ELSEVIER

Discrete Mathematics 169 (1997) 83–94

DISCRETE MATHEMATICS

Dominating sets whose closed stars form spanning trees

Jerrold W. Grossman*

Department of Mathematical Sciences, Oakland University, Rochester, MI 48309-4401, USA

Received 16 November 1994; revised 19 May 1995

Abstract

For a subset W of vertices of an undirected graph G, let $S(W)$ be the subgraph consisting of W, all edges incident to at least one vertex in W, and all vertices adjacent to at least one vertex in W. If there exists a W such that $S(W)$ is a tree containing all the vertices of G, then $S(W)$ is a *spanning star tree* of G. These and associated notions are related to connected and/or acyclic dominating sets and also arise in the study of A-trails in Eulerian plane graphs. Among the results in this paper are a characterization of those values of n and m for which there exists a connected graph with n vertices and m edges that has no spanning star tree, and a proof that finding spanning star trees is in general NP-hard.

AMS subject classification: Primary: 05C35; secondary: 05C05, 05C45, 05C85, 68R10, 90B12

Keywords: A-trail; Dominating set; Eulerian plane graph; NP-complete problem; Spanning tree; Weakly connected

1. Introduction

In this paper we introduce a new variation on domination in graphs. The motivation for this research grew not from the wealth of results on dominating sets — see, for example, special Volume 86 of *Discrete Mathematics*, also reprinted in book form [9], or the latest domination bibliography [8], with over 800 papers — but rather from an intriguing problem on Eulerian tours in planar graphs. We explain this connection at the end of the introduction.

All graphs $G = (V,E)$ in this paper are finite and contain no loops or parallel edges. We denote $|V|$ and $|E|$ by n and m, respectively. The problems we consider come in two versions — ordinary and bipartite. In the bipartite case, V is partitioned

* E-mail: grossman@oakland.edu.

SSDI 0012-365X(95)00334-7

into nonempty subsets R and B of red and blue vertices, respectively, which play nonsymmetrical roles. If W is a subset of V (a subset of R in the bipartite case), then by $S(W)$, the subgraph *weakly induced* by W, we mean the subgraph consisting of W, all edges incident to at least one vertex in W, and all vertices adjacent to at least one vertex in W. In other words, $S(W)$ is the union of the closed stars at the vertices in W, but is in general not the graph induced by the closed neighborhoods of vertices in W (because some edges joining two neighbors of a vertex in W may fail to be present). In the ordinary situation, we want $S(W)$ to be a spanning subgraph — that is, to contain all the vertices in G. In other words, we require W to be a dominating set for G. In the bipartite situation, we require that $S(W)$ contain all the blue vertices.

Furthermore, we want to impose one or both of two additional requirements: that $S(W)$ be connected, so that we are not only dominating all the [blue] vertices of the graph but also connecting them together in some sense; and/or that $S(W)$ be acyclic, so that we are not only dominating but also avoiding redundant connections in some sense. (Note that $S(W)$ is acyclic if and only if every cycle in G contains two adjacent vertices not in W.) If the connectivity condition holds, then we say that W is a *weakly connected dominating set* and that $S(W)$ is a *spanning star connector*, or SSC. (This is in general much weaker than the widely studied notion of a *connected dominating set*, in which the subgraph induced by W must be connected [14] — our condition is merely that the subgraph induced by W in the square of G be connected. Nor is an SSC the same as a set-dominating set [13] or a strong or weak connecting set [12].) If the acyclicity condition holds, then we say that W is a *strongly acyclic dominating set* and that $S(W)$ is a *spanning star forest*, or SSF. (Note that we are requiring more than that the subgraph induced by W be acyclic. Also note that we are not using the term 'star forest' in the conventional sense to mean a forest consisting of stars.) Finally, if both conditions hold, then we call $S(W)$ a *spanning star tree*, or SST. When we wish to emphasize that only the blue vertices need to be spanned in the bipartite case, we use the abbreviations BSSC, BSSF, and BSST (the letter 'B' standing simultaneously for 'blue' and 'bipartite'). In all cases we say that $S(W)$ *uses* the vertices in W.

Much study has been focused on the cardinality of a smallest dominating set for G — the domination number $\gamma(G)$. Similarly, we are interested in the minimum value of $|W|$, if any, such that $S(W)$ is an SSC, SSF, or SST. Since subscripted gammas already have other, standard, interpretations in the domination literature, we will use $s_c(G)$ (the *spanning star connector number* or the *weakly connected domination number*), $s_f(G)$ (the *spanning star forest number* or *strongly acyclic domination number*), and $s_t(G)$ (the *spanning star tree number* or *weakly connected strongly acyclic domination number*), respectively, for these parameters. Clearly, $s_c(G)$ exists if and only if G is connected, but we will soon see that $s_f(G)$ and $s_t(G)$ may fail to exist even for connected G. We denote the bipartite versions of these parameters by s_c^B, s_f^B, and s_t^B.

The primary focus in this paper is on spanning star trees. Let us call a connected graph *good* for our present purposes if it has a spanning star tree and *bad* if it does not.

One can imagine the following application for SSTs. Suppose that a floor of a building is viewed as a graph: the edges are the straight hallways, and the vertices are the locations where hallways come together (or bend). We wish to station guards at some of the vertices so that these guards can monitor all of the vertices. The connectivity condition ensures that the guards can communicate by shining their flashlights down the hallways they guard, and the acyclicity condition ensures that the communication paths are unique. A good graph is one that can be effectively guarded under these conditions.

The remaining two sections of the paper deal with the following questions. It is clear that if a connected graph on a fixed number of vertices has so few edges that it is just a tree, or so many edges that it has a spanning star, then it has an SST. In Section 2 we show that this is almost the strongest statement one can make, in the sense that there exist bad graphs if m is neither too small nor too large, in both the ordinary and the bipartite situation. (In a separate paper [6] we study the existence of good and bad regular graphs.) In fact almost all graphs, in the usual technical sense, have no SSF. We also show how to control the possible sets that weakly induce SSTs by attaching certain small graphs ('forcers' and 'preventers') to a given graph. Naturally, one is also interested in computational issues: how easy is it to find SSTs or to compute parameters like $s_t(G)$? In Section 3 we answer many of these questions, showing the (surely to be expected) NP-hardness of these problems in general, but providing polynomial-time algorithms in special cases.

As promised, we now briefly discuss the original motivation for looking at spanning star trees. Herbert Fleischner [4] introduced the notion of an *A-trail* in a 2-connected Eulerian graph embedded in the plane — a closed Eulerian tour having the property that consecutive edges bound a common face. If H is such a graph, we can color the faces of H white and green (with the unbounded face white) and form the bipartite plane graph G whose red vertices are the vertices of H and whose blue vertices are the green faces of H, with a red vertex adjacent to each blue vertex (green face) that it bounds. Then H has an A-trail if and only if G has a blue-spanning star tree. (Essentially the Eulerian tour 'runs around the outside' of the tree.) Fleischner conjectured that an A-trail always exists in any Eulerian plane triangulation (a counterexample exists if only 3-connectivity is assumed). For us, then, this is the statement that certain planar bipartite graphs, in which every blue vertex has degree 3, have BSSTs. Part of the importance of Fleischner's conjecture is that it is equivalent to the Barnette–Tutte conjecture that every planar, 3-connected, cubic, bipartite graph is Hamiltonian, which in turn is a modification of Tait's famous but ill-fated conjecture (the latter omitted the bipartiteness assumption), whose truth would have provided an easy proof of the Four Color Theorem. Fleischner's conjecture and the various questions one can ask about SSTs in planar graphs, by dealing at their most intimate levels with the subtly competing notions of connectivity and acyclicity, seem to get at the heart of our ignorance about the combinatorial structure of the plane.

2. The existence of good and bad graphs

It is trivial to find good graphs: trees, cycles, complete bipartite graphs, or graphs containing a vertex adjacent to every other vertex, for instance. The last of these examples makes the following fact clear.

Proposition 1. *There exists a good graph having n vertices and m edges if and only if* $n-1 \leqslant m \leqslant n(n-1)/2$.

It is also not hard to find examples of bad graphs. The complement of a matching on six or more vertices, or of a cycle on seven or more vertices, has no spanning star forest, because in these cases if W has just one vertex, then $S(W)$ will not span, and if any two vertices are included in W, then $S(W)$ will contain a 3-cycle or a 4-cycle. The following lemma presents a useful tool along these lines.

Lemma 2. *If every vertex of a graph G has degree greater than* $n/2$ *and less than* $n-1$, *then G has no SSF.*

Proof. The upper bound makes it impossible for $S(\{v\})$ to span, for any vertex v; and the lower bound forces $S(\{u,v\})$ to contain a 3-cycle or a 4-cycle for any two distinct vertices u and v. The examples of the good graphs K_n and $K_{n/2,n/2}$ show that we cannot weaken the hypotheses. □

Fig. 1 shows all bad graphs on six vertices (of the 112 connected 6-graphs), found by an exhaustive computer search, which utilized Brendan McKay's powerful *makeg* and *nauty* programs [11]. (By Theorem 2, below, we know that there are no bad graphs on fewer than six vertices.) Two of these graphs have 9 edges, two have 10, one has 11 and one has 12. The computer search also turned up 83 bad 7-graphs, out of 853 (6 with 10 edges, 10 with 11 edges, 22 with 12 edges, 15 with 13 edges, 14 with 14 edges, 11 with 15 edges, 4 with 16 edges, and 1 with 17 edges). And there are 1870 bad 8-graphs, out of 11,117.

Our first theorem shows that there exist bad graphs having n vertices and m edges unless m is too small or too large.

Theorem 3. *There exists a bad graph having n vertices and m edges if and only if* $n+3 \leqslant m \leqslant n(n-2)/2$.

Proof. For the 'only if' part, suppose that G is a connected graph violating the inequality. We must show that G is good. If $m > n(n-2)/2$, then some vertex of G has degree greater than $n-2$ and hence G has a spanning star. So suppose that $m \leqslant n+2$; since G is connected, the smallest possible value for m is $n-1$. In that case G itself is an SST (using all the vertices). If $m = n$, then the graph must be unicyclic. If we let W consist of all but two adjacent vertices on the cycle, together with all the

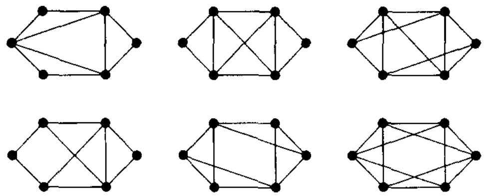

Fig. 1. All graphs on six vertices with no spanning star tree.

vertices not on the cycle, then $S(W)$ is an SST. Note that the vertices not on any cycle can always be included in W without creating any cycles (and automatically covering those vertices and connecting them to the rest of the graph), so we will assume in the remaining two cases that there are no such vertices. If $m = n + 1$, then G has two cycles. If the cycles are vertex-disjoint or share just one vertex, then we can let W contain all the vertices of G except for two adjacent vertices on each cycle (making sure to include any vertices of degree three or four). Otherwise, the cycles have at least two vertices in common, and the graph consists of two vertices, u and v, of degree three, with paths P_1, P_2, and P_3 joining them, such that at least two of the paths, say P_1 and P_2, have length greater than one (since there are no parallel edges). We take W to be the set of all vertices on P_3 other than v, together with all vertices on P_1 and P_2 other than v or the vertices adjacent to v. It is clear that $S(W)$ is an SST. To finish the 'only if' part of the proof, assume that $m = n + 2$ and again that G has no vertices not in some cycle. There are several possibilities for the structure of G up to homeomorphism (although only three that are 2-connected), and it is straightforward (if tedious) to check that in each case an SST can be found, proceeding along lines similar to those used in the $m = n + 1$ case.

We prove the 'if' part of the theorem by induction on n. The smallest case in which the inequality can be satisfied is $n = 6$ (with $9 \leqslant m \leqslant 12$), and it is straightforward to check that the graphs in Fig. 1 have no SSTs. Assume that there exist bad graphs with n vertices and m edges for $n + 3 \leqslant m \leqslant n(n-2)/2$; we must show that there exist bad graphs with $n + 1$ vertices and m edges for $n + 4 \leqslant m \leqslant (n+1)(n-1)/2$. For the first part of this range, as long as $m \leqslant n(n-2)/2 + 1$, we can simply take a bad graph on n vertices and $m - 1$ edges and attach a new pendant edge to one vertex; this new edge neither helps nor impedes the search for an SST. As for the upper part of this range, if $m \geqslant (n+1)(n+3)/4$ (for n odd) or $m \geqslant (n+1)(n+2)/4$ (for n even), then we can easily construct a graph on $n+1$ vertices in which each vertex has degree greater than $(n+1)/2$ (but still less than n) by making the vertex degrees nearly equal. Then by Lemma 2 the resulting graph is bad. We have left no gap as long as

$$\frac{(n+1)(n+3)}{4} \leqslant \frac{n(n-2)}{2} + 2 \quad \text{(for } n \text{ odd)}$$

or

$$\frac{(n+1)(n+2)}{4} \leqslant \frac{n(n-2)}{2} + 2 \quad \text{(for } n \text{ even).}$$

These inequalities are satisfied if $n \geqslant 6$, with the one exception of $n = 7$. In that case, to fill the gap we need a bad graph with 8 vertices and 19 edges, and there are in fact 145 of them — for example, a graph with degree sequence $(5,5,5,5,5,5,4,4)$ in which the two vertices of degree 4 are not adjacent. □

Similar characterizations hold in the bipartite situation.

Theorem 4. *There exists a bipartite graph with n_r red vertices, n_b blue vertices, and m edges having a BSST if and only if $n_b \leqslant m \leqslant n_r n_b$. There exists a connected bipartite graph with n_r red vertices, n_b blue vertices, and m edges having no BSST if and only if $n_b \geqslant 4$ and $n_r + n_b \leqslant m \leqslant n_r(n_b - 1)$.*

Proof. The first statement is trivial, since each of the blue vertices must be dominated and we can let one red vertex dominate all of them. We turn to the 'only if' part of the second statement, and let G be a connected bipartite graph with the given parameters. The condition $n_b \geqslant 4$ follows from the fact that G is connected. The lower bound on m is clear, since the only other possibility would be that $m = n_r + n_b - 1$, in which case G itself is a BSST (using all the red vertices). And if the upper bound on m is violated, then G must have a red vertex adjacent to every blue one.

It remains to construct a connected bipartite graph without a BSST and having the given parameters whenever the inequalities are satisfied. If $n_r = 2$, then we can let the first red vertex be adjacent to the last $n_b - 1$ blue vertices and let the second red vertex be adjacent to the first $m - n_b + 1$ blue vertices. As long as m satisfies the given double inequality, neither red vertex dominates all the blue ones, but together they weakly induce a 4-cycle (with the second and third blue vertices); thus, the resulting graph has no BSST. If $n_r > 2$, then we can join the additional red vertices to the blue vertices, in order, from the second onward, because of the given double inequality. Such vertices can never help in forming a BSST. □

We next turn to asymptotic behavior.

Theorem 5. *Almost all graphs have no SSF, and almost all bipartite graphs have no BSSF.*

Proof. We sketch the proof for the ordinary situation; the bipartite case is similar. Here we are using the random graph model in which each possible edge appears independently with probability p, $0 < p < 1$. The probability that a graph with n vertices has a vertex of degree $n-1$ is at most np^{n-1}, which approaches 0 as $n \to \infty$. Given any two vertices, the probability that they do not have a pair of common neighbors is at most $(1-p^4)^{\lfloor (n-2)/2 \rfloor}$, since every other pair of vertices has probability p^4 of being common neighbors. Therefore, the probability that there exists a pair of vertices without a pair of common neighbors is at most $\binom{n}{2}(1-p^4)^{\lfloor (n-2)/2 \rfloor}$. Since this, too, approaches 0, we conclude that almost all graphs have no SSF, because almost surely one vertex is not enough to weakly induce a spanning forest, and two (or more) vertices weakly induce a graph with 4-cycles. □

Finally, we look at some graphs that can be attached to other graphs to control which vertices are used in SSTs and/or SSFs. These will be useful in Section 3. Call a graph P a *preventer* if, for every graph H formed from the disjoint union of P and another graph G with distinguished vertex v by adding edges from P to G in a specified way, (1) H has no spanning star forest that uses v, and (2) every SSF $S(W)$ of G that does not use v can be extended to an SSF of H by adding some vertices of P to W. In the ordinary situation, we can take P to be a triangle, with two of its vertices attached to v by new edges. If $v \in W$, then clearly $S(W)$ will contain a cycle if W contains any vertex of P, and $S(W)$ will not include all the vertices of P if it does not. Furthermore, including one of the attaching vertices of P extends any SSF of G not using v to an SSF of H. In the bipartite case, the distinguished vertex of attachment will be red, of course. Here one preventer P consists of red vertices u and w, blue vertices a, b, and c, and edges au, bu, cu, and bw, with edges of attachment av, bv, and wg for some arbitrary blue vertex g in the graph to which P is being attached (we think of g as a 'ground').

In a similar way, we can force a particular [red] vertex v to be part of any subset W that weakly induces an SSF. In the bipartite case, we simply attach a new blue vertex to v. In the ordinary case, we add four new vertices to the graph, each adjacent to v, and put a 4-cycle on the new vertices. Here any SSF $S(W)$ of the original graph is also an SSF of the larger graph as long as $v \in W$, and if $v \notin W$ then the larger graph has no SSF.

By using preventers and forcers, we can construct a wide variety of bad graphs, graphs with unique SSTs, and so forth.

3. Algorithmic considerations

One can ask many algorithmic questions about weakly connected and/or strongly acyclic domination. We would like to be able to determine whether a given graph G has an SST; to find one if it does; and more explicitly to find a set of vertices W that weakly induces one. We would also like to be able to determine the cardinality of the smallest such W. And we would like to be able to answer similar questions for SSCs and SSFs, all of these in both the ordinary and the bipartite situation.

We start with a basic intractability result for bipartite graphs and then extend it to the ordinary situation.

Theorem 6. *The problem of determining whether a given connected bipartite graph has a BSSF (resp., BSST) is* NP-*complete. Determining whether it has a BSST remains* NP-*complete even if every blue vertex has some fixed degree $k \geqslant 2$, or if the graph is planar.*

Proof. The problems are clearly in the class NP, since it is easy to compute $S(W)$ from a given W and check it for connectedness and the absence of cycles. To show NP-completeness, we polynomially reduce to them the known [5] NP-complete problem EXACT COVER. Recall that an instance of EXACT COVER is a set C of subsets of a set X, and the question is whether there is a subset $C' \subseteq C$ (the exact cover) such that every element of X occurs exactly once as an element of a set in C'. Suppose that we wish to solve a given EXACT COVER problem. Construct a bipartite graph G whose blue vertices are the elements of X together with one extra vertex b, and whose red vertices are the elements of C. Each red vertex is adjacent to its elements and to b. If there is an exact cover C', then the vertices in C' weakly induce a BSSF (which is also a BSST). Conversely, any set C' of red vertices that weakly induces a BSSF (or BSST) is necessarily an exact cover, since no vertex in X can be covered twice without creating a 4-cycle in $S(C')$.

We need to modify the construction a bit in order to achieve the desired degree restriction on the blue vertices. Here we can assume without loss of generality that every element of X appears in at least one set in C. Again start with a red vertex r_S for each set S in C. This time there is a blue vertex for each occurrence of each element in a set; so if $x \in S$, then there is a blue vertex $b_{x,S}$ adjacent to r_S. Add a red vertex r_x for each $x \in X$, and make it adjacent to each blue vertex $b_{x,S}$. Add $k-2$ pendant red vertices adjacent to each $b_{x,S}$ as well, bringing the degree of each $b_{x,S}$ to exactly k. Next create a blue vertex b_x for each $x \in X$, adjacent to r_x and to $k-1$ pendant red vertices. Finally, add a blue vertex b_S for each $S \in C$, an edge from r_S to b_S, and $k-2$ pendant red vertices adjacent to b_S; and then add a red vertex r adjacent to all the b_S vertices, as well as a blue vertex b adjacent to r and $k-1$ pendant red vertices. Then each blue vertex has degree k; the graph has only $O(k|C||X|)$ vertices; and there is an exact cover if and only if our graph has a BSST.

The NP-completeness of the BSST problem when the bipartite graph is planar follows from the discussion at the end of Section 1 and the fact that the problem of determining whether an Eulerian plane graph has an A-trail is NP-complete [1,2]. □

Remark 7. Lars Døvling Andersen and Fleischner [1], working in the equivalent context of hypergraphs, give a different proof of part of Theorem 6, for the case in which the graph has no 4-cycles and each blue vertex has degree 3. They also present Carsten Thomassen's proof in the case that the red vertices all have degree 4, one blue vertex is adjacent to all red vertices, and all other blue vertices have degree at most 3.

Theorem 8. *Given a bipartite graph H, one can construct in polynomial time a graph G such that H has a BSSF (resp., BSST) if and only if G has an SSF (resp., SST).*

Proof. The construction of G is as follows. Start with H, viewed as a graph rather than a bipartite graph. First attach a preventer (see Section 2) to each (formerly) blue vertex in order to ensure that these vertices cannot be used in any SSF. Second, for each red vertex r, choose one blue vertex b (adjacent to r if possible), and add a new vertex v_r together with edges rv_r and bv_r. The required equivalence is immediate. (For the following corollary, note that this construction can be carried out in the plane if H is planar.) $\square$

Corollary 9. *The problem of determining whether a given connected graph has an SSF (resp., SST) is* NP-*complete, and the SST problem remains so even if the graph is required to be planar.*

Remark 10. It is easy to recover a set of vertices weakly inducing a given SSF, since we can simply take all those vertices whose stars are contained in the SSF.

As usual, these problems become tractable if restricted to a suitably nice class of graphs (see also the remark following Theorem 13). As one example, recall [7] that a split graph is one whose vertex set can be partitioned into an independent set and a clique, and that finding such a partition (or determining that there is none) can be carried out in polynomial time. Split graphs are a special case of chordal (triangulated) graphs. The problem of determining whether a split graph has an SSF or SST can be solved in polynomial time, since at most one vertex of the clique can be used in any SSF, and a vertex in the independent set must be used unless one of its neighbors in the clique is used. We conjecture that there is also a polynomial-time algorithm for finding SSTs or SSFs of chordal graphs, since it is unusual [10] for computational problems to separate chordal graphs from split graphs.

Next we turn to the issue of computing parameters such as $s_c(G)$, the weakly connected domination number of G, or $s_t(G)$, the cardinality of a smallest set that weakly induces a spanning tree of G. The following simple theorem gives the flavor of the results one can obtain. Further progress along these lines is reported in [3], as are various bounds on these and related parameters.

Theorem 11. *The problem of computing s_c is* NP-*hard.*

Proof. We polynomially reduce the well-known [5] NP-hard problem of computing a graph's domination number γ to the problem of computing s_c for a related graph. Given a graph G, form a new connected graph G' by adding a new vertex v' for every vertex v of G, each edge vv', a new vertex s adjacent to all the v' vertices, and a new vertex t adjacent to s. We claim that $s_c(G') = \gamma(G) + 1$. Let us look for a minimum cardinality weakly connected dominating set W for G'. Clearly either s or t must be included in W, and there is no advantage in using t, so we can assume that $s \in W$. There is also no point in using any vertex v', since using v instead puts at least all the same edges into $S(W)$. But now since s covers none of the vertices of G, it is clear that a smallest W consists of s together with a minimum dominating set of G. Note that $S(W)$ is connected because the distance from s to each vertex in G' is at most 3. $\square$

Just as computing other varieties of domination numbers becomes easy on trees, we have the following result. Note that if T is a tree, then $s_{\mathrm{f}}(T) = \gamma(T)$ and $s_{\mathrm{t}}(T) = s_{\mathrm{c}}(T)$.

Theorem 12. *There is a linear-time algorithm for computing $s_{\mathrm{t}}(T)$ if T is a tree.*

Proof. We consider T as a rooted tree, with root r, and label the vertices in post-order with pairs (a_v, b_v), where a_v is the minimum cardinality of a weakly connected dominating set for the subtree rooted at v that uses v, and b_v is the minimum cardinality of a weakly connected dominating set for the subtree rooted at v that does not use v. Each leaf receives the label $(1, \infty)$. Thereafter, b_v is the sum of a_i for all children i of v; and a_v is $1 + \sum \min(a_i, b_i')$, again summed over the children of v, where $b_i' = 0$ if i is a leaf and $b_i' = b_i$ otherwise. Then $s_{\mathrm{t}}(T) = \min(a_r, b_r)$. The number of steps in the algorithm is proportional to the number of edges in the tree — that is, linear in the size of the input. □

The analysis of the complexity of computing s_{t} in general is a bit more subtle than it was for s_{c}, since we need to worry about acyclicity. The proof uses analogues of the preventers introduced in Section 2; this time we want to make it expensive to use a particular vertex in W, not to make it impossible. In the ordinary situation, the graph E to be attached consists of an edge uw, with a large number of pendant edges incident to u; both u and w are joined by edges to the distinguished vertex v of the original graph. If v is used in forming an SST of this union, then all the pendant vertices must also be used; otherwise only u is needed. The graph E^{B} for the bipartite situation is constructed as follows: Blue vertices b and c are adjacent to red vertex w. Further, w is adjacent to a large number of blue vertices a_j, each of which is in turn adjacent to a red vertex u_j. The vertices b and c are attached by edges to v (the distinguished red vertex of the original graph), and w and all the u_j vertices are 'grounded' by inserting edges from them to some blue vertex in the original graph (the ground). With E^{B} attached, it is expensive to use v in a BSST, since using v requires the use of all the u_j (and because of the ground, it is sufficient to do this), whereas using w alone suffices if v is not used.

Theorem 13. *The problem of determining, given a graph (resp., a bipartite graph) G and an integer k, whether $s_{\mathrm{t}}(G) \leqslant k$ (resp., $s_{\mathrm{t}}^{\mathrm{B}}(G) \leqslant k$) is* NP*-complete. It remains* NP*-complete even if we know that G has an SST (resp., a BSST).*

Proof. The proof in the ordinary situation is via reduction from the problem of determining whether a graph has an SSF (Corollary 9), as follows. We are given a graph G' with n vertices and wish to determine whether G' has an SSF. Form a new graph G by adding a new vertex v for every vertex v' of G', each edge vv', and a new vertex s adjacent to all v vertices; and attaching a copy of the graph E described above to each v, with $2n+1$ pendant edges in each copy. Note that $S(W)$ is an SST of G if W consists of all the v vertices and all the pendant vertices in all the copies of E. Now if G' has an SSF $S(W')$, then we need to use at most $n+1$ additional vertices to weakly induce an SST $S(W)$ of G, since we can let W be W' together with s and the high-degree vertex in each copy of E. Thus, if G' has an SSF, then $s_t(G) \leqslant 2n+1$. On the other hand, if G' does not have an SSF, then any SST of G must use at least one of the v vertices and its associated pendant vertices, and therefore must use more than $2n+1$ vertices in all. Thus, which of these is the case will be settled if we know whether $s_t(G) \leqslant 2n+1$. Note that this reduction is polynomial, since there are only $O(n^2)$ vertices in G.

The bipartite version is similar, using reduction from the BSST problem. Suppose that we have a bipartite graph G' and wish to determine whether G' has a BSST. Form G by adding a new red vertex v_i adjacent to each blue vertex i of G, making all the v_i vertices adjacent to one new blue vertex g (which serves as the ground), and attaching a copy of the bipartite graph E^{B} constructed above to each v_i. If G' has a BSST, then we can extend it to a BSST of G by using just one of the new expensive vertices v_i (and its associated u_j's); but if not, then any BSST of G will need at least two of them. □

Remark 14. The various problems considered here can be stated in (extended) monadic second order logic. It follows [15] that there are efficient algorithms for their solution on graphs of bounded treewidth, such as series-parallel graphs, outerplanar graphs, graphs with bounded bandwidth, and chordal or interval graphs with bounded maximum clique size.

Acknowledgements

The author thanks Suzanne Zeitman for helpful conversations on several aspects of this paper.

References

[1] L.D. Andersen and H. Fleischner, The NP-completeness of finding A-trails in Eulerian graphs and of finding spanning trees in hypergraphs, Discrete Appl. Math. 59 (1995) 203–214; MR 96b:05098.

[2] S.W. Bent and U. Manber, On non-intersecting Eulerian circuits, Discrete Appl. Math. 18 (1987) 87–94; MR 88h:05058.

[3] J.E. Dunbar, J.W. Grossman, J.H. Hattingh, S.T. Hedetniemi and A.A. McRae, On weakly-connected domination in graphs, Discrete Math., to appear.

[4] H. Fleischner, Eulerian Graphs and Related Topics, Part 1, Vol. 1, Annals of Discrete Mathematics, Vol. 45 (North-Holland, Amsterdam, 1990); MR 91g:05086.

[5] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness (W.H. Freeman, San Francisco, 1979); MR 80g:68056.

[6] J.W. Grossman, Spanning star trees in regular graphs, preprint.

[7] P.L. Hammer and B. Simeone, The splittance of a graph, Combinatorica 1 (1981) 275–284; MR 84c:05050.

[8] T. Haynes, Bibliography of papers on domination in graph theory, private communication, 1994.

[9] S.T. Hedetniemi and R.C. Laskar, Topics on Domination: Reprint of Discrete Math. 86 (1990), no. 1–3, Annals of Discrete Mathematics, Vol. 48 (North-Holland, Amsterdam, 1991); CMP 1 104 281.
[10] D.S. Johnson, The NP-completeness column: an ongoing guide, J. Algorithms 6 (1985) 434–451; MR 87b:68038.
[11] B.D. McKay, Computer programs and user's guide, private communication.
[12] R.E. Newman-Wolfe, R.D. Dutton and R.C. Brigham, Connecting sets in graphs — a domination related concept, Congressus Numerantium, Vol. 67 (Nineteenth Southeastern Conference on Combinatorics, Graph Theory, and Computing, Baton Rouge, 1988) 67–76; MR 90b:05096.
[13] E. Sampathkumar and L. Pushpa Latha, Set domination in graphs, J. Graph Theory 18 (1994) 489–495; MR 95h:05123.
[14] E. Sampathkumar and H.B. Walikar, The connected domination number of a graph, J. Math. Phys. Sci. 13 (1979) 607–613; MR 81g:05077.
[15] D. Seese, Interpretability and tree automata: a simple way to solve algorithmic problems on graphs closely related to trees, in: M. Nivat and A. Podelski, eds., Tree Automata and Languages: Papers from the Workshop Held in Le Touquet, June 1990 (North Holland, Amsterdam, 1992) 83–114; MR 93j:68062.