Degree-Preserving Forests

Hajo Broersma[1], Andreas Huck[2], Ton Kloks[3], Otto Koppius[1], Dieter Kratsch[4], Haiko Müller[4], and Hilde Tuinstra[1]

[1] University of Twente
Faculty of Applied Mathematics
P.O. Box 217
7500 AE Enschede, the Netherlands
{broersma,tuinstra}@math.utwente.nl

[2] University of Hannover
Institute of Mathematics
Welfengarten 1
30167 Hannover, Germany
huck@math.uni-hannover.de

[3] Department of Applied Mathematics
Charles University
Malostranské nám. 25
11800 Praha 1, Czech Republic
kloks@kam.ms.mff.cuni.cz

[4] Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik
07740 Jena, Germany
{kratsch,hm}@minet.uni-jena.de

Abstract. We consider the degree-preserving spanning tree (DPST) problem: given a connected graph G, find a spanning tree T of G such that as many vertices of T as possible have the same degree in T as in G. This problem is a graph-theoretical translation of a problem arising in the system-theoretical context of identifiability in networks, a concept which has applications in e.g., water distribution networks and electrical networks. We show that the DPST problem is NP-complete, even when restricted to split graphs or bipartite planar graphs. We present linear time approximation algorithms for planar graphs of worst case performance ratio $1 - \epsilon$ for every constant $\epsilon > 0$. Furthermore we give exact algorithms for interval graphs (linear time), graphs of bounded treewidth (linear time), cocomparability graphs ($O(n^4)$), and graphs of bounded asteroidal number.

1 Description of the Problem and Its Practical Niche

Analysis of communication or distribution networks is often concerned with finding spanning trees (or forests) of those networks fulfilling certain criteria. Also in other contexts spanning trees show up as important tools in modeling and analyzing problems. Therefore, a myriad of problems on spanning trees have been

Luboš Prim et al. (Eds.): MFCS'98, LNCS 1450, pp. 713–721, 1998.

studied in literature (see [5,6,8,9]). This paper deals with a virtually unexplored problem concerning spanning trees which we call the degree-preserving spanning tree (DPST) problem: given a connected graph G, find a spanning tree T of G with a maximum number of degree-preserving vertices, i.e., with a maximum number of vertices having the same degree in T as in G.

Some closely related questions were studied before by Lewinter et al. [1,12,13] from a purely theoretical point of view. They published a number of short notes on the subject. For example, Lewinter [12] introduced the concept of degree-preserving spanning trees and he proved that the number of degree-preserving vertices interpolates on the set of spanning trees of a given connected graph G. In other words: if spanning trees exist with k and l degree-preserving vertices respectively and $k < l$, then there exists a spanning tree with exactly m degree-preserving vertices for every m with $k < m < l$.

Our attention was initially turned to this problem through a practical application in water distribution networks (see [14]), which makes the DPST problem a nice example of theory and practice going hand-in-hand.

Suppose that we have to determine (or control) all flows in such a network by installing and using a small number of flow meters and/or pressure gauges. The network can be regarded as an undirected connected graph G and the flow through each edge of G is described by an orientation of that edge and a non-negative flow value. Since the sum of all flow values of edges entering a vertex is always the same as the sum of all flow values of edges leaving that vertex, except for possible sources and sinks, it is not difficult to derive all flows in the network from the flows through all edges of a cotree C of G (i.e., C is obtained from G by removing the edges of a spanning tree). Hence it would suffice to install flow meters at the edges of C. However the costs of installing a flow meter is much higher than those of installing a water pressure gauge at some vertex. Alternatively, we can derive the flow through an edge from the water pressure drop between the two incident vertices. If we only use pressure gauges, and want to minimize the costs, the problem becomes that of finding a cotree whose edges are incident with a minimum number of vertices (in order to minimize the number of pressure gauges that have to be installed) or, equivalently, of finding a spanning tree T whose complement in G has as many isolated vertices as possible, i.e., T has a maximum number of degree-preserving vertices. Rahal [15] independently discovered the cotree approach in his investigation of a steady state formulation for water distribution networks.

Our problem of determining all flows in the network with minimal costs of measuring (installing pressure gauges) is a so-called identifiability problem (see Walter [16]). The concrete water distribution network that we considered has 80 vertices and 98 edges, making it a very sparse network. Our network is planar and it has outerplanarity 2. Especially this latter fact enables us to solve the DPST problem in our case by a linear time algorithm.

2 Preliminaries

Throughout let $G = (V, E)$ be a graph and let $n = |V|$ and $m = |E|$.

For a subset $S \subseteq V$ we use $G[S]$ to denote the subgraph of G induced by the vertices of S. For a subset $S \subseteq V$ we also write $G - S$ for $G[V \setminus S]$, and for a vertex x of G we write $G - x$ instead of $G - \{x\}$.

For a vertex x of G we use $N_G(x)$ to denote the set of neighbors of x in G, and we write $N_G[x] = \{x\} \cup N_G(x)$ for the closed neighborhood of x in G; the degree of x in G is $d_G(x) = |N_G(x)|$. A pendant vertex of G is a vertex with degree one in G. We omit the subscript G from the above expressions if it is clear which graph G we consider.

Definition 1. *A subset $S \subseteq V$ is* realizable *if there exists a spanning forest T of G such that the degree of every vertex $x \in S$ is preserved in T (i.e., if $d_T(x) = d_G(x)$ for every vertex $x \in S$). If T is such a spanning forest, then we call T an S-*preserving *forest. If, moreover, T is chosen in such a way that $|S|$ is maximum, then we call T a* maximum degree-preserving *forest, and $|S|$ the degree-preserving number (of T or G). The DPST problem is the problem to find for a given graph G a maximum degree-preserving spanning forest.*

As an example, the degree-preserving number of a tree, a unicyclic graph, and a complete graph ($\neq K_2$) on n vertices are respectively n, $n-2$, and 1.

Notice that to solve the DPST problem, it is sufficient to compute a maximum (cardinality) realizable set S since, given S, an S-preserving spanning forest is then easy to find. By $p(G)$ we denote the cardinality of a maximum realizable set in G. Clearly $p(G)$ is the sum of $p(C)$ taken over all 2-edge-connected components C of G. Therefore we can restrict to 2-edge-connected graphs.

Let W be a set of vertices of a graph G. By $G[\![W]\!]$ we denote the graph with vertex set $N[W]$ containing all edges of G incident with a vertex in W.

Lemma 1. *Let S be a nonempty set of vertices of a graph $G = (V, E)$. Then S is a realizable set of G if and only if $G[\![S]\!]$ is a forest.*

3 Hardness Results

A graph $G = (V, E)$ is called a *split graph* (*bipartite graph*) if V can be partitioned into an independent set I and a clique C (into two independent sets X and Y) of G. Such a graph is also denoted by $G = (I, C, E)$ ($G = (X, Y, E)$).

Let $G = (V, E)$ be a graph. We define a split graph H with independent set V and clique $E \times \{1, 2\}$ as follows. A pair $\{v, (e, i)\}$ is an edge of H if and only if $v \in V$, $e \in E$, $i \in \{1, 2\}$ and $v \in e$. It is easy to see that a set $W \subseteq V$ is an independent set of G if and only if W is a realizable set in H. Moreover, if G has no isolated vertices (i.e., vertices with degree zero), then for every realizable set W of H with $|W| > 1$ we have $W \subseteq V$. These simple observations lead to the following theorem showing that the DPST problem restricted to split graphs is NP-complete.

Theorem 1. *For a given split graph H and a given integer k it is NP-complete to decide whether H contains a realizable set of cardinality k.*

Proof. The reduction is from the NP-complete graph problem INDEPENDENT SET. As seen before a graph G has an independent set of cardinality k if and only if the corresponding split graph H has a realizable set of cardinality k. □

Next we apply the same idea to bipartite graphs. Let $G = (V, E)$ be a graph. We define a bipartite graph $B = (V \cup (E \times \{2, 4, 6, 8\}), E \times \{1, 3, 5, 7\}, F_1 \cup F_2)$, where

$$\begin{aligned} F_1 &= \{\{v, (e, i)\} : v \in V, e \in E, i \in \{1, 5\}, v \in e\} \\ F_2 &= \{\{(e, 1), (e, 2)\}, \{(e, 2), (e, 3)\}, \{(e, 3), (e, 4)\}, \{(e, 4), (e, 1)\}, \\ &\qquad \{(e, 5), (e, 6)\}, \{(e, 6), (e, 7)\}, \{(e, 7), (e, 8)\}, \{(e, 8), (e, 5)\} : e \in E\}. \end{aligned}$$

Note that for the maximum degrees $\Delta(B)$ and $\Delta(G)$ of B and G we get $\Delta(B) = \max\{4, 2 \cdot \Delta(G)\}$. Moreover, B is planar if and only if G is planar.

We observe that for every edge $e \in E$ and every realizable set S of B, $|S \cap (\{e\} \times \{1, 2, 3, 4\})| \leq 2$. In what follows we may assume $S \subseteq V \cup (E \times \{2, 3, 6, 7\})$ for all realizable sets S of B, since for every other realizable set T the set $T' = (T \cap V) \cup (E \times \{2, 3, 6, 7\})$ is also realizable and fulfills $|T| \leq |T'|$.

Next observe that $W \subseteq V$ is an independent set of G if and only if W is a realizable set of B. This leads to the following theorem showing that the DPST problem restricted to bipartite planar graphs is NP-complete.

Theorem 2. *For a given bipartite planar graph B of maximum degree six and a given integer k, it is NP-complete to decide whether B contains a realizable set of cardinality k.*

Proof. The reduction is from the INDEPENDENT SET problem restricted to cubic (i.e., 3-regular) planar graphs [9]. Let (G, k) be an instance of this NP-complete problem where $G = (V, E)$ with $|E| = m$. As seen before a planar graph G has an independent set of cardinality k if and only if the corresponding bipartite planar graph B has a realizable set of cardinality $k + 4m$. □

Our problem remains NP-complete even when restricted to bipartite planar graphs of maximum degree three [7].

The INDEPENDENT SET problem is not only NP-complete, it is also hard to approximate. More precisely for every $\epsilon > 0$, there is no polynomial time approximation algorithm for the MAXIMUM INDEPENDENT SET problem with worst case ratio $n^{1/4-\epsilon}$ unless P=NP [4], and there is no polynomial time approximation algorithm with worst case ratio $n^{1-\epsilon}$ unless co-NP=NP [11]. By the reduction used in the proof of Theorem 1, approximating an optimal solution to the DPST problem is as hard as approximating MAXIMUM INDEPENDENT SET, even when DPST is restricted to split graphs.

Theorem 3. *For every $\epsilon > 0$, there is no polynomial time algorithm to approximate a maximum realizable set of a given split graph with worst case ratio $n^{1/4-\epsilon}$ unless P=NP (respectively with worst case ratio $n^{1-\epsilon}$ unless co-NP=NP).*

4 Approximation for Planar Graphs

In this section we apply an idea of Baker [3] to establish linear time approximation algorithms for the DPST problem when restricted to planar graphs. We will prove the following theorem.

Theorem 4. *For every $\epsilon > 0$ there is a linear time approximation algorithm of worst case performance ratio $1 - \epsilon$ for the DPST problem restricted to planar graphs.*

Let $W \subseteq V$ be a set of forbidden vertices. A realizable set R of G is called *maximum W-avoiding realizable set* if $R \cap W = \emptyset$ and $|R| \geq |R'|$ for every realizable set R' of G with $R' \cap W = \emptyset$.

Let $G = (V, E)$ be a planar graph given with a fixed embedding. We partition V into levels $L_1, L_2, \ldots, L_d$. The level L_1 contains all vertices on the outer face of G. For $i > 1$, the level L_i contains all vertices on the outer face of $G - \bigcup_{j=1}^{i-1} L_j$. Let d be the largest index such that $L_d \neq \emptyset$. For technical reason set $L_i = \emptyset$ for $i > d$ or $i < 1$. A planar graph is *k-outerplanar* if and only if it has an embedding defining at most k nonempty levels.

We decompose the planar graph G into k-outerplanar graphs. Each k-outerplanar graph consists of k consecutive levels of G. More precisely, let k and r be integers with $1 \leq r \leq k$. For $i = 0, 1, \ldots, q$ with $q = \lceil \frac{d-r}{k} \rceil$ we define

$$G_{k,r,i} = G\big[\bigcup_{j=(i-1)k+r+1}^{ik+r} L_j\big] \quad \text{and} \quad W_{k,r,i} = L_{(i-1)k+r+1} \cup L_{ik+r}\,.$$

Note that $W_{k,r,i}$ contains all vertices in the outer and inner level of $G_{k,r,i}$.

Lemma 2. *For $i = 0, 1, \ldots, q$ let $R_{k,r,i}$ be a $W_{k,r,i}$-avoiding realizable set of $G_{k,r,i}$. Then $\bigcup_{i=0}^{q} R_{k,r,i}$ is a realizable set of G.*

Proof. For all i the set $W_{k,r,i}$ contains the vertices on the outer and the inner level of the k-outerplanar graph $G_{k,r,i}$. Hence the endpoints of an arbitrary edge of $G[\![R_{k,r}]\!]$ belong to the same k-outerplanar graph. □

Lemma 3. *For every $k \geq 1$ there is an index $r(k)$ with $1 \leq r(k) \leq k$ such that*

$$|R \setminus \bigcup_{i=0}^{q} W_{k,r(k),i}| \geq \tfrac{k-2}{k}\, p(G)\,.$$

Proof. Let R be a maximum realizable set of G and let $W_{k,r} = \bigcup_{i=0}^{q} W_{k,r,i}$. For every level L_j, $j = 1, 2, \ldots, d$, of G there exist at most two $r \in \{1, 2, \ldots, k\}$ with $L_j \subset W_{k,r}$. Hence $\sum_{r=1}^{k} |R \cap W_{k,r}| \leq 2|R|$, which implies that there is an $r = r(k)$ such that $|R \cap W_{k,r(k)}| \leq \frac{2}{k}|R|$. □

Let $k \geq 1$. For every $r = 1, 2, \ldots, k$ and every $i = 1, 2, \ldots, q$ let $R_{k,r,i}$ be a maximum $W_{k,r,i}$-avoiding realizable set of $G_{k,r,i}$. By Lemma 2, $R_{k,r} = \bigcup_{i=0}^{q} R_{k,r,i}$ is a realizable set of G. Consequently,

$$\max\{|R_{k,r}| : 1 \leq r \leq k\} \geq \tfrac{k-2}{k} p(G)\,.$$

For every k we develop an exact linear time algorithm computing a maximum W-avoiding realizable set for k-outerplanar graphs. Using standard techniques for graphs of bounded treewidth, it can be shown that a linear time algorithm, exists [2]. Notice that the treewidth of a k-outerplanar graph is at most $3k - 1$. Consequently, for every fixed k we obtain a linear time approximation algorithm of worst case performance ratio $\frac{k-2}{k}$.

5 Interval Graphs

Definition 2. *A graph is* chordal *if it contains no induced cycle of length more than three.*

Notice that for chordal graphs, the problem of finding a maximum realizable set is NP-complete, since the class of split graphs is a proper subclass of the class of chordal graphs. However, for the class of interval graphs, which is another important subclass of the class of chordal graphs, we can give a fast algorithm. For an introduction into these graph class we refer to [10].

Our first result shows that for chordal graphs we can restrict our search for realizable sets to independent sets. Remember that we may restrict to 2-edge connected graphs.

Theorem 5. *If G is a 2-edge connected chordal graph, then any realizable set S of G is an independent set of G.*

Proof. Let $G = (V, E)$ be a 2-edge connected chordal graph and assume $\{x, y\} \in E$ for two distinct vertices $x, y \in S$. Since G is 2-edge connected, $\{x, y\}$ is contained in a cycle of G, and, since G is chordal this implies $\{x, y\}$ is contained in some triangle of G. This contradicts Lemma 1. □

If a graph G is disconnected, then a maximum realizable set of G is simply the union of maximum realizable sets of all components of G. If a connected graph G (or a component) has a bridge e, then to compute a maximum realizable set of G delete e and compute maximum realizable sets S_1 and S_2 for both components. Let T_1 be an S_1-preserving forest and T_2 be an S_2-preserving forest. Adding e as an edge between T_1 and T_2 gives a forest T which is $S_1 \cup S_2$-preserving, and $S_1 \cup S_2$ is a maximum realizable set in G. We will use the above observations and the following properties of 2-edge connected interval graphs.

Definition 3. *An interval graph is a graph for which one can associate with each vertex an interval on the real line such that two vertices are adjacent if and only if their corresponding intervals have a nonempty intersection.*

Interval graphs can be recognized in linear time, and, given an interval graph, an interval model for it can be found in linear time [10]. In the following we assume that an interval model of the graph is given, and we identify the vertices of the graph with the corresponding intervals. Without loss of generality we may assume that no two intervals have an endpoint in common.

Definition 4. *An interval and its corresponding vertex are called* minimal *if it is minimal with respect to inclusion, i.e., if it does not contain any other interval.*

Lemma 4. *Let G be a 2-edge connected interval graph. Then there exists a maximum realizable set S of G such that for every vertex $p \in S$ the corresponding interval is minimal.*

Proof. Let S be a maximum realizable set containing a vertex x which is not minimal. Then there exists an interval y contained in the interval x. By Theorem 5 we know that a realizable set can contain only one of x and y and hence $y \notin S$. Now $N(y) \subseteq N[x]$, and hence, there exists a maximum realizable set $S' = \{y\} \cup S \setminus \{x\}$. Repeating the arguments we can prove the assertion of the theorem. □

Consider the ordering of the minimal intervals defined by the left endpoints.

Lemma 5. *Let G be a 2-edge connected interval graph with corresponding interval model and let x be the first minimal interval (i.e., with the leftmost left endpoint). There exists a maximum realizable set S of G with $x \in S$.*

Proof. Consider a maximum realizable set S of G containing only minimal intervals. If $x \in S$ there is nothing to prove. Otherwise, let y be the first interval in S. The other intervals of S lie totally to the right of y because S is an independent set by Theorem 5. The right endpoint of y must be to the right of the right endpoint of x since the interval x is minimal. It follows that $S' = \{y\} \cup S \setminus \{x\}$ is also realizable, since x lies totally left of $S \setminus \{y\}$ and $N(z) \cap N(x) \subseteq N(z) \cap N(y)$ for all $z \in S \setminus \{y\}$. □

Theorem 6. *There is a linear time algorithm to compute a maximum realizable set S for given interval graph G.*

Proof. Locate the set of bridges B in G and compute maximum cardinality realizable sets for each component of $G - B$. This can be done as follows.

Consider an interval model for a 2-edge connected component. First mark the minimal intervals. Take the minimal interval with the leftmost left endpoint as the first element of S. Consider the endpoints one by one, from left to right. We keep track of the last minimal interval in S which is totally left of the current position. We also keep a counter for the number of intervals that have one endpoint to the left of the current position and that overlap with the last interval in S. If we encounter a left endpoint of a minimal interval which starts to the right of the last interval in S so far, and if there is at most one interval overlapping the current position and the last interval of S, then we put this new minimal interval in S.

Let S' be a maximum realizable set such that $S \neq S'$. By the previous lemmas we may assume that S' contains minimal intervals only and that S and S' have a common first interval. Suppose y is the first interval of S' which is not in S, and that $x_1, x_2, \ldots, x_p$ are common intervals of S and S' and $x_{p+1} \neq y$ is the next interval of S chosen by the above procedure. We complete the proof by showing that y in S' can be replaced by x_{p+1}. This follows by the same arguments as in the proof of Lemma 5 and the following observations. By the choice of $x_1, x_2, \ldots, x_p$, for all $i, j \in \{1, \ldots, p\}$ with $i \neq j$, x_i and x_j have at most one common neighbor and $N(x_{p+1}) \cap N(x_i) \subseteq N(x_{p+1}) \cap N(x_{i+1})$ $(i = 1, \ldots, p-1)$. If the addition of x_{p+1} to $\{x_1, \ldots, x_p\}$ would cause a cycle in $G[\![\{x_1, x_2, \ldots, x_p, x_{p+1}\}]\!]$, then such a cycle would already exist in $G[\![\{x_1, \ldots, x_p\}]\!]$, a contradiction to the choice of $x_1, x_2, \ldots, x_p$. □

6 Other Classes of Graphs

In this section we list further results proven in the full version.

Theorem 7. *The degree preserving spanning tree problem is solvable in linear time for graphs of bounded treewidth.*

Definition 5. *A graph $G = (V, E)$ is a* cocomparability graph *if and only if there is an ordering $v_1, v_2, \ldots, v_n$ of V such that $i < j < k$ and $\{v_i, v_k\} \in E$ implies either $\{v_i, v_j\} \in E$ or $\{v_j, v_k\} \in E$. Hence $N(v_j) \cap \{v_i, v_k\} \neq \emptyset$ for all j with $i < j < k$. Such an ordering is called* cocomparability ordering.

Theorem 8. *There is an algorithm to compute a maximum degree-preserving forest of a cocomparability graph in time $O(n^4)$.*

Definition 6. *An independent set A is called an* asteroidal set *if for every vertex $a \in A$, the set $A \setminus \{a\}$ is contained in a component of $G - N[a]$. The* asteroidal number *of a graph G, is the maximum cardinality of an asteroidal set in G.*

Theorem 9. *There is an algorithm to solve the degree preserving spanning tree problem for any graph G in time $O(2^{k^3} n^{k+3} \log n)$, where k is the asteroidal number of G.*

References

1. Aaron, M. and M. Lewinter, 0-deficient vertices of spanning trees, *NY Acad. Sci. Graph Theory Notes* **XXVII**, (1994), pp. 31–32.
2. Arnborg S., J. Lagergren and D. Seese, Easy problems for tree-decomposable graphs, *Journal of Algorithms* **12**, (1991), pp. 308–340.
3. Baker, B. S., Approximation algorithms for NP-complete problems on planar graphs, *J. ACM* **41**, (1994), pp. 153–180.
4. Bellare, M., O. Goldreich and M. Sudan, Free bits, PCPs and non-approximability - towards tight results, *SIAM J. Comput.* **27** (1998), pp. 804–915.

5. Camerini, P. M., G. Galbiati and F. Maffioli, Complexity of spanning tree problems: Part I, *Eur. J. Oper. Res.* **5**, (1980), pp. 346–352.
6. Camerini, P. M., G. Galbiati and F. Maffioli, The complexity of weighted multi-constrained spanning tree problems, *Colloq. Math. Soc. Janos Bolyai* **44**, (1984), pp. 53–101.
7. Damaschke, P., Degree-preserving spanning trees and coloring bounded degree graphs, Manuscript 1997.
8. Dell'Amico, M., M. Labbé and F. Maffioli, Complexity of spanning tree problems with leaf-dependent objectives, *Networks* **27**, (1996), pp. 175–181.
9. Garey, M. R. and D.S. Johnson, *Computers and Intractability: A guide to the Theory of NP-completeness*, Freeman, New York, 1979.
10. Golumbic, M. C., *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
11. Håstad, J., Clique is hard to approximate within $n^{1-\epsilon}$, Proc. 37th Ann. IEEE Symp. on Foundations of Comput. Sci., (1996), IEEE Computer Society, pp. 627–636.
12. Lewinter, M., Interpolation theorem for the number of degree-preserving vertices of spanning trees, *IEEE Trans. Circ. Syst.* **CAS-34**, (1987), 205.
13. Lewinter, M. and M. Migdail-Smith, Degree-preserving vertices of spanning trees of the hypercube, *NY Acad. Sci. Graph Theory Notes* **XIII**, (1987), 26–27.
14. Pothof, I. W. M. and J. Schut, *Graph-theoretic approach to identifiability in a water distribution network*, Memorandum 1283, Faculty of Applied Mathematics, University of Twente, Enschede, the Netherlands, (1995).
15. Rahal, A co-tree flows formulation for steady state in water distribution networks, *Adv. Eng. Softw.* **22**, (1995), pp. 169–178.
16. Walter, E., *Identifiability of state space models with applications to transformation systems*, Springer-Verlag, New York NY, USA, 1982.