Do the minimum spanning trees of a weighted graph have the same number of edges with a given weight?

If a weighted graph G has two different minimum spanning trees $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$, then is it true that for any edge e in E_1, the number of edges in E_1 with the same weight as e (including e itself) is the same as the number of edges in E_2 with the same weight as e? If the statement is true, then how can we prove it?

graph-theory spanning-trees weighted-graphs

edited Jun 2 '12 at 23:23
Gilles ♦ 27.6k 7 74 145

asked Jun 2 '12 at 22:25
Aden Dong 521 4 19

One tricky but feasible approach is to show 1) Kruskal's algorithm can produce every minimal spanning tree and 2) all minimal spanning trees found by Kruskal have the same edge-weight multiset. – Raphael ♦ Mar 6 '13 at 10:38

2 Answers

Claim: Yes, that statement is true.

Proof Sketch: Let T_1, T_2 be two minimal spanning trees with edge-weight multisets W_1, W_2. Assume $W_1 \neq W_2$ and denote their symmetric difference with $W = W_1 \,\Delta\, W_2$.

Choose edge $e \in T_1 \,\Delta\, T_2$ with $w(e) = \min W$, that is e is an edge that occurs in only one of the trees and has minimum disagreeing weight. Such an edge, that is in particular $e \in T_1 \,\Delta\, T_2$, always exists: clearly, not all edges of weight $\min W$ can be in both trees, otherwise $\min W \notin W$. W.l.o.g. let $e \in T_1$ and assume T_1 has more edges of weight $\min W$ than T_2.

Now consider all edges in T_2 that are also in the cut $C_{T_1}(e)$ that is induced by e in T_1. If there is an edge e' in there that has the same weight as e, update T_1 by using e' instead of e; note that the new tree is still a minimal spanning tree with the same edge-weight multiset as T_1. We iterate this argument, shrinking W by two elements and thereby removing one edge from the set of candidates for e in every step. Therefore, we get after finitely many steps to a setting where all edges in $T_2 \cap C_{T_1}(e)$ (where T_1 is the updated version) have weights other than $g(e)$.

Now we can always choose $e' \in C_{T_1}(e) \cap T_2$ such that we can swap e and e'[1], that is we can create a new spanning tree

$$T_3 = \begin{cases} (T_1 \setminus \{e\}) \cup \{e'\} & , w(e') < w(e) \\ (T_2 \setminus \{e'\}) \cup \{e\} & , w(e') > w(e) \end{cases}$$

which has smaller weight than T_1 and T_2; this contradicts the choice of T_1, T_2 as minimal spanning trees. Therefore, $W_1 = W_2$.

1. The nodes incident of e are in T_2 connected by a path P; e' is the unique edge in $P \cap C_{T_1}(e)$.

edited Jan 14 '13 at 20:29

answered Jun 3 '12 at 14:08
Raphael ♦ 48.1k 17 116 252

3 In reference to Dave's comment, I came up with this proof after 0) believing I had a counter-example which I saw was wrong after TikZing it, 1) trying to prove the statement but failing, 2) trying to construct a counter-example based on where the proof failed and failing again, and finally 3) using the way these new examples failed to work for coming up with the proof. That's probably also why it is not as refined as it could be. – Raphael ♦ Jun 5 '12 at 11:13

yeas exactly, I don't understand what is meant by cyt induced by e in T_1 I had only seen cut like $(S, V - S)$ cut – dragoboy Sep 29 '14 at 12:07

@dragoboy Removing e disconnects T_1; one component forms S, the other the complement. – Raphael ♦ Sep 29 '14 at 13:23

Here is a slightly simpler argument that also works for other matroids. (I saw this question from another one.)

Suppose that G has m edges. Without loss of generality, assume that the weight function w takes on values in $[m]$, so we have a partition of E into sets $E_i := w^{-1}(i)$ for $i \in [m]$. We can do induction on the number j of non-empty E_i and number of vertices n in G; for $j = 1$ and any n, the statement is obvious.

A standard fact about matroids is that for every MST T there is a linear extension of the ordering induced by w so that the greedy algorithm produces T.

To close the induction, take t to be the largest number so that E_t is not empty. Set $E' = E_1 \cup \cdots \cup E_{t-1}$. Observe that any linear extension of w puts every edge in E' before any edge in E_t. According to the fact, any MST consists of a spanning forest F of the subgraph induced by E' and some edges from E_t. By the inductive hypothesis, each connected component of F has the same number of edges from each E_i for $i < t$. Since all the choices of F have the same size, the number of edges from E_t required to complete F to a spanning tree is independent of the choice of F and we are done.

edited Apr 13 at 12:48
Community ♦
1

answered Apr 9 '15 at 14:15
Louis
2,428 8 24

Can you *give* the matroid for the MST problem? I seem to remember that it is a tough thing to come up with, and I have yet to see it done (rigorously). Yes, we use greedy algorithms, but not *the* (canonical) greedy from matroid theory. – Raphael ♦ Apr 9 '15 at 19:47

That said, I think your core argument works (and does not need matroids at all): by correctness of Kruskal's algorithm and the fact that every MST *can* be obtained from a run of Kruskal with a specific (sorted) permutation of the weight multiset (rigorous proof pending), the claim follows. I write "proof pending" because it's neither trivial nor immediate: without using the claim itself it is not at all clear why Kruskal should find all MSTs. Clearly, if one *had* a different weight multiset, Kruskal would never find it! – Raphael ♦ Apr 9 '15 at 19:50

1. The matroid is the graphic matroid. Done! – Louis Apr 10 '15 at 4:27

2. You're confused. Abstractly, we are doing linear optimization over the basis polytope. One of the standard characterizations of matroids is that the greedy algorithm works for any choice of weights. All the w-minimal spanning trees are vertices of a face of this polytope. Now standard ideas from LP lead to the standard fact I mentioned. – Louis Apr 10 '15 at 4:32

1. Can you give a reference? I don't know *the* graphic matroid. 2. Now you drag LP into it, too! All I'm saying is that your answer lacks the matroid, and that without the matroid the line of reasoning seems to rely on the claim itself. If you have a way around that, please edit/clarify the answer. – Raphael ♦ Apr 10 '15 at 6:59