Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

DANIEL REHFELDT, THORSTEN KOCH

Transformations for the Prize-Collecting Steiner Tree Problem and the Maximum-Weight Connected Subgraph Problem to SAP

ZIB Report 16-36 (June 2016)

Zuse Institute Berlin
Takustr. 7
D-14195 Berlin

Telefon: +49 30-84185-0
Telefax: +49 30-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Transformations for the Prize-Collecting Steiner Tree Problem and the Maximum-Weight Connected Subgraph Problem to SAP

Daniel Rehfeldt* · Thorsten Koch

Abstract

Transformations of Steiner tree problem variants have been frequently discussed in the literature. Besides allowing to easily transfer complexity results, they constitute a central pillar of exact state-of-the-art solvers for well-known variants such as the Steiner tree problem in graphs. In this paper transformations for both the prize-collecting Steiner tree problem and the maximum-weight connected subgraph problem to the Steiner arborescence problem are introduced for the first time. Furthermore, we demonstrate the considerable implications for practical solving approaches, including the computation of strong upper and lower bounds.

1 Introduction

The classical Steiner tree problem in graphs has been investigated for long time, latest during the 11th DIMACS Challenge dedicated to the study of Steiner tree problems [1]. In practical applications, however, the problem usually arises in modified form. As a result, there exist a large number of problem variants. One of the oldest and most widely known is the prize-collecting Steiner tree problem [12], while during the last years the, related, maximum-weight connected subgraph problem has received considerable attention [3, 4, 7].

Many transformations between Steiner problem variants are known, most notably perhaps the one of the Steiner tree problem in graphs to its directed kinsman, the Steiner arborescence problem. This transformation is used by state-of-the-art solvers for the Steiner tree problem in graphs [10, 17] for row generation within branch-and-cut [13, 17] as well as for powerful reduction techniques [5, 17]. Other well-known transformations include those for the rectilinear Steiner tree problem and the group Steiner tree problem to the Steiner tree problem in graphs [6, 11]. In order to apply results that have been achieved for the classical Steiner tree problem in graphs it is particularly convenient, if possible, to transform the variant at hand to the classical problem, either in its undirected or its directed form. But while any polynomial transformation is enough to transfer complexity results, efficient transformations in the size of the transformed problem can also allow to employ existing advanced solvers for the Steiner tree problem in graphs.

In the following we will present transformations for the (rooted and unrooted) prize-collecting Steiner tree problem and the maximum-weight connected subgraph problem to the Steiner arborescence problem. These novel transformations allow to use state-of-the-art solvers for the Steiner tree problem in graphs to solve the problems. Moreover, we demonstrate how to use the transformations to employ powerful – primal and dual – heuristics to obtain strong upper and lower bounds within very short time.

*Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany, koch@zib.de, rehfeldt@zib.de

2 Transforming PCSTP to SAP

First, we define the *Steiner arborescence problem* (*SAP*), which constitutes the result of all transformations described in this paper. Given a directed graph $D = (V, A)$, costs $c : A \rightarrow \mathbb{Q}_{\geq 0}$, a set $T \subseteq V$ of terminals, and a root $r \in T$, a directed tree $S = (V_S, A_S) \subseteq D$ is required that first, for all $t \in T$ contains exactly one directed path from r to t and second, minimizes

$$C(S) := \sum_{a \in A_S} c_a. \tag{1}$$

Having defined the resulting problem, we now turn towards the other end and introduce the *prize-collecting Steiner tree problem* (*PCSTP*), a variant well-studied both theoretically and practically [12, 14]. Given an undirected graph $G = (V, E)$, edge weights $c : E \rightarrow \mathbb{Q}_{\geq 0}$, and vertex weights $p : V \rightarrow \mathbb{Q}_{\geq 0}$, a tree $S = (V_S, E_S)$ in G is required such that

$$P(S) := \sum_{e \in E_S} c_e + \sum_{v \in V \setminus V_S} p_v \tag{2}$$

is minimized. Hereinafter it is assumed for ease of presentation that at least one vertex v is of positive weight (otherwise any vertex constitutes an optimal solution).

To set the stage, we first introduce a transformation for a problem closely related to the PCSTP, the *rooted prize-collecting Steiner tree problem* (*RPCSTP*). This variant incorporates the additional condition that one distinguished node r, called *root*, is part of every feasible solution to the problem.

Transformation 1 (RPCSTP to SAP).
Input: *An RPCSTP* $P = (V, E, p, r)$
Output: *An SAP* $P' = (V', A', T', c', r')$

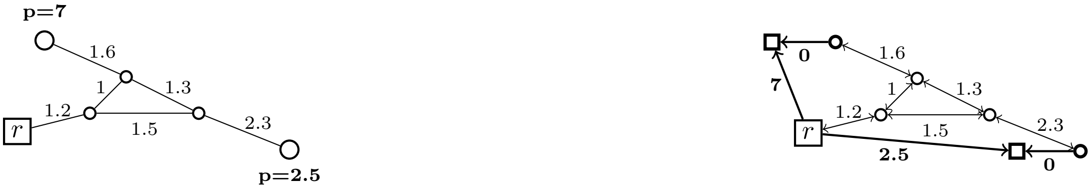
1. *Set* $V' := V$, $A' := \{(v, w) \in V' \times V' \mid \{v, w\} \in E\}$, $r' := r$ *and* $c' : A' \rightarrow \mathbb{Q}_{\geq 0}$ *with* $c'_a = c_{\{v,w\}}$ *for* $a = (v, w) \in A'$.
2. *Denote the set of all* $v \in V$ *with* $p_v > 0$ *by* $T = \{t_1, ..., t_s\}$. *For each node* $t_i \in T$, *a new node* t'_i *and an arc* $a = (t_i, t'_i)$ *with* $c'_a = 0$ *is added to* V' *and* A' *respectively.*
3. *Add arcs* (r', t'_i) *for each* $i \in \{1, ..., s\}$, *setting their respective weight to* p_{t_i}.
4. *Define the set of terminals* $T' := \{t'_1, ..., t'_s\} \cup \{r\}$.
5. ***Return*** (V', A', T', c', r').

Lemma 1 (RPCSTP to SAP). *Let* $P' = (V', A', T', c')$ *be an SAP obtained from an RPCSTP* $P = (V, E, c, p)$ *by applying Transformation 1. Each solution* $S' = (V'_{S'}, A'_{S'})$ *to* P' *can be mapped to a solution* $S = (V_S, E_S)$ *to* P *defined by:*

$$V_S := \{v \in V \mid v \in V'_{S'}\}, \tag{3}$$

$$E_S := \{\{v, w\} \in E \mid (v, w) \in A'_{S'} \text{ or } (w, v) \in A'_{S'}\}. \tag{4}$$

If S' *is an optimal solution to* P', *then* S *is an optimal solution to* P *and their objective values are equal.*



(a) RPCSTP instance

(b) Transformed SAP instance (modifications in **bold**)

Figure 1: Illustration of an RPCSTP instance with root r (left) and the equivalent SAP obtained by Transformation 1 (right). Terminals are drawn as squares and Steiner vertices as circles (with those of positive weight enlarged).

Proof. Preliminarily, define the set of all arcs of P' corresponding to edges of P as $A := \{(v,w) \in A' : \{v,w\} \in E\}$ and accordingly $A_{S'} := A'_{S'} \cap A$. Furthermore, let $T = \{t_1, ..., t_s\}$ and $T' = \{t'_1, ..., t'_s\} \cup \{r'\}$ as defined in Transformation 1. To acknowledge that (3) and (4) constitute a mapping $\mathcal{S}' \to \mathcal{S}$, it can be observed that first the root node is conserved and second that the set $A_{S'}$ (of all arcs of S' that correspond to edges in the original graph (V,E)) forms a tree, so E_S forms a tree as well.

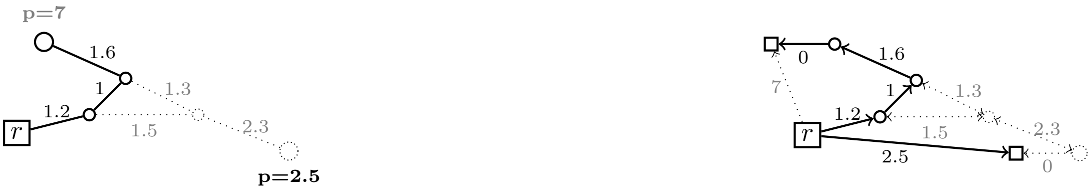
Next, assume that S' is an optimal solution to S, which implies that $t_i \in V'_{S'} \Rightarrow (t_i, t'_i) \in A'_{S'}$. Since moreover $t_i \notin V'_{S'} \Rightarrow (r', t'_i) \in A'_{S'}$ holds, it follows that

$$\sum_{a \in A'_{S'}} c'_a = \sum_{a \in A_{S'}} c'_a + \sum_{a \in A'_{S'} \setminus A_{S'}} c'_a = \sum_{e \in E_S} c_e + \sum_{v \in V \setminus V_S} p_v,$$

so the costs of S' and S are equal.

It remains to be shown that S is optimal as well. Assume that it is not, i.e. that there is an $\tilde{S} \in \mathcal{S}$ such that $P(S) < P(\tilde{S})$. We build a solution $\tilde{S}' = (V'_{\tilde{S}'}, A'_{\tilde{S}'})$ to the SAP P' as follows: First, define $A'_{\tilde{S}'}$ as the set of all arcs obtained by traversing all forward arcs (v,w) such that $\{v,w\} \in E_{\tilde{S}}$ starting from r'. Second, we add for each $t_i \in T$ reachable by arcs in $A'_{\tilde{S}'}$ the arc (t_i, t'_i) to $A'_{\tilde{S}'}$. For all other $t_i \in T$ we add (r', t'_i) to $A'_{\tilde{S}'}$. Consequently, all $t'_i \in T'$ are reachable from r' through forwards arcs and, being cycle-free and connected, $\tilde{S}'$ is a solution to P'. Moreover, we can infer that:

$$C(\tilde{S}') = P(\tilde{S}) < P(S) = C(S') \tag{5}$$



(a) RPCSTP solution

(b) SAP solution

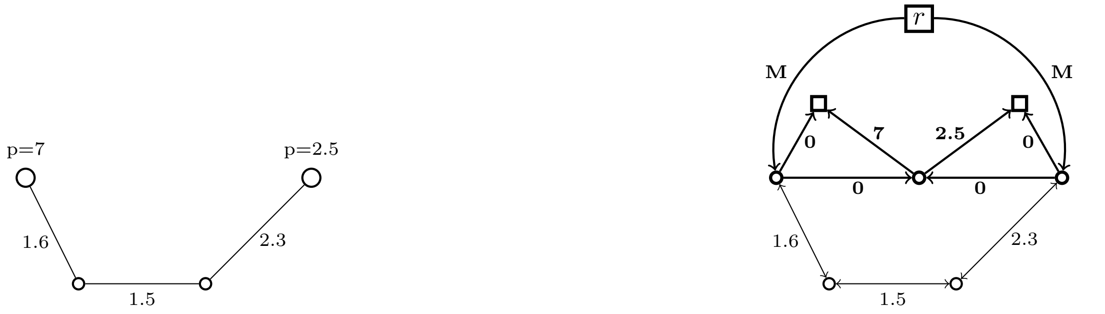
Figure 2: Illustration of a solution to the RPCSTP instance of Figure 1 (left) and an equivalent solution to the corresponding SAP (right).

which contradicts the assumption of S' being optimal. Thus S is an optimal solution to P. □

Likewise, the PCSTP can be transformed to an SAP. In [15] a transformation of the PCSTP to an SAP was introduced and formed the base of an exact solving approach. However, this SAP contains arcs of negative weight and is therefore not consistent with the common definition of an SAP (and cannot be used for the SAP derived solving approaches described in Section 4). To the best of our knowledge, the following transformation allows for the first time to solve a PCSTP as a pure SAP:

Transformation 2 (PCSTP to SAP)**.**
Input: *A PCSTP* $P = (V, E, p)$
Output: *An SAP* $P' = (V', A', T', c', r')$

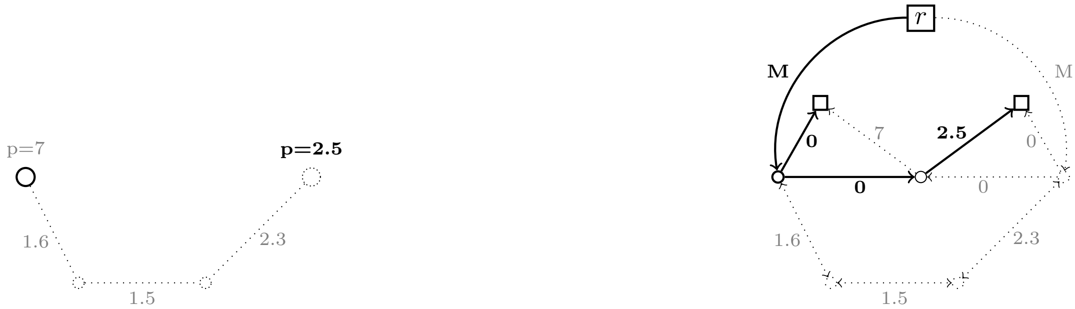
1. *Set* $V' := V$*,* $A' := \{(v, w) \in V' \times V' \mid \{v, w\} \in E\}$*, and* $c' : A' \to \mathbb{Q}_{\geq 0}$ *with* $c'_a = c_{\{v,w\}}$ *for* $a = (v, w) \in A'$*.*
2. *Add two vertices* r' *and* v'_0 *to* V'*.*
3. *Denote the set of all* $v \in V$ *with* $p_v > 0$ *by* $T = \{t_1, ..., t_s\}$ *and define* $M := \sum_{t \in T} p_t$*.*
4. *For each* $i \in \{1, ..., s\}$*:*
   - *(a) Add an arc* (r', t_i) *of weight* M *to* A'*.*
   - *(b) Add a new node* t'_i *to* V'*.*
   - *(c) Add arcs* (t_i, v'_0) *and* (t_i, t'_i) *to* A'*, both being of weight* 0*.*
   - *(d) Add an arc* (v'_0, t'_i) *of weight* p_{t_i} *to* A'*.*
5. *Define the set of terminals* $T' := \{t'_1, ..., t'_s\} \cup \{r'\}$*.*
6. ***Return*** (V', A', T', c', r')*.*



(a) PCSTP instance

(b) SAP instance (modifications in **bold**)

Figure 3: Illustration of a PCSTP instance (left) and the equivalent SAP obtained by Transformation 2 (right). Terminals are drawn as squares and Steiner vertices as circles (with those of positive weight enlarged).



(a) PCSTP solution

(b) SAP solution

Figure 4: Illustration of the (single-vertex) optimal solution to the PCSTP instance depicted in Figure 3 (left) and the equivalent solution to the corresponding SAP (right).

Lemma 2 (PCSTP to SAP)**.** *Let $P = (V, E, c, p)$ be a PCSTP and $P' = (V', A', T', c', r')$ the corresponding SAP obtained by applying Transformation 2. Denote by $\mathcal{S}$ and $\mathcal{S}'$ the sets of solutions to P and P' respectively. There exists a function $H_{PC} : \mathcal{S}' \to \mathcal{S}$ such that for each optimal solution $S' = (V'_{S'}, A'_{S'})$ to $\mathcal{S}'$ also $S := (V_S, E_S) := H_{PC}(S')$ is optimal (with respect to $\mathcal{S}$) and:*

$$V_S = \{v \in V : v \in V'_{S'}\}, \tag{6}$$

$$E_S = \{\{v, w\} \in E : (v, w) \in A'_{S'} \text{ or } (w, v) \in A'_{S'}\}. \tag{7}$$

Furthermore, if S' is optimal, it holds that $C(S') - M = P(S)$.

Proof. Consider an arbitrary $S' = (V'_{S'}, A'_{S'}) \in \mathcal{S}'$ and define the mapping $S' \mapsto H_{PC}(S') = (V_S, E_S)$ in the following way: Let $i_0 = \min\{i \in \{1, ..., s\} : (r', t_i) \in A'_{S'}\}$ and let $(V'_{\tilde{S}'}, A'_{\tilde{S}'})$ be the subtree of S' consisting of all vertices and arcs reachable from t_{i_0} through forward arcs in $A'_{S'}$. Thereupon, define $V_S = \{v \in V : v \in V'_{\tilde{S}'}\}$ and $E_S = \{\{v, w\} \in E : (v, w) \in A'_{\tilde{S}'}$ or $(w, v) \in A'_{\tilde{S}'}\}$. As $\tilde{S}'$ is cycle free and connected in P', so is S in P.

Next, assume that S' is optimal, which by definition of M implies that exactly one (r', t_i) is contained in S'. Therefore, $H_{PC}(S') = (V_S, E_S)$ corresponds to (6) and (7). To verify that $C(S') - M = P(S)$ holds, let $A := \{(v, w) \in A' : \{v, w\} \in E\}$ and $A_{S'} := A \cap A'_{S'}$. Since S' is optimal we can infer $t_i \notin V'_{S'} \iff (r', t'_i) \in A'_{S'}$ and thereupon:

$$\begin{aligned}\sum_{a \in A'_{S'}} c'_a &= \sum_{a \in A_{S'}} c'_a + \sum_{a \in A'_{S'} \setminus A_{S'}} c'_a \\ &= \sum_{e \in E_S} c_e + \sum_{a \in A'_{S'} \setminus A_{S'}} c'_a \\ &= \sum_{e \in E_S} c_e + \sum_{v \in V \setminus V_S} p_v + M,\end{aligned}$$

which implies $C(S') - M = P(S)$.

Finally, suppose that there is an $\overline{S} \in \mathcal{S}$ such that $P(\overline{S}) < P(S)$. Since we have presupposed that at least one vertex of V is of positive prize in P, we can find an $\tilde{S} = (V_{\tilde{S}}, E_{\tilde{S}})$ that satisfies $P(\tilde{S}) \leq P(\overline{S})$ and contains at least one $u \in V$ with $p_u > 0$. Thereupon, we build a solution

$\tilde{S}' = (V_{\tilde{S}'}, A'_{\tilde{S}'})$ to the SAP P' as follows: First, we define $V'_{\tilde{S}'} = \{r', u, v'_0\}$, $A'_{\tilde{S}'} = \{(r', u), (u, v'_0)\}$ and add to $A'_{\tilde{S}'}$ all arcs reachable from u through forward arcs (v, w) such that $\{v, w\} \in E_{\tilde{S}}$. Concomitantly, we add all vertices corresponding to arcs in $A'_{\tilde{S}'}$ to $V'_{\tilde{S}'}$. Second, we add for each $t_i \in T$ contained in $V'_{\tilde{S}'}$ the arc (t_i, t'_i), which is of cost 0, to $A'_{\tilde{S}'}$. For all $t_i \in T$ not connected we add the arc (v'_0, t'_i), which is of cost p_{t_i}, to $A'_{\tilde{S}'}$. Finally, we add all $t'_i \in T'$ to $V'_{\tilde{S}'}$. Consequently, all $t'_i \in T'$ are reachable from r' through forwards arcs and, being cycle-free and connected, $\tilde{S}'$ is a solution to P'. Furthermore, it holds that:

$$C(\tilde{S}') - M = P(\tilde{S}) < P(S) = C(S') - M$$

which contradicts the assumption that S' is optimal. Therefore, S is an optimal solution to P. □

3 Transforming MWCSP to SAP

The last problem considered in this paper is the *maximum-weight connected subgraph problem* (*MWCSP*), which involves a variety of real-world applications, for instance in computational biology [4]. Given an undirected graph (V, E) and weights $p : V \to \mathbb{Q}$, the objective is to find a tree $S = (V_S, E_S)$ that maximizes

$$W(S) := \sum_{v \in V_S} p_v. \tag{8}$$

Once again, it will be assumed for the ease of presentation that at least one vertex $v \in V$ is of positive weight.

In [4] a transformation of the MWCSP to the PCSTP was introduced. Consequently, we could transform the MWCSP to the SAP by applying Transformation 2 on the resulting PCSTP. However, in [18] it was shown that the transformation [4] usually results in an PCSTP with a large number of positive-weight vertices, which not only leads to a considerably larger size of the final SAP, but also renders the solving approaches described in Section 4 less efficient. Therefore, we introduce a direct transformation of the MWCSP to the SAP:

Transformation 3 (MWCSP to SAP)**.**
Input: *An MWCSP $P = (V, E, p)$*
Output: *An SAP $P'' = (V'', A'', T'', c'', r'')$*

1. *Set $V' := V$, $A' := \{(v, w) \in V' \times V' \mid \{v, w\} \in E\}$.*
2. *Set $c' : A' \to \mathbb{Q}_{\geq 0}$ such that for $a = (v, w) \in A'$:*
   $$c'_a = \begin{cases} -p_w, & \textit{if } p_w < 0 \\ 0, & \textit{otherwise} \end{cases}$$
3. *Set $p' : V' \to \mathbb{Q}_{\geq 0}$ such that for $v \in V'$:*
   $$p'_v = \begin{cases} p_v, & \textit{if } p_v > 0 \\ 0, & \textit{otherwise} \end{cases}$$
4. *Perform Transformation 2 to (V', A', c', p'), slightly changed in such a way that in step 1, instead of constructing a new arc set, A' is being used. This procedure results in an SAP $P'' = (V'', A'', T'', c'', r'')$.*
5. ***Return*** *P''.*

Lemma 3 (MWCSP to SAP)**.** *Let $P = (V, E, p)$ be an MWCSP and $P'' = (V'', A'', T'', c'', r'')$ an SAP obtained from P by Transformation 3, with solution sets $\mathcal{S}$ and $\mathcal{S}''$ respectively. Thereupon, a function $H_{MW} : \mathcal{S}'' \rightarrow \mathcal{S}$ exists such that for each solution $S'' = (V''_{S''}, A''_{S''}) \in \mathcal{S}''$ that is optimal, $S := (V_S, E_S) := H_{PC}(S'')$ is an optimal solution to the original MWCSP P and:*

$$V_S = \{v \in V : v \in V''_{S''}\}, \tag{9}$$

$$E_S = \{\{v, w\} \in E : (v, w) \in A''_{S''} \text{ or } (w, v) \in A''_{S''}\}. \tag{10}$$

Furthermore, if S'' is optimal, it holds that:

$$W(S) = \sum_{v \in V : p_v > 0} p_v - C(S'') + M. \tag{11}$$

Proof. The function H_{MW} can be defined analogously to H_{PC} in the proof to Lemma 2.

To prove (11), let $S'' = (V''_{S''}, A''_{S''})$ be an optimal solution to P'' and $S := (V_S, E_S) := H_{MW}(S'')$. Further, define $A := \{(v, w) \in A'' : \{v, w\} \in E\}$ and $A_{S''} = A \cap A''_{S''}$. First, one observes that for each $v \in V_S$ such that $p_v \leq 0$ there is exactly one incoming arc $a \in A_{S''}$, so:

$$\sum_{v \in V_S : p_v \leq 0} p_v = - \sum_{a \in A_{S''}} c''_a. \tag{12}$$

Second:

$$\sum_{v \in V_S : p_v > 0} p_v = \sum_{v \in V : p_v > 0} p_v - \sum_{v \in V \setminus V_S : p_v > 0} p_v \tag{13}$$

$$= \sum_{v \in V : p_v > 0} p_v - \sum_{a \in A''_{S''} \setminus A_{S''}} c''_a + M. \tag{14}$$

Finally, by combining (12) and (13) the equation:

$$\sum_{v \in V_S} p_v = \sum_{v \in V : p_v > 0} p_v - \sum_{a \in A''_{S''}} c''_a + M \tag{15}$$

is obtained, which coincides with (11).

Finally, verifying that S is optimal can be done analogeously to the procedure in the proof to Lemma 2. □

4 Practical Implications

When it comes to practical solving, the arguably most natural application of the transformations described in this paper is to solve the resulting SAPs to optimality. For the rooted prize-collecting Steiner tree problem this strategy works remarkably well: The Steiner problem solver SCIP-Jack [10, 18] which uses Transformation 1 to solve an RPCSTP instance as an SAP took first place in the category *exact solving of rooted prize-collecting Steiner tree problems* at the 11th DIMACS Challenge in December 2014. However, for the PCSTP and MWCSP we have found that a solving approach based on additional model constraints as described in [18] yields better results than solving the SAPs arising from the transformations described in this paper.

Besides the direct approach, the three transformations introduced in this paper bring in their wake considerable further advantages for practical (exact) solving: An important point is the possibility to transfer reductions techniques for the SAP to the respective original problem, as has successfully been done for the Steiner tree problem in graphs [5, 17]. Here we want to focus on two other aspects: The fast computation of strong upper and lower bounds. To this end, we introduce an IP formulation for the SAP, originally stated in [19]. Given an SAP (V, A, c, r), we associate with each arc $a \in A$ a variable y_a indicating whether a is contained in the Steiner arborescence ($y_a = 1$) or not ($y_a = 0$). These definitions give rise to the *directed cut formulation*:

$$\begin{aligned} \min \quad & c^T y & & & (16)\\ y(\delta^-(W)) \quad & \geq \quad 1 & & \text{for all } W \subset V,\ r \notin W, W \cap T \neq \emptyset, & (17)\\ y_a \quad & \in \quad \{0,1\} & & \text{for all } a \in A. & (18) \end{aligned}$$

In [19] a dual-ascent algorithm for the SAP was introduced that empirically not only provides strong lower bounds, but also allows for fast computation. At termination, dual-ascent provides a dual solution to the LP-relaxation of the directed cut formulation, involving directed paths along arcs of reduced cost 0 from the root to each additional terminal. With the three new transformation introduced in this paper at hand we can use the dual-ascent algorithm to obtain lower bounds for the RPCSTP, the PCSTP, and the MWCSP.

Additionally, the new transformations can be used to obtain upper bounds. To this end, we have extended the powerful heuristic *ascend-and-prune* [17] which has so far only been used for the Steiner tree problem in graphs. Consider an RPCSTP, PCSTP or MWCSP P and the SAP P' resulting from the respective transformation described in this paper. Thereupon, ascend-and-prune attempts to find a good solution to the subproblem $\tilde{P}$ consisting of the edges, vertices and terminals of P corresponding to all (directed) root-terminal paths in P' that contain only arcs of reduced cost 0. For this purpose, the reduction-based heuristic *prune* [17] is used, which we employ for the RPCSTP, the PCSTP and the MWCSP, using the reduction techniques we introduced in [18].

We have implemented the primal and dual heuristics described above in C, with the dual-ascent algorithm implemented according to [16]. In the following we provide results on six benchmark test sets, all but one from the 11th DIMACS Challenge: The RPCSTP instances are derived from the design of fiber optic networks [15] and contain up to 20,000 edges. The PCSTP test set JMP [12] (of which only two instances were part of the DIMACS Challenge) contains up to the 1500 edges and the E test set [15] (which was not part of the Challenge) up to 62,500. Finally, the MWCSP test set ACTMOD is derived from an application in computational biology [4], exhibiting almost 100,000 edges for some instances, while the MWCSP test set JMPALMK [3] comes with up to 20,000 edges. We partitioned the latter set into all instances that were part of the DIMACS competition (JMPALMK1) and the remainder (JMPALMK2). The computational setting is the same as for the DIMACS Challenge (Cluster of Intel Xeon X5672 CPUs with 3.20 GHz and 48 GB RAM).

Testset	Problem	Instances	Solved	∅ Gap [%]	∅ Time [s]	∅ DIMACS Time [s]
Cologne1	RPCSTP	14	13	0.0	0.0	4.0 [1]
Cologne2	RPCSTP	15	14	0.0	0.0	56.2 [1]
JMP	PCSTP	34	27	0.1	0.0	-
E	PCSTP	40	22	0.4	0.8	-
ACTMOD	MWCSP	8	3	1.1	0.2	4.1 [1]
JMPALMK1	MWCSP	30	30	-	0.1	0.4 [1]
JMPALMK2	MWCSP	42	41	0.8	0.1	-

Table 1: Computational results.

[1] Best solvers at DIMACS Challenge: SCIP-Jack [18] for RPCSTP; Mozartballs [8] for MWCSP.

The results are illustrated in Table 1. Columns four to six show the number of optimally solved instances, the average gap (arithmetic mean), and the average run time (shifted geometric mean [2] with shift 1) of our heuristics, while the last column lists the average run time of the best exact solver in the DIMACS Challenge for the respective problem variant. Clearly, the heuristics run very fast on all test sets, as compared to the respective best exact solver. Furthermore, more than 80 % of the instances can be solved to optimality and the (arithmetic) average gap among the unsolved instances is small. Several instances are solved to optimality by us more than a thousand times faster than by all solvers of the DIMACS Challenge. Also, we could solve several instances of the E test set that had not been solved without preprocessing before. Conclusively, the heuristics described in our paper allow to solve many benchmark instances much faster to proven optimality than exact state-of-the-art solvers and exhibit fairly small gaps for the remaining problems.

Finally, the techniques described in this paper are not only implemented as stand-alone heuristics, but have also been added to the exact Steiner tree solver SCIP-Jack and will be published as part of the next SCIP Optimization Suite [9] release.

5 Acknowledgements

This work was supported by the BMWi project *Realisierung von Beschleunigungsstrategien der anwendungsorientierten Mathematik und Informatik fur optimierende Energiesystemmodelle - BEAM-ME* (fund number 03ET4023DE). This work was supported by DFG in the framework of the Collaborative Research Centre CRC/Transregio 154, *Mathematical Modelling, Simulation and Optimization Using the Example of Gas Networks.* This work was supported by the ICT COST Action TD1207 *Mathematical Optimization in the Decision Support Systems for Efficient and Robust Energy Networks.* The work for this article has been conducted within the Research Campus Modal funded by the German Federal Ministry of Education and Research (fund number 05M14ZAM).

References

[1] 11th DIMACS Challenge. `http://dimacs11.zib.de/`. Accessed: June 5. 2016.

[2] Tobias Achterberg. *Constraint Integer Programming.* PhD thesis, Technische Universität Berlin, 2007.

[3] Eduardo Álvarez-Miranda, Ivana Ljubić, and Petra Mutzel. The maximum weight connected subgraph problem. In *Facets of Combinatorial Optimization*, pages 245–270. Springer Berlin Heidelberg, 2013.

[4] Marcus T. Dittrich, Gunnar W. Klau, Andreas Rosenwald, Thomas Dandekar, and Tobias Müller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. In *ISMB*, pages 223–231, 2008.

[5] C. Duin. *Steiner Problems in Graphs.* PhD thesis, University of Amsterdam, 1993.

[6] C. W. Duin, A. Volgenant, and S. Voss. Solving group Steiner problems as Steiner problems. *European Journal of Operational Research*, 154(1):323–329, 2004.

[7] Mohammed El-Kebir and Gunnar W. Klau. Solving the Maximum-Weight Connected Subgraph Problem to Optimality. *Computing Research Repository*, abs/1409.5308, 2014.

[8] Matteo Fischetti, Markus Leitner, Ivana Ljubic, Martin Luipersbeck, Michele Monaci, Max Resch, Domenico Salvagnin, and Markus Sinnl. Thinning out steiner trees: a node-based model for uniform edge costs. *Mathematical Programming Computations*, 2015.

[9] Gerald Gamrath, Tobias Fischer, Tristan Gally, Ambros M. Gleixner, Gregor Hendel, Thorsten Koch, Stephen J. Maher, Matthias Miltenberger, Benjamin Müller, Marc E. Pfetsch, Christian Puchert, Daniel Rehfeldt, Sebastian Schenker, Robert Schwarz, Felipe Serrano, Yuji Shinano, Stefan Vigerske, Dieter Weninger, Michael Winkler, Jonas T. Witt, and Jakob Witzig. The scip optimization suite 3.2. Technical Report 15-60, ZIB, Takustr.7, 14195 Berlin, 2016.

[10] Gerald Gamrath, Thorsten Koch, Stephen Maher, Daniel Rehfeldt, and Yuji Shinano. SCIP-Jack - A solver for STP and variants with parallelization extensions. Technical Report 15-27, ZIB, Takustr.7, 14195 Berlin, 2015.

[11] M. Hanan. On Steiner's problem with rectilinear distance. *SIAM Journal of Applied Mathematics*, 14(2):255–265, 1966.

[12] David S. Johnson, Maria Minkoff, and Steven Phillips. The Prize Collecting Steiner Tree Problem: Theory and Practice. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, pages 760–769, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[13] Thorsten Koch and Alexander Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32:207–232, 1998.

[14] Ivana Ljubic, Ren Weiskircher, Ulrich Pferschy, Gunnar W. Klau, Petra Mutzel, and Matteo Fischetti. An Algorithmic Framework for the Exact Solution of the Prize-Collecting Steiner Tree Problem. *Mathematical Programming*, 105(2-3):427–449, 2006.

[15] Ivana Ljubi. *Exact and Memetic Algorithms for Two Network Design Problems.* PhD thesis, Vienna University of Technology, 2004.

[16] Thomas Pajor, Eduardo Uchoa, and Renato F. Werneck. A Robust and Scalable Algorithm for the Steiner Problem in Graphs. *Computing Research Repository*, 2014.

[17] Tobias Polzin. *Algorithms for the Steiner problem in networks.* PhD thesis, Saarland University, 2004.

[18] Daniel Rehfeldt. A generic approach to solving the Steiner tree problem and variants. Master's thesis, Technische Universität Berlin, 2015.

[19] R.T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Mathematical Programming*, 28:271287, 1984.