Packing spanning trees and the k-tree protocol

Robert F. Bailey[a,*], Brett Stevens[b]

[a] *Department of Mathematics and Statistics, University of Regina, 3737 Wascana Parkway, Regina, Saskatchewan S4S 0A2, Canada.*
[b] *School of Mathematics and Statistics, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario K1S 5B6, Canada.*

Abstract

We provide a structural description of, and invariants for, *maximum spanning tree-packable graphs*, i.e. those graphs G for which the edge connectivity of G is equal to the maximum number of edge-disjoint spanning trees in G. These graphs are of interest for the *k-tree protocol* of Itai and Rodeh [*Inform. and Comput.* **79** (1988), 43–59].

Keywords: Spanning tree, edge connectivity, spanning tree packing number, network reliability
2010 MSC: 05C05 (primary), 68M10, 90B25 (secondary)

---

1. Introduction

In [7], Itai and Rodeh proposed a communication protocol, called the *k-tree protocol*, which allows all nodes of a network to communicate through a distinguished root node v, even when some set of $k-1$ or fewer edges are removed from the network. The protocol requires the graph G modelling the network to have two properties. First, the graph G must remain connected when any $k-1$ edges are removed, so k can be at most the *edge connectivity* of G. Second, it requires a collection of k spanning trees for G, $\{T_1, \ldots, T_k\}$, with the following property (which they called the *k-tree condition for edges*): for all vertices w distinct from v, and for any i, j where $1 \leq i < j \leq k$, the paths in T_i and T_j from v to w are internally disjoint.

---

[*] Corresponding author

Email addresses: `robert.bailey@uregina.ca` (Robert F. Bailey), `brett@math.carleton.ca` (Brett Stevens)

Itai and Rodeh conjectured that any k-edge connected graph satisfies the k-tree condition for edges, but they were only able to prove it when $k = 2$. However, if it happens that the graph has k edge-disjoint spanning trees, and so k must be at most the *spanning tree packing number* of G, then it clearly satisfies the k-tree condition. (This is not a requirement: for example, an n-cycle satisfies the 2-tree condition, but does not have two edge-disjoint spanning trees.) Consequently, the protocol can be applied if the two parameters (edge connectivity and spanning tree packing number) coincide. The purpose of this article is to investigate these graphs.

Let $G = (V, E)$ be a graph. Throughout this paper, we shall assume that graphs are connected, and we allow for the possibility of multiple edges.

An *edge cut* in G is a partition (V_1, V_2) of the vertex set of G into two nonempty subsets. In other words, an edge cut is a set where the removal of the edges between V_1 and V_2 disconnects G; if the number of such edges is k, we call it a *k-edge cut*. Therefore the edge-connectivity of G is the least value of k for which there exists a k-edge cut in G. We note that sometimes we will refer to a k-edge cut by the set of edges whose removal disconnects the graph, rather than the partition of V. Also, we say that G is *k-edge connected* if $\lambda(G) \geq k$.

The *spanning tree packing number* of G, denoted $\sigma(G)$, is the maximum number of edge-disjoint spanning trees in G. (We usually shorten this to *STP number*.) A survey of results on STP numbers can be found in Palmer [9]. In particular, graphs with given STP number were characterized independently by Nash-Williams [8] and Tutte [11], both in 1961.

Theorem 1 (Nash-Williams; Tutte)**.** *A connected graph G has at least k edge-disjoint spanning trees if and only if, for every partition of $V(G)$ into r parts, there are at least $k(r-1)$ edges between the parts.*

It is a straightforward observation that $\sigma(G) \leq \lambda(G)$: clearly, to disconnect G we must remove at least one edge from each of the $\sigma(G)$ disjoint spanning trees (and possibly some other edges as well). Also, in 1983 Gusfield [6] showed that it follows from Nash-Williams and Tutte's result that $\lambda(G) \leq 2\sigma(G)$ (see also Diestel [4, Section 3.5]).

Definition 2. A graph G is said to be *maximum spanning tree-packable*, or *max-STP* for short, if $\lambda(G) = \sigma(G)$, i.e. the edge connectivity is equal to the spanning tree packing number.

Given that both the edge-connectivity and STP number of a graph can be computed in polynomial time (see Schrijver [10, Sections 15.3 and 51.4]), it follows that determining if a graph is a max-STP graph can also be computed in polynomial time. Furthermore, it is actually possible to find the collection of $\sigma(G)$ edge-disjoint spanning trees in polynomial time; the best algorithm known for doing this is due to Gabow and Westermann [5].

Finally, some notation: if X is a subset of vertices of G, then $G[X]$ denotes the induced subgraph on X.

2. The main result

In what follows, we will obtain our main result: a structural characterization of the max-STP graphs. To do so requires us to define the following "joining" operation.

Definition 3. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be connected graphs, where $V_1 \cap V_2 = \varnothing$, and let K be a set of k edges with one end in V_1 and one end in V_2 (for some integer k). Then the *K-join* of G_1 and G_2, denoted by $G_1 *_K G_2$, is the graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2 \cup K$.

When the set K of k edges is not specified (or is not important), we speak of a *k-join* of G_1 and G_2, and denote it by $G_1 *_k G_2$.

We follow the definition with a couple of remarks. First, by construction (V_1, V_2) is a k-edge cut of $G_1 *_K G_2$, so consequently the edge-connectivity of the K-join is at most k. Second, two k-joins will not, in general, be isomorphic (unless we have a special case, such as when $k = 1$ and both G_1 and G_2 are vertex-transitive). Consider the following example:

Example 4. There are three non-isomorphic possibilities for $K_4 *_2 K_4$ (the 2-join of two copies of K_4), as shown in Figure 1.

We are interested in k-joins because they preserve the property we are concerned with, as shown in the following lemma.

Lemma 5. *Suppose we have graphs G_1 and G_2, a edge-set K such that $k = |K| \leq \sigma(G_i) \leq \lambda(G_i)$ for $i = 1, 2$, and let $G = G_1 *_K G_2$. Then $\lambda(G) = \sigma(G) = k$.*

Proof. Suppose G_1 has k edge-disjoint spanning trees $T_1, \ldots, T_k$, that G_2 has k edge-disjoint spanning trees $S_1, \ldots, S_k$, and that the $K = \{e_1, \ldots, e_k\}$. Then we can construct a set $\mathcal{T}$ of k edge-disjoint spanning trees for G, where

$$\mathcal{T} = \{T_i \cup \{e_i\} \cup S_i \mid i = 1, \ldots, k\}.$$

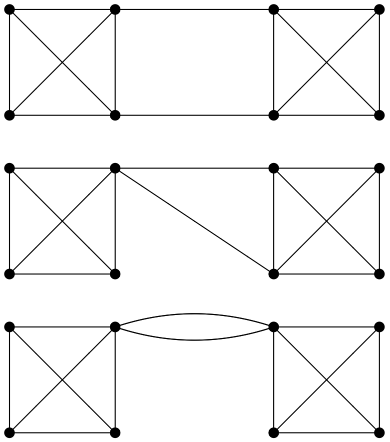


Figure 1: The three non-isomorphic possibilities for $K_4 *_2 K_4$.

Also, because any spanning tree for G must contain at least one of $e_1, \dots, e_k$, there can be no more. Hence $\sigma(G) = k$, and so also $\lambda(G) \geq k$.

Now, K is a k-edge cut in G, so therefore $\lambda(G) \leq k$. Combining this with the above shows that $\lambda(G) = k$. □

In fact, we can see that if $\lambda(G_1)$ and $\lambda(G_2)$ are both *at least* k, then by construction $\lambda(G_1 *_k G_2) = k$. This is also true if one or other of G_1 and G_2 consists of a single vertex.

The next result is a kind of converse to Lemma 5 above.

Lemma 6. *Suppose $\sigma(G) = \lambda(G) = k$. Then there exist graphs G_1 and G_2, and a set K of k edges, such that $G = G_1 *_K G_2$, and where G_1, G_2 each satisfy exactly one of the following:*

(i) *G_i has one vertex and no edges;*

(ii) $k \leq \sigma(G_i) < \lambda(G_i)$*;*

(iii) $k < \sigma(G_i) = \lambda(G_i)$*;*

(iv) $\sigma(G_i) = \lambda(G_i) = k$*.*

Proof. The graphs G_1 and G_2 are those obtained by removing a k-edge-cut from G. That there are k edge-disjoint spanning trees and a k-edge cut in G ensures that each of G_1 and G_2 inherit naturally a set of k edge-disjoint spanning trees from G. Thus the four classes arise by enumerating all possibilities for $\lambda(G_i)$ and $\sigma(G_i)$. □

We remark that these four classes (i)–(iv) partition the class of all graphs with k or more edge-disjoint spanning trees (i.e. the class determined by Nash-Williams and Tutte).

Definition 7. We call a graph *k-irreducible* if it belongs to classes (i)–(iii) of Lemma 6; we call it *k-reducible* if it belongs to class (iv).

We remark that a graph G in class (iii) will itself be a max-STP graph, but with a higher spanning tree packing number and edge-connectivity. Such a graph will also be k'-reducible, where $k' = \sigma(G) = \lambda(G) > k$.

A complication arises when two or more k-joins are made. Suppose we make a k-join $H = G_1 *_k G_2$, and then make $H *_k G_3$ (where $G_i = (V_i, E_i)$). If the k edges in the second k-join are all attached to exactly one of G_1 or G_2 (assume without loss of generality that this is G_2), then in the resulting graph $(V_1, V_2 \cup V_3)$ and $(V_1 \cup V_2, V_3)$ will both be k-edge cuts, obtained by removing the edges of the first or second k-joins respectively. However, if the second k-join attaches G_3 to some vertices in each of G_1 and G_2, only $(V_1 \cup V_2, V_3)$ is a k-edge cut. This phenomenon is demonstrated in Example 8 below.

Example 8. Figure 2 shows two ways of forming 2-joins of three copies of K_4. The graph on the left has two 2-edge cuts, while the graph on the right has only one.

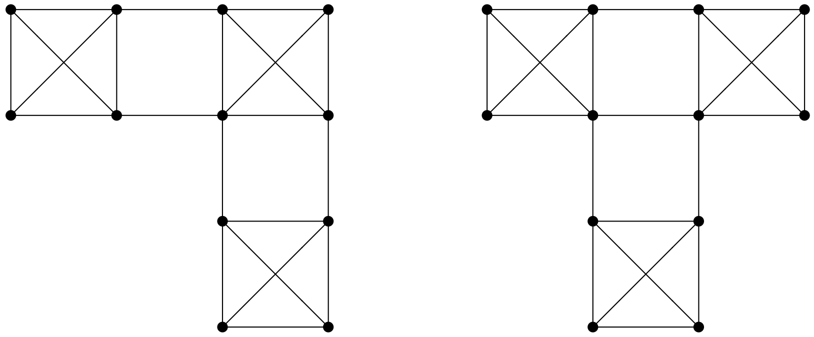


Figure 2: Two ways of forming 2-joins of three copies of K_4.

We call a sequence of k-joins *order-independent* if the joining edges of any one of them yield a k-edge cut in the resulting graph. Thus in Figure 2, the 2-joins in the graph on the left are order-independent, while those in the graph on the right are not.

The term order-independent refers to the fact that the final graph is invariant of different choices of which k-join is performed first, second, third, etc. However, which "pieces" are joined by a particular k-join remain fixed. Thus order-independence is more akin to an associative law than a commutative law.

Of course, an arbitrary k-edge-connected graph may have more than one k-edge-cut. We need to examine the possibilities for when this happens. Suppose G has two k-edge cuts (A_1, A_2) and (B_1, B_2). Label the intersections $A_1 \cap B_1 = X$, $A_2 \cap B_1 = Y$, $A_1 \cap B_2 = Z$ and $A_2 \cap B_2 = W$, as shown in Figure 3.

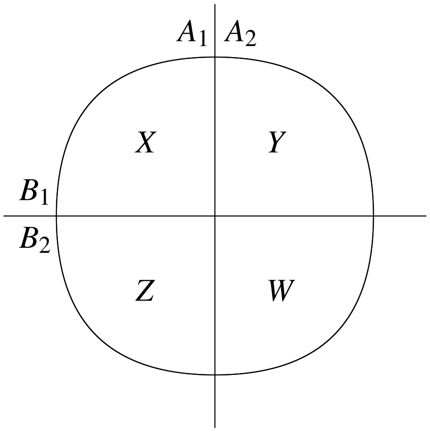


Figure 3: A graph with two edge-cuts (A_1, A_2) and (B_1, B_2).

Now, at most one of these intersections can be empty; otherwise, either one of the A_i or B_j will be empty, or the two cuts are in fact the same. Consequently, there are three possible situations.

(1) None of the intersections are empty.

Otherwise, suppose without loss of generality that $W = A_2 \cap B_2 = \varnothing$. Then we have either:

(2) In G, there is at least one edge between each pair of subsets chosen from X, Y and Z.

(3) There exists a pair of subsets chosen from X, Y and Z (without loss of generality, we assume these are Y and Z) which has no edges between them in G.

It is easy to see that there cannot be more than one such pair with no edges, as otherwise G would be disconnected.

We make the following definition to help distinguish some of these possibilities.

Definition 9. Let G be a graph with k-edge cuts (A_1, A_2) and (B_1, B_2). We say that the two cuts are *overlapping* if situation (1) or (2) above applies.

The next lemma analyses situation (1) in more detail.

Lemma 10. *Suppose $G = (V, E)$ has edge connectivity $\lambda(G) = k$, and that (A_1, A_2) and (B_1, B_2) are two k-edge-cuts of G, such that $A_i \cap B_j \neq \varnothing$ for all i and j. Then $(A_1 \cap B_1, A_2 \cup B_2)$ is also a k-edge-cut of G.*

Proof. Suppose the intersections $A_i \cap B_j$ are labelled X, Y, Z, W as in Figure 3. We want to show that $(X, Y \dot\cup Z \dot\cup W)$ is a k-edge-cut.

An important observation is that since G is assumed to be k-edge-connected, there do not exist any smaller edge-cuts. $(\star)$

We define the non-negative integer n_{xy} to be the number of edges from X to Y, and define n_{xz}, n_{xw}, n_{yz}, n_{yw} and n_{zw} in a similar fashion. By the assumption that $(X \dot\cup Y, Z \dot\cup W)$ and $(X \dot\cup Z, Y \dot\cup W)$ are k-edge-cuts, we know that

$$n_{xz} + n_{xw} + n_{yz} + n_{yw} = n_{xy} + n_{xw} + n_{yz} + n_{zw} = k. \qquad (\star\star)$$

Let $S_X = n_{xy} + n_{xz} + n_{xw}$, i.e. the total number of edges leaving X, and define S_Y, S_Z, S_W similarly. Now, it suffices for us to prove that $S_X \leq k$, as by $(\star)$ it would then be equal to k.

Now, by rearranging part of $(\star\star)$ we have that $n_{xz} + n_{xw} = k - n_{yz} - n_{yw}$, so by adding n_{xy} to both sides we obtain

$$\begin{aligned} S_X = n_{xy} + n_{xz} + n_{xw} &= (k - n_{yz} - n_{yw}) + n_{xy} \\ &= (k - n_{yz} - n_{yw}) + (k - n_{xw} - n_{yz} - n_{zw}) \\ &= 2k - (2n_{yz} + n_{xw} + n_{yw} + n_{zw}) \\ &= 2k - N, \end{aligned}$$

and $S_X = 2k - N \leq k$ if and only if $N \geq k$.

Now, since $(X \dot\cup Y \dot\cup Z, W)$ is an edge-cut of G, we must have that $S_W = n_{xw} + n_{yw} + n_{zw} \geq k$, as otherwise we would violate $(\star)$. Hence $N = 2n_{yz} + S_W \geq k$, and we are done. $\square$

We now make some remarks about the above proof. By the symmetry of the argument, it follows that we must have $S_X = S_Y = S_Z = S_W = N = k$. Consequently $n_{xw} = n_{yz} = 0$ (i.e. there can be no "diagonal" edges in Figure 3). Also, it follows that $n_{xy} = n_{xz} = n_{yw} = n_{zw} = \frac{1}{2}k$ (and, thus that k must be even). The next lemma considers what happens if we add the assumption that G is a max-STP graph.

Lemma 11. *Suppose G is a max-STP graph, with $\lambda(G) = \sigma(G) = k$. Then if G contains two k-edge-cuts, they cannot be overlapping.*

Proof. Suppose that G contains two k-edge-cuts, (A_1, A_2) and (B_1, B_2). Now, as we observed earlier, at most one of the intersections $A_i \cap B_j$ may be empty.

First, we suppose that none of these intersections are empty, and that they are labelled X, Y, Z and W as in Figure 3. Let H_1 be the induced subgraph on X and H_2 be the induced subgraph on $Y \cup Z \cup W$. By construction, we have that $G = H_1 *_k H_2$. Since $\sigma(G) = k$, we must have that $\sigma(H_1), \sigma(H_2) \geq k$. However, since $n_{yz} = 0$, H_2 must contain two edge-cuts of size $n_{yw} = n_{zw} = \frac{1}{2}k$, and thus $\sigma(H_2) \leq \frac{1}{2}k$, which is impossible.

So therefore we must have an empty intersection, and without loss of generality we suppose it is $W = A_2 \cap B_2$. As before, let n_{xy} denote the number of edges between X and Y, with n_{yz} and n_{xz} defined similarly. This time our two k-edge cuts show that $n_{xy} + n_{yz} = n_{xz} + n_{yz} = k$ (†). Let H denote the induced subgraph $G[X \cup Y]$. Since $(B_1, B_2) = (X \cup Y, Z)$ is a k-edge cut of G, we have that $G = H *_k G[Z]$, so by Lemma 6 it follows that $\lambda(H) \geq \sigma(H) \geq k$. Now, (X, Y) is an edge-cut of H of size $n_{xy} \leq k$, so therefore $n_{xy} = k$. Substituting this into (†) above, we find that $n_{yz} = 0$, so there are no edges between Y and Z.

Thus the only possible way for a max-STP graph G to contain two k-edge cuts is if they are not overlapping. □

The key observation here is that since the k-join operation preserves max-STP graphs (thanks to Lemma 5), if we have made two k-joins order-independently, then the two k-edge cuts arising from these must be non-overlapping. Following this, we are now able to state our main theorem.

Theorem 12. *Suppose G is a max-STP graph satisfying $\lambda(G) = \sigma(G) = k$. Then we have the following.*

(i) There exists a unique set $\mathcal{A}$ of k-irreducible graphs $G_1, \ldots, G_m$ (for some m).

(ii) *There exists a unique rooted tree R with m leaves labelled by $G_1, \ldots, G_m$, such that the root is labelled by G and each node is labelled by an order-independent k-join of its children.*

(iii) *For each non-leaf, labelled by H and its d children labelled $H_1, \ldots, H_d$, there exists a unique tree T_H with vertices $\{1, \ldots, d\}$ labelled by $H_1, \ldots, H_d$. Then for each edge $e = ij$ of T_H, there exists a k-edge cut K_e of H such that $H_i *_{K_e} H_j$ is an induced subgraph of H.*

(This shows that if $\lambda(G) = \sigma(G) = k$, G must be obtained by an iterated k-join of k-irreducible graphs.)

Proof. We start with G and build the rooted tree R and trees T_H recursively.

Suppose a graph Γ is the label of a node which has yet to be considered. If Γ is k-irreducible, this node will be a leaf in R, and we add Γ to $\mathcal{A}$.

If Γ is k-reducible, then $\lambda(\Gamma) = \sigma(\Gamma) = k$, and so Γ contains some collection of k-edge cuts. By Lemma 11, these must be pairwise non-overlapping. Removing the edges from all of these k-edge cuts yields a graph with some number $d \geq 2$ connected components; label these $\Gamma_1, \ldots, \Gamma_d$. By Lemma 6, each Γ_i must be k-edge connected and contains at least k edge-disjoint spanning trees.

We note that Γ is therefore an order-independent k-join of $\Gamma_1, \ldots, \Gamma_d$, and we can build a tree to specify explicitly which pairs are joined, as follows. Define T_Γ to be the graph obtained from Γ by contracting each Γ_i to a single vertex (and removing any multiple edges). Lemma 11 ensures that this graph is a tree. Now, each edge e of T_Γ corresponds to exactly one of the k-edge cuts of Γ, so we can label these k-edge cuts by the edges of T_Γ.

Finally, we add a child node of Γ to R labelled by Γ_i for each $i \in \{1, \ldots, d\}$, and apply the recursion to each of these new nodes. $\square$

Definition 13. For a given max-STP graph G, the *ingredients* of G are the triple $I(G) = (\mathcal{A}, R, \{T_H\})$.

Example 14. Consider the max-STP graph G shown in Figure 4, which has $\lambda(G) = \sigma(G) = 2$. Now, G possesses exactly one 2-edge cut, so the root vertex of R, labelled by G, has two descendents; the tree T_G associated with it is the unique tree on two vertices. Now, one of the child nodes is labelled by a copy of K_4, which is 2-irreducible, so the node becomes a leaf. The other child node is labelled by a graph H which is the order-independent 2-join of three copies of K_4. Thus this node has three child nodes, all leaves labelled by a copy of K_4, and the associated tree T_H is the unique tree on 3 vertices.

Thus the ingredients of G are $(\mathcal{A}, R, \mathcal{T})$, where $\mathcal{A}$ contains of four copies of K_4, R is as shown in Figure 4, and $\mathcal{T} = \{$ •—•, •—•—• $\}$.

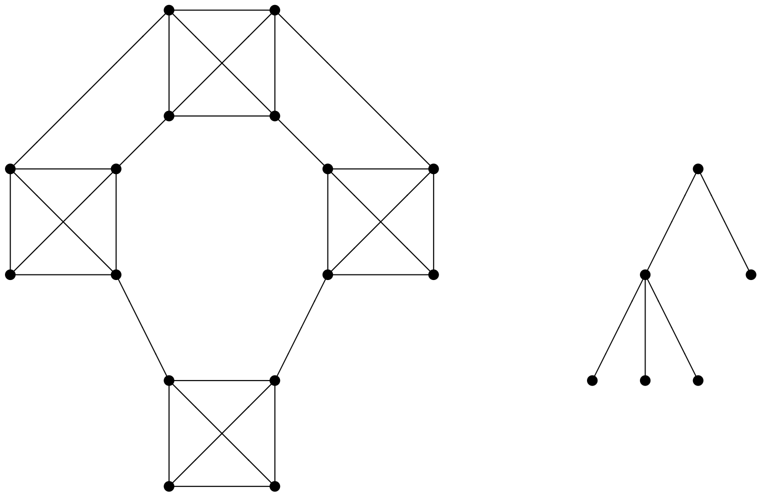


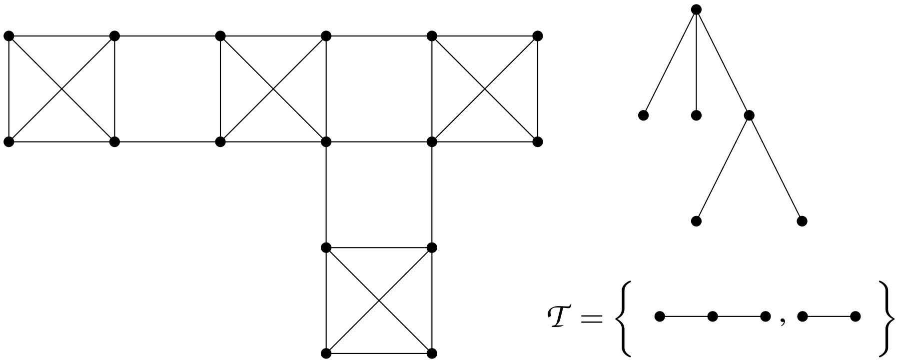
Figure 4: A max-STP graph G with $\lambda(G) = \sigma(G) = 2$, and the associated rooted tree R.

We should remark on the limitations of our result, and make the following observations about it. First, if the ingredients are different, then the graphs they arose from must also have been different; that is, if $I(G) \neq I(G')$, then $G \neq G'$. Second, if the ingredients are the same, that is $I(G) = I(G')$, then because of the non-uniqueness of the k-joining operation both $G = G'$ and $G \neq G'$ are possible. Our remaining remarks requires us to define two classes of graphs.

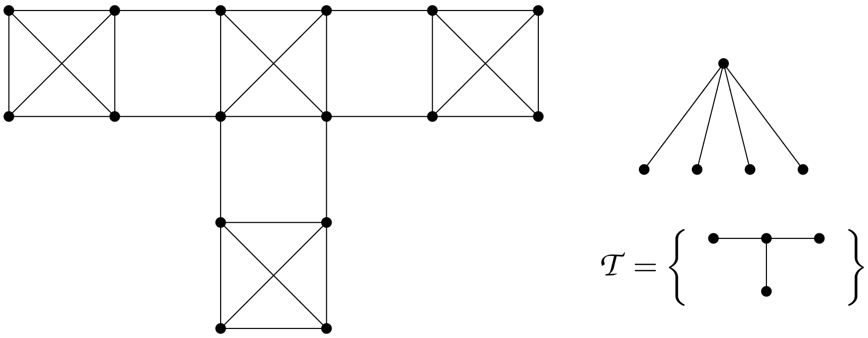
Definition 15. Let $\mathcal{A}$ be a set of k-irreducible graphs $\mathcal{A} = \{G_1, \ldots, G_m\}$, R be a rooted tree with m leaves, and $\mathcal{T} = \{T_H\}$ be an appropriate set of trees. Then the *deconstruction class* $\mathcal{D}(\mathcal{A}, R, \mathcal{T})$ is the class of graphs G whose ingredients are $(\mathcal{A}, R, \mathcal{T})$, and the *construction class* $\mathcal{C}(\mathcal{A}, R, \mathcal{T})$ is the class of all graphs G obtained by making sequences of order-independent k-joins of $G_1, \ldots, G_m$ according to the trees in $\mathcal{T}$ and the rooted tree R.

It is clear that $\mathcal{D}(\mathcal{A}, R, \mathcal{T}) \subseteq \mathcal{C}(\mathcal{A}, R, \mathcal{T})$. However, in general the two classes are not equal; that is, if $G \in \mathcal{C}(\mathcal{A}, R, \mathcal{T})$, it is not necessarily true that $I(G) = (\mathcal{A}, R, \mathcal{T})$. Example 16 below demonstrates the possibilities for what can happen.

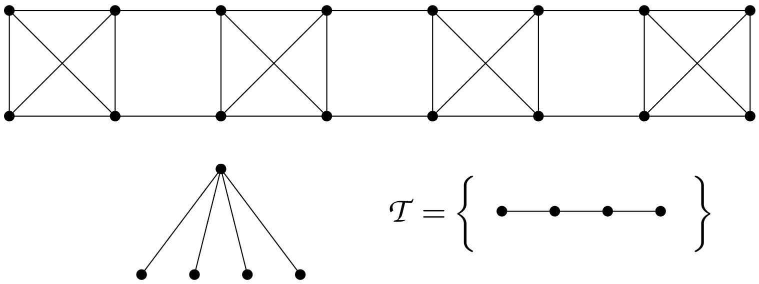
Example 16. The graphs in Figure 5 are all members of the class $\mathcal{C}(\mathcal{A}, R, \mathcal{T})$, where $(\mathcal{A}, R, \mathcal{T})$ are the ingredients of the graph in Example 14. However, they each have ingredients which differ from those in that example.



(a) Leaving some previous 2-joins order-independent of the new 2-join



(b) Leaving *all* previous 2-joins order-independent of the new 2-join



(c) Same as (b), but with a different set $\mathcal{T}$

Figure 5: Three members of the same construction class, but with different ingredients.

Furthermore, we remark that all deconstruction classes are disjoint, while two construction classes are not necessarily disjoint.

3. Conclusion

We conclude the paper with a couple of remarks about related work. First, both parameters of interest in this paper have natural generalizations from graphs to matroids: the STP number to the *base packing number*, and the edge connectivity to the *cogirth*. One can ask for a description of matroids for which these two parameters are equal, and this is investigated by Newman and the first author in [2]. The structural description obtained is similar to Theorem 12; however, they do not coincide exactly.

Second, we remark that any collection $T_1, \ldots, T_k$ of edge-disjoint spanning trees for a max-STP graph G forms an *uncovering-by-bases* (or UBB) for G (i.e. a collection of spanning trees for which the deletion of any $t \leq k-1$ edges of G leaves one of the trees intact). These are discussed by the authors in [3], where it is explained that such a collection is optimal in two ways: (i) the spanning trees are disjoint, so the UBB is as small as possible; and (ii) the number of edges which can be "uncovered" by the collection is as large as possible. (In fact, it was the study of UBBs which led the authors to the results in the present paper.) This notion also generalizes to matroids (see [1, Section 7]).

Acknowledgements

The authors wish to thank NSERC and the Ontario Ministry of Research and Innovation for their financial support. The first author is a PIMS Postdoctoral Fellow.

References

[1] R. F. Bailey, Uncoverings-by-bases for base-transitive permutation groups, *Des. Codes Cryptogr.* **41** (2006), 153–176.

[2] R. F. Bailey and M. W. Newman, On packing bases in matroids, in preparation.

[3] R. F. Bailey and B. Stevens, Uncoverings on graphs and network reliability, in preparation.

[4] R. Diestel, *Graph Theory* (2nd edition), Graduate Texts in Mathematics (173), Springer–Verlag, Berlin, 2000.

[5] H. N. Gabow and H. H. Westermann, Forests, frames, and games: algorithms for matroid sums and applications, *Algorithmica* **7** (1992), 465–497.

[6] D. Gusfield, Connectivity and edge-disjoint spanning trees, *Inform. Process. Lett.* **16** (1983), 87–89.

[7] A. Itai and M. Rodeh, The multi-tree approach to reliability in distributed networks, *Inform. and Comput.* **79** (1988), 43–59.

[8] C. St. J. A. Nash-Williams, Edge-disjoint spanning trees of finite graphs, *J. London Math. Soc.* **36** (1961), 445–450.

[9] E. M. Palmer, On the spanning tree packing number of a graph: a survey, *Discrete Math.* **230** (2001), 13–21.

[10] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer–Verlag, Berlin, 2003.

[11] W. T. Tutte, On the problem of decomposing a graph into n connected factors, *J. London Math. Soc.* **36** (1961), 221–230.