Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

EFFICIENT ESTIMATION OF TAIL PROBABILITIES OF THE TYPICAL DISTANCE IN PREFERENTIAL ATTACHMENT MODELS

Morgan R. Grant
Dirk P. Kroese

School of Mathematics and Physics
The University of Queensland
Brisbane, QLD 4072, AUSTRALIA

ABSTRACT

The properties of random graphs provide insight into the behavior of real-world complex networks. One such property is the Typical Distance, which characterizes the time required to traverse the network. For example, the Typical Distance measures how fast a virus spreads through a population. The probability that the Typical Distance is large is difficult to estimate via crude Monte Carlo. We propose a new sequential importance sampling estimator that can find the probability of a large Typical Distance in preferential attachment models, with a computational complexity that is quadratic in the number of nodes. Numerical experiments indicate that the estimator is significantly more efficient than crude Monte Carlo.

1 INTRODUCTION

The distance between two nodes in a complex network is of great interest for communication and social networks. For a communication network such as the Internet, this distance represents the delay between sending and receiving messages (Mao, Saul, and Smith 2006). In an epidemiological network this distance affects the time required for a disease to spread throughout the network (Kuperman and Abramson 2001). Evaluating the distance between nodes of a complex network is difficult because real-world complex networks are large, change frequently and usually cannot be measured accurately (Paxson 1997).

To overcome these difficulties, complex networks can be modelled with *random graphs*. For example the World Wide Web can be simulated with a *preferential attachment random graph* model (Barabási and Albert 1999), (Dorogovtsev, Mendes, and Samukhin 2000), and social networks can be simulated by the *exponential random graph* model (Robins et al. 2007). The study of random graphs as models for complex networks gained traction around the year 2000. This increase in activity motivated the investigation of random graph properties, such as connectivity (Mihail, Papadimitriou, and Saberi 2003), degree sequences (Bollobás et al. 2001) and cliques (Alon, Krivelevich, and Sudakov 1998). The property of focus for this paper is *Typical Distance*, which is defined as the average distance between any two nodes of a graph, given that a path between them exists. There exist results regarding the *expected* Typical Distance for popular random graph models (e.g, (Bollobás and Riordan 2004, Kuperman and Abramson 2001)), however no research has been conducted into the tail probability of the Typical Distance. Exact computation of this tail probability is intractable for large graphs. Furthermore, the probability that the Typical Distance exceeds some large threshold is often too small to estimate using crude Monte Carlo.

To find the probability of large Typical Distance within preferential attachment random graphs, we develop a new *sequential importance sampling* estimator. We also exploit a property of Typical Distance which allows the estimator to complete in $O(n^2)$ time, where n is the number of vertices in the random graph. Empirical tests show that the estimator is substantially more efficient than crude Monte Carlo.

The layout of the paper is as follows. Section 2 contains the formal graph theory definitions used throughout the paper. The section also states and proves a lemma that is essential for reducing the time complexity of the estimator constructed in Section 3. The focus of the paper is on the sequential importance sampling estimator developed in Section 3. This section provides the full algorithm for implementing the estimator. Numerical results are displayed and discussed in Section 4, before Section 5 concludes the paper.

2 GRAPH PRELIMINARIES

Let $\mathscr{G}$ be a *graph* with *vertex* set $\mathscr{V}_\mathscr{G} = \{1,2,\ldots,n\}$ and *edge* multiset $\mathscr{E}_\mathscr{G} = \{\{v_1,u_1\},\{v_2,u_2\},\ldots\}$, where the $v_i,u_i \in \mathscr{V}_\mathscr{G}$ may be equal. The *degree* of a vertex v is the number of edges incident with v and is written as $\deg_\mathscr{G}(v)$.

The *distance* between two vertices v and u in graph $\mathscr{G}$, denoted $d_\mathscr{G}(v,u)$, is the length of the shortest path from v to u. If there is no path from u to v, then by convention $d_\mathscr{G}(v,u) = \infty$. The set $\mathscr{C}_\mathscr{G}(v) = \{u \mid d_\mathscr{G}(v,u) < \infty\}$ is called the *component* of vertex v. Vertices u and v are said to be *connected* if $\mathscr{C}_\mathscr{G}(v) = \mathscr{C}_\mathscr{G}(u)$. As only undirected graphs will be considered in this paper, $u \in \mathscr{C}_\mathscr{G}(v)$ implies that $\mathscr{C}_\mathscr{G}(v) = \mathscr{C}_\mathscr{G}(u)$.

The *Typical Distance from v* is defined as

$$D_\mathscr{G}(v) = \mathbb{E}[d_\mathscr{G}(v,U) \mid \mathscr{C}_\mathscr{G}(v) = \mathscr{C}_\mathscr{G}(U)], \tag{1}$$

where U is a vertex chosen uniformly from $\mathscr{V}$. We define the *Typical Distance* of $\mathscr{G}$ as

$$D(\mathscr{G}) = \mathbb{E}[d_\mathscr{G}(V,U) \mid \mathscr{C}_\mathscr{G}(V) = \mathscr{C}_\mathscr{G}(U)], \tag{2}$$

where V and U are vertices chosen independently and uniformly from $\mathscr{V}$. Hence the Typical Distance of a graph is the expected distance between a pair of connected vertices.

The following lemma will be useful in Section 3.

Lemma 1 Consider graphs $\mathscr{G}$ and $\mathscr{H}$, where $\mathscr{H}$ is formed by adding an edge $\{z,w\}$ between two disconnected components $\mathscr{C}_\mathscr{G}(z)$ and $\mathscr{C}_\mathscr{G}(w)$. Then if $\mathscr{C}_\mathscr{G}(z)$ and $\mathscr{C}_\mathscr{G}(w)$ have m and n vertices respectively,

$$D_\mathscr{H}(v) = \frac{mD_\mathscr{G}(v) + n(D_\mathscr{G}(w) + d_\mathscr{G}(v,z) + 1)}{(m+n)}, \quad v \in \mathscr{C}_\mathscr{G}(z). \tag{3}$$

Proof. Let U be a uniformly chosen vertex from $\mathscr{H}$. Then,

$$\begin{aligned}
D_\mathscr{H}(v) &= \mathbb{E}[d_\mathscr{H}(v,U) \mid \mathscr{C}_\mathscr{H}(v) = \mathscr{C}_\mathscr{H}(U)] \\
&= \frac{1}{m+n} \sum_{u \in \mathscr{V}_\mathscr{H}} d_\mathscr{H}(v,u) \\
&= \frac{1}{m+n} \left(\sum_{u \in \mathscr{C}_\mathscr{G}(z)} d_\mathscr{G}(u,v) + \sum_{u \in \mathscr{C}_\mathscr{G}(w)} \{d_\mathscr{G}(v,z) + 1 + d_\mathscr{G}(w,u)\} \right) \\
&= \frac{1}{m+n} \left(mD_\mathscr{G}(v) + n(d_\mathscr{G}(u,z)+1) + \sum_{u \in \mathscr{C}_\mathscr{G}(w)} d_\mathscr{G}(w,u) \right) \\
&= \frac{mD_\mathscr{G}(v) + n(d_\mathscr{G}(u,z) + D_\mathscr{G}(w) + 1)}{m+n}.
\end{aligned}$$

□

2.1 Graph Processes

The random graph models of interest in this paper are constructed via *graph processes*. A graph process is a discrete-time process $(G_t, t = 0,1,\ldots)$ where each G_t is a random multigraph. An example of a graph

model which is constructed via graph processes is the *preferential attachment* (PA) model (Dorogovtsev, Mendes, and Samukhin 2000, van der Hofstad 2014).

The preferential attachment model will be the focus of this paper. In its simplest form, the model depends on two parameters: the number of vertices, n, and an *attachment weighting* parameter $\alpha \geq -1$. We write $\mathrm{PA}_n(\alpha)$ for the corresponding model. The special case $\mathrm{PA}_n(0)$ is the *Barabási–Albert* random graph model (Barabási and Albert 1999). Figure 1 shows typical realizations of PA_{50} for different values of α.

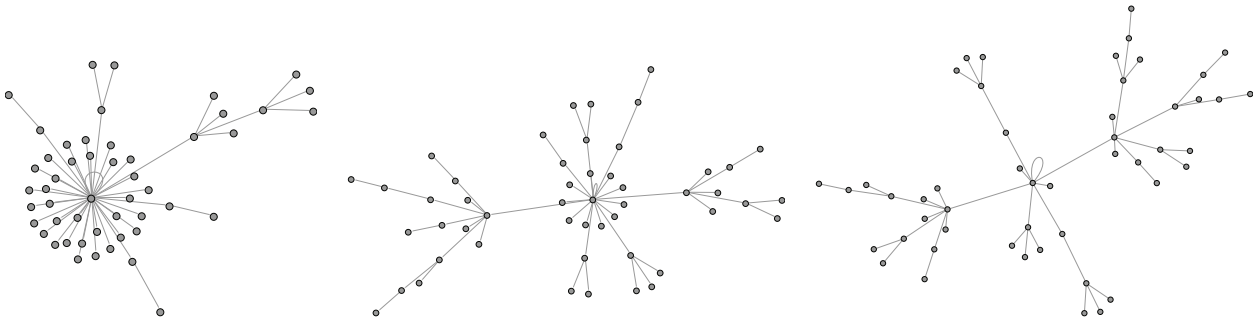

Figure 1: Examples of Preferential Attachment random graphs. The left, center and right graphs are $\mathrm{PA}_{50}(-0.75)$, $\mathrm{PA}_{50}(0)$ and $\mathrm{PA}_{50}(3)$, respectively.

The model is constructed via a graph process $(G_t, t = 0, 1, \ldots)$ using the following procedure. The process starts with G_0 empty. To advance the process from any G_t, a vertex labelled $t+1$ is added, followed by an edge $\{t+1, V_{t+1}\}$, where V_{t+1} is a vertex chosen randomly from the current vertex set $\{1, \ldots, t+1\}$ according to the probability

$$\mathbb{P}(V_{t+1} = v \mid G_t = \mathscr{G}_t) = \frac{\deg_{\mathscr{G}_t}(v) + \alpha + \delta_{v,(t+1)}}{t(2+\alpha) + 1 + \alpha} \quad v = 1, \ldots, t+1, \tag{4}$$

where $\delta_{i,j}$ is the Kronecker delta. This new graph with additional edge and vertex is G_{t+1}. Once $t = n$, we deliver G_n as a $\mathrm{PA}_n(\alpha)$ random graph.

From the definition above, knowledge of V_{t+1} and G_t implies complete knowledge of G_{t+1}. Likewise, if G_{t+1} is known, V_{t+1} can be identified as the vertex that is adjacent to vertex $t+1$. Clearly this implies that there exists a one-to-one correspondence between G_t and $\mathbf{V}_t = (V_1, \ldots, V_t)$.

To efficiently simulate a preferential attachment graph, we can use Algorithm 1. The algorithm is generalized from the one presented in (Newman 2003) to allow for $\alpha \neq 0$. To do this, we note that the density can be rewritten as

$$\mathbb{P}(V_{t+1} = v \mid G_t = \mathscr{G}_t) = \begin{cases} \frac{1+\alpha}{t(2+\alpha)+1+\alpha} + \frac{\deg_{\mathscr{G}_t}(v)-1}{t(2+\alpha)+1+\alpha} & \text{If } v < t+1 \\ \frac{1+\alpha}{t(2+\alpha)+1+\alpha} & \text{If } v = t+1, \end{cases} \tag{5}$$

thus the generation of V_{t+1} can be separated into two cases. The first has V_{t+1} proportional to $\deg_{\mathscr{G}_t}(v) - 1$ and the second has V_{t+1} chosen uniformly from vertices $v \leq t+1$. Which case to use is decided by Bernoulli trial with weight $\theta = t/(t(2+\alpha)+1+\alpha)$ and success indicating the first case. By storing V_s for all $s \leq t$

in an array which is updated after each edge is added, randomly selecting one of the entries of the array uniformly can be used to efficiently select a vertex proportional to $\deg_{\mathscr{G}_t}(v) - 1$.

Algorithm 1: $\mathrm{PA}_n(\alpha)$

Input: Number of vertices n, attachment weight α
Output: $\mathrm{PA}_n(\alpha)$ graph G_n
Initialize G_0 as a graph with n vertices and no edges
Construct G_1 by adding edge $\{1,1\}$ to G_0
$A \leftarrow [1]$
$t \leftarrow 1$
for $t < n$ **do**
  $U \leftarrow$ Uniform real number between 0 and 1
  $\theta \leftarrow t/(t(2+\alpha)+1+\alpha)$
  if $U < \theta$ **then**
    $X \leftarrow$ Uniform integer from $\{0,1,\ldots,\mathrm{size}(A)-1\}$
    $V \leftarrow A[X]$
  else
    $V \leftarrow$ Uniform integer from $\{1,2,\ldots,t+1\}$
  end
  Add edge $\{t+1,V\}$ to G_t to form G_{t+1}
  $A[t] \leftarrow V$
  $t \leftarrow t+1$
end
return G_n

3 TYPICAL DISTANCE ESTIMATION VIA SEQUENTIAL IMPORTANCE SAMPLING

The aim is to evaluate $\ell = \mathbb{P}(D(G) > \gamma)$ for γ large, where G is a $\mathrm{PA}_n(\alpha)$ random graph. In this section we develop a *Sequential Importance Sampling* (SIS) estimator that provides an accurate estimation in reasonable time. For more information regarding sequential importance sampling, we refer the reader to (Kroese, Taimre, and Botev 2011).

Recall that the vector of random vertices $\mathbf{V}_t = (V_1,\ldots,V_n)$ uniquely determines G_t. Hence there exists a function h such that $G_t = h(\mathbf{V}_t)$. Furthermore denote $f_{t+1}(v_{t+1} \mid \mathbf{v}_t) = \mathbb{P}(V_{t+1} = v_{t+1} \mid G_t = h(\mathbf{v}_t))$ for $t = 1,2,\ldots,n$. For convenience, set $S(\mathbf{V}_n) = D(h(\mathbf{V}_n))$.

With $\ell = \mathbb{P}(S(\mathbf{V}_n) > \gamma)$, we have

$$\ell = \mathbb{E}_f \, \mathbb{I}\{S(\mathbf{V}_n) > \gamma\}. \tag{6}$$

Here the subscript f indicates that the expectation is taken with respect to

$$f(\mathbf{v}_n) = f_1(v_1) f_2(v_2 \mid \mathbf{v}_1) \cdots f_n(v_n \mid \mathbf{v}_{n-1}). \tag{7}$$

SIS involves sampling from an *importance sampling* density $g(\mathbf{v}_n) = g_1(v_1) g_2(v_2 \mid \mathbf{v}_1) \cdots g_n(v_n \mid \mathbf{v}_{n-1})$ where g satisfies

$$g(\mathbf{v}_n) = 0 \implies \mathbb{I}\{S(\mathbf{V}_n) > \gamma\} f(\mathbf{v}_n) = 0. \tag{8}$$

By taking the expectation with respect to g, we can represent ℓ as

$$\ell = \mathbb{E}_g \left[\mathbb{I}\{S(\mathbf{V}_n) > \gamma\} \prod_{t=1}^{n} \frac{f(V_{t+1} \mid \mathbf{V}_t)}{g(V_{t+1} \mid \mathbf{V}_t)} \right]. \tag{9}$$

Assume it is easy to simulate from $g_1(v_1)$ to find V_1, and then simulate from $g_2(v_2 \mid v_1)$ to find V_2 and so on, until we have $\mathbf{V}_n$ from $g(\mathbf{v}_n)$. As the nominal pdf f has conditionals $f_t(v_t \mid \mathbf{v}_{t-1})$ that are easy to evaluate, we arrive at the following unbiased SIS estimator:

$$\hat{\ell} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{S(\mathbf{V}_n^{(i)}) > \gamma\} \prod_{t=1}^{n} \frac{f_{t+1}(V_{t+1}^{(i)} \mid \mathbf{V}_t^{(i)})}{g_{t+1}(V_{t+1}^{(i)} \mid \mathbf{V}_t^{(i)})}, \tag{10}$$

where $V_1^{(1)}, \ldots, V_1^{(N)}$ are independent samples from g_1 and $V_t^{(1)}, \ldots, V_t^{(N)}$ are independent samples from the respective conditional densities $g_t(\,\cdot \mid \mathbf{V}_{t-1}^{(1)}), \ldots, g_t(\,\cdot \mid \mathbf{V}_{t-1}^{(N)})$.

The remaining problem is to select $g_1, g_2, \ldots, g_n$ such that the variance of the estimator is minimized. We propose the following family of importance sampling densities:

$$g_{t+1}(v_{t+1} \mid \mathbf{v}_t) \propto \left(\deg_{G_t}(v_{t+1}) + \alpha + \delta_{v_{t+1},t+1}\right) \times \max\{1, D_{G_t}(v_{t+1})^p\}, \tag{11}$$

where $p \geq 0$ is a parameter of the density and $G_t = h(\mathbf{v}_t)$. Essentially this density means that for each new edge added to the graph, it is more likely that the edge will be incident with a vertex that has high distance from other vertices. Equation 3 shows that this will produce graphs with higher typical distance, therefore sampling from g means the event $D(G) > \gamma$ is more likely to occur. For $p = 0$ the estimator is identical to crude Monte Carlo (CMC), which will be useful in analysing the performance of the estimator.

3.1 Efficient Typical Distance Evaluation

The importance sampling density defined in (11) requires an expensive Typical Distance calculation to find $D_{G_t}(v)$ for every vertex v at each time t. A naive implementation of the estimator would find the Typical Distance using a *breadth first search* (BFS) from each vertex. Here BFS requires $O(n)$ time and there are at most n vertices, so evaluating the Typical Distance takes $O(n^2)$ time. The Typical Distance would need to be found before evaluating $g_{t+1}(v_{t+1} \mid \mathbf{v}_t)$ at each stage, hence it must be computed n times. Thus a naive implementation would take $O(n^3)$ time to execute.

Compare this against CMC, which only requires a single Typical Distance calculation after all edges are added. So CMC requires $O(n^2)$ time, which shows that a naive approach to implementing the SIS estimator is inadequate. We introduce an update scheme which allows the SIS estimator to avoid redundant calculations and achieve $O(n^2)$ time.

The update scheme works as follows. A vector $\mathbf{c}$ tracks the component of each vertex and a vector $\mathbf{b}$ maintains the value of $D_{G_t}(v)$ for all vertices v. The two vectors are initialized to $c_v = v$ and $b_v = 0$. If $c_v = c_u$ then vertex v and u are in the same component. Say an edge $\{u, w\}$ is added to G_t, with $u \leq w$. It is clear that $\mathscr{C}_{G_t}(w)$ contains only w for the preferential attachment model defined in Section 2. Hence, by Lemma 1, the value of b_w is updated to

$$b_w \leftarrow \frac{|\mathscr{C}_{G_t}(u)|(1 + b_u)}{|\mathscr{C}_{G_t}(u)| + 1}$$

and for all v such that $c_v = c_u$, we set

$$b_v \leftarrow \frac{|\mathscr{C}_{G_t}(u)|}{|\mathscr{C}_{G_t}(u)| + 1} b_v + \frac{1}{|\mathscr{C}_{G_t}(u)| + 1}(1 + d_{G_t}(v, u)).$$

Note that we use $\mathbf{c}$ to determine $\mathscr{C}_{G_t}(v)$ and $|\mathscr{C}_{G_t}(v)|$ in the above equations. Furthermore, $d_{G_t}(v, u)$ can be evaluated for all necessary v in a single breadth first search. After updating the $\mathbf{b}$ vector, $\mathbf{c}$ is updated

by setting $c^* \leftarrow \min\{c_u, c_w\}$ and then $c_v \leftarrow c^*$ for all v such that $c_v = c_u$ or $c_v = c_w$. The full algorithm incorporating this update scheme is shown in Algorithm 2.

Algorithm 2: SIS estimator for $\mathbb{P}(D(G) > \gamma)$

Input: Number of vertices n, parameter α, parameter p, threshold γ
Output: Estimate of $\mathbb{P}(D(G) > \gamma)$ for G in $\mathrm{PA}_n(\alpha)$
$W \leftarrow 0$
for $i = 1$ *to* N **do**
  Construct G_0 as a graph with n vertices
  $w = 1$
  $\mathbf{b} \leftarrow \mathbf{0}$
  for $j = 1$ *to* n **do**
    $c_j = j$
  end
  for $t = 0$ *to* $n-1$ **do**
    $V_t \leftarrow$ vertex chosen randomly with density $g_{t+1}(v \mid \mathbf{V}_t)$
    $w \leftarrow w \times \frac{f_{t+1}(V_{t+1} \mid \mathbf{V}_t)}{g_{t+1}(V_{t+1} \mid \mathbf{V}_t)}$
    Construct G_{t+1} by adding edge $\{t+1, V\}$ to G_t to make G_{t+1}
    UpdateVector$(\mathbf{c}, \mathbf{b}, G_{t+1}, V, t+1)$
  end
  if $(\sum_{j=1}^{n} b_j)/n > \gamma$ **then**
    $W \leftarrow W + w$
  end
end
return W/N

Algorithm 3: UpdateVector

Input: Component vector $\mathbf{c}$, Typical Distance vector $\mathbf{b}$, graph $\mathscr{G}$, vertex u, vertex w (assume $u < w$)
Output: Updated vectors $\mathbf{c}$ and $\mathbf{b}$.
$m \leftarrow$ number of elements of $\mathbf{c}$ with $c_v = c_u$
$b_w \leftarrow \frac{m}{m+1}(1 + b_u)$
$\Delta \leftarrow$ result from BFS starting at u (so $\Delta_v = d_t(u, v)$)
for v *in* $\{j \mid c_j = c_u\}$ **do**
  $b_v \leftarrow \frac{m}{m+1} b_v + \frac{1}{m+1}(1 + \Delta_v)$
end
$c_w = c_u$

The asymptotic time complexity of the SIS estimator with this update scheme in on par with CMC.

Proposition 1 Algorithm 2 has asymptotic time complexity $O(n^2)$, where n is the number of vertices in the random graph.

Proof. There are n iterations for a $\mathrm{PA}_n(\alpha)$ random graph. At each iteration the vertex V_t must be selected and vectors $\mathbf{b}$ and $\mathbf{c}$ must be updated. The subroutine for choosing V_t in Algorithm 2 loops over vertices $1, \ldots, t$. As $t \leq n$, this subroutine is $O(n)$. Within the same iteration we perform single BFS which takes $O(n)$ time. Counting the size of the components from $\mathbf{c}$ and then updating every element of $\mathbf{b}$ and $\mathbf{c}$ requires two separate loops over both vectors, and hence requires $O(n)$ time. As all other operations are not dependent on n, the full algorithm requires $O(n(n+n+n)) = O(n^2)$ time to run. □

4 NUMERICAL RESULTS

Tables 1 through 4 demonstrate the performance of the estimator. Here N is the number of graphs generated per estimate. To combine the relative error and time results, we record the empirical *efficiency* of each estimator (Handscomb and Hammersley 1964), defined as

$$\text{Efficiency}(\hat{\ell}) = \frac{s_{\hat{c}}^2 \, \mathrm{T}_{\hat{c}}}{s_{\hat{\ell}}^2 \, \mathrm{T}_{\hat{\ell}}}, \tag{12}$$

where $\hat{\ell}$ is the estimator of interest, $\hat{c}$ is the crude Monte Carlo estimator for the same quantity, s is the standard error of the estimator and T is the average time required per estimate. It is clear from Table 1 that the estimator outperforms CMC for appropriate parameter selection. Setting the parameter p too high causes the estimator to increase in variance, as seen in Table 1. For the problem described in Table 1 the optimal value for p appears to be between 1.2 and 2.0. Finding the optimal value can be done through a pilot run with fewer samples. Empirical tests show that the estimator is stable, so increasing the number of runs by a factor of K reduces the relative error by a factor of $\sqrt{K}$. Prior to running the trials for Table 2, 3 and 4, pilot runs showed that $p = 1$ was a reasonable choice for all three cases.

Table 1: Results for $n = 50, \alpha = 0, N = 10^7, \gamma = 6.0$.

p	0	0.4	0.8	1.2	1.6	2.0	CMC
Mean	1.80e-6	1.28e-6	1.34e-6	1.31e-6	1.30e-6	1.45e-6	1.40e-6
Relative Error	0.236	0.085	0.065	0.048	0.037	0.111	0.267
Time (seconds)	14325	13159	13201	13778	13531	14507	13735
Efficiency	0.75	12.25	19.33	35.08	60.15	5.19	1.00

Table 2: Results for $n = 25$, $\alpha = 0$, $N = 10^7$. For SIS, $p = 1$.

γ	Method	Estimate	Relative Error	Time (seconds)	Efficiency
4.50	CMC	5.02e-5	0.045	4202	1.00
	SIS	4.98e-5	0.010	3812	22.28
4.75	CMC	6.60e-6	0.123	3135	1.00
	SIS	6.88e-6	0.022	3326	28.108
5.00	CMC	5.00e-7	0.447	3110	1.00
	SIS	9.00e-7	0.052	3685	19.20

Table 3: Results for $n = 25$, $\alpha = -0.75$, $N = 10^7$. For SIS, $p = 1$.

γ	Method	Estimate	Relative Error	Time (seconds)	Efficiency
3.75	CMC	4.77e-5	0.046	4493	1.00
	SIS	4.18e-5	0.013	3922	18.43
4.00	CMC	4.40e-6	0.150	4117	1.00
	SIS	4.43e-6	0.031	3596	26.44
4.25	CMC	6.00e-7	0.408	3140	1.00
	SIS	4.07e-7	0.079	3972	45.23

Table 4: Results for $n = 25$, $\alpha = 3$, $N = 10^7$. For SIS, $p = 1$.

γ	Method	Estimate	Relative Error	Time (seconds)	Efficiency
5.00	CMC	5.15e-5	0.044	3172	1.0
	SIS	5.06e-5	0.009	3406	19.49
5.25	CMC	9.50e-6	0.102	2559	1.0
	SIS	7.96e-6	0.022	3118	26.11
5.50	CMC	1.30e-6	0.277	2527	1.0
	SIS	1.19e-6	0.052	3157	26.82

Finally, we see in Figure 2 that the use of an update scheme is essential for ensuring the SIS estimator is viable for large n.

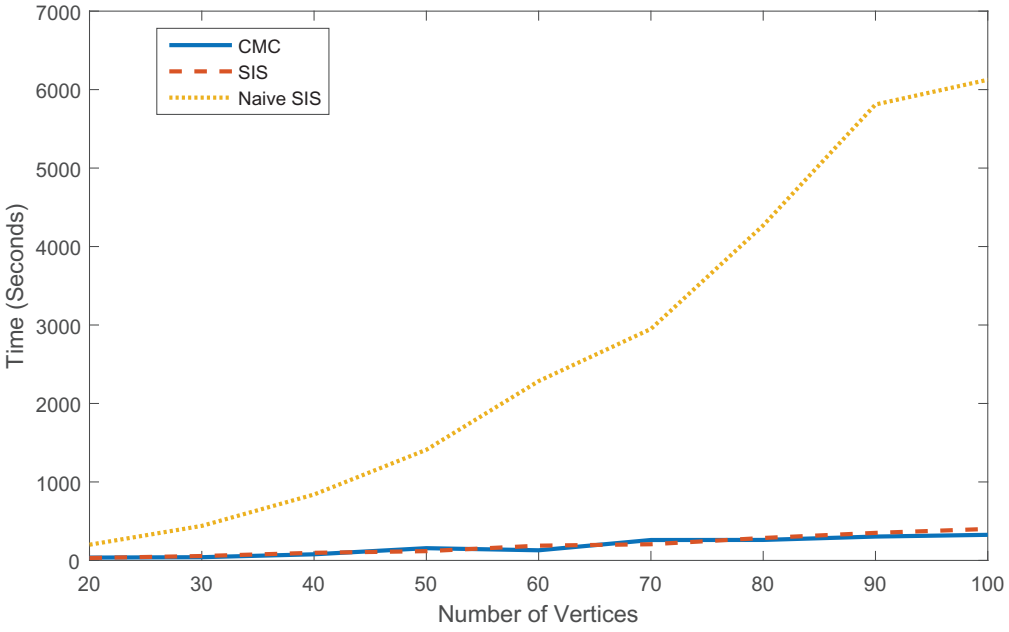

Figure 2: Time required for each algorithm to complete. Here $\alpha = 0$, $N = 10000$. Naive SIS is the implementation of the SIS algorithm described at the start of Section 3.1.

5 CONCLUSION

Here we have provided an efficient estimator for the probability that the Typical Distance between two nodes is high in a preferential attachment random graph. By using SIS with a simple update mechanism we get an $O(n^2)$ estimator which is substantially more efficient than CMC. For CMC to achieve the same relative error as SIS on graphs of fifty vertices, it would require at least fifty times as many samples.

Future work includes finding an estimator to handle preferential attachment models which permit multiple edges to be added at each time step. In addition, combining the SIS estimator with splitting to create a sequential importance resampling estimator would further enhance the estimator, assuming that the extra calculations do not slow the algorithm down considerably.

ACKNOWLEDGMENTS

This work was supported by the Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers, under grant number CE140100049.

REFERENCES

Alon, N., M. Krivelevich, and B. Sudakov. 1998. "Finding a Large Hidden Clique in a Random Graph". *Random Structures and Algorithms* 13 (3-4): 457–466.

Barabási, A.-L., and R. Albert. 1999. "Emergence of Scaling in Random Networks". *Science* 286 (5439): 509–512.

Bollobás, B., and O. Riordan. 2004. "The Diameter of a Scale-Free Random Graph". *Combinatorica* 24 (1): 5–34.

Bollobás, B., O. Riordan, J. Spencer, G. Tusnády et al. 2001. "The Degree Sequence of a Scale-Free Random Graph Process". *Random Structures & Algorithms* 18 (3): 279–290.

Dorogovtsev, S. N., J. F. F. Mendes, and A. N. Samukhin. 2000, Nov. "Structure of Growing Networks with Preferential Linking". *Phys. Rev. Lett.* 85:4633–4636.

Handscomb, D., and J. Hammersley. 1964. *Monte Carlo Methods*. Methuen.

Kroese, D. P., T. Taimre, and Z. I. Botev. 2011. *Handbook of Monte Carlo Methods*. New York: John Wiley & Sons.

Kuperman, M., and G. Abramson. 2001. "Small World Effect in an Epidemiological Model". *Physical Review Letters* 86 (13): 2909.

Mao, Y., L. K. Saul, and J. M. Smith. 2006. "IDES: An Internet Distance Estimation Service for Large Networks". *Selected Areas in Communications, IEEE Journal on* 24 (12): 2273–2284.

Mihail, M., C. Papadimitriou, and A. Saberi. 2003. "On Certain Connectivity Properties of the Internet Topology". In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, 28–35. IEEE.

Newman, M. E. 2003. "The Structure and Function of Complex Networks". *SIAM review* 45 (2): 167–256.

Paxson, V. 1997. "End-to-End Routing Behavior in the Internet". *Networking, IEEE/ACM Transactions on* 5 (5): 601–615.

Robins, G., P. Pattison, Y. Kalish, and D. Lusher. 2007. "An Introduction to Exponential Random Graph (p*) Models for Social Networks". *Social networks* 29 (2): 173–191.

van der Hofstad, Remco 2014. "Random Graphs and Complex Networks". http://www.win.tue.nl/~rhofstad/NotesRGCN.pdf.

AUTHOR BIOGRAPHIES

MORGAN R. GRANT is a PhD student in the School of Mathematics and Physics at The University of Queensland, where he obtained his honours degree in mathematics. His research interests include random graph models, rare event simulation techniques and applications of Monte Carlo simulation to solve real world problems. His email address is morgan.grant@uqconnect.edu.au.

DIRK P. KROESE is a professor of Mathematics and Statistics at The University of Queensland. He is the co-author of several influential monographs on simulation and Monte Carlo methods, including Handbook of Monte Carlo Methods and Simulation and the Monte Carlo Method, (3rd Edition). Dirk is a pioneer of the well-known Cross-Entropy method — an adaptive Monte Carlo technique, invented by Reuven Rubinstein, which is being used around the world to help solve difficult estimation and optimization problems in science, engineering, and finance. His personal website can be found under http://www.maths.uq.edu.au/~kroese/. His email address is kroese@maths.uq.edu.au.