Backwards Analysis of the Karger-Klein-Tarjan Algorithm for Minimum Spanning Trees

Timothy M. Chan

Department of Mathematics and Computer Science
University of Miami, Coral Gables, FL 33124–4250, USA
E-mail: `tchan@cs.miami.edu`

April 29, 1998

Abstract

This note gives a short proof of a sampling lemma used by Karger, Klein, and Tarjan in the analysis of their randomized linear-time algorithm for minimum spanning trees.

Keywords: Minimum spanning trees; Randomized algorithms; Backwards analysis

1 Background

The problem of computing the minimum spanning tree in a weighted undirected graph has a long history, but a linear-time solution is realized only recently from the work of Karger, Klein, and Tarjan [2]. Their algorithm is randomized and works under a restricted RAM model of computation where the only allowable operations on the weights are comparisons. The basic idea is prune-and-search: one recursively applies (i) Borůvka steps to reduce the number of vertices by a constant factor, and (ii) random-sampling steps to reduce the number of edges.

The novelty of the algorithm lies in (ii), and to be more explicit, we need a definition. Let $G = (V, E)$ be the given connected graph with n vertices and m edges. An edge (u, v) is said to be *light with respect to a spanning tree* T if $(u, v) \in T$ or (u, v) has a smaller weight than some edge along the path from u to v in T. As is easily observed, edges that are not light cannot be in the minimum spanning tree and can therefore be pruned. The light edges can be identified in linear time by known techniques for verifying minimum spanning trees.

The key idea behind the algorithm is to choose T to be the minimum spanning tree of a random sample $R \subset E$ (computable by another recursive invocation). It turns out that the expected number of light edges with respect to T is sufficiently small so that with a proper choice of sample size, the overall expected running time of the algorithm is $O(m)$. Karger, Klein, and Tarjan's proof of this sampling lemma requires a simulation of Kruskal's algorithm; essentially the same approach is taken in all subsequent descriptions that the author is aware of at the time of this writing [1, 3]. Here we present an arguably simpler, more intuitive proof of a variant of this lemma.

Before giving a precise statement of the lemma, we have to deal with two minor technical issues. First, to ensure that the minimum spanning tree is indeed unique, we assume that no two edge weights are equal. Such a nondegeneracy assumption poses no problem if ties are broken in a consistent manner. Second, to ensure that the minimum spanning tree exists, we need connectedness of the sample subgraph. One remedy of this problem is to fix a spanning tree T_0 of G (which has only $n-1$ edges) and consider the minimum spanning tree of $R \cup T_0$, which we will denote by $\mathrm{MST}(R)$. Our lemma is the following:

Lemma 1.1 (Sampling Lemma) *For a random subset $R \subset E$ of size r, the expected number of edges that are light with respect to $\mathrm{MST}(R)$ is less than mn/r.*

Our proof of Lemma 1.1 makes use of the following simple characterization of light edges:

Observation 1.2 *An edge e is light with respect to $\mathrm{MST}(R)$ if and only if $e \in \mathrm{MST}(R \cup \{e\})$.*

Proof: Straightforward. To be self-contained, we include a proof sketch of the direction we will use, namely, the "only if" part.

Assume $e \notin \mathrm{MST}(R \cup \{e\})$ and note that $\mathrm{MST}(R \cup \{e\}) = \mathrm{MST}(R)$. If e is light with respect to $\mathrm{MST}(R \cup \{e\})$, then e has a smaller weight than another edge e' in some cycle of $\mathrm{MST}(R \cup \{e\}) \cup \{e\}$. But replacing e' with e in the spanning tree $\mathrm{MST}(R \cup \{e\})$ would yield a spanning tree with a smaller weight: a contradiction! □

2 Proof of the Sampling Lemma

Pick a random edge $e \in E$ (independent of R). It suffices to show that e is light with respect to $\mathrm{MST}(R)$ with probability less than n/r. By Observation 1.2, we only have to bound the probability that $e \in \mathrm{MST}(R \cup \{e\})$.

Let $R' = R \cup \{e\}$. We take an approach called "backwards analysis:" instead of adding a random edge to R, we imagine deleting a random edge from R'. Since e is equally likely to be any element in R' and $\mathrm{MST}(R')$ has precisely $n-1$ edges, the probability that $e \in \mathrm{MST}(R')$ conditioned to fixed choice of R' is at most $(n-1)/|R'| < n/r$. As this upper bound does not depend on R', it holds unconditionally and the result is proved. □

3 Comments

Lemma 1.1 differs from Karger, Klein, and Tarjan's version of the sampling lemma in several points. First, they obtain the subset R by Bernoulli sampling, i.e., by selecting each edge independently with probability r/m. Second, they prove not just an expected bound but also a high-probability bound. Third, they avoid the issue of connectedness in the sample subgraph by considering minimum spanning forests. In the interest of preserving the simplicity of our proof, we have ignored these modifications.

It is unclear why this simple proof has been missed. In fact, a weaker form of Observation 1.2 was already noted in the original paper by Karger [1]: an edge e is light with respect to $\mathrm{MST}(R)$ precisely when $e \in \mathrm{MST}(\mathrm{MST}(R) \cup \{e\})$. In Karger's matroid terminology, e is said to *improve the basis* $\mathrm{MST}(R)$.

On the other hand, it is easily proved that given $e \notin \mathrm{MST}(R)$, the edge e is light with respect $\mathrm{MST}(R)$ if and only if $\mathrm{MST}(R \cup \{e\}) \neq \mathrm{MST}(R)$. In Sharir and Welzl's framework of "LP-type problems" [6], the condition can be interpreted as saying that *e violates R*, or equivalently, *e violates the basis* $\mathrm{MST}(R)$. This connection could explain why backwards analysis—a popular technique in computational geometry and low-dimensional linear programming [3, 4, 5]—would find an application here.

References

[1] D. R. Karger. Random sampling and greedy sparsification for matroid optimization problems. *Mathematical Programming*, to appear. Also in *Proc. 34th IEEE Sympos. Found. Comput. Sci.*, pages 84–93, 1993.

[2] D. R. Karger, P. N. Klein, and R. E. Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *J. ACM*, 42:321–328, 1995.

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

[4] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms.* Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[5] R. Seidel. Backwards analysis of randomized geometric algorithms. In *New Trends in Discrete and Computational Geometry* (J. Pach, ed.), vol. 10 of *Algorithms and Combinatorics*, Springer-Verlag, pages 37–68, 1993.

[6] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Sympos. Theoret. Aspects Comput. Sci.*, Lect. Notes in Comput. Sci., vol. 577, Springer-Verlag, pages 569–579, 1992.