Counting and Constructing Minimal Spanning Trees

Perrin Wright
Department of Mathematics
Florida State University
Tallahassee, FL 32306-3027

Abstract. We revisit the minimal spanning tree problem in order to develop a theory of construction and counting of the minimal spanning trees in a network. The theory indicates that the construction of such trees consists of many different choices, all independent of each other. These results suggest a block approach to the construction of all minimal spanning trees in the network, and an algorithm to that effect is outlined as well as a formula for the number of minimal spanning trees.

Key words and Phrases: Minimal spanning trees, equal edge replacements, equivalent edges, choices.

1991 Mathematics Subject Classification. Primary 05C05, 68R10.

1. Introduction. Let G be a connected loop-free nondirected graph, let T be a spanning tree of G, and let f be an edge of G not in T. We define $P(f,T)$, the *path of f in T*, to be the unique simple path in T that joins the vertices of f. Consider an ordered pair (e,f) of edges such that $e \in T$, $f \notin T$ and e is on $P(f,T)$. Define $S = (T \backslash \{e\}) \cup \{f\}$. Then S is also a spanning tree of G. We say that the pair (e,f) is an *edge replacement in T*, and we write $e \to f$ to signify that e is replaced by f. We write $T \to S$ and say that S has been obtained from T by an edge replacement. Observe that the pair (f,e) is an edge replacement in S and is the inverse of (e,f). Thus, $T \to S$ if and only if $S \to T$. Edge replacements are also called T-exchanges, as in [1].

Now suppose further that G has a weight function $w : E \to \{1,2,3,\ldots\}$, where E is the set of edges of G. We say that G is a *network*. A tree T in G is a *minimal spanning tree* (MST) if T is a spanning tree whose edge sum $\sum w(e)$ is the minimum over all spanning trees of G. An overview and an extensive bibliography of the MST problem can be found in [2].

If T is a MST, an *equal edge replacement* is an edge replacement (e,f) in T such that $w(e) = w(f)$. Clearly, the result is another MST. The following result is well known.

Theorem 1. *Let T and S be minimal spanning trees in the network G. Then there is a sequence $T_0, T_1, \ldots, T_n$ of minimal spanning trees in G such that $T_0 = T$, $T_n = S$ and*

$T_i \to T_{i+1}$ is an equal edge replacement for each i.

Corollary 1.1. *Any two minimal spanning trees in G have the same spectrum of edge weights, i.e., have the same number of edges of weight k, for each k.*

This author does not know the origin of Theorem 1. A version of it appears as an exercise in [4]. Corollary 1.1, in spite of its obvious appeal and impact, is not mentioned in the textbooks. The corollary suggests that it might be possible to partition the process of constructing an MST, and we will explore such a partitioning in this paper.

Proof of Theorem 1. Let k be the smallest positive integer such that T has an edge of weight k not in S. Thus, on all edges of weight less than k, T and S coincide. Let $e \in T$, $e \notin S$ and $w(e) = k$.

In $S \cup \{e\}$ a cycle is formed by e and the path $P(e, S)$. Every edge on this cycle has weight $\leq k$; otherwise, we could replace some edge in S by e to obtain a spanning tree with smaller edge sum, contrary to hypothesis. Some edge on $P(e, S)$ must have weight exactly equal to k; for if all edges of $P(e, S)$ had smaller weights they would belong to T as well, and $\{e\} \cup P(e, S)$ would be a cycle in T. Furthermore, at least one of the edges of weight k on $P(e, S)$ must not belong to T, for the same reason.

Choose an edge f on $P(e, S)$ such that $f \notin T$ and $w(f) = k$. Then (e, f) is an equal edge replacement from S_1 to S, where $S_1 = (S \backslash \{f\}) \cup \{e\}$. But S_1 is a MST with more edges in common with T than S. The proof follows by downward induction on the number of edges of T not in S. □

We glean two properties from this proof for future reference:

(1) If the edge f does not belong to the MST T, then all edges of $P(f, T)$ have weight $\leq w(f)$.

(2) If $f \in T_0$ and $f \in T_n$, then the sequence $T_0 \to T_1 \to \ldots \to T_n$ may be chosen so that $f \in T_i$ for each i.

2. Property E and electability. Let f be an edge of G and let T be a MST of G. We say that f has *Property* $E(T)$ if either $f \in T$ or the path $P(f, T)$ contains an edge g with $w(g) = w(f)$.

Property $E(T)$ appears to depend on the choice of the tree T, but the next two lemmas show otherwise.

Lemma 2.1. *Property $E(T)$ is invariant under equal edge replacements; that is, if f has Property $E(T)$ and if $T \to S$ is an equal edge replacement, then f has Property $E(S)$.*

Proof. We must show that f has Property $E(S)$. There are two cases.

Case 1. f is involved in the edge replacement. If $f \in T$, then f is replaced by an edge e

such that $w(e) = w(f)$. Then e is on $P(f, S)$, so f has Property $E(S)$.

If $f \notin T$, then f replaces some edge of T and therefore $f \in S$. Thus f has Property $E(S)$.

Case 2. f is not involved in the edge replacement. If $f \in T$, then $f \in S$ also, since f is not the edge replaced.

If $f \notin T$, then the path $P(f, T)$ contains an edge g with $w(g) = w(f)$. Let (e, e') be the edge replacement from T to S. If e is not on $P(f, T)$, then $P(f, S) = P(f, T)$, and there is nothing to prove.

If e belongs to $P(f, T)$, we must compute $P(f, S)$ (see Figure 1):

$$P(f, S) = P(f, T) \cup P(e', T) \cup \{e'\} \setminus P(f, T) \cap P(e', T).$$

Observe that $P(f, T) \cap P(e', T)$ is a simple arc whose edges include e and perhaps others, but not e'. (If $P(f, T) \cap P(e', T)$ were not a simple arc, then $P(f, T) \cup P(e', T)$ would contain a cycle, which cannot occur in T.)

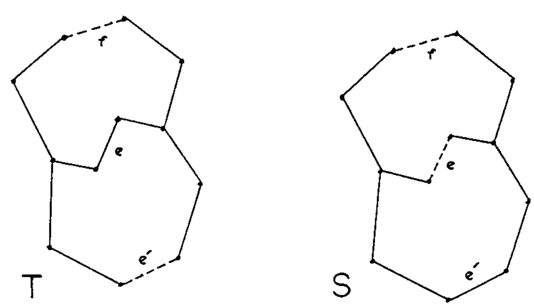

Figure 1. The relationship between $P(f, S)$, $P(f, T)$ and $P(e', T)$.

Since $e \in P(f, T)$, we have by property (1) that $w(e) \leq w(f)$. There are two subcases:

(a) $w(e) = w(f)$.

In this case, e' is the desired edge, since $e' \in P(f, S)$ and $w(e') = w(e) = w(f)$.

(b) $w(e) < w(f)$.

In this case, since (e, e') is an equal edge replacement, every edge on $P(e', T)$ has weight $\leq w(e') = w(e)$. In particular, the edges of $P(f, T) \cap P(e', T)$ have weight strictly less than $w(f)$, and thus g is not among them. Therefore $g \in P(f, S)$.

In all cases, either $f \in S$ or $P(f, S)$ contains an edge of weight $w(f)$. Thus, f has Property $E(S)$. $\square$

Lemma 2.2. *If the edge f has Property $E(T)$ for some minimal spanning tree T, then f has Property $E(S)$ for an arbitrary minimal spanning tree S.*

Proof. Suppose f has Property $E(T)$. Let S be another MST. By Theorem 1, there is a sequence of equal edge replacements from T to S. By Lemma 2.1, f has Property $E(S)$.$\square$

By virtue of Lemma 2.2 we may now generalize the notation $E(T)$ and say that the edge f has Property E if for some (all) T, f has property $E(T)$.

An edge e in G is *electable* if e belongs to some MST.

Theorem 2. *The edge e is electable if and only if e has Property E.*

Proof. If e is electable, then e belongs to a MST and hence has Property E.

If e has Property E, then for any MST T, either $e \in T$ or there is an edge e' of equal weight on $P(e,T)$. If $e \in T$, e is electable. If $e \notin T$, then we can replace e' by e to obtain another MST S such that $e \in S$. Again, e is electable. □

Theorem 2 provides a conclusive test for electability. To test the edge e, it is necessary only to examine a single MST T to determine whether e has Property $E(T)$. Nonelectable edges can be deleted from G at once, as they will never appear in an MST.

3. Equivalent edges. We define an equivalence relation $\sim$ on the electable edges of G: $e \sim f$ if and only if $e = f$ or there exists a sequence $T_0 \rightarrow T_1 \rightarrow \ldots \rightarrow T_n$ of equal edge replacements transforming e into f.

The relation $\sim$ is clearly reflexive and symmetric. If $e \sim f$ and $f \sim g$, then there are sequences $T_0 \rightarrow T_1 \rightarrow \ldots \rightarrow T_n$ and $S_0 \rightarrow S_1 \rightarrow \ldots \rightarrow S_m$ transforming e to f and f to g, respectively. By Theorem 1 and property (2), there is a sequence $T_n \rightarrow \ldots \rightarrow S_0$ leaving f fixed. The composition of these sequences establishes that $e \sim g$, and thus $\sim$ is transitive. Let $[e]$ denote the equivalence class of e under $\sim$.

Theorem 3. *Over all minimal spanning trees of G, the number of edges from $[e]$ is constant.*

Proof. If T and S are MSTs, then there is a sequence of equal edge replacements from T to S. Each equal edge replacement replaces an edge by another edge in the same equivalence class. □

Consequently, we define $i([e])$ to be the number of edges of $[e]$ which must be included in every MST. This number is well defined, by Theorem 3. We then obtain the following refinement of Corollary 1.1, whose proof is immediate.

Corollary 3.1. *If $(n_1, n_2, \ldots, n_{\max})$ is the edge weight spectrum for minimal spanning trees in G, then for each k, we have*

$$n_k = \sum i([e]),$$

where the sum is taken over all equivalence classes of edges of weight k.

At this point we observe that although precisely $i([e])$ edges from $[e]$ must be included in any MST, we are not at liberty to choose these edges arbitrarily. Figure 2 provides an illustration of this constraint. The network G has five edges, all of weight 2, and all in the same class. Every MST must contain exactly three of these edges, but not every subset with three edges can be chosen.

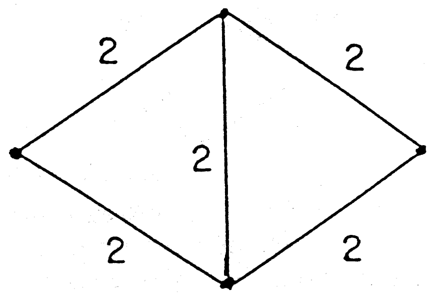

Figure 2. Not every subset with three edges is a choice.

For a given class $[e]$, we say that a subset $\{e_1, e_2, \ldots, e_r\}$ of $[e]$ is a *choice* from $[e]$ if there is some MST T such that $T \cap [e] = \{e_1, e_2, \ldots, e_r\}$. By the preceding discussion, $r = i([e])$. The next theorem is our main structural result. It shows that the various choices from $[e]$ are freely interchangeable.

Theorem 4. *Let $\{e_1, e_2, \ldots, e_r\}$ and $\{f_1, f_2, \ldots, f_r\}$ be two choices from $[e]$. Let T be a MST containing $\{e_1, e_2, \ldots, e_r\}$. Then*

$$(T \setminus \{e_1, e_2, \ldots, e_r\}) \cup \{f_1, f_2, \ldots, f_r\}$$

is also a MST.

The next lemma is the key to the proof of Theorem 4.

Lemma 4.1. *Let $T \to S \to R$ consist of the equal edge replacements (e, f) and (e', f'), respectively. If e and e' belong to different equivalence classes, then the replacements can be performed in reverse order, to obtain $T \to S' \to R$.*

Proof of Lemma 4.1.
Case 1. $w(e) = w(e') = k$. The first replacement (e, f) replaces e by f across the cycle $\{f\} \cup P(f, T)$, all of whose edges of weight k belong to $[e]$ (because each of them can be replaced by f). Likewise, the second replacement (e', f') replaces e' by f' across the cycle $\{f'\} \cup P(f', S)$, all of whose edges of weight k belong to $[e']$. Since $[e] \cap [e'] = \emptyset$, the intersection of these two cycles either has no edges or has edges only of weight $< k$. In

particular, $f' \notin \{f\} \cup P(f,T)$ and $f \notin \{f'\} \cup P(f',S)$. Thus, the replacements can be performed in reverse order (see Figure 3a).

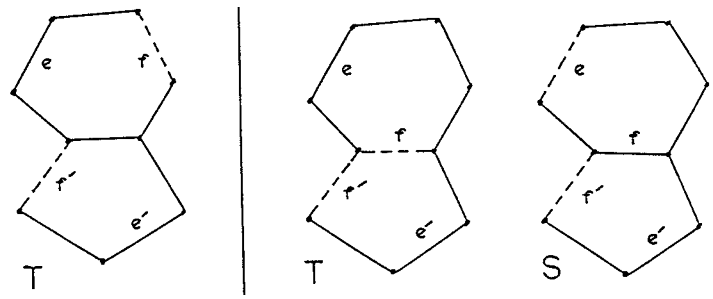

Figure 3. (a) In the tree T, the intersection of the two cycles consists of edges of lower weight only. (b) In T, the intersection of the cycles may contain f.

Case 2. $w(e) < w(e')$. The replacement (e,f) replaces e by f across the cycle $\{f\} \cup P(f,T)$, and neither e' nor f' belong to this cycle because of their greater weight (property (1)). The replacement (e',f') replaces e' by f' across the cycle $\{f'\} \cup P(f',S)$, which may contain f, but not e because $e \notin S$ and $e \neq f'$. If $P(f',S)$ does not contain f, the order of replacement can be reversed. If $P(f',S)$ contains f, $P(f',S) \cap P(e,S)$ is a simple arc containing f (a disconnected intersection would produce a cycle in S). Then

$$P(f',T) = P(f',S) \cup P(e,S) \cup \{e\} \setminus P(f',S) \cap P(e,S).$$

(See Figure 3b.) Also, $e' \in P(f',T)$ because $e' \in P(f',S)$, and $e' \notin P(e,S)$ by property (1). Thus we may perform the replacement (e',f') in T to obtain $S' = (T \setminus \{f'\}) \cup \{e'\}$.

Now $f \notin S'$ because $f \notin T$, but $e \in S'$ and e belongs to $P(f,S')$, because $P(f,S') = P(f,T)$. The replacement (e,f) can now be performed in S' to obtain R.

Case 3. $w(e') < w(e)$. Consider the inverse sequence $R \to S \to T$. Apply the argument of Case 2 to reverse the order of replacement, obtaining $R \to S' \to T$. Then invert the sequence once again. □

Proof of Theorem 4. Let T' be a MST containing $\{f_1, f_2, \ldots, f_r\}$. By Theorem 1 there is a sequence $T_0 \to T_1 \to \ldots \to T_n$ of equal edge replacements such that $T_0 = T'$ and $T_n = T$. By Lemma 4.1 we may assume that this sequence has been chosen so that the replacements involving edges of $[e]$ are done last.

Let T_m be the last tree in the sequence before the replacements involving the edges of $[e]$ are done. Then T_m agrees with T on all edges from all equivalence classes except $[e]$, and T_m contains the choice $\{f_1, f_2, \ldots, f_r\}$ from $[e]$. Thus, $T_m = (T \setminus \{e_1, e_2, \ldots, e_r\}) \cup \{f_1, f_2, \ldots, f_n\}$, which is a MST as asserted. □

For a given class $[e]$, define $c([e])$ to be the number of choices from $[e]$.

Corollary 4.1. *The number of minimal spanning trees in G is the product $\prod c([e])$, where the product is taken over all equivalence classes $[e]$ of electable edges in G.*

4. An algorithm. Theorem 4 provides a strategy for generating all the minimal spanning trees in a network. Since choices from different equivalence classes are independent of each other, it is sufficient to determine the choices from each class and then combine them in all possible ways.

Kruskal's algorithm [3] for finding a MST proceeds by choosing an unchosen edge of lowest weight in G that does not form a cycle with the chosen edges, and continuing until no more edges can be chosen. Kruskal's algorithm leaves the impression that the process of constructing minimal spanning trees is a sequential one. Theorem 4 indicates the opposite: MSTs are constructed in independent blocks.

We can now state a macroversion of Kruskal's algorithm as follows.

1) Choose any MST T. Use T to test all edges of G for Property $E(T)$, and hence for electability. Delete the nonelectable edges of G.
2) Determine the equivalence classes $[e]$ of electable edges. For each class $[e]$, determine $i([e])$ from inspection of T.
3) For each class $[e]$, determine the choices from $[e]$.
4) Arbitrarily combine these choices, one from each equivalence class, to produce all MSTs in G.

In order to make this into a workable algorithm, it is first necessary to elaborate on steps 2) and 3).

Regarding 2), more work must be done before we can determine the classes $[e]$ by working with the chosen tree T. We first define another equivalence relation $\sim_T$ that is defined in terms of T, and we then prove in Theorem 5 that $\sim_T$ is the same relation as $\sim$.

For a given MST T and for each electable edge g of weight k not in T, define $X(g)$ to be the set consisting of g together with the edges of weight k on $P(g, T)$. Then $X(g)$ contains g and all the edges that can be replaced by g in a single equal edge replacement in T.

Define an equivalence relation $\sim_T$ on the electable edges of G as follows: $e \sim_T f$ if and only if $e = f$ or e and f belong to the same transitivity class of the sets $X(g)$. The relation $\sim_T$ apparently depends on the choice of the tree T, but we now show that is not the case.

Theorem 5. *For every minimal spanning tree T, $\sim_T = \sim$.*

The significance of Theorem 5 is that the relation $\sim$, which is defined without reference to a specific tree, can be computed from any specific tree T. This means that step 2) of the algorithm can be executed with reference to the tree T chosen in step 1), as follows:

2′) For each edge $g \notin T$, list the $X(g)$ and compute their transitivity classes. These transitivity classes are equivalence classes under $\sim_T$, hence under $\sim$. If e is an edge of T that belongs to no set $X(g)$, then $[e] = \{e\}$, and e must belong to every MST.

The next lemma is needed to prove Theorem 5.

Lemma 5.1. *Let (a, b) be an equal edge replacement from T to S. Let e, f be edges of G. Then $e \sim_T f \Rightarrow e \sim_S f$. Consequently, the relation $\sim_T$ is the same for all minimal spanning trees.*

Proof. If $e \sim_T f$ and $e \neq f$, then there is a sequence $\{g_1, g_2, \ldots, g_m\}$ of edges not in T such that $e \in X(g_1)$, $f \in X(g_m)$ and $X(g_i) \cap X(g_{i+1}) \neq \emptyset$. Furthermore, $a \in T$ and $b \notin T$. Let $k = w(e) = w(f) = w(g_i)$.

We will use the notation $Y(g)$ for the set $\{g\} \cup P(g, S)$ in the tree S, in parallel with the notation $X(g)$ used for the tree T.

The proof proceeds by induction on m.

If $m = 1$, then both e and f belong to $X(g_1)$. Suppose first that $b = g_1$. Then $w(a) = w(b) = k$, and $a \in X(g_1)$. (The edge a may be either e or f.) Do the replacement (a, g_1) to obtain the tree S. Then $Y(a)$ consists of a together with the edges of weight k on $P(a, S)$, which is the same as $X(g_1)$, and both e and f belong to $Y(a)$. Thus, $e \sim_S f$.

Next, suppose that $b \neq g_1$ and $w(a) = w(b) = k$. If $a \notin X(g_1)$, then $Y(g_1) = X(g_1)$, and thus $e \sim_S f$. If $a \in X(g_1)$, consider the sets $Y(a)$ and $Y(g_1)$, whose intersection contains b. (See Figure 4.) All edges on $P(g_1, T)$ are now either on $P(g_1, S)$ or $\{a\} \cup P(a, S)$. In particular, e and f belong to $Y(g_1) \cup Y(a)$, and $b \in Y(g_1) \cap Y(a)$. By definition of $\sim_S$, $e \sim_S f$.

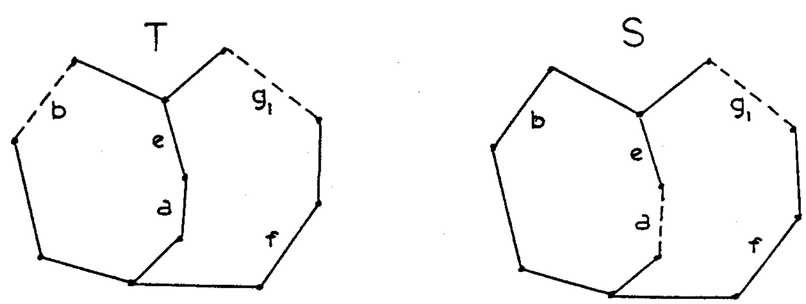

Figure 4. In the tree T, both e and f belong to $X(g_1)$. In S, e and b belong to $Y(a)$, and f and b belong to $Y(g_1)$.

Finally, suppose that $w(a) = w(b) \neq k$. If $w(a) = w(b) < k$, then $P(a, S)$ contains only edges of weight $\leq w(a)$, and hence e and f do not belong to $P(a, S)$, nor are they equal to a. Thus, e and f belong to $P(g_1, S)$ and hence to $Y(g_1)$, so $e \sim_S f$.

If $w(a) = w(b) > k$, then $Y(g_1) = X(g_1)$ and again $e \sim_S f$.

This establishes Lemma 5.1 for the case $m = 1$.

Assume inductively that the statement $e \sim_T f \Rightarrow e \sim_S f$ holds whenever the relation $e \sim_T f$ is established by a sequence of length less than m. For the chosen sequence $\{g_1, g_2, \ldots, g_m\}$, select $e' \in X(g_{m-1}) \cap X(g_m)$. Then $e \sim_T e'$, and by the inductive hypothesis, $e \sim_S e'$. It remains to show that $e' \sim_S f$. But $e' \sim_T f$, and both e' and f belong to a single $X(g)$, namely $X(g_m)$. By the proof for $m = 1$, we have $e' \sim_S f$. Thus $e \sim_S f$.

By induction, $e \sim_T f \Rightarrow e \sim_S f$ holds for any pair e, f of edges in G.

The converse $e \sim_S f \Rightarrow e \sim_T f$ is also true, because (b, a) is an equal edge replacement from S to T.

Finally, if T and S are arbitrary MSTs, there is a sequence of equal edge replacements from T to S. Consequently, $\sim_T = \sim_S$. This completes the proof of Lemma 5.1. □

Proof of Theorem 5. Given a minimal spanning tree T and edges e, f such that $e \sim_T f$, then if $e \neq f$ there exists a sequence $\{g_1, g_2, \ldots, g_m\}$ of edges not in T such that $e \in X(g_1)$, $f \in X(g_m)$ and $X(g_i) \cap X(g_{i+1}) \neq \emptyset$. For each i, $1 \leq i \leq m - 1$, choose an edge $x_i \in X(g_i) \cap X(g_{i+1})$. For each i, each element $g \in X(g_i)$ satisfies $g \sim g_i$, since if $g \neq g_i$, the replacement (g, g_i) can be done in T. Consequently, $e \sim x_1 \sim x_2 \sim \ldots \sim x_{n-1} \sim f$.

Conversely, suppose that $e \sim f$. Let (a, b) be an equal edge replacement in some MST S. Then $a \in X(b)$, hence $a \sim_S b$. Since $e \sim f$, there is a sequence $T_0 \to T_1 \to \ldots \to T_n$ of equal edge replacements transforming e into f. For each i, let e_i be the image of e under the composition $T_0 \to T_1 \to \ldots \to T_i$. Then $e \sim_{T_0} e_1 \sim_{T_1} e_2 \sim_{T_2} \cdots \sim_{T_{n-1}} e_n = f$. By Lemma 5.1, all these relations are equal to $\sim_T$, so $e \sim_T f$.

Therefore, $\sim_T = \sim$, and Theorem 5 is proved. □

There are two different routes to the execution of step 3) in the algorithm. First, consider the class $[e]$, and let $k = w(e)$. Let H be the subgraph of T consisting of the edges of weight $\leq k - 1$. For each subset of $[e]$ with $i([e])$ edges, determine whether this subset forms a cycle with the edges of H. If not, this subset is a choice. This route shares the spirit of Kruskal's algorithm, since it tests for cycles at each stage.

The second route begins with the whole tree T and removes the edges belonging to the class $[e]$ to obtain a subgraph H'. For each subset of $[e]$ with $i([e])$ edges, form the union with H' and test for connectedness. If connected, this subset is a choice. This route may be the more economical, since testing for connectedness is easier than testing for cycle formation.

5. An example. The following simple example illustrates the workings of the algorithm. In the network G shown in Figure 5, the edges are given a letter name and a weight. We arbitrarily select the tree T as shown and observe that the edges e, f and m are not electable, deleting them from the graph. The edges c, h and ℓ are electable and are indicated by the dotted lines.

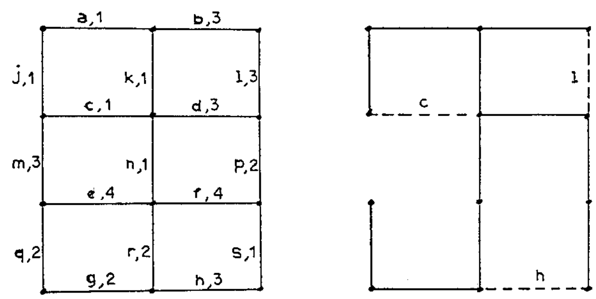

Figure 5. The MST T and other electable edges.

We compute the sets

$$X(c) = \{c, a, j, k\}$$
$$X(h) = \{h, d\}$$
$$X(\ell) = \{\ell, b, d\}$$

to obtain the equivalence classes

$[n] = \{n\}$	
$[s] = \{s\}$	
$[a] = \{a, c, j, k\}$	$w = 1$
$[p] = \{p\}$	
$[q] = \{q\}$	
$[g] = \{g\}$	
$[r] = \{r\}$	$w = 2$
$[b] = \{b, d, h, \ell\}$	$w = 3$

For each of the singleton classes, the inclusion number i and the choice number c are both equal to 1.

In $[a]$, we have, by inspection of T, that $i([a]) = 3$. Any three edges form a choice, so $c([a]) = \binom{4}{3} = 4$.

In $[b]$, we have $i([b]) = 2$. There are $\binom{4}{2} = 6$ subsets with 2 elements, but $\{d, h\}$ is not a choice. Thus $c([b]) = 5$.

Observe that if we use the first method for determining the choices from $[b]$, the subset $\{d, h\}$ is eliminated because it creates a cycle with edges of lower weight in T. In the second method, with $H' = T \setminus \{b, d\}$, the subset $\{d, h\}$ is eliminated because $H' \cup \{d, h\}$ is not connected (the vertex at the upper right corner is isolated).

Finally, by Corollary 4.1, the number of MSTs in G is

$$\prod c([e]) = (1)(1)(4)(1)(1)(1)(1)(5) = 20.$$

References

[1] Gabow, Harold, N. Two algorithms for generating weighted trees in order. *SIAM J. Comp.* **6** (1977), 139–150.

[2] Graham, Ronald L. and Hell, Pavol. On the history of the minimum spanning tree problem. *Annals of the History of Computing* **1** (1985), 43–57.

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

[4] Tucker, Alan. *Applied Combinatorics*, 2nd ed., John Wiley, 1984.