On the Spanning Trees of Weighted Graphs*

Ernst W. Mayr   C. Greg Plaxton†

Abstract

Given a weighted graph, let $W_1, W_2, W_3, \ldots$ denote the increasing sequence of all possible distinct spanning tree weights. Settling a conjecture due to Kano, we prove that every spanning tree of weight W_1 is at most $k-1$ edge swaps away from some spanning tree of weight W_k. Three other conjectures posed by Kano are proven for two special classes of graphs. Finally, we consider the algorithmic complexity of generating a spanning tree of weight W_k.

1 Introduction

The minimum spanning tree problem is a classic problem in computer science for which a large number of sequential, parallel and distributed algorithms have been devised. To the graph theorist, however, this body of work has provided little beyond the underlying proof of correctness of the original greedy algorithms due to Kruskal and Prim [Kr][Pr]. Evidently, the advances have occurred in the areas of data structures, parallel processing techniques and distributed protocols. In contrast, this paper attacks questions arising from a generalization of the minimum spanning tree concept that requires additional insight at the graph-theoretic level.

If one considers partitioning the spanning trees of a weighted graph into weight classes, a number of natural questions arise with regard to relationships between the classes. In a recent paper, Kano [Kan] posed four conjectures that were motivated by the previous work of Kawamoto, Kajitani and Shinoda [KKS]. Our main result is a proof that every minimum spanning tree is at most $k-1$ edge swaps away from some representative of the kth weight class, settling the first of Kano's conjectures. With regard to the three remaining conjectures, we offer a stronger unified conjecture and prove that it holds for two non-trivial families of graphs.

We also consider the algorithmic complexity of generating a representative of the kth weight class. For fixed k, we obtain a polynomial time algorithm. When k is part

*This work was supported in part by a grant from the AT&T Foundation and NSF grant DCR-8351757.

†Primarily supported by a 1967 Science and Engineering Scholarship from the Natural Sciences and Engineering Research Council of Canada.

of the input, the associated decision problem (KMST) is seen to be NP-hard using a reduction due to Johnson and Kashdan [JK] for the related kth best spanning tree problem (KBST). Lawler [La] gave a simple branch-and-bound algorithm for KBST with a running time that is pseudo-polynomial in k. The existence of such an algorithm provides a hopeful sign that a similar result might be possible for KMST. We do not achieve this goal, but by extending a determinant method due to Okada and Onodera [OO], we obtain an algorithm for which the running time is pseudo-polynomial in the edge weights as well as k.

2 Preliminaries

Let $G = (V, E)$ be a connected undirected graph with a real-valued weight $w(e)$ assigned to each edge $e \in E$. Let e^- and e^+ denote the endpoints of edge e. Parallel edges and self-loops are permissible. For any subset E' of E define the weight of E', denoted $w(E')$, as the sum of the weights of the edges of E', that is,

$$w(E') = \sum_{e \in E'} w(e).$$

A *spanning tree* P of G is any subset of E for which the graph (V, P) is acyclic and connected; in order to satisfy both of these properties simultaneously it is necessary that $|P| = |V| - 1$. In order to discuss the partition of the spanning trees of G into weight classes, we will make use of the following notation.

$$\begin{aligned}
T(G) &= \{P \mid P \text{ is a spanning tree of } G\} \\
W(G) &= \{x \mid w(P) = x \text{ for some } P \in T(G)\} \\
N(G) &= |W(G)| \\
W_i(G) &= \text{the } i\text{th smallest element of } W(G), 1 \leq i \leq N(G) \\
T_i(G) &= \{P \in T(G) \mid w(P) = W_i(G)\} \\
\rho(G, P) &= i, \text{ if and only if } P \in T_i(G) \\
\sigma(G, P) &= |T_i(G)|, \text{ if and only if } P \in T_i(G)
\end{aligned}$$

A spanning tree of weight $W_1(G)$ is a *minimum spanning tree* of G. In general, a spanning tree of weight $W_k(G)$ will be referred to as a *kth minimal spanning tree* (k-MST) of G. Note that some previous authors have preferred to formulate their results in terms of kth *maximal* spanning trees [Kan][KKS], an equivalent concept.

Let P and Q be spanning trees of G. Then $|P \setminus Q|$ is the *distance* between P and Q and will be denoted by $d(P, Q)$. For any nonnegative integer k, let $L_k(G, P)$ represent the set of all spanning trees Q of G such that $d(P, Q) \leq k$. Notice that $d(P, Q) = d(Q, P)$ so that $Q \in L_k(G, P)$ if and only if $P \in L_k(G, Q)$. For every $e \notin P$, let $Cyc(P, e)$ denote the fundamental cycle of G defined by e with respect to P. Given distinct edges a, b such that $a \in Cyc(P, b)$, the spanning tree $P \setminus \{a\} \cup \{b\}$ is defined to be a single *edge swap* away from P. For any two spanning trees P and Q, we note that the length of the shortest sequence of edge swaps required to transform P into Q is precisely $d(P, Q)$.

We will have occasion to make use of several well-known facts about 1-MSTs. The validity of each of these statements follows easily from the proof of correctness of Kruskal's greedy algorithm for computing a 1-MST [Kr].

Fact 1 *A spanning tree is a 1-MST of G if and only if it is a 1-MST in $L_1(G, P)$.*

Fact 2 *The unique heaviest edge in some cycle cannot belong to any 1-MST. A heaviest edge in some cycle cannot belong to every 1-MST.*

Fact 3 *The unique lightest edge in some cutset must belong to every 1-MST. A lightest edge in some cutset must belong to some 1-MST.*

Fact 4 *Every 1-MST contains the same distribution of edge weights.*

Fact 5 *Every 1-MST can be generated by Kruskal's algorithm.*

Given Fact 1, it is straightforward to verify that for any i-MST P of G, $L_{i-1}(G, P)$ contains a 1-MST. However, going in the other direction turns out to be much more difficult. Kawamoto, Kajitani and Shinoda [KKS] studied 2-MSTs and proved that for every 1-MST P there is a 2-MST Q such that $d(P, Q) = 1$. This led Kano to pose the following conjecture, for which the result of Kawamoto et al. corresponds to the special case $i = 2$.

Conjecture 1 *If P is a 1-MST of G then $L_{i-1}(G, P)$ contains an i-MST, $1 \le i \le N(G)$.*

Kano was able to prove that Conjecture 1 holds for $i = 3$ and 4. We will show that Conjecture 1 is, indeed, a theorem. Kano also proposed the following three conjectures, proving them for values of i less than or equal to 3, 4 and 3, respectively.

Conjecture 2 *If P is an i-MST in $L_i(G, P)$ then P is an i-MST of G.*

Conjecture 3 *If P is an i-MST of G then P is an i-MST in $L_{i-1}(G, P)$.*

Conjecture 4 *Let $\Gamma(i, j)$ denote the graph with vertex set $T_i(G)$ and an edge between each pair of i-MSTs P, Q such that $d(P, Q) \le j$. Then $\Gamma(i, i)$ is connected.*

3 Proofs

In this section, we prove Conjecture 1 along with a number of other results.

Lemma 3.1 *Let P and Q be spanning trees of a given graph G. Then for each edge $p \in P \setminus Q$ there is an edge $q \in Q \setminus P$ such that $p \in Cyc(P, q)$ and $q \in Cyc(Q, p)$.*

Proof: Let P, Q and p be as defined above and assume that there is no edge q satisfying the requirements of the lemma. Then there must be a path made up of edges from $P \setminus \{p\}$ that connects the two endpoints of p. But then P contains a cycle, contradicting the assumption that it is a spanning tree. □

Definition 3.1 *Given a graph $G = (V, E)$, let ξ represent G or any subset of E. If C and D are disjoint subsets of E, then the graph or set of edges produced from ξ by* contracting *the edges of C and* discarding *the edges of D will be denoted $\xi[C, D]$. Notice that there is a 1-1 correspondence between the edges of $\xi[C, D]$ and those edges of ξ that do not belong to $C \cup D$. We identify the pairs of edges determined by this correspondence, inheriting edge weights in the weighted case.*

We note that if P is a spanning tree of G, then $P[c, d]$ is a spanning tree of $G[c, d]$ if and only if $c \in P$ and $d \notin P$, or $c \notin P$ and $d \in Cyc(P, c)$.

Lemma 3.2 *Let P be a spanning trees of a weighted graph $G = (V, E)$. For any edge $e \notin P$, let $G' = G[, e]$ and $P' = P$. For any edge $e \in P$, let $G' = G[e,]$ and $P' = P[e,]$. In either case, the following statements hold:*

1. *P' is a spanning tree of G'.*
2. *$\rho(G', P') \leq \rho(G, P)$.*
3. *$\sigma(G', P') \leq \sigma(G, P)$.*

Proof: The above statements follow easily once we exhibit an injective map Φ from $T(G')$ to $T(G)$ that takes P' to P and also satisfies a "constant displacement" property, namely, there exists a real value Δ such that for all spanning trees $R' \in T(G')$,

$$w(\Phi(R')) = w(R') + \Delta. \tag{1}$$

For the case $e \notin P$, take $\Phi(R')$ to be simply R' so that equation 1 holds with $\Delta = 0$. For $e \in P$, let $\Phi(R')$ be the unique spanning tree $R \in T(G)$ such that $R[e,] = R'$ and set $\Delta = w(e)$. □

Lemma 3.3 *Let P be a 1-MST of a given weighted graph $G = (V, E)$. For any edge $e \in P$, let $P' = P[e,]$. For any edge $e \notin P$, let $P' = P[e, h]$ where h is a heaviest edge on $Cyc(P, e) \setminus \{e\}$. In either case, P' is a 1-MST of $G[e,]$.*

Proof: If $e \in P$ then Lemma 3.2 applies. Assume that $e \notin P$. Appealing to Fact 5, consider an execution of Kruskal's algorithm on input G that generates the 1-MST P. As each edge $p \in P$ gets added to P, Kruskal's algorithm running on input $G[e,]$ can correctly select p with one exception: the edge that first puts e^- and e^+ into the same component must be omitted. This edge must have weight $w(h)$ since it is a heaviest edge on the path from e^- to e^+ in P. □

Definition 3.2 *A graph $G = (V, E)$ will be called a* bispanning graph *if E is the union of two disjoint spanning trees P and Q. Such a bispanning graph will be denoted by the triple (V, P, Q).*

Definition 3.3 *A* simple bispanning graph *is a bispanning graph that contains only cycles of length 2.*

The following result was obtained previously by Kano using Hall's Theorem [Kan]. Here we provide an alternative proof that is explicitly constructive and introduces some of the ideas that will be used to prove Conjecture 1.

Theorem 1 *Let P be a 1-MST and Q be an arbitrary spanning tree of a given weighted graph $G = (V, E)$. Then there exists a bijection Φ from $P \setminus Q$ to $Q \setminus P$ such that for every edge $e \in P \setminus Q$, $\Phi(e) \in Cyc(Q, e)$ and $w(\Phi(e)) \geq w(e)$.*

Proof: It is sufficient to determine a bijection between the disjoint spanning trees $P' = P \setminus Q$ and $Q' = Q \setminus P$ of the bispanning graph $G' = G[P \cap Q, E \setminus P \setminus Q]$. This is due to the observation that $Cyc(Q', e) \subseteq Cyc(Q, e)$ for all $e \in P'$.

Let p be a heaviest edge in P' and let q be any edge in Q' for which $p \in Cyc(P', q)$ and $q \in Cyc(Q', p)$. Lemma 3.1 guarantees that we can find such a q. As P' is a 1-MST, we must have $w(Q) \geq w(P)$. Let $\Phi(p) = q$.

To determine the next component of the bijection, repeat this procedure on the bispanning graph $G'[q, p]$ with disjoint spanning trees $P'[q, p]$ and $Q'[q, p]$. $P'[q, p]$ is a 1-MST of $G'[q, p]$ by Lemma 3.3. Furthermore, for all $e \in P'[q, p]$,

$$Cyc(Q'[q, p], e) = Cyc(Q', e) \setminus \{q\} \subseteq Cyc(Q', e) \subseteq Cyc(Q, e)$$

so that subsequent assignments to Φ will be guaranteed to satisfy the condition $\Phi(\cdot) \in Cyc(Q, \cdot)$. $\square$

Lemma 3.4 *Conjecture 1 holds if and only if there is no weighted bispanning graph $B = (V, P, Q)$ such that $d(P, Q) \geq \rho(B, Q) > \rho(B, P) = 1$ and $\sigma(B, Q) = 1$.*

Proof: The "only if" directon is easy. To establish the "if" direction, we will prove the contrapositive. Given a counterexample (G, P, i) to Conjecture 1, let Q be a closest i-MST to P. Then repeated application of Lemma 3.2 proves that the bispanning graph $G' = G[P \cap Q, E \setminus P \setminus Q]$ with disjoint spanning trees $P' = P \setminus Q$ and $Q' = Q \setminus P$ also violates the conjecture, that is, $d(P', Q') \geq \rho(G', Q') > \rho(G', P') = 1$.

It remains to be proven that $\sigma(G', Q') = 1$. If not, there must be a spanning tree Q'' of G' such that $w(Q'') = w(Q')$ and $Q'' \neq Q'$. Then $d(P', Q'') \geq d(P', Q')$ by the definition of Q. On the other hand, $d(P', Q'') < d(P', Q')$ since Q'' must have at least one edge in common with P', whereas Q' has none. Thus, Q' must have unique weight in $T(G')$, as required. $\square$

Theorem 2 *There is no weighted bispanning graph $B = (V, P, Q)$ such that $d(P, Q) \geq \rho(B, Q) > \rho(B, P) = 1$ and $\sigma(B, Q) = 1$. Hence, Conjecture 1 holds by Lemma 3.4.*

Proof: Assume the theorem is false and let $B = (V, P, Q)$ be a counterexample with smallest possible $|V|$. We will establish a contradiction by exhibiting a smaller counterexample B'.

As in the proof of Theorem 1, let p be a heaviest edge in P and let q be any edge in Q for which $p \in Cyc(P, q)$ and $q \in Cyc(Q, p)$. Since P is a 1-MST and $\sigma(B, Q) = 1$, we must have $w(Q) > w(P)$. Therefore, q is the unique heaviest edge in $Cyc(P, q)$ and does not belong to any 1-MST of B by Fact 2.

Now consider the smaller bispanning graph $B' = B[q, p]$ with disjoint spanning trees $P' = P[q, p]$ and $Q' = Q[q, p]$. Clearly, $d(P', Q') = d(P, Q) - 1$. Furthermore, $\rho(B', P') = 1$ and $\sigma(B', Q') = 1$ by Lemmas 3.3 and 3.2, respectively. In order to show that B' is a counterexample to the theorem, it is sufficient to prove that $\rho(B', Q') < \rho(B, Q)$. Lemma 3.2 gives us only $\rho(B', Q') \leq \rho(B, Q)$, but in the present case that argument can be strengthened to yield the desired strict inequality. Namely, let Φ be the injective map taking $R' \in T(B')$ to $R \in T(B)$ such that $R' = R[q, p]$ and observe that Φ does not map any spanning tree of B' into $T_1(B)$. $\square$

The statements proven by the next three lemmas all have the structure of Lemma 3.4, underscoring the similarities between Kano's various conjectures. Note, however, that Lemma 3.5 is proven in one direction only.

Lemma 3.5 *Conjecture 2 holds if there is no weighted bispanning graph $B = (V, P, Q)$ such that $d(P, Q) \geq \rho(B, Q) > \rho(B, P)$ and $\sigma(B, Q) = 1$.*

Proof: We prove the contrapositive. Accordingly, let (G, R, i) be a counterexample to Conjecture 2, that is, R is an i-MST in $L_i(G, R)$ but not an i-MST of G. There must be a least integer $j < i$ such that $L_i(G, R)$ does not contain a j-MST. By Fact 1, there must be a k-MST P such that $d(P, R) \leq i - k$ for some $k < j$. Let Q be a closest j-MST to P. Since Q is not contained in $L_i(G, R)$ we have

$$d(P, Q) \geq i - d(P, R) \geq k.$$

Now consider the bispanning graph $G' = G[P \cap Q, E \setminus P \setminus Q]$ with disjoint spanning trees $P' = P \setminus Q$ and $Q' = Q \setminus P$. By repeated application of Lemma 3.2,

$$d(P', Q') \geq \rho(G', Q') > \rho(G', P') \text{ and } \sigma(G', Q') = 1$$

so the proof is complete. $\square$

Lemma 3.6 *Conjecture 3 holds if and only if there is no weighted bispanning graph $B = (V, P, Q)$ such that $d(P, Q) \geq \rho(B, P) > \rho(B, Q)$ and $\sigma(B, Q) = 1$.*

Proof: The proof is similar to that given for Lemma 3.4. In this case, assume that (G, P, i) is a counterexample to Conjecture 3, let j be any integer less than i such that $L_{i-1}(G, P)$ does not contain a j-MST, and let Q be a closest j-MST to P. □

Lemma 3.7 *Conjecture 4 holds if and only if there is no weighted bispanning graph $B = (V, P, Q)$ such that $d(P, Q) > \rho(B, P) = \rho(B, Q)$ and $\sigma(B, Q) = 2$.*

Proof: Once again, the proof is similar to that given for Lemma 3.4. Assume that (G, i) is a counterexample to Conjecture 4, and let P, Q be a closest pair of i-MSTs belonging to different connected components of $\Gamma(i, i)$. □

Given a weighted bispanning graph $B = (V, P, Q)$ such that Q has unique weight in $T(B) \setminus \{P\}$, let $i = \rho(B, P)$, $j = \rho(B, Q)$, $a = \max\{i, j\}$ and $b = \min\{i, j\}$. Unifying the "if" statements of Lemmas 3.4 through 3.7, we find that all of Kano's conjectures hold if we can guarantee that $d(P, Q) \leq \max\{a - 1, b\}$. This bound does not appear to be attainable for all values of a and b. We expect that it can be strengthened to the inequality stated below.

Conjecture 5 *Let P, Q, a, b be as defined in the preceding paragraph. Then*

$$d(P,Q) \leq \begin{cases} a, & \text{if } a = b; \\ a - 1, & \text{if } a > b = 1; \\ \left\lfloor \frac{a+b-1}{2} \right\rfloor, & \text{otherwise.} \end{cases}$$

This conjecture is strongest possible if $a - b$ is odd, $a = b$ or $b = 1$, as may be easily seen by assigning appropriate edge weights to simple bispanning graphs. Furthermore, Conjecture 5 implies the following bound for general graphs that is *optimal* in the sense that equality can be achieved for all values of a and b.

Lemma 3.8 *Let P be an i-MST of a given weighted graph G and suppose that Q is a closest j-MST to P in $T(G) \setminus \{P\}$. Let $a = \max\{i, j\}$ and $b = \min\{i, j\}$. If Conjecture 5 holds, then*

$$d(P,Q) \leq \begin{cases} a, & \text{if } a = b; \\ \max\left\{a - b, \left\lfloor \frac{a+b-1}{2} \right\rfloor\right\}, & \text{otherwise.} \end{cases}$$

Proof: Apply Lemma 3.2 until the bispanning graph $G[P \cap Q, E \setminus P \setminus Q]$ is obtained, then use Conjecture 5. □

Thus far, we have been able to prove Conjecture 5 only for certain special classes of graphs. We will now examine two such families that were obtained by restricting: the graph structure, in the first case, and the set of allowable edge weights, in the second case.

Definition 3.4 *Let P and Q be spanning trees of a given graph G. Then P and Q are related by a* parallel swap *if and only if there exists a bijection Φ from $P \setminus Q = \{e_1, \ldots, e_t\}$ to $Q \setminus P = \{\Phi(e_1), \ldots, \Phi(e_t)\}$ such that each of the 2^t sets of $|P|$ edges containing $P \cap Q$ and exactly one edge from each pair $\{e_i, \Phi(e_i)\}, 1 \leq i \leq t$, is a spanning tree of G.*

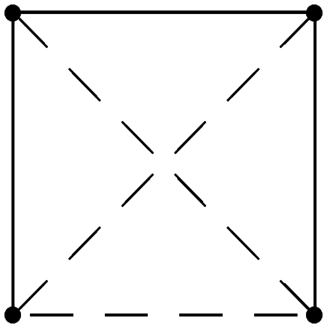

Figure 1: The smallest bispanning graph without a parallel swap.

As an aside, it is possible to prove that spanning trees P and Q are related by a parallel swap if and only if Lemma 3.1 is satisfied by a unique $q \in Q \setminus P$ for each $p \in P \setminus Q$. This means that one may readily determine whether or not two particular spanning trees are related by a parallel swap.

Theorem 3 *Let $B = (V, P, Q)$ be a bispanning graph for which P and Q are related by a parallel swap. Then Conjecture 5 holds.*

Proof: Let B be a counterexample with smallest possible $|V|$. Without loss of generality, we can assume that B is a simple bispanning graph. Observe that for all $R \in T(B)$, $\overline{R} = E \setminus R$ also belongs to $T(B)$ and $\rho(B, R) + \rho(B, \overline{R}) = N(B) + 1$. Hence, P has unique weight in $T(B) \setminus \{Q\}$. If either P or Q is a 1-MST then the result follows easily by Fact 1. If $w(P) = w(Q)$ then contract any cycle to obtain a smaller counterexample. If two possible edge swaps increase (decrease) the weight of P by different amounts, then contract the cycle corresponding to the larger change in order to obtain a smaller counterexample.

The preceding argument allows us to assume that s of the possible edge swaps in B increase the weight of P by some amount δ_1 while each of the remaining $|V| - 1 - s$ edge swaps causes a decrease of δ_2. It is straightforward to prove that a weighted bispanning graph of this form cannot provide a counterexample to Conjecture 5. □

If Conjecture 5 holds for all bispanning graphs with up to n vertices, then Conjectures 2, 3 and 4 must hold for all values of i less than or equal to $n-1$, n and $n-1$, respectively. It is easy to check that every bispanning graph $B = (V, P, Q)$ with $|V| \leq 3$ admits a parallel swap between P and Q. For $|V| = 4$, only the bispanning graph in Figure 1 does not have this property, but we can show by case analysis that Conjecture 5 holds for this particular graph as well. These observations re-establish the results obtained by Kano with respect to the small cases of Conjectures 2 through 4.

Another interesting class of weighted graphs for which we can prove Conjecture 5 are those with edge weights drawn from an arithmetic sequence of length 3. Weighted graphs of this sort arise in the analysis of LCR networks [Kaj], and Kano was able to prove Conjecture 1 for such graphs.

Theorem 4 *Let $B = (V, P, Q)$ be a weighted bispanning graph with edge weights drawn from the set $\{a, a+d, a+2d\}$. Then Conjecture 5 holds.*

Proof Sketch: Assume without loss of generality that $a = 0$ and $d = 1$, and let $B = (V, P, Q)$ be a counterexample with smallest possible $|V|$. We will briefly outline the proof of the hardest case, which turns out to occur when $W(Q) > W(P) > W_1(B)$. Let Q' be a lightest spanning tree in $L_1(Q)$. Either $W(Q') = W(Q) - 1$ or $W(Q') = W(Q) - 2$. If $W(Q') = W(Q) - 1$, it is possible to prove that $\sigma(B, Q) > 1$, a contradiction. If $W(Q') = W(Q) - 2$ then we show that P must be a 2-MST with $W(P) = W_1(B) + 1$ or else $\sigma(B, Q) > 1$. It is now sufficient to prove that there is an edge $q \in Q$ of weight 2 such that $Cyc(P, q)$ contains an edge p of weight 0, since this implies the existence of a smaller counterexample $B[q, p]$. $\square$

4 The Generation Problem

For fixed k, Theorem 2 yields a polynomial time algorithm for computing a k-MST. In particular, if $n = |V|$ and $m = |E|$ we have

Corollary 4.1 *Given a weighted graph G, a k-MST of G can be generated in $O((mn)^{k-1})$ time.*

Proof: First compute a 1-MST M. This can be done in $O(m + n \log n)$ time using Fibonacci heaps [FT], although for the present argument it would suffice to use one of the simpler 1-MST algorithms with a higher asymptotic running time. Then compute the weight of all spanning trees that are strictly fewer than k edge swaps away from M. There are at most

$$\sum_{i=0}^{k-1} \binom{n-1}{i} \binom{m-n+1}{i} = O((mn)^{k-1}) \tag{2}$$

such spanning trees and the weight of each tree can be computed incrementally in constant time by performing the enumeration in a depth-first manner. At any point in the computation we need to remember the k largest distinct weights found so far. This only increases the running time by a constant factor since k is fixed. $\square$

In order to discuss complexity issues, we introduce the language problem associated with the problem of finding a k-MST.

Definition 4.1 *Given a graph $G = (V, E)$, positive integer weights $w(e)$ for each $e \in E$, positive integers k and B. The* KMST problem *is to determine whether or not there are k spanning trees of G with distinct weights less than or equal to B.*

By omitting the word "distinct" from Definition 4.1, we get the definition of the kth best spanning tree problem (KBST).

Theorem 5 *KMST is NP-hard.*

Proof: The proof is identical to the one given in [JK] for KBST, which is not known to be in NP. The reduction is from HAMILTON CIRCUIT. □

Note that the $O((mn)^{k-1})$ algorithm described earlier does not solve KMST in time pseudo-polynomial in k. The running time of such an algorithm would have to be polynomial in $|V|$, k, $\log B$ and $\log w_{max}$, where w_{max} is the weight of the heaviest edge. Interestingly, a pseudo-polynomial time solution *is* known for KBST; the approach is to generate all spanning trees up to the kth [La]. Unfortunately, this method is not powerful enough for KMST since there may be exponentially many spanning trees belonging to classes below the kth.

We will now present a determinant-based KMST algorithm for which the running time is pseudo-polynomial in the edge weights as well as k. Let G be a graph with $n+1$ vertices $v_0, \ldots, v_n$ and m edges $e_1, \ldots, e_m$. Construct the $m \times (n+1)$ matrix A with

$$a_{ij} = \begin{cases} +1, & \text{if } e_i = (v_j, v_k) \text{ for some } k > j; \\ -1, & \text{if } e_i = (v_j, v_k) \text{ for some } k < j; \\ 0, & \text{otherwise.} \end{cases}$$

and let A_0 be the $m \times n$ matrix A with column 0 deleted. Letting A_S denote the matrix (a_{ij}) for $i \in S$ and j ranging from 1 to n, we have

$$\det A_0^T A_0 = \sum_{\substack{S \subseteq \{1,\ldots,m\} \\ |S|=n}} [\det A_S]^2$$

by the Binet-Cauchy formula (see [Kn]). Furthermore,

$$\det A_S = \begin{cases} \pm 1, & \text{if } (V,S) \text{ is a tree;} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

so that the determinant of $A_0^T A_0$ is the number of spanning trees of G. This result is due to Okada and Onodera [OO].

Similary, we can construct the $m \times (n+1)$ matrix B with

$$b_{ij} = \begin{cases} +x^{w(e_i)}, & \text{if } e_i = (v_j, v_k) \text{ for some } k > j; \\ -x^{w(e_i)}, & \text{if } e_i = (v_j, v_k) \text{ for some } k < j; \\ 0, & \text{otherwise.} \end{cases}$$

and let B_0 be the $m \times n$ matrix A with column 0 deleted. Let $p(x)$ be the determinant of $B_0^T A_0$, and let p_k be the coefficient of x^k in $p(x)$.

$$\begin{aligned} \det B_0^T A_0 &= \det\left[(x^{w(e_1)} \cdots x^{w(e_m)}) A_0^T A_0\right] \\ &= \sum_{\substack{S \subseteq \{1,\ldots,m\} \\ |S|=n}} [\det A_S]^2 x^{w(S)} \\ &= \sum p_k x^k \end{aligned}$$

Using equation 3 we find that p_k is the number of spanning trees of G with weight k, that is, $p(x)$ is the generating function for the number of spanning trees in each weight class of G.

This relationship leads us to a pair of algorithms for solving KMST. The first idea is to compute $\det B_0^T A_0$ for all integer values of x from 0 up to W_{max}, where $W_{max} = W_{N(G)}(G)$, and then interpolate to obtain $p(x)$. The interpolation is the most costly step, and it involves solving a system of $W_{max}+1$ equations with $O(W_{max}^2)$ digit integer coefficients. This can be done in time polynomial in W_{max} by the result of Edmonds [Ed], who proved that Gaussian elimination with pivoting is in $\mathcal{P}$ for exact rational arithmetic.

A second approach is to obtain an upper bound for p_k and then compute $\det B_0^T A_0$ for a single value of x that is large (or small) enough to ensure that all of the coefficients p_k can be extracted from the final result. Letting $C = B_0^T A_0$ with $x = 1$, we can derive a suitable bound on p_k as follows.

$$\begin{aligned} p_k &\leq \det C \\ &\leq \prod_{1\leq i\leq n} \left(\sum_{1\leq j\leq n} c_{ij}^2 \right)^{1/2} \end{aligned}$$

The first inequality follows from the fact that $p_k \geq 0$ for all k; the second is Hadamard's inequality.

Either of these methods gives a KMST algorithm with running time pseudo-polynomial in the edge weights as well as k. At the expense of an extra polynomial factor, we can easily convert them into algorithms for generating a k-MST.

5 Open Problems

We have succeeded in proving one of Kano's four conjectures; the remaining three would be established by a proof of Conjecture 5. The existence of a pseudo-polynomial (in k alone) algorithm for generating a k-MST remains open.

References

[Ed] J. Edmonds. Systems of distinct representatives and linear algebra. *J. of Research and the National Bureau of Standards*, **71**B (1967), 241-245.

[FT] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *JACM*, **34** (1987), 596-615.

[JK] D. B. Johnson and S. D. Kashdan. Lower bounds for selection in $X+Y$ and other multisets. *JACM*, **25** (1978), 556-570.

[Kaj] Y. Kajitani. Graph theoretical properties of the node determinant of an LCR network. *IEEE Trans. Circuit Theory*, CT-**18** (1971), 343-350.

[Kan] M. Kano. Maximum and kth maximal spanning trees of a weighted graph. Combinatorica, **7** (1987), 205-214.

[KKS] T. Kawamoto, Y. Kajitani and S. Shinoda. On the second maximal spanning trees of a weighted graph (in Japanese). *Trans. IECE of Japan*, **61**-A (1978), 988-995.

[Kn] D. E. Knuth. *The Art of Computer Programming Vol. I: Fundamental Algorithms*, Addison-Wesley, Reading, Mass.

[Kr] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.*, **7** (1956), 48-50.

[La] E. L. Lawler. A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem. *Management Sci.*, **18** (1972), 401-405.

[OO] Okada and Onodera. *Bull. Yamagata Univ.*, **2** (1952), 89-117 (cited in [Kn]).

[Pr] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical J.*, **36** (1957), 1389-1401.