A POLYNOMIAL-TIME APPROXIMATION ALGORITHM FOR ALL-TERMINAL NETWORK RELIABILITY

HENG GUO AND MARK JERRUM

Abstract. We give a fully polynomial-time randomized approximation scheme (FPRAS) for the all-terminal network reliability problem, which is to determine the probability that, in a undirected graph, assuming each edge fails independently, the remaining graph is still connected. Our main contribution is to confirm a conjecture by Gorodezky and Pak (*Random Struct. Algorithms*, 2014), that the expected running time of the "cluster-popping" algorithm in bi-directed graphs is bounded by a polynomial in the size of the input.

1. Introduction

Network reliability problems are extensively studied #**P**-hard problems [Col87] (see also [BP83, PB83, KL85, Bal86]). In fact, these problems are amongst the first of those shown to be #**P**-hard, and the two-terminal version is listed in Valiant's original thirteen [Val79]. The general setup is that in a given (undirected or directed) graph, every edge (or arc) e has an independent probability p_e to fail, and we are interested in various kinds of connectivity notions of the remaining graph. For example, the two-terminal connectedness [Val79] asks for the probability that for two vertices s and t, s is connected to t in the remaining graph, and the (undirected) all-terminal network reliability asks for the probability of all vertices being connected after edges fail. The latter can also be viewed as a specialization of the Tutte polynomial $T_G(x, y)$ with $x = 1$ and $y > 1$, yet another classic topic whose computational complexity is extensively studied [JVW90, VW92, GJ08, GJ14].

Prior to our work, the approximation complexity of network reliability problems remained elusive despite their importance. There is no known efficient approximation algorithm (for any variant), but nor is there any evidence that such an algorithm does not exist. A notable exception is Karger's fully polynomial-time randomized approximation scheme (FPRAS) for (undirected) all-terminal network *unreliability* [Kar99] (see also [HS14, Kar16, Kar17] for more recent developments). Although approximating unreliability is potentially more useful in practice, it does not entail an approximation of its complement.

In this paper, we give an FPRAS for the all-terminal network reliability problem, defined below and denoted Reliability.

Name: Reliability
Instance: A (undirected) graph $G = (V, E)$, and failure probabilities $\mathbf{p} = (p_e)_{e \in E}$.
Output: $Z_{rel}(G; \mathbf{p})$, which is the probability that if each edge e fails with probability p_e, the remaining graph is connected.

When p_e is independent of e, Reliability is an evaluation of the Tutte polynomial. The *Tutte polynomial* is a two-variable polynomial $T_G(x, y)$ associated with a graph G, which encodes much interesting information about G. As (x, y) ranges over $\mathbb{R}^2$ or $\mathbb{C}^2$ we obtain a family of graph parameters, the so-called *Tutte plane*. As already noted, the study of the computational complexity of these parameters has a long history. Reliability with a uniform failure probability $0 < p < 1$ is equivalent to evaluating the Tutte polynomial $T_G(x, y)$ on the line $x = 1$ and $y = \frac{1}{p} > 1$. Our algorithm is the first positive result on the complexity of the Tutte plane since Jerrum and Sinclair presented an FPRAS for the partition function of the ferromagnetic Ising model, which is equivalent to the Tutte polynomial on the positive branch of the hyperbola

The work described here was supported by the EPSRC research grant EP/N004221/1 "Algorithms that Count".

$(x-1)(y-1) = 2$ [JS93]. It also answers a well-known open problem from 1980s, when the #**P**-hardness of Reliability was established [Jer81, PB83] and the study of approximate counting initiated. This problem is explicitly proposed in, for example, [Wel93, Conjecture 8.7.11] and [Kar99]. We note that many conjectures by Welsh ([Wel93, Chapter 8.7] and [Wel99]) remain open, and we hope that our work is only a beginning to answering these questions.

Another related and important reliability measure is *reachability*, introduced and studied by Ball and Provan [BP83]. A directed graph $G = (V, A)$ with a distinguished root r is said to be *root-connected* if all vertices can reach r. Reachability, denoted $Z_{reach}(G, r; \mathbf{p})$ for failure probabilities $\mathbf{p} = (p_e)_{e \in A}$, is the probability that, if each arc e fails with probability p_e independently, the remaining graph is still root-connected.

We define the computational problem formally.

Name: Reachability
Instance: A directed graph $G = (V, A)$ with root r, and failure probabilities $\mathbf{p} = (p_e)_{e \in A}$.
Output: $Z_{reach}(G, r; \mathbf{p})$.

Exact polynomial-time algorithms are known when the graph is acyclic [BP83] or has a small number of cycles [Hag91]. However, in general the problem is #**P**-hard [PB83].

Ball [Bal80] showed that Reliability is equivalent to Reachability in bi-directed graphs. A bi-directed[1] graph is one where every arc has an anti-parallel twin with the same failure probability. It is shown [Bal80] that $Z_{rel}(G; \mathbf{p}) = Z_{reach}(\overrightarrow{G}, r; \mathbf{p}')$, where $\overrightarrow{G}$ and $\mathbf{p}'$ are obtained by replacing every undirected edge in G with a pair of anti-parallel arcs having the same failure probability in either direction, and r is chosen arbitrarily. See Lemma 12.

Our FPRAS for Reliability utilizes this equivalence via approximating Reachability in bi-directed graphs. The core ingredient is the "cluster-popping" algorithm introduced by Gorodezky and Pak [GP14]. The goal is to sample root-connected subgraphs with probability proportional to their weights, and then the reduction from counting to sampling is via a sequence of contractions. A cluster is a subset of vertices not including the root and without any out-going arc. The sampling algorithm randomizes all arcs independently, and then repeatedly resamples arcs going out from minimal clusters until no cluster is left, at which point the remaining subgraph is guaranteed to be root-connected. This approach is similar to Wilson's "cycle-popping" algorithm [Wil96] for rooted spanning trees, and to the "sink-popping" algorithm [CPP02] for sink-free orientations. Gorodezky and Pak [GP14] have noted that cluster-popping can take exponential time in general, but they conjectured that in bi-directed graphs, the algorithm runs within polynomial-time.

We confirm this conjecture. Let $p_{\max}$ be the maximum failure probability of edges (or arcs). Let m be the number of edges (or arcs) and n the number of vertices.

Theorem 1. *There is an FPRAS for* Reliability *(or equivalently,* Reachability *in bi-directed graphs). The expected running time is* $O\left(\varepsilon^{-2}(1-p_{\max})^{-3}m^2n^3\right)$ *for an* $(1 \pm \varepsilon)$*-approximation. There is also an exact sampler to draw (edge-weighted) connected subgraphs with expected running time* $O\big((1-p_{\max})^{-1}m^2n\big)$*.*

We analyze the cluster-popping algorithm [GP14] under the *partial rejection sampling* framework [GJL17], which is a general approach to sampling from a product distribution conditioned on avoiding a number of "bad" events. Partial rejection sampling is inspired by the Moser-Tardos algorithm for the Lovász Local Lemma [MT10]. It starts with randomizing all variables independently, and then gradually eliminating "bad" events. At every step, we need to find an appropriate set of variables to resample. We call an instance *extremal* [KS11, She85], if any two bad events are either disjoint or independent. For extremal instances, the resampling set can be simply chosen to be the set of all variables involved in occurring bad events [GJL17], and the algorithm becomes exactly the same as the Moser-Tardos resampling algorithm [MT10]. In particular, all three "popping" algorithms [Wil96, CPP02, GP14] are special cases of partial

[1] There are other definitions of "bi-directed graphs" in the literature. Our definition is sometimes also called a symmetric directed graph.

rejection sampling for extremal instances. In case of cluster-popping, the bad events are exactly minimal clusters.

The advantage of the partial rejection sampling treatment is that we have an explicit formula for the expected number of resampling events for any extremal instance [KS11, GJL17], which equals to the ratio between the probability of having exactly one bad event and the probability of avoiding all bad events. In order to bound this ratio, we use a combinatorial encoding idea and design a mapping from subgraphs with a unique minimal cluster to root-connected subgraphs. To make this mapping injective, we record an extra vertex and an arc so that we can recover the pre-image. This extra cost is upper-bounded by a polynomial in the size of the graph.

Cluster-popping only draws root-connected subgraphs in the bi-directed setting. In order to sample connected subgraphs in the undirected setting, we provide an alternative proof of the equivalence between Reliability and Reachability in bi-directed graphs, which essentially is a coupling argument. This coupling has a new consequence that, once we have a sample of a root-connected subgraph, it is easy to generate a connected subgraph according to the correct distribution.

In Section 2 we introduce the cluster-popping algorithm and the partial rejection framework. In Section 3 we analyze its running time in bi-directed graphs. For completeness, in Section 4 we include the approximate counting algorithm due to Gorodezky and Pak [GP14]. In Section 5 we give a coupling proof of the equivalence between Reliability and Reachability in bi-directed graphs. In Section 6 we use our sampling algorithm to show how to approximately count the number of connected subgraphs of a fixed size. In Section 7 we conclude by mentioning a few open problems.

2. Cluster-popping

Let $G = (V, A)$ be a directed[2] graph with root r. The graph G is called *root-connected* if there is a directed path in G from every non-root vertex to r. Let $0 < p_e < 1$ be the failure probability of arc e, and define the weight of a subgraph S to be $\mathrm{wt}(S) := \prod_{e\in S}(1-p_e)\prod_{e\notin S} p_e$. Then reachability, $Z_{reach}(G, r; \mathbf{p})$, is defined as follows,

$$Z_{reach}(G, r; \mathbf{p}) := \sum_{\substack{S\subseteq A\\ (V,S)\text{ is root-connected}}} \mathrm{wt}(S).$$

Here, $\mathbf{p} = (p_e : e \in A)$ denotes the vector of failure probabilities.

Let $\pi_G(\cdot)$ (or $\pi(\cdot)$ for short) be the distribution resulting from choosing each arc e independently with probability $1 - p_e$, and conditioning on the resulting graph being root-connected. In other words, the support of $\pi(\cdot)$ is the collection of all root-connected subgraphs, and the probability of each subgraph S is proportional to its weight $\mathrm{wt}(S)$. Then $Z_{reach}(G, r; \mathbf{p})$ is the normalizing factor of the distribution $\pi(\cdot)$. Gorodezky and Pak [GP14] have shown that approximating $Z_{reach}(G, r; \mathbf{p})$ can be reduced to sampling from $\pi(\cdot)$ when the graph is bi-directed.

The cluster-popping algorithm of Gorodezky and Pak [GP14], to sample root-connected subgraphs from $\pi(\cdot)$, can be viewed as a special case of partial rejection sampling [GJL17] for extremal instances. With every arc e of G we associate a random variable that records whether that arc has failed. Bad events are characterized by the following notion of clusters.

Definition 2. *In a directed graph (V, A) with root r, a subset $C \subseteq V$ of vertices is called a* cluster *if $r \notin C$ and there is no arc $u \to v \in A$ such that $u \in C$ and $v \notin C$.*

We say C is a minimal cluster *if C is a cluster and for any proper subset $C' \subset C$, C' is not a cluster.*

If (V, A) contains no cluster, then it is root-connected. For each vertex v, let $A_{\text{out}}(v)$ be the set of outgoing arcs from v. We also abuse the notation to write $A_{\text{out}}(S) = \bigcup_{v\in S} A_{\text{out}}(v)$ for a subset $S \subset V$ of vertices. Notice that $A_{\text{out}}(S)$ contains edges between vertices inside S. To "pop" a cluster C, we re-randomize all arcs in $A_{\text{out}}(C)$. However, re-randomizing clusters does not yield the desired distribution. We will instead re-randomize minimal clusters.

[2] It is easy to see that in a undirected graph, reachability is the same as all-terminal reliability.

Claim 3. *Any minimal cluster is strongly connected.*

Proof. Let C be a minimal cluster, and $v \in C$ be an arbitrary vertex in C. We claim that v can reach all vertices of C. If not, let C' be the set of reachable vertices of v and $C' \subsetneq C$. Since C' does not have any outgoing arcs, C' is a cluster. This contradicts to the minimality of C. □

Claim 4. *If C_1 and C_2 are two distinct minimal clusters, then $C_1 \cap C_2 = \emptyset$.*

Proof. By Claim 3, C_1 and C_2 are both strongly connected components. If $C_1 \cap C_2 \neq \emptyset$, then they must be identical. □

For every subset $C \subseteq V$ of vertices, we define a bad event B_C, which occurs if C is a minimal cluster. Observe that B_C relies only on the status of arcs in $A_{\mathrm{out}}(C)$. Thus, if $C_1 \cap C_2 = \emptyset$, then B_{C_1} and B_{C_2} are independent, even if some of their vertices are adjacent. By Claim 4, we know that two bad events B_{C_1} and B_{C_2} are either independent or disjoint. Thus the aforementioned extremal condition is met. Moreover, it was shown [GJL17, Theorem 8] that if the instance is extremal, then at every step, we only need to resample variables involved in occurring bad events. This leads to the cluster-popping algorithm of Gorodezky and Pak [GP14], which is formally described in Algorithm 1.

Algorithm 1 Cluster Popping

```
Let $S$ be a subset of arcs by choosing each arc $e$ with probability $1 - p_e$ independently.
while There is a cluster in $(V, S)$. do
    Let $C_1, \dots, C_k$ be all minimal clusters in $(V, S)$, and $C = \bigcup_{i=1}^k C_i$.
    Re-randomize all arcs in $A_{\mathrm{out}}(C)$ to get a new $S$.
end while
return $S$
```

The correctness of Algorithm 1 is first shown by Gorodezky and Pak [GP14]. It can also be easily verified using [GJL17, Theorem 8].

Theorem 5 ([GP14, Theorem 2.2])**.** *The output of Algorithm 1 is drawn from π_G.*

An advantage of thinking in the partial rejection sampling framework is that we have a closed form formula for the expected running time of these algorithms on extremal instances. Let Ω_k be the collection of subgraphs with k minimal clusters, and

$$Z_k := \sum_{S \in \Omega_k} \mathrm{wt}(S).$$

Then $Z_0 = Z_{reach}(G, r; \mathbf{p})$, since any subgraph in Ω_0 has no cluster and is thus root-connected.

Theorem 6 ([GJL17])**.** *Let T be the number of resampled events of the partial rejection sampling algorithm for extremal instances. Then*

$$\mathbb{E}\, T = \frac{Z_1}{Z_0}.$$

In particular, for Algorithm 1, T is the number of popped clusters.

Theorem 6 can be shown via manipulating generating functions. The less-than-or-equal-to direction of Theorem 6 was shown by Kolipaka and Szegedy [KS11], which is the direction we will need later. The other direction is useful to show running-time lower bounds, but that is not our focus in this paper.

3. Running time of Algorithm 1 in bi-directed graphs

Gorodezky and Pak [GP14] have given examples of directed graphs in which Algorithm 1 requires exponential time. In the following we focus on bi-directed graphs. A graph G is called *bi-directed* if $u \to v$ is present in G, then $v \to u$ is present in G as well, and the failure probabilities are the same for these two arcs. We use Bi-directed Reachability to denote

Reachability in bi-directed graphs. For an arc $e = u \to v$, let $\overline{e} := v \to u$ denote its reverse arc. Then in a bi-directed graph, $p_e = p_{\overline{e}}$.

Lemma 7. *Let $G = (V, A)$ be a root-connected bi-directed graph with root r. We have that $Z_1 \le \max_{e\in A} \left\{\frac{p_e}{1-p_e}\right\} mnZ_0$, where $n = |V|$, and $m = |A|$.*

Proof. We construct an injective mapping $\varphi : \Omega_1 \to \Omega_0 \times V \times A$. For each subgraph $S \in \Omega_1$, $\varphi(S)$ is defined by "repairing" S so that no minimal cluster is present. We choose in advance an arbitrary ordering of vertices and arcs. Let C be the unique minimal cluster in S and v be the first vertex in C. Let R denote the set of all vertices which can reach the root r in the subgraph S. Since $S \in \Omega_1$, $R \ne V$. Let $U = V \setminus R$. Since G is root-connected, there is an arc in A from U to R. Let $u \to u'$ be the first such arc, where $u \in U$ and $u' \in R$. We let

$$\varphi(S) := (S_{\text{fix}}, v, u \to u'),$$

where $S_{\text{fix}} \in \Omega_0$ is defined next. Figure 1 is an illustration of these objects.

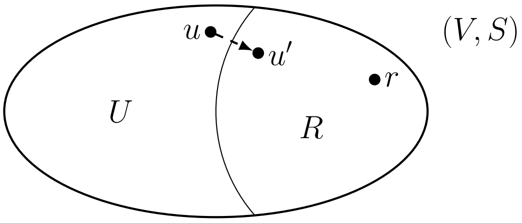

Figure 1. An illustration of R, U, and $u \to u'$.

Consider the subgraph $H = (U, S[U])$, where

$$S[U] := \{x \to y \mid x \in U,\ y \in U,\ x \to y \in S\}.$$

We consider the directed acyclic graph (DAG) of strongly connected components of H, and call it $\widehat{H}$. (We use the decoration $\widehat{}$ to denote arcs, vertices, etc. in $\widehat{H}$.) To be more precise, we replace each strongly connected component by a single vertex. For a vertex $w \in U$, let $[w]$ denote the strongly connected component containing w. For example, $[v]$ is the same as the minimal cluster C by Claim 3. We may also view $[w]$ as a vertex in $\widehat{H}$ and we do not distinguish the two views. The arcs in $\widehat{H}$ are naturally induced by $S[U]$. Namely, for $[x] \ne [y]$, an arc $[x] \to [y]$ is present in $\widehat{H}$ if there exists $x' \in [x]$, $y' \in [y]$ such that $x' \to y' \in S$.

We claim that $\widehat{H}$ is root-connected with root $[v]$. This is because $[v]$ must be the unique sink in $\widehat{H}$ and $\widehat{H}$ is acyclic. If there is another sink $[w]$ where $v \notin [w]$, then $[w]$ is a minimal cluster in H. This contradicts $S \in \Omega_1$.

Since $\widehat{H}$ is root-connected, there is at least one path from $[u]$ to $[v]$. Let $\widehat{W}$ denote the set of vertices of $\widehat{H}$ that can be reached from $[u]$ in $\widehat{H}$ (including $[u]$), and $W := \{x \mid [x] \in \widehat{W}\}$. Then W is a cluster and $[u]$ is the unique source in $\widehat{H}[\widehat{W}]$. As $\widehat{H}$ is root-connected, $[v] \in \widehat{W}$. Define

$$S_{\text{flip}} := \left\{x \to y \;\middle|\; [x] \ne [y],\ x, y \in W, \text{ and } x \to y \in S\right\},$$

which is the set of edges to be flipped. Notice that $S[W]$ is different from S_{flip}, namely all arcs that are inside strongly connected components are ignored in S_{flip}. Now we are ready to define S_{fix}. We reverse all arcs in S_{flip} and add the arc $u \to u'$ to fix the minimal cluster. Formally, let

$$S_{\text{fix}} := S \cup \{u \to u'\} \cup \{y \to x \mid x \to y \in S_{\text{flip}}\} \setminus S_{\text{flip}}.$$

Figure 2 is an example of these objects we defined.

Let $\widehat{H}_{\text{fix}}$ be the graph obtained from $\widehat{H}$ by reversing all arcs induced by S_{flip}. Observe that $[u]$ becomes the unique sink in $\widehat{H}_{\text{fix}}[\widehat{W}]$ (and $[v]$ becomes the unique source).

We verify that $S_{\text{fix}} \in \Omega_0$. For any $x \in R$, x can still reach r in (V, S_{fix}) since the path from x to r in (V, S) is not changed. Since $u \to u' \in S_{\text{fix}}$, u can reach $u' \in R$ and hence r. For any

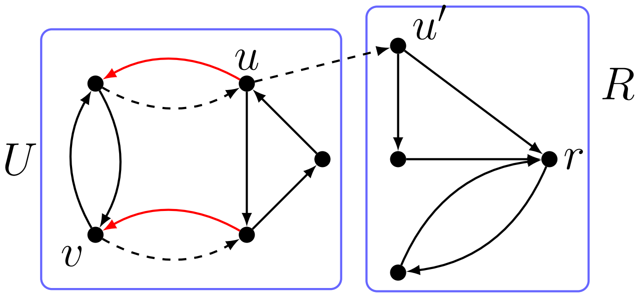

Figure 2. An example of S_{flip} (red arcs) in the subgraph (V, S). Dashed arcs are to be added to S_{fix}. The underlying graph has more arcs than are drawn here.

$y \in W$, y can reach u as $[u]$ is the unique sink in $\widehat{H}_{\mathrm{fix}}[\widehat{W}]$. For any $z \in U \setminus W$, z can reach $v \in W$ since the path from z to v in (V, S) is not changed.

Next we verify that φ is injective. To do so, we show that we can recover S given S_{fix}, $u \to u'$, and v. First remove $u \to u'$ from S_{fix}. The set of vertices which can reach r in $(V, S_{\mathrm{fix}} \setminus \{u \to u'\})$ is exactly R in (V, S). Namely we can recover U and R. As a consequence, we can recover all arcs in S that are incident with R, as these arcs are not changed.

What is left to do is to recover arcs in $S[U]$. To do so, we need to find out which arcs have been flipped. We claim that $\widehat{H}_{\mathrm{fix}}$ is acyclic. Suppose there is a cycle in $\widehat{H}_{\mathrm{fix}}$. Since $\widehat{H}$ is acyclic, the cycle must involve flipped arcs and thus vertices in $\widehat{W}$. Let $[x] \in \widehat{W}$ be the lowest one under the topological ordering of $\widehat{H}[\widehat{W}]$. Since $\widehat{W}$ is a cluster, the outgoing arc $[x] \to [y]$ along the cycle in $\widehat{H}_{\mathrm{fix}}$ must have been flipped, implying that $[y] \in \widehat{W}$ and $[y] \to [x]$ is in $\widehat{H}[\widehat{W}]$. This contradicts to the minimality of $[x]$.

Since $\widehat{H}_{\mathrm{fix}}$ is acyclic, the strongly connected components of $H_{\mathrm{fix}} := (U, S_{\mathrm{fix}}[U])$ are identical to those of $H = (U, S[U])$. Hence contracting all strongly connected components of H_{fix} results in exactly $\widehat{H}_{\mathrm{fix}}$. All we need to recover now is the set $\widehat{W}$. Let $\widehat{W}'$ be the set of vertices reachable from $[v]$ in $\widehat{H}_{\mathrm{fix}}$. It is easy to see that $\widehat{W} \subseteq \widehat{W}'$. We claim that actually $\widehat{W} = \widehat{W}'$. For any $[x] \in \widehat{W}'$, there is a path from $[v]$ to $[x]$ in $\widehat{H}_{\mathrm{fix}}$. Suppose $[x] \notin \widehat{W}$. Since $[v] \in \widehat{W}$, we may assume that $[y]$ is the first vertex along the path such that $[y] \to [z]$ where $[z] \notin \widehat{W}$. Thus $[y] \to [z]$ has not been flipped and is present in $\widehat{H}$. However, this contradicts the fact that $\widehat{W}$ is a cluster in $\widehat{H}$.

To summarize, given S_{fix}, $u \to u'$, and v, we may uniquely recover S. Hence the mapping φ is injective. Moreover, flipping arcs does not change the weight as $p_e = p_{\overline{e}}$, and only adding the arc $u \to u'$ would. We have that $\mathrm{wt}(S_{\mathrm{fix}}) = \frac{1-p_{u\to u'}}{p_{u\to u'}}\mathrm{wt}(S)$. The lemma follows. □

We remark that an alternative way of repairing S in the proof above is to reverse all arcs in $S[W]$ without defining S_{flip}. The key point is that doing so leaves the strongly connected components intact. However this makes the argument less intuitive.

Let $p_{\max} = \max_{e \in A} p_e$. Combining Theorem 6 and Lemma 7, we have the following theorem. Notice that for each popping, we resample only a subset of arcs.

Theorem 8. *Let T be the expected number of popped clusters in Algorithm 1. For a root-connected bi-directed graph $G = (V, A)$, $\mathbb{E}\,T \le \frac{p_{\max}}{1-p_{\max}} mn$, where $n = |V|$, and $m = |A|$. The expected running time is asymptotically at most $\frac{p_{\max}}{1-p_{\max}} m^2 n$.*

4. Approximate counting

We include the approximate counting algorithm of Gorodezky and Pak [GP14] for completeness. Let $G = (V, A)$ be an instance of Bi-directed Reachability with root r and parameters $\mathbf{p}$. We construct a sequence of graphs $G_0, .., G_{n-1}$ where $n = |V|$ and $G_0 = G$. Given G_{i-1}, choose two arbitrary adjacent vertices u_i and v_i, remove all arcs between u_i and v_i (in either direction), and identify u_i and v_i to get $G_i = (V_i, A_i)$. Namely we contract all arcs between u_i and v_i, but parallel arcs in the resulting graph are preserved. If one of u_i and v_i is r, the new vertex is labelled r. Thus $G_{n-1} = (\{r\}, \emptyset)$. Since A_i is always a subset of A, we denote by $\mathbf{p}_i$ the parameters $\mathbf{p}$ restricted to A_i.

For $i = 1, \ldots, n-1$, define a random variable R_i as follows:

$$R_i := \begin{cases} 1 & (V_{i-1}, S_{i-1}) \text{ is root-connected in } G_{i-1}; \\ 0 & \text{otherwise}, \end{cases}$$

where $S_{i-1} \subset A_{i-1}$ is a random root-connected subgraph drawn from the distribution $\pi_{G_i}(\cdot)$, together with all arcs e between u_i and v_i added independently with probability $1 - p_e$. It is easy to see that

$$\mathbb{E}\, R_i = \frac{Z_{reach}(G_{i-1}, r; \mathbf{p}_{i-1})}{Z_{reach}(G_i, r; \mathbf{p}_i)},$$

and

$$Z_{reach}(G, r; \mathbf{p}) = \prod_{i=1}^{n-1} \mathbb{E}\, R_i.$$

Let $p_{\max} = \max_{e \in A} p_e$ and $s = \lceil 5(1 - p_{\max})^{-2}(n-1)\varepsilon^{-2} \rceil$ where s is the desired precision. We estimate $\mathbb{E}\, R_i$ by the empirical mean of s independent samples of Z_i, denoted by $\widetilde{R}_i$, and let $\widetilde{Z} = \prod_{i=1}^{n-1} \widetilde{R}_i$ and $Z = Z_{reach}(G, r; \mathbf{p})$. Gorodezky and Pak [GP14] showed the following.

Proposition 9 ([GP14, Section 9])**.** $\Pr\left(\left| Z - \widetilde{Z} \right| > \varepsilon Z \right) \le 1/4$.

In order to sample Z_i, we use Algorithm 1 to draw independent samples of root-connected subgraphs. Theorem 8 implies that each sample takes at most $\frac{p_{\max}}{1-p_{\max}} m^2 n$ time in expectation. We need $O\left(\frac{n}{\varepsilon^2 (1-p_{\max})^2} \right)$ samples for each Z_i. Putting everything together, we obtain the following theorem.

Theorem 10. *There is an FPRAS for* Bi-directed Reachability*. The expected running time is $O\left(\varepsilon^{-2} (1 - p_{\max})^{-3} m^2 n^3 \right)$ for an $(1 \pm \varepsilon)$-approximation.*

A natural question is what if $1 - p_{\max}$ is close to 0. We can answer this in the case of uniform failure probabilities, i.e., when $p_e = p$ for all $e \in A$. Let $p^* = 1 - 1/(3m)$. If $p \le p^*$ then run Algorithm 1 as usual, to produce a sample in time $O(m^3 n)$. Otherwise, run the algorithm with modified weights $p_e = p^*$ for all $e \in A$, and let the output be S, so that (V, S) is a root-connected subgraph. Suppose $n - 1 \le k < m$. Note that any root-connected subgraph with k edges can be augmented to one with $k+1$ edges in at most m ways. Also, any root-connected subgraph with $k+1$ edges can be obtained by augmentation from at least one with k edges. Thus,

$$\Pr(|S| = k+1) \le \frac{m(1-p^*)}{p^*} \Pr(|S| = k) < \tfrac{1}{2} \Pr(|S| = k).$$

It follows that $\Pr(|S| = n-1) \ge \frac{1}{2}$, i.e., there is a significant probability of observing an arborescence or directed tree. Of course, the output distribution is not quite the one we want, but we can deal issue with that by (usual) rejection sampling: simply retain S with probability

$$\left[\frac{p^*(1-p)}{p(1-p^*)} \right]^{|S|-n+1},$$

and otherwise run Algorithm 1 again to produce a fresh sample. Note that the rejection probability is at most $\frac{1}{2}$, so the expected overall running time is still $O(m^3 n)$. This deals with exact sampling. Since the number of arborescences can be computed exactly in polynomial time, an FPRAS for the case $p > p^*$ follows easily. It is not clear whether this method can be adapted to varying failure probabilities $\mathbf{p}$.

5. Coupling between reliability and bi-directed reachability

In this section, we give an alternative proof of Ball's equivalence between Reliability and Bi-directed Reachability [Bal80, Corollary 1]. Our proof constructs a coupling, between the (edge-weighted) distribution of connected subgraphs in the undirected setting, and the (edge-weighted) distribution of root-connected subgraphs in the bi-directed setting. This coupling, together with Algorithm 1, yields an efficient exact sampler for connected subgraphs.

We use $\{u,v\}$ to denote an undirected edge, and (u,v) or (v,u) to denote a directed one (namely an arc). Let $G=(V,E)$ be an undirected graph, and $\mathbf{p}=(p_e)_{e\in E}$ be a vector of failure probabilities. Let $\overrightarrow{G}=(V,A)$ be the bi-directed graph obtained by replacing every edge in G with a pair of anti-parallel arcs. Namely, $A=\{(u,v),(v,u)\mid\{u,v\}\in E\}$. Moreover, let $p_{(u,v)}=p_{(v,u)}=p_{\{u,v\}}$ and denote these failure probabilities by $\mathbf{p}'$. For $S\subseteq E$ (or $S\subseteq A$), let $\mathrm{wt}(S):=\prod_{e\in S}(1-p_e)\prod_{e\in E\setminus S}p_e$ (or $\mathrm{wt}(S):=\prod_{e\in S}(1-p_e)\prod_{e\in A\setminus S}p_e$).

Consider the following coupling between the product distribution over edges of G and the one over arcs of $\overrightarrow{G}$. We reveal edges in a breadth-first search (BFS) fashion in both graphs, from the same "root" vertex r. If an edge $\{u,v\}$ is present in the subgraph of G, we couple it with the arc (u,v) or (v,u), whose direction is pointing towards r in the subgraph of $\overrightarrow{G}$. The arc in the other direction is drawn independently from everything else. The key observation is that to decide the set of vertices that can reach r, at any point, only one direction of a bi-directed edge is useful and the other is irrelevant. One can verify that in the end, the subgraph of G is connected if and only if the subgraph of $\overrightarrow{G}$ is root-connected. We will formalize this intuition next.

Fix an arbitrary ordering of V, which will be used for the exploration, and let the first vertex be a distinguished root r. Let $\mathcal{P}(S)$ denote the power set of S for a set S. Define a mapping $\Phi:\mathcal{P}(E)\to\mathcal{P}(A)$ as follows. For $S\subseteq E$, we explore all vertices that can reach r in (V,S) in a deterministic order, and add arcs to $\Phi(S)$ in the direction towards r. To be more specific, we maintain the set of explored and the set of active vertices, denoted by V_e and V_a, respectively. At the beginning, $V_e=\emptyset$ and $V_a=\{r\}$. Given V_e and V_a, let v be the first vertex (according to the predetermined ordering) in V_a. For all $u\in V\setminus V_e$, if $\{u,v\}\in S$, add (u,v) to $\Phi(S)$ and add u to V_a (u may be in V_a already). Then move v from V_a to V_e. This process ends when all vertices that can reach r in (V,S) are explored. Let σ_S be the arriving order of V_e. We will call σ_S the traversal order. We remark that if $\{u,v\}\in S$ then exactly one of the arcs (u,v) and (v,u) is in $\Phi(S)$, and otherwise neither arc is in $\Phi(S)$.

Strictly speaking, the exploration above is not a BFS (V_a may contain a newly added vertex that is lower in the predetermined ordering than all other older vertices). To perform a BFS we need to in addition maintain a layer ordering, which seems unnecessary. The key properties of the exploration are: 1) all edges incident to the current vertex are processed together, as a group; 2) V_e is always connected (or root-connected for Ψ below).

Similarly, define $\Psi:\mathcal{P}(A)\to\mathcal{P}(E)$ as follows. For $S'\subseteq A$, we again maintain V_e and V_a, and initialize $V_e=\emptyset$ and $V_a=\{r\}$. Given V_e and V_a, let v be the first vertex in V_a. For all $u\in V\setminus V_e$, if $(u,v)\in S'$, add $\{u,v\}$ to $\Psi(S')$ and add u to V_a. Then move v from V_a to V_e. This process ends when all vertices that can reach r in (V,S') are explored. Analogously, let $\sigma_{S'}$ be the arriving order of V_e. We remark that if $(u,v)\notin S'$, and v is visited before u, then $\{u,v\}\notin\Psi(S')$, even in case of $(v,u)\in S'$.

Let $\Omega:=\{S\subseteq E\mid(V,S)\text{ is connected}\}$, and correspondingly $\overrightarrow{\Omega}:=\{S\subseteq A\mid(V,S)\text{ is root-connected}\}$. We have the following lemma.

Lemma 11. *Let Φ, Ψ, Ω, and $\overrightarrow{\Omega}$ be defined as above. Then the following holds:*

(1) if $S\in\Omega$, then $\Phi(S)\in\overrightarrow{\Omega}$;
(2) if $S'\in\overrightarrow{\Omega}$, then $\Psi(S')\in\Omega$;
(3) if $S\in\Omega$, then $\Psi(\Phi(S))=S$;
(4) $\Psi(\overrightarrow{\Omega})=\Omega$;

(5) for any $S \in \Omega$,

$$\mathrm{wt}(S) = \sum_{S' \in \Psi^{-1}(S)} \mathrm{wt}(S').$$

Proof. (1) It is easy to verify that, at any point of the construction of Φ, all vertices in V_e can reach r, in both (V, S) and $(V, \Phi(S))$. If $S \in \Omega$, then $V_e = V$ at the end of Φ. Hence $(V, \Phi(S))$ is root-connected, and $\Phi(S) \in \overrightarrow{\Omega}$.

(2) This item is completely analogous to item (1).

(3) If $\{u, v\} \in S$ and u is processed first during the exploration, then $(v, u) \in \Phi(S)$. The traversal orderings σ_S and $\sigma_{\Phi(S)}$ are the same. Hence, during the construction of $\Psi(\Phi(S))$, u is still processed first, and $\{v, u\} \in \Psi(\Phi(S))$. On the other hand, if $\{u, v\} \notin S$, then neither (u, v) nor (v, u) is in $\Phi(S)$ and thus $\{u, v\} \notin \Psi(\Phi(S))$.

(4) This item is a straightforward consequence of items (1), (2), and (3).

(5) By item (3), we have that $\Phi(S) \in \Psi^{-1}(S)$. Let

$$\Phi_c(S) := \left\{(u, v) \mid (u, v) \notin \Phi(S) \text{ and } v < u \text{ in the traversal order } \sigma_{\Phi(S)}\right\}.$$

Note that $\Phi(S) \cup \Phi_c(S)$ covers all unordered pairs of vertices as $S \in \Omega$. Moreover,

$$\prod_{e \in \Phi(S)} (1 - p_e) \prod_{e \in \Phi_c(S)} p_e = \mathrm{wt}(S). \tag{1}$$

Call S' consistent with $\Phi(S)$ if $\Phi(S) \subseteq S'$ and $S' \cap \Phi_c(S) = \emptyset$.

We claim that $S' \in \Psi^{-1}(S)$ if and only if S' is consistent with $\Phi(S)$. Suppose S' is not consistent with $\Phi(S)$. Consider the exploration of $\Phi(S)$ and S' in the construction of Ψ simultaneously. Since S' is not consistent with $\Phi(S)$, either $\Phi(S) \setminus S' \neq \emptyset$ or $S' \cap \Phi_c(S) \neq \emptyset$. Let v be the first vertex during the exploration so that there is an arc $(u, v) \in \Phi(S) \setminus S'$, or $(u, v) \in S' \cap \Phi_c(S)$ for some $u \notin V_e$. Since $S \in \Omega$, all vertices will be processed, and such a v must exist. (In the latter case, since $(u, v) \in \Phi_c(S)$, v is active first.) If $(u, v) \in \Phi(S) \setminus S'$, then $\{u, v\} \notin \Psi(S')$ but $\{u, v\} \in \Psi(\Phi(S))$. If $(u, v) \in S' \cap \Phi_c(S)$, $\{u, v\} \notin \Psi(\Phi(S))$ but $\{u, v\} \in \Psi(S')$. In either case, $\Psi(S') \neq \Psi(\Phi(S)) = S$ (by item (3)).

On the other hand, if $\Phi(S) \subseteq S'$ and $S' \cap \Phi_c(S) = \emptyset$, then we can trace through the construction of $\Psi(\Phi(S))$ and $\Psi(S')$ to verify that $\Psi(S') = \Psi(\Phi(S)) = S$.

The claim together with (1) implies that

$$\begin{aligned}\sum_{S' \in \Psi^{-1}(S)} \mathrm{wt}(S') &= \sum_{S' \text{ is consistent with } \Phi(S)} \mathrm{wt}(S') \\ &= \prod_{e \in \Phi(S)} (1 - p_e) \prod_{e \in \Phi_c(S)} p_e = \mathrm{wt}(S).\end{aligned}$$

□

Lemma 12. $Z_{rel}(G; \mathbf{p}) = Z_{reach}(\overrightarrow{G}, r; \mathbf{p}')$.

Proof. First notice that

$$Z_{rel}(G; \mathbf{p}) = \sum_{S \in \Omega} \mathrm{wt}(S)$$

and

$$Z_{reach}(\overrightarrow{G}, r; \mathbf{p}') = \sum_{S \in \overrightarrow{\Omega}} \mathrm{wt}(S).$$

By item (4) of Lemma 11, $\Psi(\overrightarrow{\Omega}) = \Omega$, implying that $\left(\Psi^{-1}(S)\right)_{S\in\Omega}$ is a partition of $\overrightarrow{\Omega}$. Combining this with item (5) of Lemma 11,

$$\begin{aligned}\sum_{S\in\Omega}\mathrm{wt}(S) &= \sum_{S\in\Omega}\sum_{S'\in\Psi^{-1}(S)}\mathrm{wt}(S')\\ &= \sum_{S'\in\overrightarrow{\Omega}}\mathrm{wt}(S').\end{aligned}$$

The lemma follows. □

Lemma 12 is first shown by Ball [Bal80, Corollary 2] via modifying edges one by one. Instead, our proof is essentially a coupling argument and has a new consequence that Algorithm 1 can be used to sample edge-weighted connected subgraphs. Recall our notation $\pi_G(\cdot)$, and generalise it to undirected graphs. Thus, for an undirected (or directed) graph G, $\pi_G(\cdot)$ is the distribution resulting from drawing each edge (or arc) e independently with probability $1-p_e$, and conditioning on the graph drawn being connected (or root-connected).

Lemma 13. *If a random root-connected subgraph S' is drawn from $\pi_{\overrightarrow{G}}(\cdot)$, then $\Psi(S')$ has distribution $\pi_G(\cdot)$.*

Proof. Since $S' \in \overrightarrow{\Omega}$, by item (2) of Lemma 11, $\Psi(S') \in \Omega$. Moreover, for any $s \in \Omega$,

$$\begin{aligned}\Pr[\Psi(S')=s] &= \sum_{s'\in\Psi^{-1}(s)}\Pr[S'=s']\\ &= \sum_{s'\in\Psi^{-1}(s)}\frac{\mathrm{wt}(s')}{Z_{reach}(\overrightarrow{G},r;\mathbf{p}')}\\ &= \frac{\mathrm{wt}(s)}{Z_{rel}(G;\mathbf{p})} = \pi_G(s),\end{aligned}$$

where we used item (5) of Lemma 11 and Lemma 12 in the last line. □

There is also a coupling going the reversed direction of Lemma 13, by drawing a random connected subgraph S from $\pi_G(\cdot)$, mapping it to $\Phi(S)$, and excluding all arcs in $\Phi_c(S)$. All other arcs are drawn independently. The resulting S' has distribution $\pi_{\overrightarrow{G}}(\cdot)$. Its correctness is not hard to prove, given Lemma 11, but it is not the direction of use to us and we omit its proof.

Theorem 10 and Lemma 12 imply the counting part of Theorem 1. Theorem 8 and Lemma 13 imply the sampling part of Theorem 1.

6. Counting connected subgraphs of a specified cardinality

In this section, we show that the sampling algorithm in Theorem 1 also leads to an FPRAS for the number of connected subgraphs of any fixed size.

For a connected (undirected) graph $G = (V, E)$, as usual let $n = |V|$ and $m = |E|$. For $n-1 \le t \le m$, let $H_t \subset E$ be the set of connected (and spanning) subgraphs of size t, and $N_t = |H_t|$. Notice that $N_m = 1$, and N_{n-1} is the number of spanning trees, which can be computed in polynomial time exactly due to Kirchhoff's matrix-tree theorem.

The complements of connected subgraphs are independent sets of the co-graphic matroid associated with G, and co-graphic matroids are representable [Oxl92]. Hence, by a breakthrough result of Huh and Katz [HK12] (see also [Len13] for a detailed derivation), $(N_t)_t$ is a log-concave sequence.

Proposition 14. *For any $n \le t \le m-1$,*

$$N_{t-1}N_{t+1} \le N_t^2.$$

We remark that log-concavity of such a sequence has now been established for all matroids [AHK18], but here we only need the case of representable matroids.

Once we have the log-concavity and the sampling algorithm in Theorem 1, we can apply a technique of Jerrum and Sinclair [JS89, Section 5] to efficiently approximate N_t for any $n-1 \le t \le m$.

Theorem 15. *For any $n-1 \le t \le m$, there is an FPRAS for N_t.*

Here we sketch the outline of the algorithm. We will only consider a uniform failure probability p over all edges in the following. Also, we make no attempt to optimise the exponent in the polynomial running time. The basic idea is to tune p in the sampler of Theorem 1 so that connected subgraphs of the desired size show up frequently enough. First notice that Proposition 14 implies that the ratios $\frac{N_{t-1}}{N_t}$ is monotonically increasing. It is straightforward to see that

$$\frac{N_{n-1}}{N_n} \ge \frac{1}{m}, \qquad \text{and} \qquad \frac{N_{m-1}}{N_m} \le m.$$

Let $r_t = \frac{N_{t-1}}{N_t}$. Hence,

$$\frac{1}{m} \le r_n \le r_{n+1} \le \cdots \le r_m \le m. \tag{2}$$

We will use $r = \frac{1-p}{p}$ to denote the edge weight when the failure probability of an edge is p. With a little abuse of notation, let $\pi_r(\cdot)$ be the distribution over connected subgraphs when each edge is removed with probability $p = \frac{1}{1+r}$ independently. (So $\pi_r(\cdot)$ is a product distribution on the edges, conditioned on the result being connected.) It is easy to see that for a connected subgraph $R \subset E$, $\pi_r(R) \propto r^{|R|}$. We note that $\pi_{r_t}(H_{t-1}) = \pi_{r_t}(H_t)$, and for any $i < t$,

$$\frac{\pi_{r_t}(H_t)}{\pi_{r_t}(H_i)} = r_t^{t-i} \cdot \frac{N_t}{N_i} = r_t^{t-i} \cdot \prod_{j=i+1}^{t} \frac{N_j}{N_{j-1}} = r_t^{t-i} \cdot \prod_{j=i+1}^{t} r_j^{-1} \ge r_t^{t-i} r_t^{i-t} = 1,$$

where we used (2). Similarly, for any $i > t$, $\pi_{r_t}(H_t) \ge \pi_{r_t}(H_i)$. Note that $\sum_{i=n-1}^{m} \pi_{r_t}(H_i) = 1$. We conclude that

$$\pi_{r_t}(H_{t-1}) = \pi_{r_t}(H_t) \ge \frac{1}{m}. \tag{3}$$

Thus, if we run the sampling algorithm of Theorem 1 with $p_t = \frac{1}{1+r_t}$, there is a significant probability to see subgraphs in H_t.

To utilise the argument above, we need to know r_t. This can be done inductively, since

$$\pi_{r_t}(H_{t-2}) = \frac{r_{t-1}}{r_t} \cdot \pi_{r_t}(H_{t-1}) \ge \frac{1}{m^3},$$

where we used (2) and (3). Rewrite N_t as

$$N_t = N_m \cdot \prod_{i=m}^{t+1} \frac{N_{i-1}}{N_i} = \prod_{i=m}^{t+1} r_i,$$

and our estimator of N_t will be the product of estimators for r_i where $i \in [t+1, m]$. A complete description is given in Algorithm 2. We should set T to be a sufficiently large number (but still polynomial in n) so that the variances of the estimators are small enough. Notice that N_{m-1} is easy to compute since it is just the number of edges in G that are not bridges.

The analysis of Algorithm 2 is identical to the proof of [JS89, Theorem 5.3] and thus omitted. Theorem 15 is a direct consequence of Algorithm 2.

Algorithm 2 Approximately count connected subgraphs of a fixed size $t \in [n, m-2]$

```
Let $\widetilde{r} \leftarrow \frac{N_{m-1}}{N_m}$ and $\widetilde{N} = N_{m-1}$.
for $i = m-2, m-1, \dots, t$ do
  if $\widetilde{r} \notin [1/2m, 2m]$ then
    return 0 {Note the bounds in (2)}
  end if
  Draw $T$ samples from $\pi_{\widetilde{r}}(\cdot)$ using Algorithm 1, yielding a set $Y$.
  if $|Y \cap H_i| = 0$ or $|Y \cap H_{i+1}| = 0$ then
    return 0
  end if
  Let $\widetilde{r} \leftarrow \widetilde{r} \cdot \frac{|Y \cap H_i|}{|Y \cap H_{i+1}|}$ and $\widetilde{N} \leftarrow \widetilde{N}/\widetilde{r}$.
end for
return $\widetilde{N}$
```

7. Concluding remarks

In this paper we give an FPRAS for Reliability (or, equivalently, Bi-directed Reachability), by confirming a conjecture of Gorodezky and Pak [GP14]. We also give an exact sampler for edge-weighted connected subgraphs with polynomial running time in expectation. The core ingredient of our algorithms is the cluster-popping algorithm to sample root-connected subgraphs, namely Algorithm 1. We manage to analyze it using the partial rejection sampling framework.

Reliability is equivalent to counting weighted connected subgraphs, which is the evaluation of the Tutte polynomial $T_G(x, y)$ for points $x = 1$ and $y > 1$. An interesting question is about the dual of this half-line, namely for points $x > 1$ and $y = 1$, whose evaluation is to count weighted acyclic subgraphs. It is well known that for a planar graph G, $T_G(x, 1) = T_{G^*}(1, x)$ where G^* is the planar dual of G [Oxl92]. Hence, Theorem 1 implies that in planar graphs, $T_G(x, 1)$ can be efficiently approximated for $x > 1$. Can we remove the restriction of planar graphs?

Another interesting direction is to generalize Algorithm 1 beyond bi-directed graphs. What about Eulerian graphs? Is approximating Reachability **NP**-hard in general?

Acknowledgements

We thank Mark Huber for bringing reference [GP14] to our attention, Mark Walters for the coupling idea leading to Lemma 12, David Harris for proposing the question answered in Section 6, and Igor Pak for comments on an earlier version. We also thank the organizers of the "LMS – EPSRC Durham Symposium on Markov Processes, Mixing Times and Cutoff", where part of the work is carried out.

References

[AHK18] Karim Adiprasito, June Huh, and Eric Katz. Hodge theory for combinatorial geometries. *Ann. of Math. (2)*, 188(2):381–452, 2018. 11

[Bal80] Michael O. Ball. Complexity of network reliability computations. *Networks*, 10(2):153–165, 1980. 2, 8, 10

[Bal86] Michael O. Ball. Computational complexity of network reliability analysis: An overview. *IEEE Trans. Rel.*, 35(3):230–239, 1986. 1

[BP83] Michael O. Ball and J. Scott Provan. Calculating bounds on reachability and connectedness in stochastic networks. *Networks*, 13(2):253–278, 1983. 1, 2

[Col87] Charles J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, 1987. 1

[CPP02] Henry Cohn, Robin Pemantle, and James G. Propp. Generating a random sink-free orientation in quadratic time. *Electron. J. Combin.*, 9(1):10:1–10:13, 2002. 2

[GJ08] Leslie Ann Goldberg and Mark Jerrum. Inapproximability of the Tutte polynomial. *Inf. Comput.*, 206(7):908–929, 2008. 1

[GJ14] Leslie Ann Goldberg and Mark Jerrum. The complexity of computing the sign of the Tutte polynomial. *SIAM J. Comput.*, 43(6):1921–1952, 2014. 1

[GJL17] Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the Lovasz local lemma. In *STOC*, pages 342–355, 2017. 2, 3, 4

[GP14] Igor Gorodezky and Igor Pak. Generalized loop-erased random walks and approximate reachability. *Random Struct. Algorithms*, 44(2):201–223, 2014. 2, 3, 4, 6, 7, 12

[Hag91] Jane N. Hagstrom. Computing rooted communication reliability in an almost acyclic digraph. *Networks*, 21(5):581–593, 1991. 2

[HK12] June Huh and Eric Katz. Log-concavity of characteristic polynomials and the Bergman fan of matroids. *Math. Ann.*, 354(3):1103–1116, 2012. 10

[HS14] David G. Harris and Aravind Srinivasan. Improved bounds and algorithms for graph cuts and network reliability. In *SODA*, pages 259–278. SIAM, 2014. 1

[Jer81] Mark Jerrum. On the complexity of evaluating multivariate polynomials. *Ph.D. dissertation.* Technical Report CST-11-81, Dept. Comput. Sci., Univ. Edinburgh, 1981. 2

[JS89] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989. 11

[JS93] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM J. Comput.*, 22(5):1087–1116, 1993. 2

[JVW90] François Jaeger, Dirk L. Vertigan, and Dominic J. A. Welsh. On the computational complexity of the Jones and Tutte polynomials. *Math. Proc. Cambridge Philos. Soc.*, 108(1):35–53, 1990. 1

[Kar99] David R. Karger. A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. *SIAM J. Comput.*, 29(2):492–514, 1999. 1, 2

[Kar16] David R. Karger. A fast and simple unbiased estimator for network (un)reliability. In *FOCS*, pages 635–644, 2016. 1

[Kar17] David R. Karger. Faster (and still pretty simple) unbiased estimators for network (un)reliability. In *FOCS*, pages 755–766, 2017. 1

[KL85] Richard M. Karp and Michael Luby. Monte-Carlo algorithms for the planar multiterminal network reliability problem. *J. Complexity*, 1(1):45–64, 1985. 1

[KS11] Kashyap Babu Rao Kolipaka and Mario Szegedy. Moser and Tardos meet Lovász. In *STOC*, pages 235–244, 2011. 2, 3, 4

[Len13] Matthias Lenz. The f-vector of a representable-matroid complex is log-concave. *Adv. in Appl. Math.*, 51(5):543–545, 2013. 10

[MT10] Robin A. Moser and Gábor Tardos. A constructive proof of the general Lovász Local Lemma. *J. ACM*, 57(2), 2010. 2

[Oxl92] James G. Oxley. *Matroid theory.* Oxford University Press, 1992. 10, 12

[PB83] J. Scott Provan and Michael O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.*, 12(4):777–788, 1983. 1, 2

[She85] James B. Shearer. On a problem of Spencer. *Combinatorica*, 5(3):241–245, 1985. 2

[Val79] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979. 1

[VW92] Dirk Vertigan and Dominic J. A. Welsh. The compunational complexity of the Tutte plane: the bipartite case. *Combin. Probab. Comput.*, 1:181–187, 1992. 1

[Wel93] Dominic J. A. Welsh. *Complexity: knots, colourings and counting*, volume 186 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 1993. 2

[Wel99] Dominic J. A. Welsh. The Tutte polynomial. *Random Struct. Algorithms*, 15(3-4):210–228, 1999. 2

[Wil96] David B. Wilson. Generating random spanning trees more quickly than the cover time. In *STOC*, pages 296–303, 1996. 2

(Heng Guo) School of Informatics, University of Edinburgh, Informatics Forum, Edinburgh, EH8 9AB, United Kingdom.

Email address: hguo@inf.ed.ac.uk

(Mark Jerrum) School of Mathematical Sciences, Queen Mary, University of London, Mile End Road, London, E1 4NS, United Kingdom.

Email address: m.jerrum@qmul.ac.uk