Bader and K. Madduri. GTgraph: A suite of synthetic graph generators, 2006.

[3] D. Bader and K. Madduri. A graph-theoretic analysis of the human protein-interaction network using multicore parallel algorithms. *Parallel Computing*, 34:627–639, 2008.

[4] Z. Bi and C. Faloutsos F. Korn. The "DGX" distribution for mining massive, skewed data. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 17–26. ACM, 2001.

[5] D. Chakrabarti and C. Faloutsos. Graph Patterns and the R-MAT Generator. In D. Cook and L. Holder, editors, *Mining Graph Data*, pages 65–95. John Wiley and Sons, 2007.

[6] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A Recursive Model for Graph Mining. In *SIAM International Conference on Data Mining*, 2004.

[7] V.P. Chistyakov. On the calculation of the power of the test of empty boxes. *Theory of Probability and its Applications*, 9:648–653, 1964.

[8] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume I.* John Wiley and Sons, third edition, 1968.

[9] Shawndra Hill and Akash Nagle. Social Network Signatures: A Framework for Re-Identification in Networked Data. *SSRN eLibrary*, 2009.

[10] H. Hwang and S. Janson. Local limit theorems for finite and infinite urn models. *The Annals of Probability*, 36(3):992–1022, 2008.

[11] N. Johnson and S. Kotz. *Urn Models and Their Applications.* John Wiley and Sons, 1977.

[12] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos. Realistic, mathematically tractable graph generation and evolution, using Kronecker multiplication. *Knowledge Discovery in Databases: PKDD 2005*, pages 133–145, 2005.

[13] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11:985–1042, 2010.

[14] M. Sasak, Liang Zhao, and H. Nagamochi. Security-aware beacon based network monitoring. In *Communication Systems, 2008. ICCS 2008. 11th IEEE Singapore International Conference on*, pages 527–531, Nov. 2008.

[15] Matthew C. Schmidt, Nagiza F. Samatova, Kevin Thomas, and Byung-Hoon Park. A scalable, parallel algorithm for maximal clique enumeration. *Journal of Parallel and Distributed Computing*, 69:417–428, 2009.

[16] A. Shiryaev. *Probability.* Springer-Verlag, second edition, 1996.

[17] Willi-Hans Steeb and Tan Kiat Shi. *Matrix Calculus and Kronecker product with applications and C++ programs*, chapter 2, page 55. World Scientific, 1997.