Covering Minimum Spanning Trees of Random Subgraphs*

Michel X. Goemans and Jan Vondrák

Abstract

We consider the problem of covering the minimum spanning tree (MST) of a random subgraph of G by a sparse set of edges, with high probability. The two random models that we consider are subgraphs induced by a random subset of vertices, each vertex included independently with probability p, and subgraphs generated as a random subset of edges, each edge with probability p.

Let n be the number of vertices in G. We show that in both cases, there is a covering set Q of cardinality $O(n \log_b n)$ where $b = 1/(1-p)$ (and p is possibly a function of n) and this is asymptotically optimal. More generally, we show a similar bound on the covering set in a matroid, which contains the minimum-weight basis of a random subset with high probability. Also, we give a randomized algorithm which calls an MST subroutine only a polylogarithmic number of times, and finds the covering set with high probability.

1 Introduction

In a variety of optimization settings, one has to repeatedly solve instances of the same problem in which only part of the input is changing. It is important in such cases to perform a precomputation that involves only the static part of the input and possibly assumptions on the dynamic part, and which allows to speed-up the repeated solution of instances. The precomputation could possibly be computationally intensive.

In telecommunication networks for example, the topology may be considered fixed but the demands of a given customer (in a network provisioning problem) may vary over time. The goal is to exploit the topology without knowing the demands. The same situation happens in performing multicast in telecommunication networks; we need to solve a minimum spanning tree or Steiner tree problem to connect a group of users, but the topology or graph does not change when connecting different groups of users. Or, in flight reservation systems, departure and arrival locations and times change for each request but schedules do not (availability and prices do change as well but on a less frequent basis). In production planning or job shop scheduling environment, the same collection of items have to be produced over and over again, and given say daily or weekly production demands for each item, we need to most efficiently schedule the different tasks to produce the items in the required quantities. Yet another example is for delivery companies; they have to solve daily vehicle routing problems in which the road network does not change but the locations of customers to serve do.

Examples of such *repetitive* optimization problems with both static and dynamic inputs are countless and in many cases it is unclear if one could take any advantage from the advance knowledge of the static part of the input. One situation which has been much studied (especially from a practical point-of-view) is for $s-t$ shortest path queries in large-scale navigation systems or Geographic Information Systems. In that setting, it is too slow to compute from scratch the shortest path whenever a query comes in. Various preprocessing steps have been proposed, often creating a hierarchical view of the network, see for example [7]. In this extended abstract, we have a modest (but nevertheless challenging) goal and study another simple combinatorial optimization problem, the minimum spanning tree (MST) problem in the situation when instances are repeatedly drawn either randomly or deterministically from a fixed given graph.

More formally, assume we are given an edge-weighted graph $G = (V, E)$ with n vertices and m edges and we would like to (repeatedly) find the minimum spanning tree of either a vertex induced subgraph $H = G[W]$, $W \subseteq V$ (the *vertex case*) or of a subgraph $H = (V, F)$, $F \subseteq E$ (the *edge case*). In general, we need to consider the minimum spanning forest, i.e. the minimum spanning tree on each component, since the subgraph might not be connected. In the random setting that the technical part of this abstract focuses on, we assume that in the vertex case, each vertex appears in W independently with probability p. In the edge case, each edge appears in F independently with probability

*This work was supported by NSF contracts ITR-0121495 and CCR-0098018. Authors' address: MIT, Dept of Mathematics, Cambridge, MA 02139. {goemans,vondrak}@math.mit.edu

p. The question we ask is whether there exists a *sparse set* of edges which contains the minimum spanning forest of the random subgraph with high probability.

The basic case is when p is a constant ($p = 0.5$ corresponds to the uniform case), but our results also apply to the case in which p is a function of n. An especially interesting case is when $p = 1 - o(1)$ (low probability of failure). In the deterministic setting, we assume that W is obtained from V by removing up to k vertices, or F is obtained from E by removing at most k edges. Then we require our set to contain the minimum spanning forest of *all* such subgraphs. Our results for the deterministic setting follow quite surprisingly from the probabilistic results.

Overview of results. Consider the random setting where either vertices or edges are sampled with probability p. Let $b = 1/(1-p)$. We prove that, for every weighted graph (no matter what the weights are), there exists a *sparse* set Q of $O(n \log_b n)$ edges which contains the MST on $G[W]$ (or (V, F)) almost surely. More precisely, we can guarantee the existence of a set Q covering the MST on $H = G[W]$ or $H = (V, F)$ with probability at least $1 - \frac{1}{n^c}$ for a set Q of cardinality $e(c+1)n \log_b n + O(n)$. See Theorem 2.1 and Corollary 4.1. For a constant p, we get $|Q| = O(n \log n)$, and this is asymptotically optimal, since Q has to be $c \log n$-connected in the vertex case (or $c \log n$-edge-connected in the edge case). Furthermore, there exist weighted graphs for which one needs $\Omega(n \log n)$ edges even to achieve a constant probability of covering the MST, see Section 5. For low probability of failure, in particular $p = 1 - \frac{1}{n^\gamma}$, we obtain a set Q of linear size, and this is asymptotically optimal as well.

We believe that our proof technique is quite interesting in its own right. We define $\sigma_p(u, v)$ to be the probability that (u, v) belongs to the MST conditioning upon the event that (u, v) is in the random subgraph. Observe that $1 - \sigma_p(u, v)$ is equal to the $u-v$-reliability probability in the subgraph with all edges cheaper than (u, v). We show a *boosting lemma* which states that, as p decreases, $\sigma_p(u, v)$ increases very rapidly. More precisely, we show that for $(1-p) = (1-q)^k$, $\sigma_q(u, v) \geq \sigma_p(u, v)^{1/k}$ (see Lemma 2.3).

Our boosting lemma is analogous to a result of Bollobás and Thomason [4]; however, they use a different random model, in which a random subset of a fixed cardinality is selected. They use the Kruskal-Katona theorem (and its simplification due to Lovász) to derive their lemma, while our lemma has an elementary probabilistic proof. We could use their lemma in some of our proofs as well (for instance, it would seem quite natural to apply it to prove Corollary 2.2, which is concerned with the minimum spanning trees after removing a fixed number of vertices), but this would produce an additional factor of $\log n$ which we are able to shave off with our boosting lemma. We also have another form of the boosting lemma, derived from the Kruskal-Katona theorem, which states that for any $q < p$, if $\sigma_p \geq (1-p)^k$, then $\sigma_q \geq (1-q)^k$. We will treat these inequalities in more detail in the journal version of this paper.

From the boosting lemma, we deduce that not many edges (u, v) can have a "large" value of $\sigma_p(u, v)$, otherwise the sum of these probabilities in a suitably chosen random subgraph (with sampling probability roughly $p/\log n$) would exceed the expected size of the minimum spanning forest. This means that we can choose all the edges with sufficiently large $\sigma_p(u, v)$ in our set Q, and this ensures the MST-covering property that we need. In order to find Q, we could try to calculate $\sigma_p(u, v)$ for each edge; however, as $1 - \sigma_p(u, v)$ corresponds to the $u - v$ reliability in an arbitrary graph, this is #P-hard [10]. It is even unknown how to efficiently approximate the $u - v$ reliability. However, in our case, we only need to check if $\sigma_p(u, v)$ is (polynomially) large enough and this can be done simply by random sampling. This leads to a randomized algorithm for computing Q that makes a polynomial number of calls to an MST subroutine. However, we can reduce the number of MST calls to polylogarithmic by using the boosting lemma, and choosing the edges which appear sufficiently often. With high probability, we find a covering set of asymptotically optimal size $O(n \log_b n)$, and the running time of the algorithm is $O(m \log_b n \log n)$. If we are interested in deterministic algorithms only, we are only able to construct in polynomial time a covering set Q of cardinality $O(n^{3/2} \ln n)$ in the vertex case and $ne^{O(\sqrt{\ln n})}$ in the edge case, and this uses specific properties of graphs; this is omitted from this extended abstract.

Going back to our original motivation, since we are able to construct a set Q of size $O(n \log_b n)$ covering almost all MSTs, we can therefore with high probability find the MST of any random subgraph by focusing on this precomputed set of $O(n \log_b n)$ edges, hence leading to an algorithm whose running time is almost linear in n instead of linear in m. This is almost a quadratic speed-up if the original graph is dense.

In the deterministic setting in which the subgraph is obtained by deleting at most k vertices (or edges), we show that there exists a set Q of cardinality at most $e(k+1)n$ which contains the MST of *all* these subgraphs, see Corollary 2.2. In the edge case, it is easy to see that the cardinality Q can be bounded by $(k+1)n$ since we can remove an MST of G, repeat this $k+1$ times, and let Q be the union of these

MSTs. The vertex case, though, does not appear as easy to prove. Observe the difference in behavior between the random and deterministic cases. For $p = 0.5$, we have shown that $\Theta(n \ln n)$ edges are sufficient to cover most subgraphs while we need $\Theta(n^2)$ to cover all of them. Interestingly, our proof for the deterministic bound of $e(k+1)n$ follows almost immediately from the probabilistic boosting lemma.

Our results only use the fact that if an edge e belongs to the MST of H it also belongs to the MST of any subgraph of H containing e. Thus all our results extend to any matroid, see Section 4.

Literature discussion. Not assuming that all the input data is known in advance or assuming it changes over time is a typical paradigm in the areas of optimization and algorithms. Although the minimum spanning tree problem (as a prototypical combinatorial optimization problem) has been considered in a wide variety of settings with incomplete or changing data, it has not been under the particular viewpoint considered here.

In dynamic graph algorithms, one assumes that the graph is dynamically changing and one needs to update the solution of the problem after each input update. For a minimum spanning tree problem in which edges can be inserted or deleted, the best known dynamic algorithm has amortized cost $O(\log^4 n)$ per operation [6]. This is not efficient here though, since our instances are changing too drastically.

In the NP-hard Probabilistic MST problem [2], each vertex is also present independently with a given probability and the goal is to find a spanning tree such that the Steiner tree obtained by removing the edges not needed to connect the random set of vertices has minimum expected cost. Our different model has the advantage of giving a minimum spanning tree (instead of a suboptimal Steiner tree) at the expense of a logarithmic increase in running time.

In practice, graph optimization problems are often solved on a sparse subgraph, and edges which are not included are then *priced* to see if they could potentially improve the solution found, see for example [1] for the matching problem. Our results can therefore be viewed as a theoretical basis for this practice in the case of the MST, and gives precise bounds on the sparsity required.

2 The upper bound for random induced subgraphs

In this section, we deduce an upper bound of $O(n \log_b n)$ on the size of an MST covering set for the vertex variant, where $b = 1/(1-p)$. This is optimal (up to a constant factor) for a wide range of sampling probabilities p.

The basic property of MSTs we use is the observation that for any given edge, being contained in the minimum-weight spanning forest of a random subgraph is a *monotone event*. The following lemma is an easy consequence of the fact that an edge is in the minimum spanning tree unless its endpoints are connected by a path containing edges of smaller weight.

LEMMA 2.1. *For an edge $(u,v) \in E$, let Ω_{uv} be the probability space of subsets of $V \setminus \{u,v\}$. Let S_{uv} denote the event that (u,v) is in the minimum spanning forest of a random induced subgraph, conditioned on the event that (u,v) is in the subgraph. I.e.,*

$$S_{uv} = \{A \in \Omega_{uv} : (u,v) \in MST(A \cup \{u,v\})\}.$$

Then S_{uv} is a down-monotone event on Ω_{uv}:

$$A \in S_{uv}, B \subseteq A \Longrightarrow B \in S_{uv}.$$

Next, we prove a general inequality for down-monotone events.

LEMMA 2.2. *Let X be a finite set and let $X(p)$ denote a random subset of X where each element is chosen independently with probability p. Let $\mathcal{F}$ be a down-monotone event on the subsets of X, and let $\sigma_p = Pr[X(p) \in \mathcal{F}]$. Then for any two probabilities p_1, p_2 such that $(1-p_1)(1-p_2) = 1-p$,*

$$\sigma_p \leq \sigma_{p_1} \sigma_{p_2}.$$

Proof. Let $Y_1 = X(p_1)$ and $Y_2 = X(p_2)$, sampled independently. Let $Y = Y_1 \cup Y_2$. Observe that Y contains each element independently with probability $1-(1-p_1)(1-p_2) = p$.

Due to the monotonicity property, $Y \in \mathcal{F}$ only if $Y_1 \in \mathcal{F}$ and $Y_2 \in \mathcal{F}$. Also, Y_1 and Y_2 are sampled independently, and therefore

$$\begin{aligned} Pr[Y \in \mathcal{F}] &\leq Pr[Y_1 \in \mathcal{F} \ \& \ Y_2 \in \mathcal{F}] \\ &= Pr[Y_1 \in \mathcal{F}] Pr[Y_2 \in \mathcal{F}]. \end{aligned}$$

For any $q < p$, we can choose p_2 such that $(1-q)(1-p_2) = 1-p$, and therefore Lemma 2.2 implies the monotonicity of σ_p.

COROLLARY 2.1. *For any $q < p$, we have $\sigma_q \geq \sigma_p$.*

In particular, this together with Lemma 2.2 implies that $\sigma_{p_1+p_2} \leq \sigma_{p_1}\sigma_{p_2}$. As another direct corollary to Lemma 2.2, we deduce our *boosting lemma* stating how the probability of a monotone event increases as the sampling probability is decreased.

LEMMA 2.3. *For $0 < p < 1$, $k \in \mathbf{N}$ and $(1-p) = (1-q)^k$,*

$$\sigma_q \geq (\sigma_p)^{1/k}.$$

Proof. Let $1-p_i = (1-q)^i, i = 1 \ldots k$. By iteration of the previous lemma,

$$\sigma_p \leq \sigma_{p_{k-1}}\sigma_q \leq \sigma_{p_{k-2}}\sigma_q\sigma_q \ldots \leq (\sigma_q)^k.$$

In the following, we will use the following estimate for $q = 1-(1-p)^{1/k}$.

LEMMA 2.4. *Let $1-p = (1-q)^k$. Then*

$$\frac{1}{q} \leq 1 - \frac{k}{\ln(1-p)}.$$

Proof.

$$\begin{aligned}\frac{1}{q} - 1 &= \frac{1}{1-(1-p)^{1/k}} - 1 \\ &= \frac{1}{e^{-1/k\ln(1-p)} - 1} \leq -\frac{k}{\ln(1-p)},\end{aligned}$$

using the fact $e^x \geq 1+x$.

Now let's go back to minimum spanning trees. As noted before, for any edge $(u,v) \in E$, the event that $(u,v) \in MST(W)$, conditioned on $u,v \in W$, is down-monotone. Let's denote

$$\sigma_p(u,v) = Pr[(u,v) \in MST(W) \mid u,v \in W]$$

where $W = V(p)$ contains each vertex independently with probability p.

LEMMA 2.5. *For a weighted graph G on n vertices, and an integer $k \geq 1$, let*

$$E_k = \{(u,v) \in E : \sigma_p(u,v) \geq e^{-k}\}.$$

Then

$$|E_k| < \left(\frac{k}{\ln b} + 1\right)\ en$$

where $b = 1/(1-p)$.

Proof. Let q be probability such that $1-p = (1-q)^k$. Sample a random subset $S = V(q)$. For any edge $(u,v) \in E_k$, we have $\sigma_p(u,v) \geq e^{-k}$, and therefore, by the boosting lemma,

$$\sigma_q(u,v) = Pr[(u,v) \in MST(S) \mid u,v \in S] \geq e^{-1}.$$

Unconditioning, we get $Pr[(u,v) \in MST(S)] \geq q^2e^{-1}$, and thus

$$\mathbf{E}[|MST(S)|] \geq |E_k|q^2e^{-1}.$$

On the other hand, the size of the minimum spanning forest on S is at most the number of vertices in S, and so $\mathbf{E}[|MST(S)|] < \mathbf{E}[|S|] = qn$. From these inequalities, we get

$$|E_k| < \frac{en}{q} \leq \left(1 + \frac{k}{\ln b}\right)\ en$$

using Lemma 2.4.

THEOREM 2.1. *Let G be any weighted graph on n vertices, $0 < p < 1$, and $c > 0$. Let $b = 1/(1-p)$. Then there exists a set $Q \subseteq E$ of size*

$$|Q| = e(c+1)n\log_b n + O(n)$$

such that

$$Pr[MST(V(p)) \subseteq Q] > 1 - \frac{1}{n^c}.$$

Proof. Let $h = \lceil (c+1)\ln n + \ln p + 2\rceil$ and $Q = E_h = \{(u,v) \in E : \sigma_p(u,v) \geq e^{-h}\}$. If we arrange the edges in the order of decreasing $\sigma_p(u,v)$ and partition the sequence into blocks B_k, where the first k blocks contain $\lfloor (1+k/\ln b)\ en\rfloor$ edges, Lemma 2.5 implies that $\sigma_p(u,v) \leq e^{1-k}$ for $e \in B_k$. The tail sum starting from block $h+1$ is

$$\begin{aligned}&\sum_{k=h+1}^{\infty}\sum_{(u,v)\in B_k}\sigma_p(u,v) \\ &\leq \left\lceil\frac{en}{\ln b}\right\rceil\sum_{k=h+1}^{\infty}e^{1-k} \\ &\leq \left(\frac{en}{\ln b}+1\right)\frac{e^{-h}}{1-e^{-1}} \\ &\leq \left(\frac{1}{\ln b}+\frac{1}{en}\right)\frac{1}{n^cp(e-1)} < \frac{1}{n^cp^2},\end{aligned}$$

using $p < \ln b$ and $n \geq 1$. We choose Q to contain the first h blocks, therefore

$$\begin{aligned}&Pr[MST(V(p)) \subseteq Q] \\ &\geq 1 - \sum_{k=h+1}^{\infty}\sum_{(u,v)\in B_k}p^2\sigma_p(u,v) > 1 - \frac{1}{n^c}.\end{aligned}$$

Due to Lemma 2.5,

$$\begin{aligned}|Q| &\leq \left(\frac{(c+1)\ln n + \ln p + 3}{\ln b} + 1\right)\ en \\ &= e(c+1)n\log_b n + O(n),\end{aligned}$$

using the inequality $(\ln p + 3)/\ln b < 7$ for $0 < p < 1$.

In particular, for constant probability of vertex failure we get $|Q| = O(n\log n)$. For low probability of vertex failure $1-p = n^{-\gamma}$, $0 < \gamma \leq 1$, we get $|Q| = O(n/\gamma)$. As we show later, these bounds are asymptotically optimal.

Directly from Lemma 2.5, we also get the following interesting implication for the "deterministic version" of the problem, where any k vertices can be removed arbitrarily.

COROLLARY 2.2. *For any weighted graph on n vertices, and $k \in \mathbf{N}$, there exists a set $Q \subseteq E$ of size*

$$|Q| < (k+1)en$$

which contains the minimum spanning forest $MST(V \setminus A)$ for any A of size at most k.

Proof. Choose $p = 1 - 1/e$ and $Q = E_k$ (from Lemma 2.5). Then $|Q| < (k+1)en$ and any edge (u,v) which appears in $MST(V \setminus A)$ for some $|A| \leq k$, has $\sigma_p(u,v) \geq e^{-k}$, therefore $(u,v) \in Q$.

It is easy to see that for $k = 1$, the bound can be strengthened to $2n - 3$. It is also easy to construct instances for any k where Q must contain $(n-1)+(n-2)+\cdots+(n-k-1)$ edges (see Section 5 for one such instance), and we conjecture this to be the maximum number of edges needed.

Observe also that, in this deterministic setting, the set Q defined as the union over all MSTs on vertex sets of cardinality at least $n-k$ can be found in polynomial time. Indeed, for every edge (u,v), one can test if it is possible to destroy all cheaper paths by removing at most k vertices by computing the vertex connectivity between u and v in the graph of cheaper edges. This, however, does not seem to translate easily into a bound on $|Q|$.

3 Algorithmic construction of covering sets

We can find the covering set Q with an efficient randomized algorithm, which takes advantage of the boosting lemma as well. It is a Monte Carlo algorithm, in the sense that it finds a correct solution with high probability, but the correctness of the solution cannot be verified easily. The algorithm works as follows:

Given $0 < p < 1$, $c > 0$ and $b = 1/(1-p)$:

- Let $k = \lceil (c+2)\ln n \rceil$ and $q = 1 - (1-p)^{1/k}$.
- Repeat the following for $i = 1, \ldots, r = \lceil 32eq^{-2}\ln n \rceil$:
  - Sample $S_i \subseteq V$, each vertex independently with probability q.
  - Find $T_i = MST(S_i)$.
- For each edge, include it in Q if it appears in at least $16\ln n$ different T_i's.

We first discuss the running time of the algorithm. The number of iterations is $r = O(q^{-2}\log n)$. Each time, we generate the random subgraph $G[S_i]$ (by generating S_i in expected time $O(qn)$ and the induced set of edges in expected time $O(qm)$, by scanning the adjacency lists of selected vertices). Then we find the minimum spanning forest in expected time $O(q^2 m)$ (since the expected number of edges is $q^2 m$, see [8]). Therefore the expected running time of each iteration is $O(qm)$, leading to a total running time of $O(m/q \log n) = O(m \log_b n \log n)$, using the bound for q given in Lemma 2.4.

THEOREM 3.1. *This algorithm finds with high probability a set $Q \subseteq E$ such that*

$$|Q| \leq 4e(c+2)n\log_b n + O(n)$$

and $Pr[MST(V(p)) \subseteq Q] > 1 - \frac{1}{n^c}$.

Proof. We have $k = \lceil (c+2)\ln n \rceil$, $1 - p = (1-q)^k$, $r = \lceil \frac{32e}{q^2}\ln n \rceil$ and $E_k = \{e \in E : \sigma_p(e) \geq e^{-k}\}$. We claim that $E_k \subseteq Q$ with high probability. Let $S_i = V(q)$ and $T_i = MST(S_i)$, $1 \leq i \leq r$. By the boosting Lemma 2.3, for any $e \in E_k$,

$$Pr[e \in T_i] \geq q^2 e^{-1}.$$

Denoting by t_e the number of T_i's containing edge e, we get $\mathbf{E}[t_e] \geq rq^2e^{-1} \geq 32\ln n$. By Chernoff bound (see [9, Theorem 4.2]), we derive $Pr[t_e < 16\ln n] < e^{-4\ln n} = \frac{1}{n^4}$, and thus $Pr[\exists e \in E_k; t_e < 16\ln n] < \frac{1}{n^2}$. Therefore with high probability, all edges in E_k are included in Q. If that happens, Q is a good covering set, since

$$\begin{aligned} Pr[MST(W) \subseteq Q] &\geq 1 - \sum_{e \in E \setminus Q} Pr[e \in MST(W)] \\ &\geq 1 - \binom{n}{2} e^{-k} > 1 - \frac{1}{n^c}. \end{aligned}$$

Now we estimate the size of Q. Since we are sampling $S_i = V(q)$, we have $\mathbf{E}[|S_i|] = qn$, and $\mathbf{E}\left[\sum_{i=1}^r |T_i|\right] \leq \mathbf{E}\left[\sum_{i=1}^r |S_i|\right] \leq rqn$. We can use a Chernoff bound again [9, Theorem 4.1] to estimate that

$$Pr\left[\sum_{i=1}^r |S_i| > 2rqn\right] < \left(\frac{e}{4}\right)^{rqn} < e^{-rqn/3} < e^{-n/3}.$$

In Q, we include only edges which appear in at least $16\ln n$ different T_i's, and $|T_i| \leq |S_i|$, so the number of such edges is, with high probability,

$$\begin{aligned} |Q| &\leq \frac{\sum |S_i|}{16\ln n} \leq \frac{rqn}{8\ln n} \leq \lceil 4enq^{-1} \rceil \\ &\leq 4e(c+2)n\log_b n + O(n), \end{aligned}$$

using Lemma 2.4 again.

4 The upper bound for matroids

Next, we consider the variant of the problem where the subgraph is generated by taking a random subset of edges $E(p)$. We approach this problem more generally, in the context of *matroids*. The matroid in this case would be the graphic matroid defined by all forests on the ground set E. By analogy, we call the elements of E *edges*. In general, consider a weighted matroid $(E, \mathcal{M}, w)$, where $w : E \to \mathbf{R}$. Let m denote the size of the ground set E and n the rank of $\mathcal{M}$, i.e. the size of the largest independent set. If the weights are distinct, then any subset $F \subseteq E$ has a unique minimum-weight basis $MB(F)$, which in the case of graphs corresponds to the minimum spanning forest. These bases satisfy exactly the monotonicity property that we used previously.

LEMMA 4.1. *For a fixed element $e \in E$, let Ω_e be the probability space of subsets of $E \setminus \{e\}$. Let S_e denote the event that e is in the minimum-weight basis of a random subset $F \subseteq E$, conditioned on $e \in F$, i.e.,*

$$S_e = \{A \in \Omega_e : e \in MB(A \cup \{e\})\}.$$

Then S_e is a down-monotone event on Ω_e:

$$A \in S_e, B \subseteq A \Longrightarrow B \in S_e.$$

Thus, we can apply the same machinery to matroids. Denote

$$\sigma_p(e) = Pr[e \in MB(F) \mid e \in F]$$

where $F = E(p)$ is a random subset of edges, sampled with probability p. We get analogous statements. It is interesting to notice that the bounds given in these lemmas depend only on the rank of the matroid, irrespective of the size of the ground set.

LEMMA 4.2. *For a weighted matroid $(E, \mathcal{M}, w)$, of rank n, probability $p < 1$ and integer $k \geq 1$, let*

$$E_k = \{e \in E : \sigma_p(e) \geq e^{-k}\}.$$

Then $|E_k| \leq \left(\frac{k}{\ln b} + 1\right)\ en$, where $b = 1/(1-p)$.

Proof. Let q be probability such that $1 - p = (1-q)^k$. Sample a random subset $S = E(q)$. For any element $e \in E_k$, the boosting lemma implies

$$\sigma_q(e) = Pr[e \in MB(S) \mid e \in S] \geq e^{-1}.$$

Therefore $Pr[e \in MB(S)] \geq qe^{-1}$ and

$$\mathbf{E}[|MB(S)|] \geq |E_k| qe^{-1}.$$

On the other hand, $|MB(S)| \leq n$, which yields

$$|E_k| \leq \frac{en}{q} = \frac{en}{1-(1-p)^{1/k}} \leq \left(\frac{k}{\ln b} + 1\right)\ en,$$

using Lemma 2.4.

THEOREM 4.1. *For any weighted matroid $(E, \mathcal{M}, w)$ of rank n, $0 < p < 1$, $c > 0$, and $b = 1/(1-p)$, there exists a set $Q \subseteq E$ of size*

$$|Q| \leq e(c+1)n \log_b n + O(n)$$

such that $Pr[MB(E(p)) \subseteq Q] = 1 - O\left(\frac{1}{n^c}\right)$.

Proof. Let $h = \lceil (c+1) \ln n \rceil$ and $Q = E_h = \{e \in E : \sigma_p(e) \geq e^{-h}\}$. Again, we arrange the elements in the order of decreasing $\sigma_p(e)$ and partition into blocks B_k at positions $\lfloor (1 + k/\ln b)en \rfloor$. Lemma 4.2 implies that $\sigma_p(e) \leq e^{1-k}$ for $e \in B_k$. The tail sum starting from block $h+1$ is

$$\begin{aligned} \sum_{k=h+1}^{\infty} \sum_{e \in B_k} \sigma_p(e) &\leq \left(\frac{en}{\ln b} + 1\right) \sum_{k=h+1}^{\infty} e^{1-k} \\ &\leq \left(\frac{en}{\ln b} + 1\right) \frac{e^{-h}}{1-e^{-1}} = O\left(\frac{1}{n^c p}\right). \end{aligned}$$

Consequently,

$$Pr[MB(F) \subseteq Q] = 1 - O\left(\frac{1}{n^c}\right).$$

The forests in a graph on $n+1$ vertices form a matroid of rank n, and minimum-weight bases correspond to minimum spanning forests. Therefore this solves the edge version of the problem as well:

COROLLARY 4.1. *For any weighted graph G on $n+1$ vertices, $0 < p < 1$, $c > 0$ and $b = 1/(1-p)$, there exists a set $Q \subseteq E(G)$ of size*

$$|Q| \leq e(c+1)n \log_b n + O(n)$$

such that for $F = E(p)$,

$$Pr[MST(F) \subseteq Q] = 1 - O\left(\frac{1}{n^c}\right).$$

Also, we have a randomized algorithm finding the covering set for any weighted matroid $(E, \mathcal{M}, w)$; the algorithm makes $O(\log_b n \log m)$ calls to a minimum basis oracle.

- Let $k = \lceil (c+2) \ln n \rceil$, $1 - p = (1-q)^k$.
- Repeat the following for $i = 1, \ldots, r = \lceil 16eq^{-1} \ln m \rceil$:
  - Sample $S_i \subseteq E$, each element independently with probability q.
  - Find $T_i = MB(S_i)$.
- For each edge, include it in Q if it appears in at least $8 \ln m$ different T_i's.

THEOREM 4.2. *This algorithm finds with high probability a set $Q \subseteq E$ such that*

$$|Q| \le 2e(c+2)n \log_b n + O(n)$$

and

$$Pr[MB(E(p)) \subseteq Q] = 1 - O\left(\frac{1}{n^c}\right).$$

Proof. We have $k = \lceil (c+2)\ln n \rceil$, $1-p = (1-q)^k$, $r = \lceil 16eq^{-1} \ln m \rceil$ and $E_k = \{e \in E : \sigma_p(e) \ge e^{-k}\}$. We claim that $E_k \subseteq Q$ with high probability. Let $S_i = E(q)$ and $T_i = MB(S_i)$. By the boosting Lemma 2.3, for any $e \in E_k$,

$$Pr[e \in T_i] \ge qe^{-1}.$$

Denoting by t_e the number of T_i's containing edge e, we get $\mathbf{E}[t_e] \ge rqe^{-1} \ge 16 \ln m$. By Chernoff bound (see [9, Theorem 4.2]), we derive $Pr[t_e < 8\ln m] < e^{-2\ln m} = \frac{1}{m^2}$, and thus $Pr[\exists e \in E_k; t_e < 8\ln m] < \frac{1}{m}$. Therefore with high probability, all edges in E_k are included in Q. If that happens, Q is a good covering set by the same argument as in the proof of Theorem 4.1. Indeed, we include all elements with $\sigma_p(e) \ge 1/n^{c+2}$ and the tail sum over the remaining elements is $O(1/n^c)$.

Now we estimate the size of Q. We have $\sum_{i=1}^{r} |T_i| \le rn$. Every element $e \in Q$ appears in $8\ln m$ different T_i's, therefore

$$|Q| \le \frac{\sum |T_i|}{8 \ln m} \le \lceil 2enq^{-1} \rceil \le 2e(c+2)n \log_b n + O(n),$$

using Lemma 2.4.

5 Lower bounds

For both vertex and edge variants of the problem, consider the complete graph K_n with the edge weights $w_{ij} = n|j-i| + i$ for $e = (i,j)$, $i < j$. In other words, if the vertices are placed equidistantly on a line, the edges are ordered primarily by their lengths. We first deal with the vertex variant.

LEMMA 5.1. *Let $0.5 \le p < 1$ (possibly a function of n) and $b = 1/(1-p)$. For the weighted graph described above, if Q contains $MST(V(p))$ with probability bounded away from zero, then*

$$|Q| \ge (1 - o(1))\ n \log_b n.$$

Proof. We claim that all edges of length smaller than $l = \log_b n - 4\log_b \ln n$ except at most $n/\ln n$ of them must be in Q. Assume that $n/\ln n$ such edges are not in Q and call this set S. We choose a subset $S' \subseteq S$ such that the intervals $[i,j]$ for $(i,j) \in S'$ are disjoint. Since the interval for any edge in S cannot overlap with more than $2\log_b^2 n$ other edges in S, we can choose such a subset of size $|S'| \ge \frac{n}{2\log_b^2 n \ln n}$. For any $e \in S'$,

$$Pr[e \in MST(V(p))] \ge p^2(1-p)^l \ge \frac{\ln^4 n}{4n}$$

since this event occurs, if the two vertices of e are chosen in $V(p)$ and none of the vertices between them are chosen. Moreover, the vertices involved in these events are disjoint for different edges in S', therefore the events are independent. Therefore,

$$Pr[S' \cap MST(W) = \emptyset] \le \left(1 - \frac{\ln^4 n}{4n}\right)^{|S'|} \le e^{-\Omega(\ln n)},$$

which means that with high probability, Q doesn't contain $MST(V(p))$.

The same result holds for the edge variant as well.

LEMMA 5.2. *Let $0.5 \le p < 1$ (possibly a function of n) and $b = 1/(1-p)$. For the weighted graph described above, if Q contains $MST(E(p))$ with probability bounded away from zero, then*

$$|Q| \ge \left(\frac{1}{2} - o(1)\right) n \log_b n.$$

Proof. We claim that for at least $(1 - 1/\ln n)\ n$ vertices, the incident edges shorter than $l = \frac{1}{2}(\log_b n - 3\log_b \ln n)$ must be contained in Q. Assume that it is not so, $|S| \ge n/\ln n$ and for each vertex $i \in S$, $e^*(i)$ is one of the incident edges shorter than l, which is not in Q. Choose $S' \subseteq S$ of size $|S'| \ge \frac{n}{\ln n \log_b n}$ such that $\forall i,j \in S'; |j-i| \ge \log_b n$. Then for a vertex $i \in S'$, let A_i be the event that $e^*(i) \in E(p)$, but no other incident edge shorter than l is in $E(p)$. We have

$$Pr[A_i] \ge p(1-p)^{2l} \ge \frac{\ln^3 n}{2n}.$$

The events for different vertices in S' are independent, because the edges involved are disjoint. Therefore

$$Pr\left[\bigcap_{i \in S'} \overline{A_i}\right] \le \left(1 - \frac{\ln^3 n}{2n}\right)^{|S'|} \le e^{-\Omega(\ln n)}$$

which means that with high probability, A_i occurs for some $i \in S'$. However, $MST(E(p))$ must contain the shortest edge in $E(p)$ incident with each vertex, therefore with high probability, Q doesn't contain $MST(E(p))$.

Acknowledgments

The authors would like to thank one the referees for comments that allowed to improve and simplify the exposition.

References

[1] D. Applegate and W. Cook: Solving Large-Scale Matching Problems, In D. Johnson and C.C. McGeoch, eds., Network Flows and Matchings, AMS 1993.

[2] D. Berstimas: The Probabilistic Minimum Spanning Tree, Networks 20 (1990), 245–275.

[3] B.Bollobás: Combinatorics - set systems, hypergraphs, families of vectors, and combinatorial probability, Cambridge University Press 1986.

[4] B.Bollobás, A.Thomason: Threshold functions, Combinatorica 7 (1987), 35–38.

[5] B. Chazelle: A Minimum Spanning Tree Algorithm with Inverse-Ackermann Type Complexity, J. ACM 47 (2000), 1028–1047.

[6] J. Holm, K. De Lichtenberg and M. Thorup: Poly-Logarithmic Deterministic Fully-Dynamic Algorithms for Connectivity, Minimum Spanning Tree, 2-Edge, and Biconnectivity, J. ACM 48(4) (2001), 723–760.

[7] N. Jing, Y.W. Huang, E.A.Rundensteiner: Hierarchical Encoded Path Views for Path Query Processing: An Optimal Model and Its Performance Evaluation, IEEE T. on Knowledge and Data Engineering 10(3) (1998), 409–432.

[8] D.R. Karger, P.N. Klein and R.E. Tarjan: A Randomized Linear-Time Algorithm to Find Minimum Spanning Trees, J. ACM 42(2) (1995), 321–328.

[9] R. Motwani and P. Raghavan: Randomized Algorithms, Cambridge University Press 1995.

[10] L.Valiant: The complexity of enumeration and reliability problems, SIAM J. on Computing 8 (1979), 410–421.