Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Journal of Discrete Algorithms 5 (2007) 622–634
JOURNAL OF DISCRETE ALGORITHMS
www.elsevier.com/locate/jda

Steiner diagrams and k-star hubs

Ulrich Blasum [a], Winfried Hochstättler [b], Peter Oertel [c,*,1], Gerhard J. Woeginger [d]

[a] *GUS SP GmbH, Cologne, Germany*
[b] *FernUniversität in Hagen, Germany*
[c] *Ford-Werke AG, Cologne, Germany*
[d] *Universiteit Eindhoven, The Netherlands*

Received 31 October 2005; received in revised form 15 May 2006; accepted 30 June 2006

Available online 7 May 2007

Abstract

In this paper, two problems derived from reload problems in transport logistics are introduced and studied. Given a transitive digraph $G = (V, A, w)$ with nonnegative arc weights (the transport network) and a set of directed node pairs B (the demand), the objective of the Steiner Diagram Problem is to find an acyclic set of arcs S of minimum cost that contains a directed path for each pair in B. This problem is $\mathcal{NP}$-complete in the general case and has some interesting structural properties that make it polynomially solvable if the size of B is bounded by a constant, the triangle inequality holds in A and A is transitively closed. A special case of this problem is a weighted edge cover problem with k cost functions on the vertices. It is shown that this problem is $\mathcal{NP}$-complete for $k \geqslant 3$. An efficient algorithm for the case $k = 2$ is given.

Keywords: Steiner trees; Hub location; Network design

1. Introduction

Reload problems play an increasing role in modern transport logistics. It seems natural in a Pickup and Delivery Problem to allow the transport good to be temporarily stored at some intermediate stop. However, appropriate mathematical models are quite complex [8]. Thus, to gain more insight into these kinds of problems, we derived two combinatorial optimization problems. Although they are too idealized to directly model an application, they seem to be quite natural and to capture important structural properties. Moreover, in one case the algorithmic solution of our relaxed problem has been successfully incorporated in a heuristic addressing a real world reload problem.

Both problems arose by distinguishing two kinds of reload problems. Those where a reload is possible, basically, at any point of the network give rise to the Steiner Diagram Problem (SDP). On the other hand, there are applications where goods may be reloaded only once at a small number of designated "hubs", originating the k-Star Hub Problem (k-SHP).

* Corresponding author.
E-mail addresses: ublasum@guskoeln.de (U. Blasum), Winfried.Hochstaettler@FernUni-Hagen.de (W. Hochstättler), poertel@ford.com (P. Oertel), gwoegi@win.tue.nl (G.J. Woeginger).
1 Supported by Alfried Krupp von Bohlen und Halbach Stiftung.

1570-8667/$ – see front matter © 2006 Published by Elsevier B.V.
doi:10.1016/j.jda.2006.06.001

While both problems are $\mathcal{NP}$-complete in general, we have found important cases to be solvable in polynomial time. In particular, this is the case for the SDP under additional restrictions on the underlying graph if the number of transportation requests is bounded. For the k-Star Hub Problem we show that it is efficiently solvable with standard techniques for at most two reload vertices and $\mathcal{NP}$-complete already for at least three "hubs".

The paper is organized as follows. In the next section the motivating application and previous literature will be discussed. Then, the Steiner Diagram Problem will be introduced. It will be shown that the problem is $\mathcal{NP}$-complete, even if all vertices of the graph are incident to demand nodes (no "Steiner vertices"), and structural properties of its optimum solutions will be studied. For this purpose, it is shown that the number of "junctions" is bounded in the optimum if the number of demands is bounded. As a consequence a bounded number of candidate solutions can be enumerated if the triangle inequality holds in the underlying graph and it is transitively closed. This suffices to prove that the problem is polynomially solvable. Both assumptions are quite natural for our application. In Section 4 the k-Star Hub Problem is defined. We analyze its complexity and present an efficient algorithm for $k \leqslant 2$. Finally, we close with some concluding remarks.

We assume familiarity with graph theory and computer science. The notation is fairly standard as in [6]. Additionally, for $G = (V, A)$ a digraph, $B \subseteq A$, $V[B]$ will denote the set of vertices incident to some edge of B. We will say that B is *transitively closed* if $(u, v) \in B$ and $(v, w) \in B$, imply $(u, w) \in B$. For any $v \in V$ we denote by $\delta_B^+(v)$ (resp. $\delta_B^-(v)$) the set of arcs in B with head (resp. tail) v. $\delta^+(v) = \delta_A^+(v)$ and $\delta^-(v) = \delta_A^-(v)$.

2. Motivation from application

Classical routing models like the Vehicle Routing Problem (VRP) or the Pickup and Delivery Problem (PDP) [13,18,19] assume that goods are transported from supplier to customer by a single truck. In contrast, in this paper we are interested in problems where it is customary to reload goods along the way, i.e. goods can be dropped off at so called hubs and then hauled on by another means of transportation. These strategies are common in large shipping operations, e.g. postal services or air lines.

Due to the complexity of these types of logistics systems, these problems are often addressed on the strategic level in optimization.

Most notably in this area are the so called *Facility Location Problems* [5,15]. Instances of these problems consist of a network containing a set of customers and possible locations for the hubs. The goal is to select a set of hubs and assign each customer to a hub, so that capacity constraints on each hub can be satisfied and some given cost function is minimized. Among these problems are the *n-median* and *n-center* problems, where n locations with minimum average resp. maximum distance to their assigned customers must be found.

The *Hub Location Problem* (HLP), e.g. [17], is an extension of the above. Here, the demand consists of transportation requests between the customer locations. Again, a suitable set of hubs and assignments of customer locations to hubs is to be selected. While in the Facility Location Problems cost is mainly determined by the distance of the customer to its associated hub, here the transportation cost of the requests is considered in the cost function. This cost is assumed to be linear in the amount of transported goods.

Guelat et al. [9] consider a network design problem, where an origin-destination matrix states the supply resp. demand at each customer location for a number of different goods. In the (not necessarily simple) network each arc represents transportation by different available carriers. The aim is to find a multi-commodity flow that minimizes total transportation cost. The problem is made more complex by constraints on the path decomposition of the solution. These constraints state that reloads are allowed only at hubs and may forbid usage of certain carriers by certain goods on a given link.

However, these problems find their application in strategic planning. Our goal was to work with problems that can arise in daily operations. Therefore, we assume that the locations of reload hubs are predetermined.

Grünert et al. [8] present the *Vehicle and Request Flow Network Design Problem* (VRFNDP), a detailed model intended for tactical planning tasks in letter mail transportation. The routing of requests is represented by a multi-commodity network flow that must be covered by the tours of the vehicles. Since this model has been developed with a particular application in mind, all relevant constraints for operational planning have been taken into account, i.e. letters have to arrive at a hub in good time, so sorting and reloading can take place before they are forwarded to the next hub or the customer. The model that is presented shortly can be considered a relaxation of the VRFNDP [16].

To make reload problems more accessible to combinatorial analysis, we decided to relax constraints on truck capacity and start and end points of truck routes. In such a setting, feasibility and an estimate of the solution cost can be determined without knowing the actual truck routes, since—provided the solution is acyclic—we can cover the used arcs with chains, yielding a routing. Acyclicity is necessary to avoid a deadlock situation, where trucks wait indefinitely for a request to arrive at a certain hub [16].

Problems of this nature may arise when, e.g. vehicle spare parts have to be delivered quickly to facilitate repairs for immobilized vehicles. In these instances, truck capacity is not so much an issue, but it is still desirable to consolidate transportation as much as possible.

For the sake of simplicity, in this paper reloading will be allowed at any vertex of the network, which results in the Steiner Diagram Problem (SDP). This problem proved to be an interesting variation of the Generalized Directed Steiner Network Problem (GDSNP) [3]. Optimum solutions of the SDP have some nice structural properties that those of a GDSNP do not have in general. The results indicate that the number of possible hubs in such a reload problem has no impact on the theoretical complexity of the problem. It is possible to refine the definition of the SDP in order to restrict reload actions to designated vertices only. Details can be found in [16].

The k-Star Hub Problem (k-SHP) was derived for certain instances of our transportation problem, where a small number of hubs has been installed in advance and one can choose between transporting goods directly or via a hub. Savings can be accomplished, if several requests start or end at the same node and are transported via the same hub, since assuming sufficient transportation capacity, each link must be established and paid only once.

In cooperation with a car manufacturer we have optimized instances with one preinstalled hub. Here an optimum solution for the 1-SHP was used to decide whether a demand should be routed directly or through a hub, reducing the problem to a classical PDP. Then an initial solution was constructed. This solution was further improved by a tabu search heuristic [7] that allows local changes of the transportation strategy. This strategy yields solutions whose cost is lower than those of previously known solutions by 5 to 10 percent [10,16].

3. The Steiner Diagram Problem

3.1. Definition and basic classification

Problem 1 *(Steiner Diagram Problem).* INPUT: Let (V, A, w) be a digraph with nonnegative arc weights $w : A \to \mathbb{Z}^+$ and B a set of (directed) node pairs.

OBJECTIVE: We say that a set $S \subseteq A$ is *spanning* for B, if there is a directed path in S for each pair in B. If S is spanning and acyclic, we call it *feasible*. Find a feasible set S such that $w(S) = \sum_{a \in S} w(a)$ is minimized. Such an S we call a *Steiner diagram*.

As we consider only nonnegative arc weights, we may assume that S is a Hasse diagram, i.e. an acyclic digraph without transitive edges (hence the name Steiner *Diagram* Problem). Note that there need not exist a feasible solution in every instance of the SDP, trivially so if the set of directed (demand) pairs is not acyclic.

As mentioned in the introduction, the acyclicity requirement is derived from applications in vehicle routing since only an acyclic solution can be translated into a feasible vehicle routing schedule. A thorough discussion of this requirement and its motivation can be found in [16].

The SDP is a generalization of the so-called Steiner Arborescence Problem (SAP):

Problem 2 *(Steiner Arborescence Problem [11]).* INPUT: Let (V, A, w) be a complete digraph with nonnegative arc weights, $t_0 \in V$ a designated *root node* and $T \subseteq V$ a set of *terminals*.

OBJECTIVE: Find a subset $S \subseteq A$ that contains a path from t_0 to each $t \in T$, such that $w(S) = \sum_{a \in S} w(a)$ is minimized.

Acyclicity, automatically guaranteed in the latter problem, must be required explicitly for a Steiner Diagram Problem. As a generalization of the SAP the SDP is $\mathcal{NP}$-complete, as well [11]. Yet, when there are no *Steiner nodes*, i.e. vertices that are neither root nor terminal nodes, the Steiner Arborescence Problem reduces to a Minimum Spanning Arborescence Problem, and thus is polynomially solvable.

Without the acyclicity condition the Steiner Diagram Problem is known as Generalized Directed Steiner Network Problem (GDSNP) [3].

A special version of the GDSNP is the Directed Steiner Network Problem (DSNP) [20]. In the DSNP one has to find an "equivalent" subgraph for a given vertex set $T \subseteq V$. So, if in (V, A) there is a path from one vertex in T to another, such a path will be in the subgraph as well. Whenever two vertices are not connected by a path in the original graph, they will be disconnected in the subgraph as well. In the GDSNP, not all the connections for a given vertex set have to be retained, but only those that are prescribed by B.

Surveys of many different kinds of Steiner Problems on both directed and undirected graphs can be found in [11,20].

3.2. Complexity of the SDP

In contrast to the SAP, the DSNP remains hard even if $T = V$ [20]. From this fact it follows easily that the GDSNP is $\mathcal{NP}$-complete even if $V[B] = V$. The same holds for the SDP, as will be proved below. Since the graph used in the proof is acyclic, this proves the fact for the GDSNP as well.

Theorem 3. *The Steiner Diagram Problem is $\mathcal{NP}$-complete even when $V[B] = V$, A is transitively closed and the triangle inequality holds in G.*

Proof. Obviously, the SDP is in NP. To prove hardness, a reduction from SAT will be used [12]. Let $c_1, \dots, c_n$ be the clauses and $v_1, \dots, v_m$ the variables in an instance of SAT.

We construct an instance of the SDP that has a solution of cost at most $2m^2 + m$ if and only if the SAT instance is satisfiable.

For each clause c_i define $C_i = \{c_i^S, c_i^E\}$, for each variable v_j $V_j = \{v_j^S, v_j^E, v_j^Y, v_j^N\}$, let the vertex set be

$$V = \bigcup_{i \leqslant n} C_i \cup \bigcup_{j \leqslant m} V_j.$$

For each variable v_j, we define the arcs:

$$A_j^V = \{(v_j^S, v_j^Y), (v_j^Y, v_j^E), (v_j^S, v_j^N), (v_j^N, v_j^E)\}$$

constituting the truth setting components (see Fig. 1). For satisfaction testing we put for each clause C_i

$$A_i^C = \{(c_i^S, v_j^S) \mid v_j \in c_i\} \cup \{(v_j^Y, c_i^E) \mid v_j \text{ is positive in } c_i\} \\ \cup \{(v_j^N, c_i^E) \mid v_j \text{ is negative in } c_i\}.$$

The arc set A is defined as the minimal transitively closed superset of $(\bigcup_{i \leqslant n} A_i^C \cup \bigcup_{j \leqslant m} A_j^V)$. Note that (c_i^S, c_i^E) and (v_j^S, v_j^E) are arcs of the graph for each i and j.

The weight function w on A is defined as follows:

$$w(a) = \begin{cases} 0 & \text{if } a = (c_i^S, v_j^S) \text{ and } v_j \text{ occurs in } c_i, \\ 0 & \text{if } a = (v_j^Y, c_i^E) \text{ and } v_j \text{ is positive in } c_i, \\ 0 & \text{if } a = (v_j^N, c_i^E) \text{ and } v_j \text{ is negative in } c_i, \\ 1 & \text{if } a = (v_j^S, v_j^Y) \text{ or } a = (v_j^S, v_j^N), \\ m & \text{if } a = (v_j^Y, v_j^E) \text{ or } a = (v_j^N, v_j^E). \end{cases}$$

All other arcs have induced cost, i.e. the cost of the shortest path from source to tail, thus the triangle inequality holds. Note that each (c_i^S, c_i^E) has a cost of 1, each (v_j^S, v_j^E) one of $m + 1$.

Finally, we set

$$B^V = \bigcup_{j \leqslant m} \{(v_j^S, v_j^E), (v_j^Y, v_j^E), (v_j^N, v_j^E)\}$$

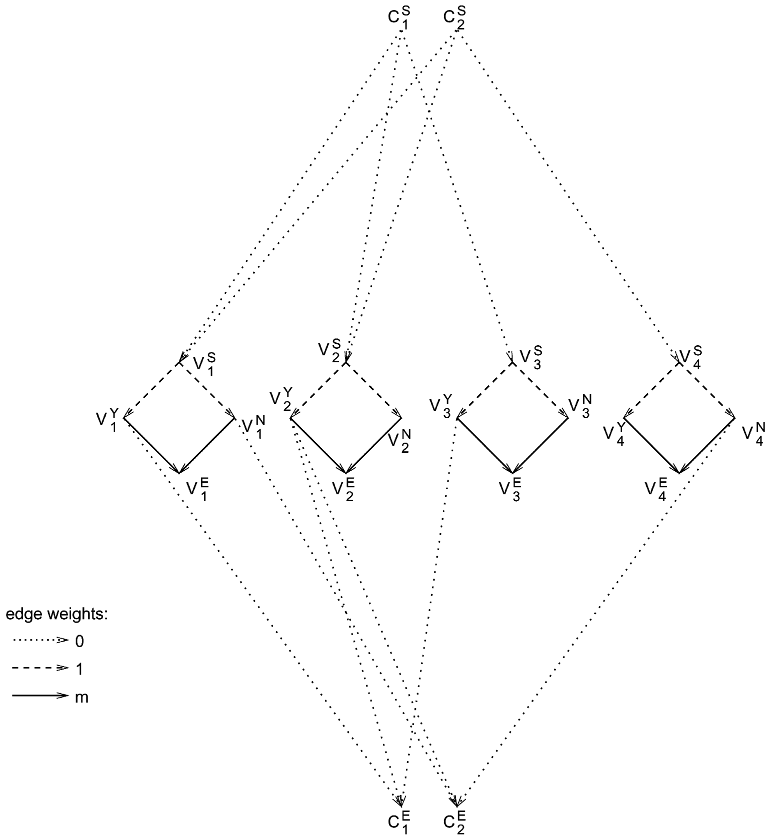

Fig. 1. Relevant edges in the construction for $(v_1 \vee v_2 \vee v_3) \wedge (\neg v_1 \vee v_2 \vee \neg v_4)$.

for the truth setting component and

$$B^C = \bigcup_{i \leqslant n} \{(c_i^S, c_i^E)\}$$

for the satisfaction testing and let $B = B^V \cup B^C$ be the demand set.

Observe that for each j any solution must contain both (v_j^Y, v_j^E) and (v_j^N, v_j^E), giving rise to a cost of $2m^2$. Therefore a solution of cost no more than $2m^2 + m$ must also contain exactly one of (v_j^S, v_j^Y) or (v_j^S, v_j^N). The latter choice determines a truth assignment for the variables of the SAT instance. No other arcs with non-zero cost can be in such a solution. Thus, any path for the demand (c_i^S, c_i^E) must pass through the truth setting component for a variable $v_j \in c_i$, using (v_j^S, v_j^Y) or (v_j^S, v_j^N). Similarly, we get a Steiner diagram of cost $2m^2 + m$ from any satisfying truth assignment. □

3.3. Structural properties of Steiner diagrams

The purpose of this section is to describe and prove some properties of Steiner diagrams that imply the existence of a polynomial time algorithm if the size of B is bounded, A is transitively closed, and the triangle inequality holds. These conditions hold naturally in the setting of our applications.

We show that the number of *junctions*, i.e. vertices with at least two entering arcs, of a minimally feasible Steiner diagram is bounded by the square of the size of B, this result is independent of the network topology. If the triangle inequality holds and A is transitively closed, there is an optimum solution S such that any vertex incident to an arc in S is either incident to a demand node pair or a junction or has at least two leaving arcs. Thus, by symmetry, if $|B|$ is a constant, an optimum solution visits only a constant number of vertices. This gives a polynomial bound on the number of possible vertex sets in a solution. By enumerating the Hasse diagrams on these sets we get the desired result. Note that if A is not transitively closed, computing shortest paths between the junctions, might yield directed circuits in the union of the paths.

We will first sketch the ideas of the proof: We say that a directed path P in a solution S *satisfies* the demand $(u, v) \in B$, if P contains a u–v-path. Although in an optimum solution the path satisfying a given demand need not be unique, for each arc $a \in S$ there is some demand $b \in B$ such that a is on every path satisfying b (Proposition 6). Thus, for every junction there are two demands, such that any two paths satisfying these two demands enter the vertex through two different arcs. Due to the acyclicity of the solution this happens at most once for any pair of demands. Therefore each junction can be uniquely identified by any one such pair (Lemma 8). This bounds the number of junctions from above by the number of possible pairings $\binom{|B|}{2}$. We tighten this bound by seeing that for three demands, at most two of the three possible pairings can be joined by different hubs. This is essentially due to the fact that the paths can be chosen in such a way that the third pairing happens in one of the other two junctions as well (Lemma 9).

Definition 4. Let $G = (V, A, w)$, B be an instance of the SDP and $S \subseteq A$ be a feasible solution. The set of *junctions* $\mathcal{J}_S^+$ of S is defined as

$$\mathcal{J}_S^+ = \{v \in V \mid \delta_S^+(v) \geqslant 2\}.$$

We will say that a set S is *minimally feasible*, if removing any arc from S causes it to be infeasible.

As mentioned above, the following observation is the crucial one for our algorithm:

Theorem 5. *Let S be a minimally feasible solution for an instance $G = (V, A, w)$, B of the Steiner Diagram Problem. Then the number of junctions of S is bounded from above by $\frac{1}{4}|B|^2$, i.e. $|\mathcal{J}_S^+| \leqslant \frac{1}{4}|B|^2$.*

To prove this theorem we need some preparation. Any arc used in the solution must serve a purpose, i.e. there must be a demand that can only be routed via this arc:

Proposition 6. *Let $S \subseteq A$ be minimally feasible for a Steiner Diagram Problem and $a \in S$. Then there exists $b \in B$, such that a is contained in any path in S satisfying b.*

Proof. Assume for a contradiction that there is some $a \in S$ such that for all b there is some path avoiding a. Then $S \setminus \{a\}$ is still feasible, contradicting minimality of S. □

Thus, we can label the vertices by their entering demands:

Definition 7. Let S be a feasible set for a Steiner Diagram Problem $G = (V, A, w)$, B. For any arc $a \in S$, let its *(arc) label* $\mu(a) \subseteq B$ denote the set of demand node pairs b, such that a is on any path in S satisfying b. For $v \in V$ we define its *(node) label* as $\lambda(v) = \bigcup_{a \in \delta_S^+(v)} \mu(a)$.

By the preceding Proposition 6 a label $\lambda(v)$ is empty if and only if $\delta_S^+(v) = \emptyset$.

The next two lemmas establish the properties that give our bound on the number of junctions. The first one states that two paths satisfying different demands can enter at most one common vertex through different arcs.

Lemma 8. *Let S be a minimally feasible set for a Steiner Diagram Problem $G = (V, A, w)$, B. If $v \neq w \in V$, $b_1 \neq b_2 \in B$ and $\{b_1, b_2\} \subseteq \lambda(v) \cap \lambda(w)$, then there exists an arc $a = (r, u)$ such that $u \in \{v, w\}$ and $\{b_1, b_2\} \subseteq \mu(a)$.*

Proof. Any two paths P_1, P_2 satisfying b_1 resp. b_2 both visit v and w. As S is acyclic, they must visit them exactly once and in the same order, say v precedes w. Assume, P_1 and P_2 enter w through different arcs a_1, a_2. But then, there exists a path satisfying b_1 that does not use a_1, a contradiction. □

This means that any choice of two demands that enter the vertex through different arcs defines an injection from the set of junctions into $\binom{B}{2}$, the set of all pairs in B. This already implies a bound of $\frac{1}{2}|B|^2$ on the number of junctions.

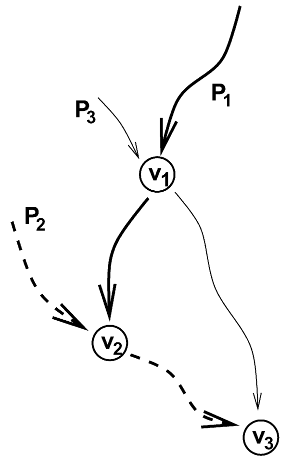

Fig. 2. Illustration of the situation in Lemma 9.

The stronger bound in Theorem 5 is achieved, because for any triple $\{b_1, b_2, b_3\} \subseteq B$ the image of such an injection contains at most two of the three possible pairings. This is implied by

Lemma 9. *Let S be a minimally feasible set for a Steiner Diagram Problem $G = (V, A, w)$, B. For any three nodes $v_1, v_2, v_3 \in V$ we have*

$$\big(\forall 1 \leqslant i < j \leqslant 3\colon\ \lambda(v_i) \cap \lambda(v_j) \neq \emptyset\big) \Rightarrow \bigcap_{k=1}^{3} \lambda(v_k) \neq \emptyset.$$

Proof. Suppose to the contrary that there are $v_1, v_2, v_3 \in V$, such that

$$b_1 \in \lambda(v_1) \cap \lambda(v_2), \qquad b_2 \in \lambda(v_2) \cap \lambda(v_3), \qquad b_3 \in \lambda(v_1) \cap \lambda(v_3)$$

while $\bigcap_{k=1}^{3} \lambda(v_k)$ contains neither of b_1, b_2, b_3 (see Fig. 2). Let P_1, P_2 and P_3 be paths satisfying b_1, b_2 resp. b_3. These paths define a total order on v_1, v_2, v_3, we may assume $v_1 < v_2 < v_3$ and that v_3 has been chosen at shortest possible distance from v_1 wrt P_3, so P_2 and P_3 enter v_3 through different arcs a_2, a_3 and $b_3 \in \mu(a_3)$. Thus, we can replace the v_1, v_3 segment of P_3 by the v_1, v_2 segment of P_1 and the v_2, v_3 segment of P_2 yielding a path satisfying b_3 not using a_3, a contradiction. □

Proof of Theorem 5. We define a graph $H = (B, F)$ on the vertex set B by adding one edge (b_1, b_2) to F for each $v \in \mathcal{J}_S^+$, selecting $b_1 \neq b_2 \in \lambda(v)$ such that there are $a_1 \neq a_2 \in \delta_S^+(v)$ and $b_1 \in \mu(a_1)$ and $b_2 \in \mu(a_2)$; i.e. b_1 and b_2 label different arcs entering v. By the definition of $\mathcal{J}_S^+$ such b_1 and b_2 can always be found.

Since one edge is defined for each junction in S, this creates a surjective mapping from $\mathcal{J}_S^+$ into F. By Lemma 8 and the remark immediately following it, this mapping is also injective, giving that $|\mathcal{J}_S^+| = |F|$.

By Lemma 9 and its preceding comment H contains no triangles. Turan's Theorem (e.g. [14, §10.30]) states that a triangle free graph has at most $\frac{1}{4}|B|^2$. Thus,

$$|\mathcal{J}_S^+| = |F| \leqslant \frac{1}{4}|B|^2. \qquad \square$$

Remark 10. By symmetry we have as well

$$|\mathcal{J}_S^-| = \big|\{v \in V \mid \delta_S^-(v) \geqslant 2\}\big| \leqslant \frac{1}{4}|B|^2.$$

For the structural properties until now we have only used the minimality of S.

In order to derive the following algorithm we have to ensure that a solution need only contain nodes incident to demands and junctions. Therefore, we require that the triangle inequality holds in the underlying graph and that it is transitively closed.

Corollary 11. *Let N be an integer. Let SDP(N) denote the class of Steiner Diagram Problems $G = (V, A, w)$, B where $|B| \leqslant N$, with A transitively closed and G satisfying the triangle inequality. Then SDP(N)$\in \mathcal{P}$.*

Proof. Let S be a solution for a Steiner Diagram Problem $G = (V, A, w)$, B.

Let $v \in V[S] \setminus V[B]$. If $\delta_S^+(v) = 0$ or $\delta_S^-(v) = 0$, then the arcs in S incident to v can be removed, yielding a solution S' of the SDP. If $\delta_S^+(v) = 1$ and $\delta_S^-(v) = 1$, then there are $u, w \in V$ such that $(u, v), (v, w) \in S$. Since G is transitively closed $S' = (S \setminus \{(u, v), (v, w)\}) \cup \{(u, w)\}$ is a solution of the SDP.

By repeatedly applying these two steps, we get a solution S_0, such that no vertex $v \in V[S_0] \setminus V[B]$ has both $\delta_{S_0}^+(v) \leqslant 1$ and $\delta_{S_0}^-(v) \leqslant 1$. By the triangle inequality the cost of S_0 does not exceed the cost of S.

By Theorem 5 and Remark 10 there is a minimum cost solution such that $|V[S] \setminus V[B]| = |\mathcal{J}_S^+ \cup \mathcal{J}_S^-| \leqslant \frac{1}{2}|B|^2$. Obviously, the number of posets on a given set of at most $\frac{1}{2}|B|^2 + 2|B|$ vertices is a constant.

Therefore, we can find a solution using the following "algorithm":

for all candidate sets V' ($|V'| \leqslant \frac{1}{2}|B|^2$):
  for all Hasse diagrams on $V' \cup V[B]$:
    if $A' \subseteq A$:
      compute the cost of A';
choose the solution with minimum cost;

Since the number of candidate sets is bounded by $\sum_{i=1}^{\frac{1}{2}|B|^2+2|B|} \binom{|V|}{i}$, which is a polynomial in $|V|$ if $|B|$ is a constant, the algorithm runs in polynomial time. □

Whenever one of the additional conditions on G is dropped, there are instances where this algorithm fails. By the transitive closedness and the triangle inequality the arc connecting any two vertices is a shortest path as well. In graphs without one of those properties finding node-disjoint shortest paths to avoid circuits may become an issue.

Remark 12. Note that Lemma 8 fails for the GDSNP, since it uses acyclicity of the solution. It is easy to construct an instance of GDSNP where the optimum solution is a directed cycle.

Finally, we note that our bound on the number of junctions is tight:

Theorem 13. *There is a class of Steiner Diagram Problems, whose graph is transitively closed and where the triangle inequality holds, where an optimum solution contains $\frac{1}{4}|V[B]|^2$ junctions.*

Proof. The main idea is to construct a solution S for a set of n independent demand node pairs $B = \{(s_1, t_1), \ldots, (s_n, t_n)\}$, such that two paths satisfying (s_i, t_i), (s_j, t_j) meet in node v_{ij}^+ if and only if i and j have different parity.

Let n be even. Our vertex set consists of

$$V = \{s_1, \ldots, s_n\} \cup \{t_1, \ldots, t_n\} \cup \{v_{ij}^+ \mid i \text{ odd}, j \text{ even}\} \cup \{v_{ij}^- \mid i \text{ odd}, j \text{ even}\}.$$

In the following we present the arc set S supposed to form the solution. The optimality of this arc set is guaranteed by defining their arc weights as 1 and assigning all the transitive arcs the induced weight.

We have three different types of arcs

$$R = \{(s_i, v_{1i}^+) \mid i \text{ even}\} \cup \{(s_i, v_{i2}^+) \mid i \text{ odd}\}, \qquad T = \{(v_{in}^-, t_i) \mid i \text{ odd}\} \cup \{(v_{n-1,i}^-, t_i) \mid i \text{ even}\},$$
$$I = \{(v_{ij}^+, v_{ij}^-) \mid i \text{ odd}, j \text{ even}\} \cup \{(v_{ij}^-, v_{i,j+2}^+) \mid i \text{ odd}, j \text{ even}\} \cup \{(v_{ij}^-, v_{i+2,j}^+) \mid i \text{ odd}, j \text{ even}\}.$$

Now $S = R \cup T \cup I$, the situation is depicted in Fig. 3 for small n.

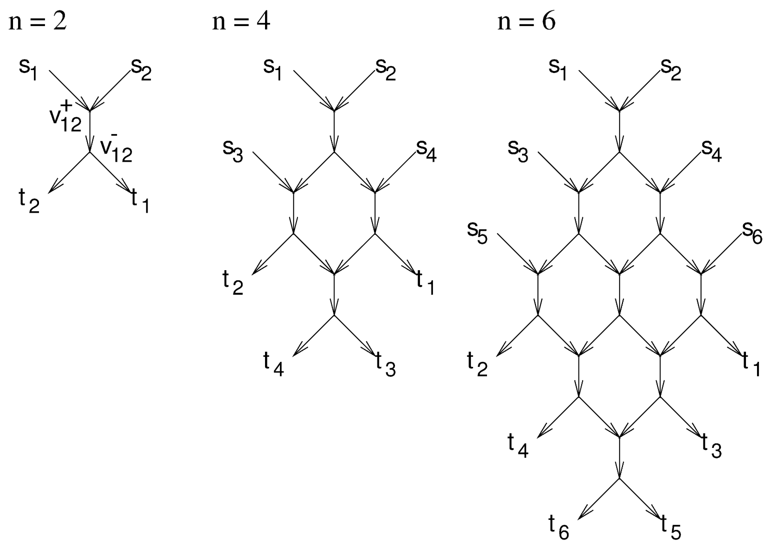

Fig. 3. The graph for up to 6 demand pairs.

To see that S is indeed a Steiner diagram, let $s_i =: v^-_{-1,i}$ for i even, $s_i =: v^-_{i,0}$ for i odd, $t_i =: v^+_{n+1,i}$ for i even and $t_i =: v^+_{i,n+2}$ for i odd. Now, it is immediate that no index ever decreases along a directed arc and that the unique (s_i, t_i) path in S is given as the graph induced by all vertices with index i. Furthermore,

$$\sum_{v\in S}\binom{\delta^+_S(v)}{2} = \sum_{v\in S}\binom{\delta^-_S(v)}{2} = \frac{1}{4}n^2.$$

With an analogous construction for odd n, we get a bound of Theorem 5 of [2]. □

4. The *k*-Star Hub Problem

The SDP is a combinatorial formulation of reload problems. It models general routing and reloading strategies. Thus, it must be $\mathcal{NP}$-hard because of the routing aspect alone. Therefore, a problem was derived that does not have a routing aspect and concentrates on the decision whether to reload a given request. In addition, it is often not viable to reload a request arbitrarily.

For example in the automotive industry spare parts delivery to facilitate repairs of immobilized vehicles can be time critical and parts must be supplied overnight. These deliveries are low volume and sourced from a central depot, but also from several secondary locations in the case of special parts.

The supply chain in this case provides several hub locations where parts can be consolidated; alternatively they can be delivered directly to the repair shop. The decision whether to transport via one of the hubs or directly will be based on the specific demand on a particular day. However, requests will never be reloaded more than once to keep the logistics operations manageable.

In the case of these express deliveries, vehicle capacity usually is not a limiting factor, since the volumes are not high enough. Any vehicle transporting to or from a hub versus a direct delivery will be able to support the necessary volumes, so it must only be decided which transportation links to establish to deliver the parts.

The model derived for this setting is called k-Star Hub Problem. A set of requests is given, that can have common stops for pickup or delivery, and k hubs, where reload actions can take place. Each request can either be delivered directly, incurring a given fee, or it can be taken to a hub and then carried on to the delivery stop. In the latter case, a link between the pickup stop and the hub and also a link between the hub and the delivery stop has to be paid for. If a link has been established between a stop and a hub, all requests that have this stop in common can use it as well without additional cost.

This can also be interpreted as a special case of the SDP where the underlying graph contains nodes originating and receiving demands and the hubs. The arcs in this graph connect the originating nodes to the hubs and the receiving nodes and the hubs to the receiving nodes. In our formulation of the problem weight functions on the nodes take the place of hubs.

Problem 14 *(k-Star Hub Problem).* INPUT: Given a graph $G = (V, E)$, a nonnegative integer weight function $w_0 : E \to \mathbb{Z}^+$ on the edges and k nonnegative integer weight functions on the vertices $w_1, \dots, w_k : V \to \mathbb{Z}^+$.

OBJECTIVE: We will say $V' \subseteq V$ *satisfies* an edge $e = (u, v)$ if $\{u, v\} \subseteq V'$. Find a set of edges $F \subseteq E$ and k subsets of the vertices $V_1, \dots, V_k \subseteq V$ such that for all $e = (u, v) \in E$ either $e \in F$ or some V_i satisfies e, and

$$\sum_{e \in F} w_0(e) + \sum_{i=1}^{k} \sum_{v \in V_i} w_i(v)$$

is minimized.

In this problem each edge corresponds to a demand with an associated transportation cost.

To see that this can be transformed into a SDP, add a node for each of the k weight function on the nodes. The weights on the nodes in the graph can be seen as the cost of edges connecting the node to one of k hubs. A solution determines for each edge (u, v), whether its associated demand is transported directly ($(u, v) \in F$) or via a hub ($\{u, v\} \subseteq V_i$).

4.1. Hardness of the 3-SHP

It will be demonstrated that the k-SHP is solvable in polynomial time if there are at most two cost functions, but $\mathcal{NP}$-complete for $k \geqslant 3$.

The latter statement follows from a result due to Dahlhaus, Johnson, Papadimitriou, Seymour and Yannakakis [4] that the following problem, a generalization of the Min-Cut Problem, is $\mathcal{NP}$-complete for fixed $k \geqslant 3$:

Problem 15 *(Multiterminal Cut Problem—MCP).*

Instance Given a graph (V, E), a set $S = \{s_1, \dots, s_k\} \subseteq V$ of *terminals* and a positive weight function $w : E \to \mathcal{N}$ on the edges.

Question Find a minimum weight set of edges $E' \subseteq E$, such that $(V, E \setminus E')$ has k components, each containing exactly one terminal.

Theorem 16. *The 3-Star Hub Problem is $\mathcal{NP}$-complete.*

Proof. Obviously, the 3-SHP is in NP.

Let (X, F), $S = \{s_1, s_2, s_3\} \subseteq X$, $w : F \to \mathcal{N}$ be an instance of MCP with three terminals and $X = \{x_1, \dots, x_n\}$.

We construct an instance of 3-SHP such that the MCP instance has a solution of cost not exceeding C, if and only if the constructed 3-SHP instance has a solution of cost not exceeding $L + C$.

The idea of the construction is to make each vertex set of a solution of the 3-SHP translate into one component of a solution of MCP. Adding a leaf to each vertex of (X, F) and appropriate cost functions on the leaf edges ensure that each vertex is assigned to exactly one vertex set and each vertex set contains exactly one terminal.

Let $U = \{u_i \mid x_i \in X\}$ be a copy of X and put $V = X \cup U$. Also, let $D = \{(x_i, u_i) \mid x_i \in X\}$ be the edges connecting the leaves, then $E = F \cup D$. Together, this defines the graph of the instance (V, E).

Let $K = \sum_{e \in F} w(e) + 1$ and $L = |V|K$. The weight function on the edges $e \in E$ then is:

$$w_0(e) = \begin{cases} w(e) & \text{if } e \in F, \\ L + K + 1 & \text{if } e \in D. \end{cases}$$

The weight functions w_1, w_2, w_3 on the vertices $v \in V$ are given as follows:

$$w_i(v) = \begin{cases} K & \text{if } v \in (X \setminus S) \cup \{s_i\}, \\ L + K + 1 & \text{if } v \in S \setminus \{s_i\}, \\ 0 & \text{if } v \in U. \end{cases}$$

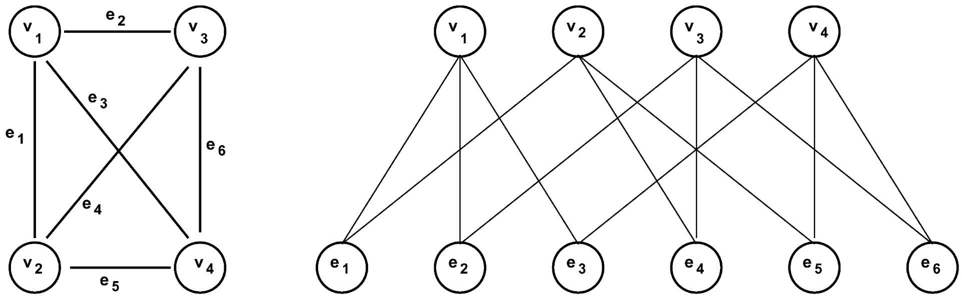

Fig. 4. Subdividing the edges in the 1-SHP yields a vertex cover problem.

A given solution of MCP of cost C defines a partition of X into three components. Let each of these components, along with their adjoint leaf nodes, define one of V_1, V_2, V_3. This solution of the 3-SHP has a cost of $|V|K = L$ for the vertex sets combined and C for the edges.

On the other hand, let V_1, V_2, V_3 be the vertex sets of a solution of the 3-SHP of cost $L + C$. Since C is the maximum cost for any solution of the corresponding MCP instance, we assume $C < K$. All edges in D must be covered, so we have $X \subseteq V_1 \cup V_2 \cup V_3$. Also, $s_i \in V_i$ $(i = 1, 2, 3)$ and any node in X is in at most one node set, otherwise a minimum cost of at least $L + K$ would be incurred.

Thus, we have a partition of X and the cost of V_1, V_2, V_3 combined is L. The edges not satisfied by one of V_1, V_2, V_3 define a multi-terminal cut of cost no more than C. □

Trivially, the above theorem implies hardness of the k-SHP with $k \geqslant 3$.

4.2. *Tractability of the* 2*-SHP*

Now, we proceed to show that the problem is polynomially solvable if $k \leqslant 2$. The 1-SHP is seen to be a bipartite vertex cover problem by subdividing all edges (see Fig. 4). For each edge in the original graph a vertex cover in the subdivided graph will contain either the node added by subdividing the edge or both nodes adjacent to it.

While the vertex cover problem is $\mathcal{NP}$-complete on general graphs [6], the dual of a bipartite weighted vertex cover problem is a bipartite B-matching problem (e.g. [1]). To solve this kind of problem we can model it as a network flow problem. By max flow–min cut duality a weighted vertex cover then corresponds to a minimum cut in this network.

Following this idea we construct a network which can be used to solve the problem in the case $n = 2$. Let V' and V'' be two isomorphic copies of V and E' and E'' two copies of E. Let the node set of our network be $N = V' \cup V'' \cup E' \cup E'' \cup \{s, t\}$, where s and t are the source and sink node respectively.

Put $A^{V'} = \{(s, v') \mid v' \in V'\}$, $A^{V''} = \{(v'', t) \mid v'' \in V''\}$, $A^E = \bigcup_{e_j \in E}(e'_j, e''_j)$. For each $e = \{u, v\} \in E$ put $A'_e = \{(u', e'), (v', e')\}$ and $A''_e = \{(e'', u''), (e'', v'')\}$. Then let the arc set of the network be (see Fig. 5)

$$A = A^{V'} \cup \bigcup_{e \in E} A'_e \cup A^E \cup \bigcup_{e \in E} A''_e \cup A^{V''}.$$

The capacity c of these arcs is defined by the following function:

$$c(a) = \begin{cases} w_1(v) & \text{if } a = (s, v') \in A^{V'}, \\ w_0(e) & \text{if } a = (e', e'') \in A^E, \\ w_2(v) & \text{if } a = (v'', t) \in A^{V''}, \\ \infty & \text{else.} \end{cases}$$

For this network a minimum cut corresponds to an optimum solution for the 2-SHP, as follows:

Clearly a minimum cut cannot contain an arc of unlimited capacity. Then for each arc $a = (e'_j, e''_j)$ with $e_j = (u, v)$ either a or (u'', t) as well as (v'', t) or (s, u') as well as (s, v') must be cut. Thus a min-cut corresponds to a feasible solution of the 2-SHP.

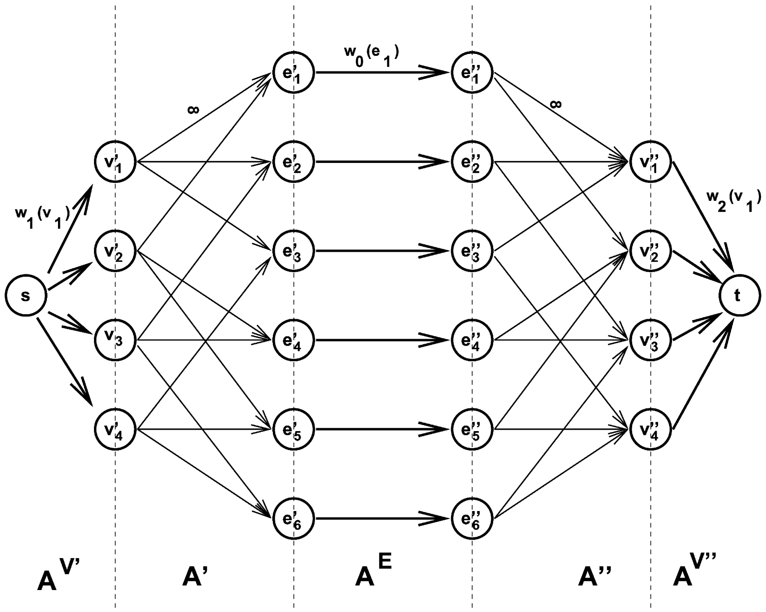

Fig. 5. The network for a 2-SHP with underlying graph as in Fig. 4 with two cost functions.

On the other hand a solution F, V_1, V_2 of the 2-SHP defines an arc set

$$C = \bigcup_{e \in F} (e', e'') \cup \bigcup_{v \in V_1} (s, v') \cup \bigcup_{v \in V_2} (v'', t).$$

For any edge $e_j = (u, v)$ we must have either $e_j \in F$, $\{u, v\} \subseteq V_1$ or $\{u, v\} \subseteq V_2$. Any directed path from s to t passes exactly one arc (e'_j, e''_j) with $e = (u, v)$ and one of $\{u', v'\}$ and one of $\{u'', v''\}$, thus is cut by C.

Remark 17. Note that this method can be generalized to any pair of hyper-graphs with a bijection on their edge sets.

5. Conclusion

The Steiner Diagram Problem is a generalization of the Steiner Arborescence Problem. We have shown that the SDP is $\mathcal{NP}$-complete even without Steiner nodes, while in this case the latter problem reduces to the computation of a minimum spanning arborescence. Still, we can show that the SDP is polynomial for a constant size of B. Unfortunately, our algorithm is of a theoretical nature and finding a reasonable one as well as one for arbitrary graphs is left as an open problem.

Furthermore, we have given complexity results for the k-Star Hub Problem. For this problem we have shown that it is polynomially solvable for $k \leqslant 2$ but $\mathcal{NP}$-complete for $k \geqslant 3$.

References

[1] R. Ahuja, T. Magnanti, J. Orlin, Network Flows, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] B. Bollobás, Graph Theory, Springer, Berlin, 1979.

[3] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, M. Li, Approximation algorithms for directed Steiner problems, J. Algorithms 33 (1) (1999) 73–91.

[4] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour, M. Yannakakis, The complexity of multiterminal cuts, SIAM Journal of Computing 23 (4) (1994) 864–894.

[5] R.L. Francis, L.F. McGinnis, J. White, Locational analysis, European Journal of Operational Research 12 (1983) 220–252.

[6] M. Garey, D.S. Johnson, Computers and Intractability, W.H. Freeman and Company, 1979.

[7] F. Glover, M. Laguna, Tabu Search, Kluwer, Boston, 1997.

[8] T. Grünert, H.-J. Sebastian, Planning models for long-haul operations of postal and express shipment companies, European Journal of Operational Research 122 (2) (2000) 289–309.

[9] J. Guelat, M. Florian, T.G. Crainic, A multimode multiproduct network assignment model for strategic planning of freight flows, Transportation Science 24 (1990) 25–39.
[10] W. Hochstättler, C. Mues, P. Oertel, Algorithmen für Speditionsroutingprobleme mit Umlademöglichkeit, Tagungsband der 6, Fachtagung Logistik, Magdeburg, 2000.
[11] F.K. Hwang, D.S. Richards, P. Winter, The Steiner Tree Problem, Annals of Discrete Mathematics, North-Holland, Amsterdam, 1992.
[12] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, 1972, pp. 85–103.
[13] G. Laporte, I.H. Osman, Routing problems: A bibliography, Annals of Operations Research 61 (1995) 227–262.
[14] L. Lovász, Combinatorial Problems and Exercises, North-Holland, Amsterdam, 1993.
[15] P. Mirchandani, R. Francis (Eds.), Discrete Location Theory, John Wiley and Sons, New York, 1990.
[16] P. Oertel, Routing with reloads, PhD thesis, Universität zu Köln, 2000.
[17] J. Sohn, S. Park, Efficient solution procedure and reduced size formulations for p-hub location problems, European Journal of Operational Research 108 (1998) 118–126.
[18] M.W.P. Savelsbergh, M. Sol, The general pickup and delivery problem, Transportation Science 29 (1995) 17–29.
[19] P. Toth, D. Vigo (Eds.), Vehicle Routing, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, PA, 2002.
[20] P. Winter, Steiner problems in networks, Networks 17 (1987) 129–167.