Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

GERALD GAMRATH, THORSTEN KOCH, STEPHEN J MAHER,
DANIEL REHFELDT, YUJI SHINANO

SCIP-Jack – A solver for STP and variants with parallelization extensions

ZIB Report 15-27 (April 2015)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

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

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

SCIP-Jack – A solver for STP and variants with parallelization extensions*

Gerald Gamrath[†]· Thorsten Koch · Stephen J. Maher
Daniel Rehfeldt · Yuji Shinano

Abstract

The Steiner tree problem in graphs is a classical problem that commonly arises in practical applications as one of many variants. While often a strong relationship between different Steiner tree problem variants can be observed, solution approaches employed so far have been prevalently problem specific. In contrast, this paper introduces a general purpose solver that can be used to solve both the classical Steiner tree problem and many of its variants without modification. This is achieved by transforming various problem variants into a general form and solving them using a state-of-the-art MIP-framework. The result is a high-performance solver that can be employed in massively parallel environments and is capable of solving previously unsolved instances.

1 Introduction

The *Steiner tree problem in graphs* (STP) is one of the classical $\mathcal{NP}$-hard problems [1]. Given an undirected connected graph $G = (V, E)$, costs $c : E \to \mathbb{Q}^+$ and a set $T \subset V$ of *terminals*, the problem is to find a minimum weight tree $S \subseteq G$ which spans T.

The STP is said to have a variety of practical applications. However, applications that involve solving the pure STP are rarely encountered in practice. The lack of real-world applications for the pure STP is highlighted by the fact that from the thousands of instances collected by the authors in the SteinLib [2] very few have practical origins. However, there exist numerous applications that include STPs as a subproblem or that are formulated as a particular variant.

The announcement of the 11th DIMACS Challenge initiated our work with an investigation into the STP solver, Jack-III, described in [3]. The model and code of Jack-III provided a base for the development of a general STP solver – being able to solve many of the problem variants. However, Jack-III is a 15 year old code; as such, many modern developments regarding STP solution methods and MIP solving techniques were not available. Our approach to address this limitation of Jack-III was to combine the model of [3] with the start-of-the-art MIP-framework SCIP [4]. Employing SCIP naturally facilitated the incorporation of many algorithm developments from the past 15 years.

A major contribution of this paper is the development of a general STP solver. This is in contrast to the many problem specific solvers observed within the literature. Furthermore, SCIP provides a massively parallel MIP-framework that is employed with this general solver.

*The work done for this article was supported by the BMBF Research Campus Modal SynLab and by a Google Faculty Research Award.

†Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany, {gamrath, koch, maher, rehfeldt, shinano}@zib.de

This combined with algorithmic improvements allows us to solve several previously unsolved instances. Throughout this paper we show:

- in Section 2, the impact of transitioning from a simple self-made branch-and-cut code to the use of a full fledged, state-of-the-art MIP-framework,
- in Section 3, how to employ the versatility of MIP models to solve a class of related problem variants, and
- in Section 4, the potential from using hundreds of CPU cores to solve a single problem.

This demonstrates how worthwhile it can be to revisit topics after some time. Further details on this can be found in [5, 6].

In general it can be stated that a branch-and-cut based Steiner tree solver has three major components. First, as with the TSP [7], preprocessing is extremely important. Apart from some pathological instances specifically constructed to defy presolving techniques, such as the PUC [8] and I640 [9] test sets, preprocessing is often able to significantly reduce instances. Results presented in the PhD thesis of Polzin [10] report an average reduction in the number of edges of 78 %, with many instances solved completely by presolving.

Second, heuristics are needed to find good or even optimal solutions. In our experiments, for 92 % of the instances the final solution was found by a heuristic.

Finally, at the core is the branch-and-cut procedure used to compute a lower bound and prove optimality. The ability to solve 536 out of 626 solved instances without branching shows the strength of the relaxation of the employed model. A surprising observation is that in many cases we either solve an instance within very few nodes or not at all, even after processing a large number of branch-and-bound nodes. This indicates that strengthening the relaxation is neccessary to solve these instances sucessfully.

2 From simple hand tailored to off-the-shelf state-of-the-art

The model employed in our new solver SCIP-Jack uses the *directed cut formulation* described in [3]. This formulation provides a tight linear programming (LP) relaxation. It is built upon the directed equivalent of the STP, the *Steiner arborescence problem* (*SAP*): Given a directed graph $D = (V, A)$, a root $r \in V$, costs $c : A \to \mathbb{Q}^+$ and a set $T \subset V$ of terminals, a directed tree $S \subseteq D$ is required such that for all $t \in T$, (V_S, A_S) contains exactly one directed path from r to t. Each STP can be transformed to an SAP replacing each edge by two anti-parallel arcs of the same cost and distinguishing an arbirtrary terminal as the root. This results in a one-to-one correspondence between the respective solution sets, see Appendix A.

Introducing variables y_a for $a \in A$ with the interpretation $y_a := 1$, if a is in the Steiner arborecence, and $y_a := 0$ otherwise, we obtain the integer program:

Formulation 1. *Directed Cut Formulation*

$$
\begin{array}{rcllr}
\min \vec{c}^T y & & & & (1) \\
y(\delta^+(W)) & \geq & 1, & \textit{for all } W \subset V, r \in W, (V \setminus W) \cap T \neq \emptyset & (2) \\
y(\delta^-(v)) & \left\{ \begin{array}{l} = \\ = \\ \leq \end{array} \right. & \begin{array}{l} 0, \textit{if } v = r; \\ 1, \textit{if } v \in T \setminus r; \\ 1, \textit{if } v \in N; \end{array} & \textit{for all } v \in V & (3) \\
y(\delta^-(v)) & \leq & y(\delta^+(v)), & \textit{for all } v \in N; & (4) \\
y(\delta^-(v)) & \geq & y_a, & \textit{for all } a \in \delta^+(v), v \in N; & (5) \\
0 \leq y_a & \leq & 1, & \textit{for all } a \in A; & (6) \\
y_a & \in & \{0,1\}, & \textit{for all } a \in A, & (7)
\end{array}
$$

where $N = V \setminus T$, $\delta^+(X) := \{(u,v) \in A | u \in X, v \in V \setminus X\}$, $\delta^-(X) := \delta^+(V \setminus X)$ for $X \subset V$ i.e., $\delta^+(X)$ is the set of all arcs going out of and $\delta^-(X)$ the set of all arcs going into X. Further details of Formulation 1 are given in [3]. It is shown in [10] that the flow-cuts (3)-(5) are indeed facets.

Since the model potentially contains an exponential number of constraints a separation approach is employed. Violated constraints are separated during the execution of the branch-and-cut algorithm. Jack-III employed this problem formulation along with a model-specific branch-and-bound search. Strong branching [11] was used with a depth-first search node selection.

Our implementation of SCIP-Jack is based on the academic MIP solver SCIP. Besides being one of the fastest non-commercial MIP solvers [12], SCIP is a general branch-and-cut framework. The plugin-based design of SCIP provides a simple method of extension to handle a variety of specific problem classes.

In the case of SCIP-Jack, the first plugins implemented were a *reader* to read problem instances and *problem data* to store the graph and build the model within SCIP. Within these plugins it was possible to re-use the reading methods, data structures, and preprocessing algorithms of Jack-III. However, each of these had to be extended as part of the implementation in SCIP-Jack. The heart of the new implementation is a *constraint handler* that checks solutions for feasibility and separates any violated model constraints. Again, we re-use the separation methods of the 15-year old code, while SCIP provides a filtering of cuts to improve numerical stability and dynamic aging of the generated cuts. Additionally, the general-purpose separation methods that exist within SCIP are used, which include Gomory and mixed-integer rounding cuts.

Jack-III includes many STP-specific preprocessing techniques, as described in [3]. However, for SCIP-Jack there have been a number of additions to incorporate some of the developments of the last 15 years and provide reduction techniques for directed STP variants. The preprocessing techniques implemented in SCIP-Jack include: the degree test I and II (DT) and nearest vertex test presented by Beasley [13]; the short link (SL), longest edge (LE) and bound tests (BT) developed by Polzin [10]; the special distance test (SD) presented by Duin and Volgenant [14]; and the nearest special vertex (NSV) and non-terminals of degree three tests (NTD3) presented by Duin and Volgenant [15]. In addition, a modified form of the bound test has been implemented for the hop-constrained directed Steiner tree variant – using the hop constraint as an upper bound on a graph with unit edge weights – and the nearest vertex for optimal arcs test (NVO) [16] is employed for the directed variants. The implemented techniques strive to reduce the size of an original problem instance in terms of both vertices and edges, allowing the retransformation of each solution to the original problem space and preserving at least one optimal solution.

The branch-and-bound search is organized by SCIP. The default hybrid branching rule [17] is used, which combines strong branching and pseudo costs with history information. Node selection is performed with respect to a best estimate criterion – interleaved with best bound and depth-first search phases [18].

Three STP-specific primal heuristics have been implemented in SCIP-Jack– the repetitive shortest path heuristic (RSPH), based on [19] and in a modified form on [10], an improvement heuristic (VQ) [20] and a novel recombination heuristic (RC). The repetitive shortest path heuristic is both coherent and empirically successful: Starting with a single vertex, in each step the current subtree is connected to a nearest terminal by a shortest path. This procedure is reiterated until all terminals are spanned. The heuristic has already been implemented in Jack-III, where it is used not only as an initial heuristic but also, with altered edge weights, during the branch-and-cut. Specifically, given an LP optimal solution $x \in \mathbb{Q}^{\mathbb{E}}$, the heuristic is called with the edge weights $(1 - x_e) \cdot c_e$ for all $e \in E$. Thus, a stimulus for the heuristic to choose edges contained in the LP solution is provided. Moreover, the heuristic is started from several distinct vertices, making it empirically much more potent (by default 100 start vertices for the intial call, 50 after the branch-and-bound root node and 10 otherwise). In regards to the SCIP-Jack implementation, the heuristic is employed with some alterations that bring empirical advantages. First, terminals are preferred as starting points. Second, ties are broken pseudo-randomly and when no new LP-solution is available, except for the initial run, each edge cost is additionally multiplied by a pseudo-random number between 1.0 and 2.5. Finally, for problems for which at least five percent of the vertices are terminals (after preprocessing) a variation based on the concept of Voronoi regions (see [10]) is used. This follows the same scheme but often provids a significant computational advantage. The heuristic is called before and after the processing of a (branch-and-bound) node, after each cut loop and after each LP solving during a cut loop.

The improvement heuristic VQ is a combination of the three local search heuristics – vertex insertion, key-path exchange, and key-vertex elimination – as described in [20]. The greatest impact is achieved by the latter two. For the implementation in SCIP-Jack some alterations were required in order to adapt the algorithms, originally designed for undirected problems, to our model (Formulation 1). The basic idea of vertex insertion (denoted by V) is to connect further vertices to an existing Steiner tree in such a way that expensive edges can be removed. Key-vertices with respect to a tree S are either terminals or vertices of degree at least three in S. Correspondingly, a key-path is a path in S connecting two key-vertices and otherwise containing only non-key-vertices. In key-path exchange attempts are made to replace existing key-paths by others that are less costly. Similarly, for key-vertex elimination in each step a non-terminal key-vertex and all adjoining key-paths (except for the key-vertices at their respective ends) are extracted and an attempt is made to reconnect the disconnected subtrees at a lower cost. As in [20], the combination of key-path exchange and key-vertex elimination is denoted by Q. VQ is called for a newly found solution whenever the latter is among the five best known solutions.

The third heuristic, RC, comprises in essence a recombination of several solutions. In the following RC is described in the context of an STP but it can be naturally extended to cover the different STP variants discussed in this paper. Preliminarily, we define the set of solutions to be considered for recombination by $\mathcal{L}$; in the case of SCIP-Jack $\mathcal{L}$ comprises the, at most 50, best found solutions. The heart of RC is the *n-merging* ($n \geq 2$) operation subsequently defined for a given solution S: S is merged with pseudo-randomly selected $n - 1$ solutions out of $\mathcal{L} \setminus \{S\}$ to form a new graph G_S consisting of all edges and vertices that are part of at least one of the n solutions. Applying the reduction techniques provided by SCIP-Jack to G_S, a reduced graph G'_S is obtained. Then, a solution to G'_S is computed in two steps. First, the cost of each edge is pseudo-randomly reduced for each solution it appears in, as suggested by [21], and RSPH is employed resulting in a solution S'. Second, retrieving the original arc costs, VQ is used on S'.

Finally, S' is retransformed to the original solution space.

The RC heuristic is clustered around the n-merging operation: Given a new solution S, in one *run* consecutively three 2-, two 3- and 4-, and one 5-merges are performed. When a solution S' is generated during an i-merging with a smaller cost than S, we set $S := S'$ and attempt to add S' to $\mathcal{L}$. Moreover, in this case the i-merging is performed again in a new run that is started after the conclusion of the current run. The total number of runs is limited to ten. RC is called whenever r new solutions have been found compared to its last execution. Intially r is set to 4 and modified throughout the solution process, setting $r := 0$ if a solution has been improved during the execution of RC and $r := r + 1$ otherwise.

Along with the three employed heuristics, each newly generated solution S is *pruned* i.e., it is substituted by a minimum spanning tree constructed on the vertices of S and nonterminals of degree one are, repeatedly, removed. Thereby, solutions generated by any of the three heuristics or from an LP-relaxation can be improved.

The combination of RPSPH, VQ and RC considerably helps generate good primal solutions quickly and is able to find optimal solutions to most problems. In 97 % of all cases the first solution was found by the TM heuristic, which is the first executed, or one of its modified forms developed for the STP variants. For all other feasible instances, SCIP constructed a feasible trivial solution prior to the first call of the TM heuristic, which was significantly improved by the execution of this heuristic. The final primal bound was found in 49 % of the instances by TM, in 23 % by VQ, and in 20 % by RC. It must be noted, however, that especially RC but also VQ are more effective for harder instances, where the optimal solution is not found quickly; e.g., in the hard PUC test set, 11 of the final primal solutions were found by TM, 19 by RC and even 20 by VQ; of the latter, 14 solutions were an improvement of a solution that had previously been found by RC.

2.1 Computational experiments

Several thousand STP instances of different variants were collected as part of the DIMACS challenge. Given our aim to develop a general STP solver, computational experiments on ten variants of the STP will be presented throughout this paper.

All computational experiments described were performed on a cluster of Intel Xeon X5672 CPUs with 3.20 GHz and 48 GB RAM, running Kubuntu 14.04. We used a development version of SCIP 3.1 with SoPlex [22] version 2.0.1 as underlying LP solver. We limited the preprocessing time by two hours and allowed another two hours for the subsequent branch-and-cut process. If an instance is not solved to optimality within the time limit, we report the gap, which is defined as $\frac{|\mathrm{pb}-\mathrm{db}|}{\max\{|\mathrm{pb}|,|\mathrm{db}|\}}$ for final primal and dual bound pb and db, respectively. The average gap is obtained as an arithmetic mean while averages of the number of nodes or the solving time are computed by taking the shifted geometric mean [18] with a shift of 1.0.

Prior to discussing the different STP variants solved by SCIP-Jack, we first demonstrate the solver performance on pure STP instances. Five STP test sets have been used for the computational experiments. Four of them, SP [2], I320, I640 [9], and PUC [8], are computationally difficult test sets from SteinLib. The I320 and I640 test sets contain randomly generated sparse graphs selected to defy preprocessing. Similarly, the PUC test set contains artificially designed combinations of odd wheels and odd circles, that are difficult to solve by linear programming approaches. Additionally, we employ SCIP-Jack to solve the vienna-i-advanced test set [23], which contains instances newly submitted to the DIMACS Challenge that have already been preprocessed with techniques presented in [23].

A summary of the computational performance of SCIP-Jack on pure STP instance is presented in Table 1. Each line in the table shows aggregated results for one test set, as specified

Table 1: Computational results for STP instances

			optimal		timeout	
test set	#	solved	∅ nodes	∅ time [s]	∅ nodes	∅ gap [%]
SP	8	6	4.4	5.7	14.7	0.6
I320	100	82	4.9	12.2	1006.6	0.5
I640	100	65	7.2	25.1	63.0	0.8
PUC	50	7	1139.5	138.2	57.0	4.3
vienna-i-advanced	85	45	1.5	430.3	1.0	0.6

in the first column. The second column, labeled #, lists the number of instances in the test set, the third column states how many of them were solved to optimality within the time limit. The average number of branch-and-bound nodes and the average running time in seconds of these instances are presented in the next two columns, named optimal. The last two columns, labeled timeout, show the average number of branch-and-bound nodes and the average gap for the remaining instances, i.e., all instances that hit the time limit. In the next section, similar tables will be presented for different STP variants. Thereby, the timeout columns are omitted if all instances have been solved to optimality. Detailed instance-wise computational results of all experiments can be found in Appendix B.

SCIP-Jack solves five of the eight instances of the SP test set, and about eighty and sixty percent of the I320 and I640 test set, respectively. On average, for the solved instances only a couple of nodes and a few seconds are required. However, there are also instances which need a significant amount of branching – up to 4000 nodes for instances solved within the time limit and more than 10 000 nodes for some instances that time out. The PUC test set appears much more difficult for SCIP-Jack. This is unsurprising since more than half of the instances in this set still remain unsolved. SCIP-Jack only solves eight of 50 instances and none at the root node. About half of the instances in the vienna test set are solved by SCIP-Jack within the time limit, most of them at the root node or after a couple of branchings. These results show the ability and limitations of SCIP-Jack to solve pure STP instances.

3 From single problem to class solver

SCIP-Jack is developed as a general STP solver – being able to solve many problem variants. An overview of the problem variants solved by SCIP-Jack is given in Table 2. This table also presents the heuristics and presolving techniques that are applied to each of the problem variants. Specific transformation approaches have been employed in order to solve each variant

Table 2: Problem variants solved by SCIP-Jack

Variant	Abbreviation	Preprocessing	Heuristics
Steiner Tree Problem in Graphs	STP	DT, NV, SVQ, LE, BT, SD, NSV, NTD3	RSPH, VQ, RC
Steiner Arborescence	SAP	DT, SD	RSPH, RC
Rectilinear Steiner Minimum Tree	RSMTP	None	RSPH, VQ, RC
Node-weighted Steiner Tree	NWSTP	DT, NVO, SD, NTD3	RSPH, RC
Prize-collecting Steiner Tree	PCSTP	DT, NVO, SD, NTD3	RSPH, VQ, RC
Rooted Prize-collecting Steiner Tree	RPCSTP	DT, NVO, SD, NTD3	RSPH, VQ, RC
Maximum-weight Connected Subgraph	MWCSP	DT, NVO, SD, NTD3	RSPH, RC
Degree-constrained Steiner Tree	DCSTP	None	RSPH
Group Steiner Tree	GSTP	DT, NV, SVQ, LE, BT, SD, NSV, NTD3	RSPH, VQ, RC
Hop-constrained directed Steiner Tree	HCDSTP	DT, BT	RSPH, RC

using SCIP-JACK. The details of these transformations will be described in detail. Throughout this section the weights of an (undirected) edge e and an (directed) arc a are denoted by c_e and c_a respectively and the weight of a vertex v by p_v.

3.1 The Steiner Arborescence Problem

The Steiner tree solver, SCIP-JACK, transforms each Steiner tree problem to a (bidirected) Steiner arborescence problem (SAP). As such, the branch-and-cut substruction and the RSPH and RC heuristics can be used for general SAPs with only minor modifications. However, due to the missing bidirection with equal cost the VQ heuristic and several presolving techniques cannot be applied. While presolving techniques have been implemented for directed STP instances, their implementation is not applicable to all forms of directed variants.

Computational Results Computational experiments have been performed on three test sets of Steiner arborescense problems. These instances were derived from a genetic application [24]. The results are summarized in Table 3. The test sets contain small SAP instance, with the largest consisting of 602 nodes, 1716 edges and 86 terminals. Because of their size SCIP-JACK solves all instance within fractions of a second without requiring any branching.

Table 3: Computational results for SAP instances

test set	#	solved	∅ nodes	∅ time [s]
gene	6	6	1.0	0.1
geneh	4	4	1.0	0.1
gene2002	9	9	1.0	0.1

3.2 The Rectilinear Steiner Minimum Tree Problem

The *rectilinear Steiner minimum tree problem* ($RSMTP$) can be described as follows: given a number of $n \in N$ points in the plane, find a shortest tree consisting only of vertical and horizontal line segments, containing all n points. The *RSMTP* is $\mathcal{NP}$-hard, as proved in [25], and has been the subject of various publications, [26, 27, 28]. In addition to this two-dimensional variant, a generalization of the problem to the d-dimensional case, with $d \geq 2$, will be considered, having real-world applications in up to eight dimensions, e.g. in cancer research [29].

Hanan [30] reduced the RSMTP to the *Hanan-grid* obtained by constructing vertical and horizontal lines through each given point of the RSMTP. It is proved in [30] that there is at least one solution to an RSMTP that is a subgraph of the grid. Hence, the RSMTP can be reduced to an STP. Subsequently, this construction and its multi-dimensional generalisation [31] is exploited in order to adapt the RSMTP to our solver. Given a d-dimensional, $d \in \mathbb{N} \setminus \{1\}$, RSMTP represented by a set of $n \in N$ points in $\mathbb{Q}^d$, the first step involves building a d-dimensional Hanan-grid. Using the resulting Hanan-grid an STP $P = (V, E, T, c)$ can be constructed.

There are a few necessary remarks regarding the implementation in SCIP-JACK. First, by default preprocessing is not used for RSMTP problems within our solver. This is spawned by the problem specific structure, which proved to be distinctively recalcitrant to the reduction techniques employed by SCIP-JACK leading to an exceptionally poor performance of the presolving. It is explained in [32] that in certain cases problem specific presolving is initially required to render RSMTP instance amenable to standard STP reductions techniques. At present, such problem specific reduction techniques are not available in SCIP-JACK. A potential reduction technique for the RSMTP is the full Steiner trees (FST) generation [33]. Apart from this, a transformed RSMTP instance is handled equivalently to a usual STP instance by SCIP-Jack. Second, we do not expect this simple approach to be competitive with highly specialized solvers, such as GeoSteiner [26] in the cases $d = 2$ and $d = 3$. However, the motivation for our implementation was to provide solutions to RSMTP instances in dimensions $d \geq 4$, since there seem to be a lack of specialized solvers. Still, it is not practical to apply the grid transformation for large instances in high dimension, as the number of both vertices and edges increases exponentially with the number of dimensions.

A variant of the RSMTP is the *obstacle-avoiding rectilinear Steiner minimum tree problem* (*OARSMTP*). These problems require that the minimum-length rectilinear tree does not pass through the interior of any specified axis-aligned rectangles, denoted as *obstacles*. SCIP-Jack is easily extended to solve the OARSMTP with a simple modification to the Hanan grid approach applied to the RSMTP. This modification involves removing all vertices that are located in the interior of an obstacle together with their incident edges. Since there was no competition for this variant in the DIMACS challenge and for the OARSMTP, unlike the RSMTP, optimal solutions to all instances submitted to the challenge have already been published, we refrain from conducting any computational experiments, although SCIP-Jack is able to solve them.

Computational Results The experiments on the RSMTP involved solving nine of the test sets submitted to the DIMACS Challenge. These test sets contain instances ranging from less than ten to 10000 points and from two to eight dimensions. Specifically, the test sets included the two-dimensional estein instances with up to 60 nodes, the solids test set with three-dimensional instances whose terminals are the vertices of the five platonic solids, and the cancer instances with up to eight dimensions. Computational results are summarized in Table 4 with the detailed results listed in the appendix.

All of the estein instances that are solved to optimality, except estein20-3, estein20-4, estein20-7, and estein40-4, require only a single node. While the first three named instances are solved with 11, 5 and 3 nodes respectively, estein40-4 needs 2667 nodes and an extension of the time limit so that optimality can be reached after almost 13 hours. As the number of vertices in the graphs increase, the runtime and the number of unsolved instances increases. For all but four of the unsolved instances, SCIP-Jack achieves an optimality gap within 1%. SCIP-Jack manages to solve all of the solids instances to optimality. The only instance that requires a significant amount of branching, with 14249 nodes, is the largest instance, modelling a dodecahedron. Finally, the cancer instances demonstrates the ability of SCIP-Jack to handle and solve instances with up to eight dimensions. Since these instances were of particular interest for us, we used a higher time limit of 36 hours. This enabled us to solve 12 of the 14 instances to optimality at the root node. To the best of our knowledge we are the first ones to solve those instances to optimality. Of these instances, two require runtimes larger than two hours – up to 51344 seconds – to achieve optimality. Of particular interest are the cancer4_6D and cancer5_6D instances which find the optimal solution early in the solution process, after 134 and 5.8 seconds respectively, but exhibit slow improvement in the dual bound. Most of the run time is spent solving LPs, specifically 97.93 % and 98.19 % of the total runtime, respectively. The cancer12_8D instance displays different behaviour, spending 92.49 % of the time in the local heuristic. Finally, cancer11_8D demonstrates a limitation of the grid approach. This instances is transformed into a problem containing almost 5 million nodes and 65 million edges, which is prohibitive for employing SCIP-Jack on a modest machine. As a result, SCIP-Jack quickly runs out of memory while solving this instance.

Table 4: Computational results for RSMTP instances

			optimal		timeout	
test set	#	solved	∅ nodes	∅ time [s]	∅ nodes	∅ gap [%]
estein1	46	46	1.0	0.3	–	–
estein10	15	15	1.0	0.3	–	–
estein20	15	15	1.5	7.3	–	–
estein30	15	15	1.0	129.9	–	–
estein40	15	15	2.2	1221.5	–	–
estein50	15	10	1.0	2766.4	2.4	0.2
estein60	15	1	1.0	6283.3	1.0	0.8
solids	5	5	14.6	7.3	–	–
cancer	14	12	1.0	139.1	1.0	29.0

3.3 The Node-Weighted Steiner Tree Problem

The *node-weighted Steiner tree problem* (*NWSTP*) is a generalization of the Steiner tree problem in graphs where the edges and, additionally, the vertices are assigned non-negative weights. The objective is to interconnect all terminals while minimizing the weight summed over both vertices and edges spanned by the corresponding tree.

The NWSTP is formally stated by: Given an undirected graph $G = (V, E)$, node costs $p : V \to \mathbb{Q}_{\geq 0}$, edge costs $c : E \to \mathbb{Q}_{\geq 0}$, and a set $T \subset V$ of terminals, the objective is to find a tree $S = (V_S, E_S)$ that spans T while minimizing

$$C(S) := \sum_{e \in E_S} c_e + \sum_{v \in V_S} p_v.$$

The NWSTP can be transformed to SAP by substituting each edge by two anti-parallel arcs. Then, observing that in a tree there cannot be more than one arc going into the same vertex, the weight of each vertex is added to the weight of each of its ingoing arcs.

Transformation 1 (NWSTP to SAP)**.**
Given an NWSTP $P = (V, E, T, c, p)$ construct an SAP $P' = (V', A', T', c', r')$ as follows:

1. *Set $V' := V$, $T' := T$, $A' := \{(v, w) \in V' \times V' : \{v, w\} \in E\}$.*
2. *Define $c' : A' \to \mathbb{Q}_{\geq 0}$ by $c'_a = c_{\{v,w\}} + p_w$, for $a = (v, w) \in A'$.*
3. *Choose a root $r' \in T'$ arbitrarily.*

Lemma 1 (NWSTP to SAP)**.** *Let $P = (V, E, T, c, p)$ be an NWSTP and $P' = (V', A', T', c')$ an SAP obtained by applying Transformation 1 on P. Denote by $\mathcal{S}$ and $\mathcal{S}'$ the set of solutions to P and P' respectively. Then $\mathcal{S}'$ can be bijectively mapped onto $\mathcal{S}$ by applying*

$$V_S := \{v \in V : \ v \in V'_{S'}\} \tag{8}$$
$$E_S := \{\{v, w\} \in E : \ (v, w) \in A'_{S'} \text{ or } (w, v) \in A'_{S'}\} \tag{9}$$

for $S' = (V'_{S'}, A'_{S'}) \in \mathcal{S}'$ and it holds:

$$c'(A'_{S'}) + p_{r'} = c(E_S) + p(V_S). \tag{10}$$

The resulting problem is an SAP, which can be handled by SCIP-Jack using the configurations described in the Steiner arborescence section.

Computational Results Two NWSTP instances derived from a computational biology application are part of the DIMACS Challenge. The two instances differ drastically in their size: the first has more than 200,000 nodes—55,000 of them terminals—and almost 2.5 million edges, while the smaller instance comprises 386 nodes, 1477 edges, and 35 terminals.

The size of the first instance is very prohibitive for SCIP-Jack. The large number of edges significantly degrades the performance of the preprocessing routines. Additionally, the memory requirements of this instance quickly exceeds the limits of SCIP-Jack when applying the default settings on a modest machine. To evaluate the ability of SCIP-Jack to solve this particular instance, a runtime of 72 hours was used on a machine with 386 GB memory. In order to reduce the presolving time, only a subset of nodes are examined in the special distance and the nearest vertex tests. This is done by randomly selecting a starting node and then examining every hundredth node from that point. After the application of the reduction techniques, the resulting graph contains 187,933 nodes and 986,703 edges. This equates to a 8.6 % and 60.4 % decrease in the number of nodes and edges respectively. SCIP-Jack fails to solve this instance to optimality, but it does achieve a primal bound of 656,970.94 with an optimality gap of 0.0049%. The much smaller second instance is solved by SCIP-Jack in the root node within 0.1 secounds.

3.4 The Prize-Collecting Steiner Tree Problem

In contrast to the classical Steiner tree problem, the required tree for the *prize-collecting Steiner tree problem* (*PCSTP*) needs only to span a (possibly empty) subset of the terminals. However, a non-negative penalty is charged for each terminal not contained in the tree. Hence, the objective is to find a tree of minimum weight, given by both the sum of its edge costs and the penalties of all terminals not spanned by the tree. For a profound discussion on the PCSTP that details real-world applications and introduces a sophisticated specialized solver, see [34].

A formal definition of the problem is stated as: Given an undirected graph $G = (V, E)$, edge-weights $c : E \to \mathbb{Q}_{\geq 0}$, and node-weights $p : V \to \mathbb{Q}_{\geq 0}$, a tree $S = (V_S, E_S)$ in G is required such that

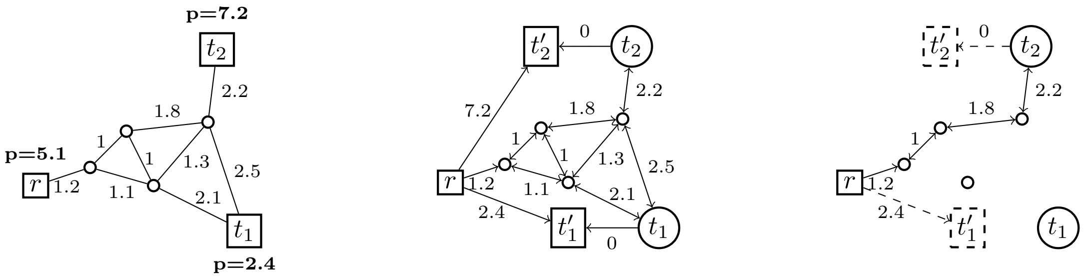
$$P(S) := \sum_{e \in E_S} c_e + \sum_{v \in V \setminus V_S} p_v \tag{11}$$

is minimized.

Before discussing the prize-collecting Steiner tree problem, we introduce a variation, the *rooted prize-collecting Steiner tree problem* (*RPCSTP*), which incorporates the additional condition that one distinguished node, denoted the *root*, must be part of every feasible solution to the problem. The RPSCTP can be transformed into an SAP as follows:

Transformation 2 (RPCSTP to SAP).
Given an RPCSTP $P = (V, E, p, r)$ construct an SAP $P' = (V', A', T', c', r')$ as follows:

1. *Set $V' := V$, $A' := \{(v, w) : \{v, w\} \in E\}$, $r' := r$ and $c' : A' \to \mathbb{Q}_{\geq 0}$ 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 E' 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 $T' := \{t'_1, ..., t'_s\}$.*



(a) RPCSTP instance (b) transformed instance (c) feasible solution

Figure 1: Illustration of a price collecting Steiner tree instance with root r (left), the equivalent SAP problem obtained by transformation 2 (middle), and a solution to the SAP instance with value 8.6 (right).

Having performed Transformation 2, for each terminal t'_i of the SAP P' there are exactly two incoming arcs $(t_i, t_i{}')$ and (r', t'). To allow a bijection between the respective solution sets of P and P' each solution $S' = (V'_{S'}, A'_{S'}) \in P'$ that contains t_i should also contain $(t_i, t_i{}')$, more succinctly:

$$\forall i \in \{1, ..., s\} : t_i \in V'_{S'} \Longrightarrow (t_i, t_i{}') \in A'_{S'} \tag{12}$$

Condition (12) is satisfied by all optimal solutions to P' and each feasible solution can be easily modified to accomplish this, concomitantly improving its solution value.

Lemma 2 (RPCSTP to SAP)**.** *Let $P' = (V', A', T', c')$ be an SAP obtained from an RPCSTP $P = (V, E, c, p)$ by applying Transformation 2. Denote by $\mathcal{S}$ and $\mathcal{S}'$ the set of solutions to P and P', satisfying condition* (12)*, respectively. P' can be mapped bijectively onto P by*

$$V_S := \{v \in V : \ v \in V'_{S'}\} \tag{13}$$
$$E_S := \{\{v, w\} \in E : \ (v, w) \in A'_{S'} \text{ or } (w, v) \in A'_{S'}\} \tag{14}$$

for $S' = (V'_{S'}, A'_{S'}) \in \mathcal{S}'$. The solution value is preserved.

Figure 1 presents a simple example of the RPCSTP and the transformation process. Note that for each of the terminals a dummy node is created with a single direction arc of zero cost added between the two. Also, there is a single direction arc from the root to each of the terminals with its selection in the tree indicating the loss of the terminal prize.

Transformation 3 can be extended to cover the PCSTP in two steps: First, an artifical root node r' is added and the transformation is performed. Second, arcs (r', t_i) with cost zero are added. They connect the artifical root with all terminals of the original problem and allow to choose a root for the solution. To this end, only one of these arcs can be part of a feasible solution, which is enforced by the following constraint:

$$\sum_{a \in \delta^+(r'), c'_a = 0} y_a = 1. \tag{15}$$

Furthermore, to allow a bijection between the original and the transformed problem, for all t_i included in a solution the arc (r', t_i) with the smallest index i is required to be part of the solution. This condition can be expressed using the following class of constraints:

$$\sum_{a \in \delta^-(t_j)} y_a + y_{(r',t_i)} \leq 1 \quad i = 1, ..., s;\ j = 1, ..., i-1. \tag{16}$$

An SAP that requires the conditions (12), (15) and (16) is henceforth referred to as *root constrained Steiner arborescence problem* (*rcSAP*). The constraints (15) and (16) can be incorporated into the cut-formulation (Formulation 1) without further alterations and each solution can be modified in order to meet condition (12). Although additional $\frac{s(s-1)}{2}$ constraints have to be introduced to fulfill (16), the solving time is considerably reduced, since concomitantly a plethora of symmetric solutions is excluded.

Transformation 3 (PCSTP to rcSAP)**.**
Given an PCSTP $P = (V, E, c, p)$ construct an rcSAP $P' = (V', A', T', c', r')$ as follows:

1. *Add a vertex v_0 to V and set $r := v_0$.*
2. *Apply Transformation 2 to obtain $P' = (V', A', T', c', r')$.*
3. *Add arcs $a = (r', t_i)$ with $c'_a := 0$ for each $t_i \in T$.*
4. *Add constraints* (15) *and* (16)*.*

Lemma 3 (PCSTP to rcSAP)**.** *Let $P = (V, E, c, p)$ be an PCSTP and $P' = (V', A', T', c', r')$ the corresponding rcSAP obtained by applying Transformation 3. Denote by $\mathcal{S}$ and $\mathcal{S}'$ the sets of solutions to P and P' respectively. Each solution $S' \in \mathcal{S}'$ can be bijectively mapped to a solution $S \in \mathcal{S}$ defined by:*

$$V_S := \{v \in V : \ v \in V'_{S'}\} \tag{17}$$

$$E_S := \{\{v, w\} \in E : \ (v, w) \in A'_{S'} \ \text{or} \ (w, v) \in A'_{S'}\}. \tag{18}$$

The solution value is preserved.

Due to its structure of both the transformed PCSTP and transformed RPCSTP only a limited set of reduction techniques are employed by SCIP-Jack, see table 2. However, all heuristic can be deployed, albeit with some alterations. For the RSPH in the case of a transformed PCSTP, i.e. an rcSAP, instead of commencing from different vertices, the starting point is always the (artficial) root. In each run all arcs between the root and non-terminals (denoted by (r', t) in Transformation 3) are temporarily removed, except for one. A tree is then computed on this new graph, using the same process as the original constructive heuristic. Instead of starting from a new terminal, the choice of the arc to remain in the graph is varied. By adjusting the shortest path data of the nodes adjacent to the root, the recomputation of the shortest paths in every iteration is not necessary. The VQ requires an adaption for both the RPCSTP and the PCSTP: All terminals are temporarily removed from the (transformed) graph and VQ is executed with all t_i, as defined in Transformation 2, marked as key vertices. Finally, the pruning has to be slightly adjusted such that (12), and for the PCSTP also (16), is satisfied by each solution.

Computational Results Table 5 shows aggregated results for three of the PCSTP test sets provided for the DIMACS Challenge. For the JMP test set all instances are solved without branching in less than one minute. Also all instances of the CRR test set are solved and only one needs a branch-and-bound search consisting of only 3 nodes. The third test set we consider is the PUCNU test set derived from the PUC test set. Since SCIP-Jack is already unable to solve

Table 5: Computational results for PCSTP instances

			optimal		timeout	
test set	#	solved	∅ nodes	∅ time [s]	∅ nodes	∅ gap [%]
JMP	34	34	1.0	2.0	–	–
CRR	80	80	1.0	7.2	–	–
PUCNU	18	7	4.0	25.3	19.6	3.9

many of the orginal instances the same results are observed for the respective PCSTP versions. Seven of the instances are solved to optimality, with three of those requiring branching. However, the remaining 11 instances terminate within the time limit with optimality gaps in the range 1.2 % to 13 %.

The average results for the RPCSTP instances are displayed in Table 6. All instances are solved at the root node in about 18.64 seconds on average. The first half, originating from the cologne1 test set, needs up to 22 seconds, the harder instances from the cologne2 test set are solved in 10 up to 341 seconds. In the DIMACS challenge, SCIP-Jack was very competitive against the other submitted solvers for this variant, achieving the best result in one of the challenge categories.

Table 6: Computational results for RPCSTP instances

			optimal	
test set	#	solved	∅ nodes	∅ time [s]
cologne1	14	14	1.0	5.3
cologne2	15	15	1.0	55.8

3.5 The Maximum-Weight Connected Subgraph Problem

At first glance, the maximum-weight connected subgraph problem (*MWCSP*) bears little resemblance to the Steiner problems introduced so far: Given an undirected graph (V, E) with (possibly negative) node weights p, the objective is to find a tree that maximizes the sum of its node weights. However, it is possible to transform this problem into a prize-collecting Steiner tree problem. One transformation is given in [35]. In this paper, we present an alternative transformation which leads to a significant reduction in the number of terminals for the resulting PCSTP.

In the following it is assumed that at least one vertex is assigned a negative and one a positive cost. Otherwise the problem can be reduced to finding a minimum spanning tree.

Transformation 4 (MWCSP to rcSAP)**.**
Let $P = (V, E, p)$ be an MWCSP, construct an rcSAP $P'' = (V'', A'', T'', c'', r'')$:

1. *Set $V' := V$, $A' := \{(v, w) : \{v, w\} \in E\}$.*
2. *$c' : A' \to \mathbb{Q}_{\geq 0}$ such that for $a = (v, w) \in A'$:*
   $$c'_a = \begin{cases} -p_w, & \text{if } p_w < 0 \\ 0, & \text{otherwise} \end{cases}$$
3. *$p' : V' \to \mathbb{Q}_{\geq 0}$ such that for $v \in V'$:*
   $$p'(v) = \begin{cases} p_v, & \text{if } p_v > 0 \\ 0, & \text{otherwise} \end{cases}$$

4. *Perform Transformation 3 to (V', A', c', p'), slightly changed in such a way, that in step 2 instead of constructing a new arc set, A' is being used. The resulting rcSAP gives us $P'' = (V'', A'', T'', c'', r'')$.*

Lemma 4 (MWCSP to rcSAP)**.** *Let $P = (V, E, p)$ be an MWCSP and $P'' = (V'', A'', T'', c'', r'')$ an rcSAP obtained from P by Transformation 4. Then each solution S'' to P'' can be bijectively mapped to a solution S to P. The latter is obtained by:*

$$V_S := \{v \in V : \; v \in V''_{S''}\} \tag{19}$$

$$E_S := \{\{v, w\} \in E : \; (v, w) \in A''_{S''} \text{ or } (w, v) \in A''_{S''}\} \tag{20}$$

Furthermore, for the objective value $C(S)$ of S and the objective value $C''(S'')$ of S'' the following equality holds:

$$C(S) = \sum_{v \in V : p_v > 0} p_v - C''(S''). \tag{21}$$

Since most of the vertex weights are nonpositive for all (real-world) DIMACS instances, Transformation 4 results in problems with significantly less terminals compared to the tranformation described in [35]. The differences in the number of terminals resulting from the two transformations are presented in Table 7. The computational settings of SCIP-Jack are identical for those of the PCSTP, except for the use of VQ, as the latter can not so easily be adapted to handle anti-parallel arcs of different weight and is therefore disabled.

Computational Results We performed computational experiments on the ACTMOD test set containing eight instances and the 72 instances of the JMPALMK test set, see Table 8. The results demonstrate the ability of SCIP-Jack to solve this problem variant – solving all but one of the ACTMOD instances to optimality at the root node. For the remaining instance SCIP-Jack is unable to construct an optimal solution at the root node and 3180 branch-and-bound nodes are processed until it is found; optimality is proven only eight nodes later after 4287.2 seconds. It should be noted that the performance of SCIP-Jack on the ACTMODPC test set, which contains the same problems, but already transformed to PCSTP by the transformation described in [35], is significantly worse.

Of the JMPALMK test set, SCIP-Jack solves all but one instance to optimality within the time limit. Similar to the ACTMOD test set, all instances are solved to optimality at the root node. To determine whether SCIP-Jack is able to solve the remaining instance, a longer time limit of 36 hours has been applied. Similar to the large RSMTP instances, a good primal solution is found by the recombination heuristic after 421 seconds and the remaining time is spent solving LPs to improve the dual bound.

Table 7: Number of terminals after transforming

instance	Transformation 4	transformation from [35]
drosophila001	71	5226
drosophila005	194	5226
drosophila0075	250	5226
HCMV	55	3863
lymphoma	67	2034
metabol_expr_mice_1	150	3523
metabol_expr_mice_2	85	3514
metabol_expr_mice_3	114	2853

Table 8: Computational results for MWCS instances

test set	#	solved	optimal ∅ nodes	optimal ∅ time [s]	timeout ∅ nodes	timeout ∅ gap [%]
ACTMOD	8	8	4.0	87.8	–	–
JMPALMK	72	71	1.0	91.2	1.0	0.1

3.6 The Degree-Constrained Steiner Tree Problem

The *degree-constrained Steiner tree problem* (*DCSTP*), is an STP with an additional degree constraint for each node. The objective is to find a minimum solution to the STP such that the degree of each node in the Steiner tree is not larger than the given limit. The DCSTP is implemented by just adding the additional degree constraints for each node as linear constraints to the directed-cut-formulation (Formulation 1). Note that this degree restriction does not comply with the usual SCIP-Jack presolving routines so that we do not perform presolving on these instances. We use a variation of the constructive heuristic, altered in such a way that while choosing a new (shortest) path to be added to the current tree it is checked: First, whether attaching this path would violate any degree constraints and second, whether after having added this path at least one additional edge could be added (or all terminals are spanned). If no such path can be found, a vertex of the tree is pseudo randomly chosen that allows to add at least one adjacent edge, and such an edge leading to a vertex of high degree and being of small cost is chosen.

Computational Results Computational experiments are performed on the 20 instances in the TreeFam test set of the DIMACS Challenge with a time limit of two hours. The results for the individual instances are presented in Table 9. Besides the size of the problem, we print dual and primal bound, the gap in percent, the number of cut separation rounds (column C) and processed branch-and-bound nodes (column N), as well as the solving time in seconds. For instances solved to optimality, we omit the gap and print the optimal objective value in bold in the center of the primal and dual bound columns (or infeasible, if infeasibilty was proven). SCIP-Jack finds the optimal solution to five instances and proves the infeasibilty of another two. The remaining 13 instances are unable to be solved by SCIP-Jack within the time limit. Most of the time for these instances is spent in the added STP constraint handlers, 52.55 % on average. Also, an average of 706.81 branch-and-bound nodes are required for these instance. This result is attributed to the lack of a more refined constructive heuristic.

Table 9. Detailed computational results for the DCSTP, test set TreeFam.

Instance	$\|V\|$	$\|A\|$	$\|T\|$	Dual	Primal	Gap %	C	N	t [s]
TF101057-t1	52	2652	35	**infeasible**			0	1	0.0
TF101057-t3	52	2652	35	**2756**			41	1361	23.5
TF101125-t1	304	92112	155	**infeasible**			0	1	2.2
TF101125-t3	304	92112	155	53676.2948	55615	3.6	104	1225	>7200.3
TF101202-t1	188	35156	72	79309.1733	80834	1.9	93	4480	>7200.2
TF101202-t3	188	35156	72	77771.3126	78233	0.6	195	2755	>7200.1
TF102003-t1	832	691392	407	190042.514	393395	107.0	40	5	>7201.5
TF102003-t3	832	691392	407	176144.713	189504	7.6	52	5	>7226.3
TF105035-t1	237	55932	104	34525.1898	40597	17.6	68	4261	>7200.2
TF105035-t3	237	55932	104	32436.7151	33018	1.8	103	1389	>7200.9
TF105272-t1	476	226100	223	131135.094	268525	104.8	56	82	>7203.5
TF105272-t3	476	226100	223	122819.718	129316	5.3	93	43	>7200.6

cont. next page

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Dual	Primal	Gap %	C	N	t [s]
TF105419-t1	55	2970	24	**18668**			31	23987	331.1
TF105419-t3	55	2970	24	**18223**			57	41	6.1
TF105897-t1	314	98282	133	105417.543	170309	61.6	59	331	>7202.2
TF105897-t3	314	98282	133	96192.5645	98529	2.4	78	502	>7201.2
TF106403-t1	119	14042	46	**54124**			89	1071	364.3
TF106403-t3	119	14042	46	**53760**			158	14	57.3
TF106478-t1	130	16770	54	54970.8772	55274	0.6	62	56359	>7200.1
TF106478-t3	130	16770	54	54750.0926	55007	0.5	89	100830	>7200.0

3.7 The Group Steiner Tree Problem

The *group Steiner tree problem* (*GSTP*) is another generalization of the Steiner tree problem, originating from VLSI design [36], where the concept of terminals as a set of vertices to be interconnected is extended to a set of vertex groups: Given an undirected graph $G = (V, E)$, edge costs $c : E \to \mathbb{Q}_{\geq 0}$ and a series of vertex subsets $T_1, ...T_s \subset V$, $s \in \mathbb{N}$, a minimum cost tree spanning at least one vertex of each subset is required. By interpreting each terminal t as a subset $\{t\}$, every STP can be considered as an GSTP, the latter likewise being $\mathcal{NP}$-hard. On the other hand, it is possible to transform each GTSP instance $(V, E, T_1, .., T_s, c)$ to an STP using the following scheme:

Transformation 5 (GSTP to STP)**.**
Given an GSTP $P = (V, E, T_1, ...T_s, c)$ construct an STP $P' = (V', E', T', c')$ as follows:

1. *Set $V' := V$, $E' := E$, $T' = \emptyset$, $c' := c$, $K := \sum_{e \in E} c_e + 1$.*
2. *For $i = 1, ..., s$ add a new node t'_i to V' and T' and for all $v_j \in T_i$ add an edge $e = \{t'_i, v_j\}$, with $c'_e := K$.*

Let $(V, E, T_1, ...T_s, c)$ be an GSTP and $P' = (V', A', T', c')$ an STP obtained by applying Transformation 5 on P. A solution S' to P' can then be reduced to a solution S to P by deleting all vertices and edges of S not in (V, E). The GSTP P can in this way be solved on the STP P' as shown in [36] and [37].

This approach has already been deployed by [38] to solve group Steiner tree problems and demonstrated to be competitive with specialized solvers at the time of publishing. In the case of SCIP-JACK, to solve an GSTP Transformation 5 is applied and the resulting problem is treated as a normal STP and is solved without any alteration.

Computational Results Computational results for two test sets of unpublished group Steiner tree instances derived from a real world wire routing problem are presented in Table 10. SCIP-Jack solves all but two of the first test set, with runtimes ranging from 3.3 to 563 seconds. Five of the instances solved to optimality only require a single node, with the remaining instance solved in 403 nodes. Two instances of this set terminate within the time limit of two hours with large optimality gaps, gstp34f2 and gstp39f2 with 1.8 % and 9.1 % respectively. The same performance does not recur on the second test set. Two instances, gstp73f2 and andre76f2, are

Table 10: Computational results for GSTP instances

			optimal		timeout	
test set	#	solved	∅ nodes	∅ time [s]	∅ nodes	∅ gap [%]
GSTP1	8	6	3.8	48.2	255.2	5.5
GSTP2	10	2	1.8	6692.3	17.9	3.4

solved within the time limit, with all others terminating after many branch-and-bound nodes – 17.95 nodes on average.

3.8 The Hop-Constrained Directed Steiner Tree Problem

The *hop-constrained directed Steiner tree problem* (*HCDSTP*) searches for an SAP with the additional constraint that the number of selected arcs must not exceed a predetermined bound, called *hop limit*. The cut formulation (Formulation 1) used by SCIP-Jack is simply extended to cover this variation by adding one extra linear inequality bounding the sum of all binary arc variables.

Still, the hop limit has significant implications for the preprocessing and heuristics approaches. Many of the presolving techniques remove or include edges from the graph if a less costly path can be found, regardless whether this involves taking more edges. Hence, the preprocessing techniques of this type currently implemented in SCIP-Jack are not able to produce a valid graph reduction. However, in order to perform some reductions on the HCSTP instances, a modified bound test, as described in Section 2, is employed.

Similar to the presolving techniques, the heuristics implemented in SCIP-Jack for the other variants do not take into account the hop limit. As such, any identified solution may not be feasible. Therefore, a simple variation of the constructive heuristic is used for this STP variant: Each arc a, having original costs c_a, is assigned the new cost $c'_a := 1 + \lambda \frac{c_a}{c_{max}}$, with $\lambda \in \mathbb{Q}_+$ and $c_{max} := \max_{a \in A} c_a$. Initially λ is set to 3 but its value is decreased or increased after each iteration of the constructive heuristic, depending on whether the last computed solution exceeds or is below the hop limit, respectively. This modification to λ is performed relatively to the deviation of the number of edges from the hop limit.

Computational Results Three different test sets are used for the computational experiments consisting of the gr12, gr14 and gr16 instances used in the evaluation of the DIMACS challenge. The gr12 test set contains 19 instances and SCIP-Jack is able to solve all of them in less than 722 seconds. On average, these instances require 15.29 seconds of runtime and 2.31 nodes. Only four instances of this test set were not solved at the root node. This performance is not repeated for the gr14 test set, with only six instances solved to optimality within the time limit. All of the unsolved instances terminate with large optimality gaps, ranging from 9.6 % to 38 %, after 34.62 nodes on average. Finally, SCIP-Jack is unable to solve any of the instances from the gr16 test set. All of these instance terminate within the time limit with a optimality gap of at least 27.4 %. For these larger instances, SCIP-Jack terminates while solving the root LP for all but one instance. One possible cause of this performance is the inability to apply the common reduction techniques and heuristics implemented in SCIP-Jack.

Table 11: Computational results for HCDSTP instances

			optimal		timeout	
test set	#	solved	∅ nodes	∅ time [s]	∅ nodes	∅ gap [%]
gr12	19	19	2.3	15.3	–	–
gr14	21	6	9.2	1055.6	57.8	20.5
gr16	20	0	–	–	1.1	81.3

Table 12: Comparison of SCIP-Jack with SoPlex and CPLEX as LP solver.

		SCIP-Jack			SCIP-Jack/CPLEX			relative change [%]		
Instance	Type	C	N	t [s]	C	N	t [s]	C	N	t [s]
cc3-4p	STP	159	22265	552.5	160	13113	166.9	+0.63	-41.10	-69.79
cc6-2u	STP	80	19	12.9	82	27	9.4	+2.50	+42.11	-27.13
i320-044	STP	154	1	5.2	154	1	5.8	–	–	+11.54
i320-245	STP	112	1217	3827.7	121	621	943.9	+8.04	-48.97	-75.34
i640-124	STP	1008	19	4886.8	854	35	3861.8	-15.28	+84.21	-20.97
i640-232	STP	60	1	20.5	59	1	9.2	-1.67	–	-55.12
l030a	STP	127	1	2968.2	122	3	1499.2	-3.94	+200.00	-49.49
l065a	STP	76	1	44.7	78	1	13.9	+2.63	–	-68.90
gene442	SAP	8	1	0.1	9	1	0.1	+12.50	–	–
gene575	SAP	22	1	0.2	21	1	0.2	-4.55	–	–
estein1-33	RSMTP	49	1	5.0	56	1	2.4	+14.29	–	-52.00
estein20-2	RSMTP	63	1	1.9	63	1	1.2	–	–	-36.84
estein20-3	RSMTP	87	11	25.0	90	5	10.4	+3.45	-54.55	-58.40
estein30-11	RSMTP	103	1	42.8	98	1	10.8	-4.85	–	-74.77
estein30-3	RSMTP	269	1	293.9	238	1	61.9	-11.52	–	-78.94
estein40-0	RSMTP	148	1	425.3	136	1	87.8	-8.11	–	-79.36
estein40-8	RSMTP	269	1	2633.1	274	1	438.0	+1.86	–	-83.37
estein50-9	RSMTP	270	1	2949.8	277	1	885.8	+2.59	–	-69.97
dodecahedron	RSMTP	138	14249	6269.7	122	5849	1002.7	-11.59	-58.95	-84.01
icosahedron	RSMTP	42	7	5.2	46	5	1.5	+9.52	-28.57	-71.15
cancer7_6D	RSMTP	516	1	2834.5	486	1	449.6	-5.81	–	-84.14
cancer9_6D	RSMTP	133	1	14.2	148	1	18.2	+11.28	–	+28.17
TF105419-t3	DCSTP	57	41	6.1	54	68	4.0	-5.26	+65.85	-34.43
TF106403-t1	DCSTP	89	1071	364.3	73	784	237.4	-17.98	-26.80	-34.83
gstp33f2	GSTP	41	1	4.4	41	1	3.0	–	–	-31.82
gstp38f2	GSTP	62	403	4064.5	116	7	171.5	+87.10	-98.26	-95.78
K200	PCSTP	17	1	1.4	18	1	1.5	+5.88	–	+7.14
K400.10	PCSTP	53	1	14.9	53	1	8.0	–	–	-46.31
drosophila001	MWCSP	1626	3188	4287.2	494	24	1589.8	-69.62	-99.25	-62.92
lymphoma	MWCSP	91	1	9.8	85	1	11.3	-6.59	–	+15.31
1000-a-0.6-d-0.5-e-0.25	MWCSP	66	1	2485.6	26	1	418.8	-60.61	–	-83.15
500-a-0.62-d-0.5-e-0.25	MWCSP	7	1	7.5	6	1	9.4	-14.29	–	+25.33
i104M2	RPCSTP	159	1	2.6	161	1	2.3	+1.26	–	-11.54
i203M4	RPCSTP	459	1	341.9	440	1	92.8	-4.14	–	-72.86
C20-A	PCSTP	14	1	14.8	9	1	12.4	-35.71	–	-16.22
D10-B	PCSTP	297	1	935.1	264	1	654.3	-11.11	–	-30.03
cc3-5nu	PCSTP	38	1	1.1	74	1	1.0	+94.74	–	-9.09
cc6-3nu	PCSTP	264	8	833.9	283	4	424.5	+7.20	-50.00	-49.09
wo10-cr200-se8	HCDSTP	319	101	722.0	430	5	87.3	+34.80	-95.05	-87.91
wo11-cr200-se3	HCDSTP	724	83	2728.5	781	3	625.6	+7.87	-96.39	-77.07
wo12-cr100-se7	HCDSTP	89	1	2.1	86	1	1.4	-3.37	–	-33.33
wo12-cr100-se9	HCDSTP	392	1	302.1	421	1	115.4	+7.40	–	-61.80
sh. geom mean		107.48	8.09	89.35	104.04	5.13	39.57	-3.21	-36.52	-55.71

3.9 Using CPLEX as underlying LP solver

SCIP-Jack is an extension of SCIP and as such provides the branch-and-cut search, but requires an external LP solver for solving the linear programming relaxations. Until now, we have used SoPlex for this, which is the default solver employed by SCIP. However, SCIP provides interfaces to many different LP solvers, among them the commercial ones. In this section we shortly discuss the impact of exchanging the academic LP solver SoPlex for the commercial solver CPLEX 12.6[1].

To this end, we selected two instances from most of the previously discussed test sets, one where SCIP-Jack had a long running time, but solved it to optimality well before the time limit, and one which was solved fast, but still needed at least one second (except for the SAP instances, which were all solved in fractions of a second). By this selection we left space for improvements as well as deteriorations when running SCIP-Jack with CPLEX as LP solver. A different LP

[1] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

solver might lead to different optimal LP solutions being computed, which can change the overall branch-and-cut process as we can observe in Table 12. There is much variation in the number of cutting plane separation rounds at the root nodes, but stays almost the same on average. On the other hand, the number of branch-and-bound nodes is reduced by 36.5 % in the shifted geometric mean when using CPLEX. And finally, the solving time is smaller with CPLEX for most of the instances, leading to a reduction of the average solving time by more than 55 %. We want to note that these results may be biased by selecting the instances based on the results of only one solver (and in particular by partly choosing exactly the instances SCIP-Jack seems to have troubles with), but they definitely show a potential to speed up SCIP-Jack by using a commercial LP solver.

4 From single core to distributed parallel

SCIP has two parallel extensions ParaSCIP [39] and FiberSCIP [40], which are built by using the *Ubiquity Generator Framework* (UG) [40]. In order to parallelize a problem-specific solver, users of SCIP can simply modify their developed plugins by adding a small glue code and linking to one of the UG libraries. This glue code consists of an additional class with a function that makes calls to include all SCIP plugins required for the sequential version of the code. Importantly, no modification to the sequential version of the problem-specific solver is required.

In this way, users obtain their own problem specific parallel optimization solver, which can do parallel tree search on a distributed memory computing environment. The main features of UG are: several *ramp-up* mechanisms (the ramp-up is the process from the begining of the computation until all available solvers become busy), a dynamic load balancing mechanism for parallel tree search and a check-pointing and restarting mechanism. For more details about the parallelization provided by UG, see [39, 40].

We present computational results for the PUC test set from SteinLib. However, it must be noted that the parallel version of SCIP-Jack can handle all of the variants presented throughout this paper. The main purpose of the parallel runs is to provide optimal solutions to as many instances as possible. As mentioned above, the parallelization of a problem-specific solver only requires a small glue code. As such, the parallel version of SCIP-Jack is identical to the sequential version. Using this simple approach, it is possible to employ large supercomputing resources to apply SCIP-Jack to solve computationally difficult Steiner tree problems. For the computations, we used clusters and supercomputers as they were available. The largest computation performed for these experiments involved up to 864 cores, which was only required for eight instances (`bip52p, bip62u, bipa2p, bipa2u, cc11-2p, cc12-2p, cc3-12p, hc9p`). However, all other computations were conducted with 192 or less solvers. In contrast to the previous experiments, we used CPLEX 12.6 as the underlying LP solver. As a reference to the scalability of ParaSCIP, the largest computation previously performed was an 80,000 cores run on Titan at ORNL. We expect SCIP-Jack to also run on such a large scale computing environment, though at this stage we have only conducted relatively small scale computational experiments.

Table 13 shows the results on the instances of the PUC test set as of 17th April 2015. We list the number of nodes, edges, and terminals, as well as the best primal bound known at the beginning of the challenge (August 2014), and the primal solution value obtained by our experiments with the parallel version of SCIP-Jack, which employs the LP solver of CPLEX 12.6. Prior to the experiments performed using SCIP-Jack, 32 instances of the PUC test set remained unsolved. Three of these instances have been solved by SCIP-Jack to proven optimality, which have been underlined and marked with an asterisk in Table 13. For a further

Table 13: Primal bound improvements on the PUC instances

instance	$\|V\|$	$\|E\|$	$\|T\|$	best	SCIP-Jack	instance	$\|V\|$	$\|E\|$	$\|T\|$	best	SCIP-Jack
bip42p	1200	3982	200	24657	24657*	cc3-5u	125	750	13	36	36*
bip42u	1200	3982	200	236	236*	cc5-3p	243	1215	27	7299	7299*
bip52p	2200	7997	200	24535	**24526**	cc5-3u	243	1215	27	71	71*
bip52u	2200	7997	200	234	234	cc6-2p	64	192	12	3271	3271*
bip62p	1200	10002	200	22870	**22843**	cc6-2u	64	192	12	32	32*
bip62u	1200	10002	200	220	**219**	cc6-3p	729	4368	76	20456	**20270***
bipa2p	3300	18073	300	35379	**35326**	cc6-3u	729	4368	76	197	197*
bipa2u	3300	18073	300	341	**338**	cc7-3p	2187	15308	222	57088	57117
bipe2p	550	5013	50	5616	5616*	cc7-3u	2187	15308	222	552	552
bipe2u	550	5013	50	54	54*	cc9-2p	512	2304	64	17296	**17199**
cc10-2p	1024	5120	135	35379	**35227**	cc9-2u	512	2304	64	167	167*
cc10-2u	1024	5120	135	342	343	hc10p	1024	5120	512	60494	**59797**
cc11-2p	2048	11263	244	63826	**63636**	hc10u	1024	5120	512	581	**575**
cc11-2u	2048	11263	244	614	618	hc11p	2048	11264	1024	119779	**119689**
cc12-2p	4096	24574	473	121106	122099	hc11u	2048	11264	1024	1154	**1151**
cc12-2u	4096	24574	473	1179	1184	hc12p	4096	24576	2048	236949	**236080**
cc3-10p	1000	13500	50	12860	**12837**	hc12u	4096	24576	2048	2275	**2262**
cc3-10u	1000	13500	50	125	126	hc6p	64	192	32	4003	4003*
cc3-11p	1331	19965	61	15609	15648	hc6u	64	192	32	39	39*
cc3-11u	1331	19965	61	153	153	hc7p	128	448	64	7905	7905*
cc3-12p	1728	28512	74	18838	18997	hc7u	128	448	64	77	77*
cc3-12u	1728	28512	74	186	187	hc8p	256	1024	128	15322	15322*
cc3-4p	64	288	8	2338	2338*	hc8u	256	1024	128	148	148*
cc3-4u	64	288	8	23	23*	hc9p	512	2304	256	30258	**30242**
cc3-5p	125	750	13	3661	3661*	hc9u	512	2304	256	292	292

16 instances, SCIP-Jack improved the best known solution. All instances where the best known primal bound has been improved are marked in bold. Finally, all previously solved instances of the PUC test set have also been solved by SCIP-Jack to proven optimality, which have been marked by an asterisk (without underline). This demonstrates an overall strong performance of the parallel version of SCIP-Jack in solving the computationally difficult set of instances.

5 Conclusions

We have shown that embedding a 15-year old solver for Steiner trees into a state-of-the-art MIP solving framework can have a significant impact in several dimensions. First, the amount of problem specific code is notably reduced while at the same time the number of general solution methods available, e.g., cutting planes, has increased and will be kept up-to-date just by the continuous improvements in the framework. Furthermore, the opportunity to solve instances in a massively parallel distributed memory environment has been added at minimal cost.

The use of a general MIP solver allows us to be extremely flexible with the model to be solved. We were able to support solving ten variants of the Steiner tree problem with nearly the same code, and the support of further restrictions in the model is straightforward. We attempted to solve the open instances of the difficult PUC test set using the massively parallel extensions included with SCIP. As a result, we were able to solve three previously unsolved instances and improve the best known solution for another 16 instances. Still, there is potential for future work to improve the performance of the solver. In particular, the inclusion of recently developed reduction techniques is expected to further reduce the solution runtimes. Also, there are many reduction techniques and heuristic approaches that can be employed to specific variants. Using the plugin structure of SCIP we hope to include some of these heuristics and reduction techniques in the future.

And finally, to the best of our knowledge this is the first time that a powerful exact Steiner tree solver is available in source code to the scientific community. We hope that this will foster the use of Steiner trees in modelling real-world phenomena as has already been the case in genetics.

6 Acknowledgements

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). It has been further supported by a Google Faculty Research Award.

We are thankful to The Institute of Statistical Mathematics (ISM), Tokyo, and the HLRN-III facility at ZIB, Berlin, for providing supercomputing resources and their respective staff for their helpful support. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

References

[1] Karp, R.: Reducibility among combinatorial problems. In Miller, R., Thatcher, J., eds.: Complexity of Computer Computations. Plenum Press (1972) 85–103

[2] Koch, T., Martin, A., Voß, S.: SteinLib: An updated library on Steiner tree problems in graphs. In Du, D.Z., Cheng, X., eds.: Steiner Trees in Industries. Kluwer (2001) 285–325

[3] Koch, T., Martin, A.: Solving Steiner tree problems in graphs to optimality. Networks **32** (1998) 207–232

[4] Achterberg, T.: SCIP: Solving constraint integer programs. Mathematical Programming Computation **1**(1) (2009) 1–41

[5] Borndörfer, R., Hoàng, N.D., Karbstein, M., Koch, T., Martin, A.: How many Steiner terminals can you connect in 20 years? In Jünger, M., Reinelt, G., eds.: Facets of Combinatorial Optimization. Springer (2013) 215–244

[6] Koch, T., Martin, A., Pfetsch, M.E.: Progress in academic computational integer programming. In Jünger, M., Reinelt, G., eds.: Facets of Combinatorial Optimization. Springer (2013) 483–506

[7] Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J.: The traveling salesman problem: a computational study. Princeton University Press, Princeton, New Jersey (2011)

[8] Rosseti, I., de Aragão, M., Ribeiro, C., Uchoa, E., Werneck, R.: New benchmark instances for the Steiner problem in graphs. In: Extended Abstracts of the 4th Metaheuristics International Conference (MIC'2001), Porto (2001) 557–561

[9] Duin, C.: Steiner Problems in Graphs. PhD thesis, University of Amsterdam (1993)

[10] Polzin, T.: Algorithms for the Steiner problem in networks. PhD thesis, Saarland University (2004)

[11] Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J.: On the solution of traveling salesman problems. Documenta Mathematica Journal der Deutschen Mathematiker-Vereinigung **Extra Volume ICM III** (1998) 645–656

[12] Mittelmann, H.: Benchmarks for optimization software (last accessed Mar. 9, 2015) `http://plato.asu.edu/bench.html`.

[13] Beasley, J.E.: An algorithm for the steiner problem in graphs. Networks **14**(1) (1984) 147–159

[14] Duin, C., Volgenant, A.: An edge elimination test for the steiner problem in graphs. Operations Research Letters **8**(2) (1989) 79 – 83

[15] Duin, C.W., Volgenant, A.: Reduction tests for the steiner problem in graphs. Networks **19**(5) (1989) 549–567

[16] Maculan, N., Souza, P., Candia Vejar, A.: An approach for the steiner problem in directed graphs. Annals of Operations Research **33**(6) (1991) 471–480

[17] Achterberg, T., Berthold, T.: Hybrid branching. In van Hoeve, W.J., Hooker, J.N., eds.: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 6th International Conference, CPAIOR 2009. Volume 5547 of Lecture Notes in Computer Science., Springer (May 2009) 309–311

[18] Achterberg, T.: Constraint Integer Programming. PhD thesis, Technische Universität Berlin (2007)

[19] Takahashi, H., A., M.: An approximate solution for the steiner problem in graphs. Math. Jap. **24** (1980) 573 – 577

[20] Uchoa, E., Werneck, R.F.F.: Fast local search for steiner trees in graphs. In Blelloch, G.E., Halperin, D., eds.: ALENEX, SIAM (2010) 1–10

[21] Ribeiro, C.C., Uchoa, E., Werneck, R.F.: A hybrid grasp with perturbations for the steiner problem in graphs. INFORMS JOURNAL ON COMPUTING **14** (2001) 200–2

[22] Wunderling, R.: Paralleler und objektorientierter Simplex-Algorithmus. PhD thesis, Technische Universität Berlin (1996)

[23] Leitner, M., Ljubic, I., Luipersbeck, M., Prossegger, M., Resch, M.: New real-world instances for the steiner tree problem in graphs. Technical report, ISOR, Uni Wien (2014)

[24] Johnston, J., Kelley, R., Crawford, T., Morton, D., Agarwala, R., Koch, T., Schäffer, A., Francomano, C., Biesecker, L.: A novel nemaline myopathy in the Amish caused by a mutation in troponin T1. American Journal of Human Genetics (October 2000) 814–821

[25] Garey, M., Johnson, D.: The rectilinear Steiner tree problem is NP-complete. SIAM Journal of Applied Mathematics **32** (1977) 826–834

[26] Warme, D., Winter, P., Zachariasen, M.: Exact algorithms for plane Steiner tree problems: A computational study. In Du, D.Z., Smith, J., Rubinstein, J., eds.: Advances in Steiner Trees. Kluwer (2000) 81–116

[27] Zachariasen, M., Rohe, A.: Rectilinear group steiner trees and applications in VLSI design. Technical Report 00906, Institute for Discrete Mathematics (2000)

[28] Emanet, N.: The rectilinear steiner tree problem (2010)

[29] Chowdhury, S.A., Shackney, S., Heselmeyer-Haddad, K., Ried, T., Schffer, A.A., Schwartz, R.: Phylogenetic analysis of multiprobe fluorescence in situ hybridization data from tumor cell populations. Bioinformatics **29**(13) (2013) 189–198

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

[31] Snyder, T.L.: On the exact location of steiner points in general dimension. SIAM J. Comput. **21**(1) (1992) 163–180

[32] Winter, P.: Reductions for the rectilinear steiner tree problem. Networks **26**(4) (1995) 187–198

[33] Zachariasen, M.: Rectilinear full steiner tree generation. Networks **33**(2) (1999) 125–143

[34] Ljubi, I.: Exact and memetic algorithms for two network design problems (2004)

[35] Dittrich, M.T., Klau, G.W., Rosenwald, A., Dandekar, T., Müller, T.: Identifying functional modules in protein-protein interaction networks: an integrated exact approach. In: ISMB. (2008) 223–231

[36] Hwang, F., Richards, D., Winter, P.: The Steiner tree problem. Annals of Discrete Mathematics **53** (1992)

[37] Vo, S.: A survey on some generalizations of steiner's problem. 1st Balkan Conference on Operational Research Proceedings **1** (1988) 41–51

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

[39] Shinano, Y., Achterberg, T., Berthold, T., Heinz, S., Koch, T.: ParaSCIP: a parallel extension of SCIP. In Bischof, C., Hegering, H.G., Nagel, W., Wittum, G., eds.: Competence in High Performance Computing 2010. (2012) 135 – 148

[40] Shinano, Y., Heinz, S., Vigerske, S., Winkler, M.: FiberSCIP - a shared memory parallelization of SCIP. Technical Report 13-55, ZIB, Takustr.7, 14195 Berlin (2013)

A Proofs

This section is concerned with providing proofs to the Lemmata stated in the course of this paper. First, the transformation used by SCIP-Jack to convert a given STP to an SAP is specified. This transformation is well-known, see, e.g. [3], but we provide a formal proof since our subsequent proofs re-use the same arguments. Then, for each transformation introduced in this paper, a one-to-one correspondence between the solution sets of the original and the transformed problem is proven as well as the linear relation between the respective solutions values. This implies that all these problems can be solved on their transformed solution spaces.

Transformation 0 (STP to SAP)**.**
Given an STP $P = (V, E, T, c)$, construct an SAP $P' = (V', A', T', c', r')$ as follows:

1. *Set $V' := V$, $T' := T$, $A' := \{(v, w) \in V' \times V' : \{v, w\} \in E\}$.*
2. *Define $c' : A' \to \mathbb{Q}_{\geq 0}$ by $c'_a = c_{\{v,w\}}$, for $a = (v, w) \in A'$.*
3. *Choose a root $r' \in T'$ arbitrarily.*

Lemma 0 (STP to SAP). *Let $P = (V, E, T, c)$ be an STP and $P' = (V', A', T', c')$ an SAP obtained by applying Transformation 0 on P. Denote by $\mathcal{S}$ and $\mathcal{S}'$ the sets of solutions to P and P' respectively. Then $\mathcal{S}'$ can be mapped bijectively onto $\mathcal{S}$ by applying*

$$V_S := \{v \in V : \ v \in V'_{S'}\} \tag{22}$$

$$E_S := \{\{v, w\} \in E : \ (v, w) \in A'_{S'} \ or \ (w, v) \in A'_{S'}\} \tag{23}$$

for $(V'_{S'}, A'_{S'}) \in \mathcal{S}'$, at equal costs.

Proof. First, it can be observed that (22) and (23) form indeed a mapping $\mathcal{S}' \to \mathcal{S}$, since each arc of a solution to P' is substituted by its undirected counterpart. To see the one-to-one correspondence let $S = (V_S, E_S) \in \mathcal{S}$ and procede as follows:
Surjective. Initially set $V'_{S'} := V_S$ and $A'_{S'} := \emptyset$. Traverse (V_S, E_S), e.g. using breadth-first search, starting from r' and add for each $w \in V_S$ visited from $v \in V_S$ the arc (v, w) to $A'_{S'}$. $S' := (V'_{S'}, A'_{S'})$ is a solution to P' and by applying (22) and (23), S is obtained.
Injective. S' is the only solution to P' that is mapped by (22) and (23) to S: Each $\tilde{S}' \in \mathcal{S}'$, $\tilde{S}' \neq S'$ contains at least one arc (v, w) such that $(v, w) \notin A'_{S'}$ and $(w, v) \notin A'_{S'}$, since only substituting arcs in $A'_{S'}$ by there anti-parallel counterparts would not allow directed paths from the root to all vertices. Therefore, $\tilde{S}'$ is not mapped onto S.
Finally, since for each $\{v, w\} \in E_S$ either $(v, w) \in A'_{S'}$ or $(w, v) \in A'_{S'}$ and vice versa, the costs of S' and S are equal. □

A.1 Proof of Lemma 1 (NWSTP to SAP)

Proof. Proving that (8) and (9) form a bijection is equivalent to the procedure in the proof of Lemma 0, since compared to the latter only the weights are altered. To acknowledge (10) one readily observes that for each node of S' except for the root there is exactly one incoming arc, so:

$$\sum_{(v,w) \in A'_{S'}} c'_{(v,w)} = \sum_{(v,w) \in A'_{S'}} \big(c_{\{v,w\}} + p_w\big) = \sum_{\{v,w\} \in E_S} c_{\{v,w\}} + \sum_{w \in V_S} p_w - p_{r'},$$

which implies (10). □

A.2 Proof of Lemma 2 (RPCSTP to SAP)

Proof. To acknowledge that (13) and (14) constitute a mapping $\mathcal{S}' \to \mathcal{S}$ it can be observed that first the root node is conserved and second the set of all arcs corresponding to edges in the original graph (V, E) forms a tree. To prove that a bijection is given, let $S = (V_S, E_S) \in \mathcal{S}$ and $T = \{t_1, ..., t_s\}$ as defined in Transformation 2.
Surjective. Initially, set $V'_{S'} := V_S$ and $A'_{S'} := \emptyset$. Analogously to the proof of Lemma 0, add for each edge in E_S an arc to $A'_{S'}$ in such a way that finally there is for each $v' \in V'_{S'}$ a directed path from r' to v'. Thereafter, for each $i \in \{1, ...s\}$ set $a_i := (t_i, t'_i)$ if $t_i \in V_S$, otherwise $a_i := (r', t'_i)$ and add a_i to $A'_{S'}$. $S' := (V'_{S'}, A'_{S'})$ is a solution to P' and by applying (13) and (14), we obtain S.
Injective. Define the set of all arcs of P' corresponding to the edges of P as $A := \{(v, w) \in A' : \ \{v, w\} \in E\}$ and accordingly $A_{S'} := A'_{S'} \cap A$. Since (12) has been assumed, it holds that: $(t_i, t'_i) \in A'_{S'} \Leftrightarrow t_i \in V'_S$ and $(r', t'_i) \in A'_{S'} \Leftrightarrow t_i \notin V'_S$. This implies that $A'_{S'}$ is already determined by $A_{S'}$. Now let $\tilde{S}' = (\tilde{V}'_S, \tilde{A}'_S) \in \mathcal{S}'$, $\tilde{S}' \neq S'$. Consequently, there is at least one arc $(v, w) \in \tilde{A}'_S$

such that $(w,v) \notin A_{S'}$ and $(w,v) \notin A_{S'}$ and therefore is $\tilde{S}'$ not mapped to S.
Finally, using the above notation one observes 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. □

A.3 Proof of Lemma 3 (PCSTP to rcSAP)

Proof. Likewise to the proof of Lemma 2 one observes that (17) and (18) constitute a mapping $\mathcal{S}' \to \mathcal{S}$. Let $S = (V_S, E_S) \in \mathcal{S}$ and $T = \{t_1, ..., t_s\}$ defined as in Transformation 3.
Surjective. Initially, define $V'_{S'} := V_S$, $A'_{S'} := \{(r, t_{i_0})\}$, with $i_0 := \min\{i \mid t_i \in V'_{S'}\}$. Then extend $A'_{S'}$ analogously to the proof of Lemma 2. The so constructed $S' := (V'_{S'}, A'_{S'})$ is a solution to P' and applying (17) and (18) S is obtained.
Injective. Parallelly to the proof of Lemma 2 it can be shown that for a solution $\tilde{S}' \neq S'$ to P' there must be at least one arc $(v,w) \in A_{\tilde{S}'}$ such that $(v,w) \notin A_{S'}$ and $(w,v) \notin A_{S'}$ with A defined as in the proof of Lemma 2. Therefore it follows that $\tilde{S}'$ is not mapped to S.
The equality of the solution values of S and S' can be seen likewise. □

A.4 Proof of Lemma 4 (MWCS to rcSAP)

Proof. The one-to-one correspondence between the sets of solutions to P and P'' can be seen analogously to the proof of Lemma 3.
To prove (21) let $S = (V_S, E_S)$ be a solution to P and $S'' = (V''_{S''}, A''_{S''})$ the corresponding solution to P'', obtained by applying (19) and (20). Further, define $A := \{(v,w) \in A'' : \{v,w\} \in E\}$. First, one observes that for each $v \in S$ such that $p_v \leq 0$ there is exactly one incoming arc $a \in A_{S''}$, so:

$$\sum_{v \in V_S : p_v \leq 0} p_v = -\sum_{a \in A_{S''}} c''_a. \tag{24}$$

Second:

$$\sum_{v \in V_S : p_v > 0} p_v = \sum_{v \in V : p_v > 0} p_v - \sum_{v \in V \setminus V_S : p_v > 0} p_v = \sum_{v \in V : p_v > 0} p_v - \sum_{a \in A''_{S''} \setminus A_{S''}} c''_a. \tag{25}$$

Finally, adding (24) and (25) the equation:

$$\sum_{v \in V_S} p_v = \sum_{v \in V : p_v > 0} p_v - \sum_{a \in A''_{S''}} c''_a \tag{26}$$

is obtained, which coincides with (21). □

B Detailed Computational Results

This section presents detailed instance-wise results of our experiments for all test sets discussed in Sections 2 and 3. We list the original and the presolved problem size, i.e., number of nodes $|V|$, arcs $|A|$, and terminals $|T|$ as well as the preprocessing time (column t [s] in the Presolved columns). Moreover, we show the Dual and Primal bound upon termination and the corresponding

Gap in percent. If an instance was solved to optimality, we print the optimal value centered in the bound columns, and omit the gap; we print "–" as gap if no primal bound was present at the time of termination. Additionally, we list the number of cut separation rounds at the root node (C), the number of branch-and-bound nodes (N), and the total solving time in seconds (last column). The total solving time includes the preprocessing time. A timeout is marked by ">" before the termination time. In case of RSMTP for which SCIP-JACK does not perform preprocessing, we omit the statistics about the presolved model.

Table 14. Detailed computational results for the STP, test set SP.

	Original			Presolved									
Instance	$\|V\|$	$\|A\|$	$\|T\|$	$\|V\|$	$\|A\|$	$\|T\|$	t [s]	Dual	Primal	Gap %	C	N	t [s]
antiwheel5	10	30	5	10	30	5	0.0	**7**			5	1	0.0
design432	8	40	4	8	40	4	0.0	**9**			8	1	0.0
oddcycle3	6	18	3	6	18	3	0.0	**4**			3	1	0.0
oddwheel3	7	18	4	7	18	4	0.0	**5**			5	1	0.0
se03	13	42	4	13	42	4	0.0	**12**			4	1	0.0
w13c29	783	4524	406	783	4524	406	0.2	**507**			578	755	92518.8
w23c23	1081	6348	552	1081	6348	552	0.4	689	697	1.2	570	122	>129600.9
w3c571	3997	20556	2284	3997	20556	2284	1.9	2853	2854	0.0	3864	1	>129602.0

Table 15. Detailed computational results for the STP, test set I320.

	Original			Presolved									
Instance	$\|V\|$	$\|A\|$	$\|T\|$	$\|V\|$	$\|A\|$	$\|T\|$	t [s]	Dual	Primal	Gap %	C	N	t [s]
i320-001	320	960	8	120	472	8	0.0	**2672**			54	1	0.1
i320-002	320	960	8	152	614	8	0.0	**2847**			39	1	0.1
i320-003	320	960	8	166	646	8	0.0	**2972**			44	1	0.4
i320-004	320	960	8	145	592	8	0.0	**2905**			45	1	0.2
i320-005	320	960	8	151	612	8	0.0	**2991**			40	1	0.3
i320-011	320	3690	8	320	3686	8	0.1	**2053**			127	1	2.5
i320-012	320	3690	8	320	3690	8	0.1	**1997**			150	1	0.9
i320-013	320	3690	8	320	3690	8	0.1	**2072**			97	1	1.8
i320-014	320	3690	8	320	3690	8	0.1	**2061**			93	3	12.0
i320-015	320	3690	8	320	3690	8	0.1	**2059**			143	1	5.8
i320-021	320	102080	8	320	5006	8	0.2	**1553**			417	1	17.2
i320-022	320	102080	8	320	5010	8	0.5	**1565**			337	1	11.9
i320-023	320	102080	8	320	5008	8	0.3	**1549**			309	1	10.6
i320-024	320	102080	8	320	5008	8	0.5	**1553**			312	1	10.6
i320-025	320	102080	8	320	5006	8	0.3	**1550**			461	1	18.3
i320-031	320	1280	8	222	1040	8	0.0	**2673**			83	1	0.8
i320-032	320	1280	8	245	1118	8	0.0	**2770**			87	1	1.3
i320-033	320	1280	8	235	1104	8	0.0	**2769**			49	1	0.2
i320-034	320	1280	8	223	1052	8	0.0	**2521**			36	1	0.1
i320-035	320	1280	8	154	706	8	0.0	**2385**			36	1	0.2
i320-041	320	20416	8	320	20388	8	0.9	**1707**			281	1	9.2
i320-042	320	20416	8	320	19822	8	0.5	**1682**			135	1	5.7
i320-043	320	20416	8	319	17082	8	0.4	**1723**			197	1	25.4
i320-044	320	20416	8	320	19252	8	0.7	**1681**			154	1	5.2
i320-045	320	20416	8	320	20366	8	0.4	**1686**			89	1	3.9
i320-101	320	960	17	147	592	16	0.0	**5548**			26	1	0.1
i320-102	320	960	17	153	602	14	0.0	**5556**			35	1	0.5
i320-103	320	960	17	156	610	17	0.0	**6239**			25	1	0.1
i320-104	320	960	17	152	604	17	0.0	**5703**			23	1	0.5
i320-105	320	960	17	158	618	16	0.0	**5928**			38	1	0.8
i320-111	320	3690	17	320	3690	17	0.1	**4273**			81	7	35.6
i320-112	320	3690	17	320	3690	17	0.1	**4213**			85	159	66.7
i320-113	320	3690	17	320	3690	17	0.1	**4205**			83	53	44.7
i320-114	320	3690	17	320	3690	17	0.1	**4104**			86	5	28.8
i320-115	320	3690	17	319	3688	17	0.1	**4238**			77	3	9.3
i320-121	320	102080	17	320	101848	17	4.3	**3321**			304	1	85.0

cont. next page

Instance	Original $\|V\|$	$\|A\|$	$\|T\|$	Presolved $\|V\|$	$\|A\|$	$\|T\|$	t [s]	Dual	Primal	Gap %	C	N	t [s]
i320-122	320	102080	17	320	101840	17	4.3	**3314**			386	1	89.6
i320-123	320	102080	17	320	101842	17	4.1	**3332**			564	1	119.8
i320-124	320	102080	17	320	101846	17	4.3	**3323**			380	1	95.5
i320-125	320	102080	17	320	101846	17	4.3	**3340**			599	1	122.3
i320-131	320	1280	17	250	1132	17	0.0	**5255**			40	1	1.2
i320-132	320	1280	17	249	1126	15	0.0	**5052**			61	1	0.6
i320-133	320	1280	17	240	1108	16	0.0	**5125**			57	1	1.0
i320-134	320	1280	17	241	1120	17	0.0	**5272**			30	1	0.6
i320-135	320	1280	17	254	1144	17	0.0	**5342**			73	1	8.7
i320-141	320	20416	17	320	20386	17	0.8	**3606**			151	491	799.7
i320-142	320	20416	17	320	20400	17	1.2	**3567**			139	22	151.1
i320-143	320	20416	17	320	20388	17	1.0	**3561**			156	7	127.8
i320-144	320	20416	17	320	20378	17	0.8	**3512**			114	1	7.9
i320-145	320	20416	17	320	20384	17	1.0	**3601**			136	363	440.8
i320-201	320	960	34	150	574	32	0.0	**10044**			33	1	0.3
i320-202	320	960	34	168	638	31	0.0	**11223**			31	1	1.8
i320-203	320	960	34	156	608	32	0.0	**10148**			18	1	0.2
i320-204	320	960	34	161	626	33	0.0	**10275**			26	1	0.6
i320-205	320	960	34	150	572	30	0.0	**10573**			21	1	0.2
i320-211	320	3690	34	320	3690	34	0.1	**8039**			68	204	150.3
i320-212	320	3690	34	320	3690	34	0.1	**8044**			60	137	114.4
i320-213	320	3690	34	320	3686	34	0.1	**7984**			69	96	123.5
i320-214	320	3690	34	319	3688	34	0.1	**8046**			105	1741	1330.7
i320-215	320	3690	34	319	3684	34	0.1	**8015**			76	3980	1841.8
i320-221	320	102080	34	320	101050	34	4.2	**6679**			335	29	1327.7
i320-222	320	102080	34	320	101040	34	4.2	**6686**			474	41	1228.8
i320-223	320	102080	34	320	101034	34	4.1	**6695**			318	177	3138.7
i320-224	320	102080	34	320	101036	34	4.3	**6694**			359	71	1546.4
i320-225	320	102080	34	320	101036	34	4.2	**6691**			341	59	1901.1
i320-231	320	1280	34	243	1116	32	0.1	**9862**			61	1	3.3
i320-232	320	1280	34	245	1120	34	0.0	**9933**			65	5	14.0
i320-233	320	1280	34	245	1124	34	0.0	**9787**			29	1	0.6
i320-234	320	1280	34	242	1110	34	0.0	**9517**			56	1	1.8
i320-235	320	1280	34	249	1126	34	0.0	**9945**			36	1	1.5
i320-241	320	20416	34	320	20240	34	1.0	**7027**			113	461	2261.5
i320-242	320	20416	34	320	20278	34	1.1	7035.51143	7072	0.5	113	1503	>7201.1
i320-243	320	20416	34	320	20268	34	1.2	7015.51741	7044	0.4	109	1804	>7201.2
i320-244	320	20416	34	320	20232	34	1.2	7042.57489	7078	0.5	112	2475	>7201.3
i320-245	320	20416	34	320	20228	34	1.1	**7046**			112	1217	3827.7
i320-301	320	960	80	155	564	58	0.0	**23279**			20	1	1.0
i320-302	320	960	80	157	566	54	0.0	**23387**			21	1	0.8
i320-303	320	960	80	161	592	59	0.0	**22693**			26	1	1.2
i320-304	320	960	80	141	542	46	0.0	**23451**			33	1	0.9
i320-305	320	960	80	136	502	56	0.0	**22547**			23	5	1.1
i320-311	320	3690	80	320	3648	80	0.1	17857.7228	17945	0.5	80	12632	>7200.1
i320-312	320	3690	80	320	3608	80	0.2	18034.4826	18122	0.5	72	12901	>7200.2
i320-313	320	3690	80	320	3600	80	0.2	17925.3527	17991	0.4	63	14346	>7200.2
i320-314	320	3690	80	320	3626	80	0.2	17957.2926	18104	0.8	74	9577	>7200.2
i320-315	320	3690	80	320	3642	80	0.1	17864.986	17987	0.7	67	9809	>7200.1
i320-321	320	102080	80	320	95960	80	3.9	15621.4971	15648	0.2	136	94	>7203.9
i320-322	320	102080	80	320	95962	80	4.0	15604.8195	15646	0.3	146	83	>7205.7
i320-323	320	102080	80	320	95952	80	3.9	15627.1891	15654	0.2	126	82	>7203.9
i320-324	320	102080	80	320	95988	80	4.0	15620.82	15667	0.3	129	116	>7205.6
i320-325	320	102080	80	320	95966	80	4.0	15620.3533	15649	0.2	148	76	>7205.0
i320-331	320	1280	80	251	1092	74	0.1	**21517**			47	11	27.2
i320-332	320	1280	80	247	1096	74	0.1	**21674**			27	3	4.2
i320-333	320	1280	80	258	1136	75	0.1	**21339**			31	5	7.6
i320-334	320	1280	80	255	1130	76	0.1	**21415**			22	1	1.9
i320-335	320	1280	80	254	1130	76	0.1	**21378**			47	5	9.6
i320-341	320	20416	80	320	19344	80	0.9	16160.2855	16312	0.9	79	363	>7201.0
i320-342	320	20416	80	320	19358	80	1.0	16158.675	16228	0.4	86	1582	>7201.2
i320-343	320	20416	80	320	19340	80	0.9	16178.5919	16318	0.9	77	761	>7201.1

cont. next page

Instance	Original $\|V\|$	$\|A\|$	$\|T\|$	Presolved $\|V\|$	$\|A\|$	$\|T\|$	t [s]	Dual	Primal	Gap %	C	N	t [s]
i320-344	320	20416	80	320	19304	80	1.1	16184.2733	16302	0.7	78	716	>7201.2
i320-345	320	20416	80	320	19380	80	1.1	16156.3342	16289	0.8	83	412	>7201.3

Table 16. Detailed computational results for the STP, test set I640.

Instance	Original $\|V\|$	$\|A\|$	$\|T\|$	Presolved $\|V\|$	$\|A\|$	$\|T\|$	t [s]	Dual	Primal	Gap %	C	N	t [s]
i640-001	640	1920	9	314	1262	9	0.1	**4033**			54	1	0.7
i640-002	640	1920	9	301	1224	9	0.1	**3588**			45	1	0.4
i640-003	640	1920	9	301	1224	9	0.1	**3438**			29	1	0.3
i640-004	640	1920	9	302	1234	9	0.1	**4000**			77	1	0.8
i640-005	640	1920	9	318	1272	9	0.1	**4006**			61	1	0.6
i640-011	640	8270	9	640	8270	9	0.5	**2392**			176	1	2.6
i640-012	640	8270	9	640	8270	9	0.2	**2465**			176	1	7.1
i640-013	640	8270	9	640	8270	9	0.5	**2399**			186	1	4.2
i640-014	640	8270	9	640	8270	9	0.4	**2171**			82	1	1.3
i640-015	640	8270	9	640	8270	9	0.4	**2347**			191	5	13.2
i640-021	640	408960	9	640	11376	9	1.5	**1749**			910	1	223.1
i640-022	640	408960	9	640	11378	9	1.6	**1756**			627	1	132.2
i640-023	640	408960	9	640	11374	9	1.8	**1754**			677	1	19.0
i640-024	640	408960	9	640	11376	9	1.4	**1751**			652	1	150.4
i640-025	640	408960	9	640	11396	9	1.5	**1745**			837	1	203.8
i640-031	640	2560	9	483	2234	9	0.1	**3278**			86	1	1.0
i640-032	640	2560	9	475	2226	9	0.1	**3187**			91	1	0.5
i640-033	640	2560	9	484	2244	9	0.1	**3260**			116	1	1.2
i640-034	640	2560	9	478	2226	9	0.1	**2953**			59	1	0.9
i640-035	640	2560	9	478	2232	9	0.2	**3292**			108	1	1.4
i640-041	640	81792	9	640	81788	9	4.2	**1897**			245	1	69.3
i640-042	640	81792	9	640	80556	9	4.0	**1934**			355	259	575.8
i640-043	640	81792	9	640	81702	9	4.0	**1931**			370	185	464.2
i640-044	640	81792	9	640	81790	9	4.2	**1938**			353	259	670.2
i640-045	640	81792	9	640	80520	9	4.1	**1866**			305	1	62.2
i640-101	640	1920	25	320	1264	25	0.1	**8764**			50	1	2.2
i640-102	640	1920	25	312	1240	25	0.1	**9109**			31	1	0.5
i640-103	640	1920	25	305	1232	24	0.1	**8819**			48	1	0.9
i640-104	640	1920	25	301	1224	23	0.1	**9040**			42	1	1.1
i640-105	640	1920	25	324	1270	25	0.1	**9623**			67	5	16.7
i640-111	640	8270	25	640	8270	25	0.5	**6167**			112	375	375.4
i640-112	640	8270	25	640	8270	25	0.6	**6304**			100	127	296.5
i640-113	640	8270	25	640	8270	25	0.3	**6249**			111	879	1221.3
i640-114	640	8270	25	640	8270	25	0.3	**6308**			99	281	435.4
i640-115	640	8270	25	640	8270	25	0.6	**6217**			114	1165	1419.1
i640-121	640	408960	25	640	408416	25	32.4	**4906**			1146	1	1835.4
i640-122	640	408960	25	640	408422	25	33.9	**4911**			786	45	5810.0
i640-123	640	408960	25	640	408416	25	33.0	**4913**			897	29	6578.4
i640-124	640	408960	25	640	408416	25	35.0	**4906**			1008	19	4886.8
i640-125	640	408960	25	640	408422	25	34.0	4907.02083	4920	0.3	827	25	>7234.0
i640-131	640	2560	25	481	2234	25	0.2	**8097**			57	1	2.6
i640-132	640	2560	25	480	2228	24	0.1	**8154**			89	1	13.8
i640-133	640	2560	25	482	2236	25	0.1	**8021**			46	1	1.8
i640-134	640	2560	25	485	2244	25	0.1	**7754**			62	1	3.0
i640-135	640	2560	25	479	2226	25	0.1	**7696**			49	1	4.0
i640-141	640	81792	25	640	81714	25	7.1	5148.6372	5199	1.0	217	307	>7207.4
i640-142	640	81792	25	640	81722	25	7.2	5144.5473	5193	0.9	251	281	>7207.2
i640-143	640	81792	25	640	81732	25	7.4	5151.17333	5194	0.8	260	172	>7207.4
i640-144	640	81792	25	640	81716	25	7.3	5155.20996	5205	1.0	236	179	>7207.4
i640-145	640	81792	25	640	81726	25	7.2	5167.51435	5218	1.0	223	298	>7207.2
i640-201	640	1920	50	313	1244	47	0.1	**16079**			37	1	1.4
i640-202	640	1920	50	320	1252	48	0.1	**16324**			24	1	1.1
i640-203	640	1920	50	325	1272	47	0.1	**16124**			36	1	3.3
i640-204	640	1920	50	323	1268	48	0.1	**16239**			34	1	1.8

cont. next page

Instance	Original $\|V\|$	$\|A\|$	$\|T\|$	Presolved $\|V\|$	$\|A\|$	$\|T\|$	t [s]	Dual	Primal	Gap %	C	N	t [s]
i640-205	640	1920	50	327	1276	48	0.1	**16616**			50	1	4.5
i640-211	640	8270	50	640	8270	50	0.6	11837.491	11991	1.3	99	1689	>7200.6
i640-212	640	8270	50	640	8270	50	0.6	**11795**			91	4729	7188.5
i640-213	640	8270	50	640	8268	50	0.6	11781.2798	11881	0.8	93	4953	>7200.6
i640-214	640	8270	50	640	8270	50	0.6	11777.0935	11898	1.0	88	2162	>7200.6
i640-215	640	8270	50	640	8262	50	0.6	11946.1458	12097	1.3	97	2714	>7200.6
i640-221	640	408960	50	640	406630	50	31.7	9782.83677	9821	0.4	444	5	>7235.2
i640-222	640	408960	50	640	406642	50	32.2	9768.47938	9806	0.4	422	5	>7235.8
i640-223	640	408960	50	640	406630	50	31.7	9777.26927	9811	0.3	445	4	>7236.1
i640-224	640	408960	50	640	406626	50	33.7	9774.44139	9805	0.3	470	6	>7238.9
i640-225	640	408960	50	640	406636	50	31.9	9774.87963	9807	0.3	418	4	>7236.0
i640-231	640	2560	50	492	2260	50	0.2	**15014**			75	53	81.5
i640-232	640	2560	50	493	2260	49	0.1	**14630**			60	1	20.5
i640-233	640	2560	50	506	2282	47	0.2	**14797**			104	5	61.1
i640-234	640	2560	50	486	2232	49	0.1	**15203**			36	1	3.3
i640-235	640	2560	50	484	2244	50	0.1	**14803**			103	77	149.3
i640-241	640	81792	50	640	81398	50	7.1	10142.2037	10230	0.9	197	44	>7207.2
i640-242	640	81792	50	640	81410	50	7.2	10111.9081	10195	0.8	172	57	>7207.3
i640-243	640	81792	50	640	81422	50	7.3	10140.5972	10215	0.7	176	45	>7208.1
i640-244	640	81792	50	640	81366	50	7.4	10140.822	10263	1.2	180	34	>7208.1
i640-245	640	81792	50	640	81424	50	7.0	10141.6661	10239	1.0	187	40	>7207.1
i640-301	640	1920	160	335	1234	124	0.1	**45005**			47	1	4.3
i640-302	640	1920	160	298	1144	110	0.1	**45736**			33	1	4.5
i640-303	640	1920	160	341	1262	126	0.2	**44922**			20	1	1.3
i640-304	640	1920	160	329	1216	127	0.2	**46233**			31	1	3.9
i640-305	640	1920	160	299	1114	116	0.2	**45902**			26	1	4.2
i640-311	640	8270	160	640	8070	160	0.7	35311.4404	35889	1.6	91	680	>7200.7
i640-312	640	8270	160	639	8064	160	0.7	35316.7338	35903	1.7	80	1522	>7200.7
i640-313	640	8270	160	640	8086	160	0.5	35209.6647	35553	1.0	81	1927	>7200.5
i640-314	640	8270	160	640	8076	160	0.7	35137.1839	35703	1.6	68	1958	>7200.7
i640-315	640	8270	160	640	8062	160	0.7	35309.7281	35720	1.2	100	2276	>7200.8
i640-321	640	408960	160	640	383906	160	29.3	30991.775	31126	0.4	163	2	>7237.2
i640-322	640	408960	160	640	383924	160	29.2	30985.6518	31127	0.5	145	3	>7229.3
i640-323	640	408960	160	640	383896	160	31.2	30998.2544	31130	0.4	152	1	>7234.3
i640-324	640	408960	160	640	383940	160	29.2	30997.4746	31100	0.3	162	2	>7236.6
i640-325	640	408960	160	640	383940	160	30.5	30986.5479	31092	0.3	170	1	>7245.3
i640-331	640	2560	160	489	2208	146	0.3	**42796**			102	270	173.6
i640-332	640	2560	160	504	2258	152	0.2	**42548**			85	39	94.5
i640-333	640	2560	160	502	2232	147	0.3	**42345**			102	285	242.6
i640-334	640	2560	160	511	2276	155	0.3	**42768**			45	815	563.5
i640-335	640	2560	160	516	2294	153	0.1	**43035**			78	404	308.0
i640-341	640	81792	160	640	77124	160	6.3	31855.8661	32108	0.8	95	11	>7208.7
i640-342	640	81792	160	640	76946	160	6.4	31807.0506	31994	0.6	125	19	>7206.7
i640-343	640	81792	160	640	77022	160	6.6	31821.1132	32049	0.7	99	14	>7206.6
i640-344	640	81792	160	640	77252	160	6.4	31820.7127	32056	0.7	100	13	>7209.7
i640-345	640	81792	160	640	77144	160	6.6	31806.7002	32048	0.8	103	22	>7209.2

Table 17. Detailed computational results for the STP, test set PUC.

Instance	Original $\|V\|$	$\|A\|$	$\|T\|$	Presolved $\|V\|$	$\|A\|$	$\|T\|$	t [s]	Dual	Primal	Gap %	C	N	t [s]
bip42p	1200	7964	200	990	7236	200	1.1	24463.3338	24703	1.0	52	10122	>7201.2
bip42u	1200	7964	200	990	7544	200	0.6	233.004998	237	1.7	39	8626	>7200.7
bip52p	2200	15994	200	1819	14676	200	2.9	24226.5605	24688	1.9	59	3422	>7203.2
bip52u	2200	15994	200	1819	15226	200	1.8	229.625821	234	1.9	56	1756	>7201.8
bip62p	1200	20004	200	1199	20000	200	1.7	22458.1748	23026	2.5	75	285	>7202.2
bip62u	1200	20004	200	1199	20002	200	1.2	213.774582	221	3.4	99	535	>7201.2
bipa2p	3300	36146	300	3140	35594	300	8.6	34693.3718	35938	3.6	89	32	>7211.6
bipa2u	3300	36146	300	3140	35826	300	4.9	329.455373	343	4.1	135	22	>7205.0
bipe2p	550	10026	50	550	10026	50	0.7	5585.6418	5616	0.5	173	18611	>7200.7
bipe2u	550	10026	50	550	10026	50	0.6	**54**			24052	83	5584.0
cc10-2p	1024	10240	135	1024	10240	135	0.9	34478.2417	35929	4.2	137	1	>7202.0
cc10-2u	1024	10240	135	1024	10240	135	0.7	334.237404	345	3.2	153	1	>7201.8
cc11-2p	2048	22526	244	2048	22526	244	3.0	62116.7127	64691	4.1	113	1	>7204.0
cc11-2u	2048	22526	244	2048	22526	244	1.9	602.515847	622	3.2	151	1	>7201.9
cc12-2p	4096	49148	473	4096	49148	473	12.0	118443.08	123824	4.5	72	1	>7212.5

cont. next page

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	t [s]	Dual	Primal	Gap %	C	N	t [s]
cc12-2u	4096	49148	473	4096	49148	473	7.0	1148.9518	1215	5.7	82	1	>7207.6
cc3-10p	1000	27000	50	1000	27000	50	1.3	12136.2552	13166	8.5	185	1	>7201.7
cc3-10u	1000	27000	50	1000	27000	50	1.0	117.352873	128	9.1	271	1	>7201.1
cc3-11p	1331	39930	61	1331	39930	61	2.3	14721.0475	16075	9.2	170	1	>7203.1
cc3-11u	1331	39930	61	1331	39930	61	1.7	143.103992	158	10.4	236	1	>7201.8
cc3-12p	1728	57024	74	1728	57024	74	4.1	17752.6768	19406	9.3	135	1	>7204.3
cc3-12u	1728	57024	74	1728	57024	74	3.2	171.666667	188	9.5	176	1	>7203.3
cc3-4p	64	576	8	64	576	8	0.0	**2338**			159	22265	552.5
cc3-4u	64	576	8	64	576	8	0.0	**23**			159	935	87.8
cc3-5p	125	1500	13	125	1500	13	0.0	3418.89492	3661	7.1	159	16636	>7200.0
cc3-5u	125	1500	13	125	1500	13	0.0	33.0769691	36	8.8	186	18423	>7200.0
cc5-3p	243	2430	27	243	2430	27	0.1	7153.63969	7308	2.2	178	1770	>7200.1
cc5-3u	243	2430	27	243	2430	27	0.1	69.2272065	71	2.6	252	1155	>7200.1
cc6-2p	64	384	12	64	384	12	0.0	**3271**			73	593	29.7
cc6-2u	64	384	12	64	384	12	0.0	**32**			80	19	12.9
cc6-3p	729	8736	76	729	8736	76	0.3	20131.6849	20544	2.0	329	42	>7200.5
cc6-3u	729	8736	76	729	8736	76	0.4	195.562252	201	2.8	391	1	>7200.4
cc7-3p	2187	30616	222	2187	30616	222	3.8	55258.9195	58079	5.1	88	1	>7203.8
cc7-3u	2187	30616	222	2187	30616	222	2.2	535.609797	563	5.1	100	1	>7202.3
cc9-2p	512	4608	64	512	4608	64	0.3	16868.6735	17436	3.4	191	1	>7200.3
cc9-2u	512	4608	64	512	4608	64	0.2	163.53675	172	5.2	189	1	>7203.1
hc10p	1024	10240	512	1024	10240	512	1.1	59220.5539	60999	3.0	58	76	>7201.1
hc10u	1024	10240	512	1024	10240	512	0.6	567.777778	591	4.1	109	4	>7200.6
hc11p	2048	22528	1024	2048	22528	1024	3.2	117382.476	121632	3.6	59	1	>7203.3
hc11u	2048	22528	1024	2048	22528	1024	1.9	1124.4254	1195	6.3	37	1	>7202.1
hc12p	4096	49152	2048	4096	49152	2048	13.8	232375.793	245016	5.4	28	1	>7214.2
hc12u	4096	49152	2048	4096	49152	2048	7.6	2217.66667	2368	6.8	29	1	>7208.4
hc6p	64	384	32	64	384	32	0.0	**4003**			50	17443	128.7
hc6u	64	384	32	64	384	32	0.0	**39**			60	6919	65.8
hc7p	128	896	64	128	896	64	0.0	7779.21214	7905	1.6	47	224077	>7200.0
hc7u	128	896	64	128	896	64	0.0	74.1012897	77	3.9	159	100408	>7200.0
hc8p	256	2048	128	256	2048	128	0.1	15155.2576	15322	1.1	62	21684	>7200.1
hc8u	256	2048	128	256	2048	128	0.0	145.173838	148	1.9	87	8447	>7200.0
hc9p	512	4608	256	512	4608	256	0.3	29908.5709	30317	1.4	56	638	>7200.3
hc9u	512	4608	256	512	4608	256	0.2	286.875	292	1.8	191	105	>7200.2

Table 18. Detailed computational results for the STP, test set vienna-i-advanced.

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	t [s]	Dual	Primal	Gap %	C	N	t [s]
I001a	14675	44110	941	12786	39572	922	289.0	55014956.6	55295701	0.5	314	1	>7490.5
I002a	23800	71516	1282	21012	64822	1266	951.7	57291696.6	58570109	2.2	79	1	>8155.2
I003a	16270	47838	2336	13705	41494	2301	386.2	96464083.9	96576382	0.1	63	1	>7586.2
I004a	867	2476	263	646	1830	239	0.9	**42990860**			57	1	6.8
I005a	1677	4860	491	1191	3478	426	2.4	**53974585**			54	1	17.0
I006a	13339	39064	1842	11592	34920	1820	224.9	136159015	136198404	0.0	121	1	>7425.0
I007a	6873	20598	599	5959	18368	594	65.6	**37370196**			647	1	2981.7
I008a	6522	19258	708	5546	16920	705	56.3	**33153078**			118	1	2702.3
I009a	14977	44870	1053	13004	40174	1041	319.7	47997891.7	48395828	0.8	207	1	>7521.3
I010a	13041	39090	782	10702	33344	762	227.8	207874799	207889674	0.0	397	1	>7428.8
I011a	9298	27370	1202	7547	23070	1181	95.2	**63848241**			114	19	2338.8
I012a	3500	10428	387	2434	7674	371	13.0	**20593258**			96	1	166.0
I013a	7147	21216	670	5814	17808	653	72.1	**37689678**			344	1	2314.4
I014a	3577	10622	364	2561	8038	353	13.2	**19455897**			134	1	51.9
I015a	20573	61082	2119	16756	51760	2100	518.0	145944116	146208119	0.2	84	1	>7718.0
I016a	27214	79648	3434	22687	68534	3378	958.2	164268459	165104658	0.5	50	1	>8158.3
I017a	7571	23142	386	6649	20940	384	67.5	**19021186**			291	1	792.5
I018a	12258	36028	1549	10237	31070	1540	170.8	67254075.8	67328733	0.1	160	1	>7370.8
I019a	11693	35248	732	9123	29050	727	149.7	49497149.3	49578991	0.2	232	1	>7350.5
I020a	6405	19128	508	4785	15136	498	49.7	**24770758**			123	1	574.0
I021a	5195	15722	295	3730	12086	289	29.2	**17025666**			151	1	685.5
I022a	8869	27102	356	7581	23968	354	108.4	24534245.8	24538643	0.0	606	1	>7308.9
I023a	13724	41726	403	12365	38428	393	259.3	17290518.8	17381764	0.5	1084	1	>7459.9
I024a	32357	96500	2511	27449	84872	2482	1413.4	165323708	170528288	3.1	9	1	>8622.7
I025a	10055	29922	833	7729	24248	828	125.7	232789880	232792769	0.0	146	1	>7325.7

cont. next page

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	t [s]	Dual	Primal	Gap %	C	N	t [s]
I026a	18155	53136	2661	14975	45352	2618	403.1	927806877	928050138	0.0	78	1	>7603.2
I027a	40772	121110	3490	33309	103138	3453	2286.4	971230677	976868278	0.6	63	1	>9486.5
I028a	43690	132922	1597	38588	120938	1588	2826.1	379208507	384102366	1.3	9	1	>10034.6
I029a	32979	99254	1946	27367	85586	1928	1564.0	481997040	492250107	2.1	6	1	>8772.6
I030a	12941	38558	1093	9820	30820	1081	214.9	**321646787**			127	1	2968.2
I031a	21054	62820	1832	16470	51584	1783	560.4	577739460	578293199	0.1	147	1	>7760.4
I032a	21345	62706	2454	17484	53236	2398	534.5	143932244	144196409	0.2	152	1	>7734.8
I033a	8500	25400	548	7093	21958	541	82.1	**31604828**			169	1	1379.4
I034a	9128	27336	606	6976	22028	592	96.6	**28842122**			271	1	4600.9
I035a	13129	38840	1428	10746	32960	1415	215.9	102024413	102037997	0.0	150	1	>7417.6
I036a	17036	50964	1258	13669	42908	1237	319.6	103987479	104931471	0.9	71	1	>7521.8
I037a	5886	17738	392	4603	14694	390	34.1	**29768713**			136	1	967.7
I038a	7733	22956	798	6187	19168	782	63.8	**48470499**			124	5	2150.2
I039a	3719	11066	306	2973	9244	299	15.9	**22582804**			133	1	281.9
I040a	18837	56312	1501	15275	47716	1482	501.4	87063311.3	88139862	1.2	48	1	>7704.2
I041a	22466	67736	1014	18106	57260	998	553.9	60474335.1	61290862	1.4	128	1	>7756.1
I042a	23925	71612	1923	19672	61338	1901	729.2	138112322	144851591	4.9	9	1	>7933.9
I043a	4511	13480	335	3582	11228	333	19.9	**24407752**			141	1	386.3
I044a	31500	93514	2954	25870	79936	2916	1371.8	230608587	232169220	0.7	63	1	>8571.9
I045a	6775	20454	378	5523	17444	376	54.9	**23565890**			344	1	436.9
I046a	32376	96108	3154	26144	81110	3116	1314.5	232632113	233831973	0.5	86	1	>8514.7
I047a	10622	30880	1791	8965	26812	1763	133.5	121059462	121097024	0.0	128	1	>7333.5
I048a	4920	14712	320	3735	11864	309	25.8	**15853402**			143	1	361.4
I049a	15045	45426	821	11921	38062	811	230.1	35198857.4	35291465	0.3	211	1	>7431.3
I050a	17787	52352	2232	14815	45062	2206	469.6	176956007	177303855	0.2	79	1	>7669.6
I051a	12130	35784	1337	10082	30812	1319	162.0	86007743.3	86019257	0.0	159	1	>7362.0
I052a	160	474	23	93	282	17	0.0	**2091965**			20	1	0.1
I053a	693	2046	102	533	1656	99	0.6	**7323696**			58	1	1.6
I054a	540	1634	25	396	1278	22	0.2	**15841596**			92	1	1.5
I055a	4701	13958	483	3554	11044	466	24.8	**144164924**			126	1	238.0
I056a	290	878	34	190	602	32	0.0	**14171206**			33	1	0.2
I057a	13078	38736	1346	10604	32706	1320	218.8	**412746415**			197	7	4638.4
I058a	7877	23314	997	6035	18678	968	65.5	**305024188**			140	1	805.1
I059a	2800	8314	286	1803	5640	272	9.8	**107617854**			97	1	34.3
I060a	18991	57072	1158	14709	46792	1150	459.5	335323138	337307756	0.6	66	1	>7661.6
I061a	20958	62930	1337	17786	55432	1328	555.4	362553620	363049760	0.1	70	1	>7758.3
I062a	23714	70610	2812	18044	56522	2753	641.6	791642678	792976980	0.2	105	1	>7841.6
I063a	9600	28084	1291	7602	23088	1260	112.8	**459801704**			165	8	3956.2
I064a	31712	93422	3182	27514	83506	3168	1405.0	185165176	186871758	0.9	44	1	>8605.1
I065a	1185	3512	119	918	2852	116	1.4	**32965718**			76	1	44.7
I066a	4551	13642	417	3348	10690	410	18.8	**174219813**			155	1	223.8
I067a	10318	31176	579	8626	27118	565	111.6	**175540750**			407	1	6761.1
I068a	12191	36046	1302	9481	29272	1275	182.5	**420730046**			171	7	2240.7
I069a	3508	10312	452	2858	8716	446	11.8	**135161583**			103	1	519.8
I070a	6739	20128	511	5255	16636	507	44.7	**136700139**			150	1	2676.3
I071a	12772	37772	1281	10214	31572	1260	170.8	**382539099**			134	1	1620.1
I072a	11628	34822	851	8819	28104	844	149.3	**289019226**			173	1	5663.3
I073a	7510	21746	1337	6219	18480	1280	74.7	**663004987**			114	1	2892.1
I074a	4441	13124	548	3290	10130	528	20.4	**165573383**			116	1	255.6
I075a	23195	68724	2498	18596	57572	2449	572.3	814660646	815423018	0.1	97	1	>7772.4
I076a	4909	14536	498	3685	11496	488	29.1	**166249692**			212	1	1268.4
I077a	9153	26726	1490	8048	24012	1474	109.8	**472503150**			127	3	6931.1
I078a	5864	17324	692	5004	15186	686	40.8	**185525490**			130	13	1002.1
I079a	7933	23614	497	5954	18768	491	76.1	150497192	150509740	0.0	266	1	>7276.4
I080a	7589	22512	499	5717	17866	494	66.7	**164299652**			155	1	1653.9
I081a	10747	32058	751	8416	26384	736	126.7	247459910	247530140	0.0	151	1	>7329.3
I082a	5850	17386	435	4290	13508	427	41.2	**147407632**			165	1	1477.2
I083a	34221	100602	4138	27216	83082	4034	1563.6	1401751030	1405645440	0.3	87	1	>8763.6
I084a	17050	50402	1918	13341	41172	1887	386.5	627079904	627196185	0.0	153	1	>7587.5
I085a	2780	8246	243	2068	6492	237	6.6	**80628079**			116	1	91.5

Table 19. Detailed computational results for the SAP, test set gene.

Instance	Original $\lvert V\rvert$	Original $\lvert A\rvert$	Original $\lvert T\rvert$	Presolved $\lvert V\rvert$	Presolved $\lvert A\rvert$	Presolved $\lvert T\rvert$	Presolved t [s]	Optimum	C	N	t [s]
gene41x	335	910	43	193	626	43	0.0	**126**	10	1	0.1
gene42	335	912	43	190	618	43	0.0	**126**	11	1	0.1
gene61a	395	1024	82	218	668	80	0.0	**205**	7	1	0.1
gene61b	570	1616	82	365	1204	80	0.0	**199**	14	1	0.1
gene61c	549	1580	82	369	1220	82	0.0	**196**	16	1	0.1
gene61f	412	1104	82	240	752	80	0.0	**198**	9	1	0.1

Table 20. Detailed computational results for the SAP, test set geneh.

Instance	Original $\lvert V\rvert$	Original $\lvert A\rvert$	Original $\lvert T\rvert$	Presolved $\lvert V\rvert$	Presolved $\lvert A\rvert$	Presolved $\lvert T\rvert$	Presolved t [s]	Optimum	C	N	t [s]
gene425	425	1108	86	237	730	84	0.0	**214**	7	1	0.1
gene442	442	1188	86	261	820	84	0.0	**207**	8	1	0.1
gene575	575	1648	86	381	1260	86	0.0	**207**	22	1	0.2
gene602	602	1716	86	393	1298	84	0.0	**209**	15	1	0.1

Table 21. Detailed computational results for the SAP, test set gene2002.

Instance	Original $\lvert V\rvert$	Original $\lvert A\rvert$	Original $\lvert T\rvert$	Presolved $\lvert V\rvert$	Presolved $\lvert A\rvert$	Presolved $\lvert T\rvert$	Presolved t [s]	Optimum	C	N	t [s]
microtri1	347	952	47	200	650	47	0.0	**128**	9	1	0.1
microtri3	400	1112	47	256	824	46	0.0	**146**	19	1	0.1
microtri5	416	1124	47	246	782	46	0.0	**150**	16	1	0.1
microtri6	419	1164	47	265	856	46	0.0	**146**	17	1	0.1
microtri7	437	1172	47	265	826	46	0.0	**159**	11	1	0.1
microtri8	484	1412	47	324	1092	46	0.0	**151**	26	1	0.2
microtri9	297	792	47	171	538	46	0.0	**131**	9	1	0.0
microtri10	319	836	47	175	546	46	0.0	**136**	9	1	0.1
microtri11	382	1024	47	228	716	47	0.0	**152**	9	1	0.1

Table 22. Detailed computational results for the RSMTP, test set estein1.

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Optimum	C	N	t [s]
estein1-00	15	44	5	**1.87**	7	1	0.0
estein1-01	12	34	6	**1.64**	6	1	0.0
estein1-02	28	90	7	**2.36**	15	1	0.0
estein1-03	64	224	8	**2.54**	25	1	0.2
estein1-04	12	34	6	**2.26**	7	1	0.0
estein1-05	24	76	12	**2.42**	8	1	0.0
estein1-06	30	98	12	**2.48**	7	1	0.0
estein1-07	24	74	12	**2.36**	11	1	0.0
estein1-08	15	44	7	**1.64**	4	1	0.0
estein1-09	36	120	6	**1.77**	17	1	0.0
estein1-10	30	98	6	**1.44**	6	1	0.0
estein1-11	27	84	9	**1.8**	11	1	0.0
estein1-12	42	142	9	**1.5**	14	1	0.0
estein1-13	36	120	12	**2.6**	11	1	0.0
estein1-14	100	360	14	**1.48**	24	1	0.4
estein1-15	9	24	3	**1.6**	6	1	0.0
estein1-16	48	164	10	**2**	17	1	0.1
estein1-17	182	674	62	**4.04**	19	1	0.3
estein1-18	168	620	14	**1.88**	27	1	0.7
estein1-19	6	14	3	**1.12**	2	1	0.0
estein1-20	15	44	5	**1.92**	10	1	0.0
estein1-21	16	48	4	**0.63**	7	1	0.0
estein1-22	16	48	4	**0.65**	8	1	0.0
estein1-23	16	48	4	**0.3**	8	1	0.0
estein1-24	9	24	3	**0.23**	5	1	0.0
estein1-25	9	24	3	**0.15**	4	1	0.0

cont. next page

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Optimum	C	N	t [s]
estein1-26	16	48	4	**1.33**	6	1	0.0
estein1-27	12	34	4	**0.24**	6	1	0.0
estein1-28	9	24	3	**2**	4	1	0.0
estein1-29	28	90	12	**1.1**	10	1	0.0
estein1-30	130	474	14	**2.59**	26	1	0.8
estein1-31	195	724	19	**3.12**	40	1	2.3
estein1-32	132	482	18	**2.68**	31	1	0.6
estein1-33	272	1022	19	**2.41**	49	1	5.0
estein1-34	240	898	18	**1.51**	44	1	1.7
estein1-35	6	14	4	**0.9**	4	1	0.0
estein1-36	49	168	8	**0.9**	20	1	0.0
estein1-37	100	360	14	**1.66**	26	1	0.3
estein1-38	100	360	14	**1.66**	23	1	0.3
estein1-39	64	224	10	**1.55**	25	1	0.1
estein1-40	144	526	20	**2.24**	24	1	0.5
estein1-41	81	288	15	**1.53**	21	1	0.2
estein1-42	195	724	16	**2.55**	43	1	1.6
estein1-43	196	728	17	**2.52**	53	1	2.7
estein1-44	270	1014	19	**2.2**	52	1	2.7
estein1-45	16	48	16	**1.5**	1	1	0.0

Table 23. Detailed computational results for the RSMTP, test set estein10.

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Optimum	C	N	t [s]
estein10-0	100	360	10	**2.292075**	34	1	0.6
estein10-10	100	360	10	**2.223952**	33	1	0.5
estein10-11	100	360	10	**1.962632**	29	1	0.2
estein10-12	100	360	10	**1.948392**	24	1	0.2
estein10-13	100	360	10	**2.185612**	27	1	0.4
estein10-14	100	360	10	**1.864192**	41	1	0.3
estein10-1	100	360	10	**1.913409**	39	1	0.5
estein10-2	100	360	10	**2.600368**	32	1	0.3
estein10-3	100	360	10	**2.046109**	45	1	0.3
estein10-4	100	360	10	**1.881893**	22	1	0.1
estein10-5	100	360	10	**2.654077**	44	1	0.4
estein10-6	100	360	10	**2.602508**	37	1	0.2
estein10-7	100	360	10	**2.50562**	37	1	0.4
estein10-8	100	360	10	**2.206235**	48	1	0.3
estein10-9	100	360	10	**2.39361**	26	1	0.2

Table 24. Detailed computational results for the RSMTP, test set estein20.

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Optimum	C	N	t [s]
estein20-0	400	1520	20	**3.370387**	49	1	3.9
estein20-10	400	1520	20	**2.712391**	79	1	4.8
estein20-11	400	1520	20	**3.04514**	75	1	9.6
estein20-12	400	1520	20	**3.443865**	63	1	2.5
estein20-13	400	1520	20	**3.406237**	115	1	13.9
estein20-14	400	1520	20	**3.230378**	100	1	10.2
estein20-1	400	1520	20	**3.263948**	56	1	3.7
estein20-2	400	1520	20	**2.784744**	63	1	1.9
estein20-3	400	1520	20	**2.762439**	87	11	25.0
estein20-4	400	1520	20	**3.403317**	82	5	16.2
estein20-5	400	1520	20	**3.601423**	63	1	4.1
estein20-6	400	1520	20	**3.493487**	102	1	12.2
estein20-7	400	1520	20	**3.801638**	85	3	12.4
estein20-8	400	1520	20	**3.673995**	72	1	5.8
estein20-9	400	1520	20	**3.402477**	80	1	8.7

Table 25. Detailed computational results for the RSMTP, test set estein30.

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Optimum	C	N	t [s]
estein30-0	900	3480	30	**4.069296**	185	1	119.6
estein30-10	900	3480	30	**4.164799**	144	1	136.2
estein30-11	900	3480	30	**3.841669**	103	1	42.8
estein30-12	900	3480	30	**3.740663**	130	1	66.9
estein30-13	900	3480	30	**4.2897**	107	1	65.2
estein30-14	900	3480	30	**4.303555**	173	1	213.0
estein30-1	900	3480	30	**4.090005**	150	1	105.6
estein30-2	900	3480	30	**4.312045**	202	1	288.9
estein30-3	900	3480	30	**4.215096**	269	1	293.9
estein30-4	900	3480	30	**4.173974**	195	1	161.6
estein30-5	900	3480	30	**3.995514**	262	1	266.7
estein30-6	900	3480	30	**4.376138**	131	1	82.2
estein30-7	900	3480	30	**4.169121**	205	1	248.4
estein30-8	900	3480	30	**3.713363**	198	1	124.6
estein30-9	900	3480	30	**4.268661**	109	1	78.2

Table 26. Detailed computational results for the RSMTP, test set estein40.

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Optimum	C	N	t [s]
estein40-0	1600	6240	40	**4.484154**	148	1	425.3
estein40-10	1600	6240	40	**4.673421**	222	1	1545.8
estein40-11	1600	6240	40	**4.384339**	184	1	702.9
estein40-12	1600	6240	40	**5.188453**	222	1	673.9
estein40-13	1600	6240	40	**4.916698**	163	1	554.9
estein40-14	1600	6240	40	**5.082803**	222	1	1024.1
estein40-1	1600	6240	40	**4.681131**	210	1	745.0
estein40-2	1600	6240	40	**4.997415**	257	1	1480.5
estein40-3	1600	6240	40	**4.528989**	272	1	970.2
estein40-4	1600	6240	40	**5.194038**	350	2667	46589.2
estein40-5	1600	6240	40	**4.97534**	295	1	839.3
estein40-6	1600	6240	40	**4.563901**	188	1	491.7
estein40-7	1600	6240	40	**4.874601**	286	1	1500.4
estein40-8	1600	6240	40	**5.176179**	269	1	2633.1
estein40-9	1600	6240	40	**5.713686**	215	1	1391.5

Table 27. Detailed computational results for the RSMTP, test set estein50.

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Dual	Primal	Gap %	C	N	t [s]
estein50-0	2500	9800	50	**5.494867**			222	1	2376.5
estein50-10	2500	9800	50	5.25225975	5.253293	0.0	378	1	>7200.2
estein50-11	2500	9800	50	5.3137051	5.343239	0.6	350	1	>7200.0
estein50-12	2500	9800	50	**5.389099**			301	1	4462.9
estein50-13	2500	9800	50	5.34799157	5.360222	0.2	409	1	>7200.0
estein50-14	2500	9800	50	**5.218085**			213	1	1966.4
estein50-1	2500	9800	50	**5.548422**			344	1	5744.6
estein50-2	2500	9800	50	**5.469105**			356	1	6852.8
estein50-3	2500	9800	50	**5.153576**			189	1	1141.0
estein50-4	2500	9800	50	**5.518601**			238	1	1778.4
estein50-5	2500	9800	50	**5.58043**			275	1	6292.8
estein50-6	2500	9800	50	4.97961005	4.999921	0.4	330	1	>7202.8
estein50-7	2500	9800	50	**5.375465**			172	1	848.2
estein50-8	2500	9800	50	5.34430057	5.345677	0.0	348	28	>7200.1
estein50-9	2500	9800	50	**5.403795**			270	1	2949.8

Table 28. Detailed computational results for the RSMTP, test set estein60.

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Dual	Primal	Gap %	C	N	t [s]
estein60-0	3600	14160	60	**5.376143**			285	1	6283.3
estein60-10	3600	14160	60	5.60674269	5.631764	0.4	257	1	>7200.1
estein60-11	3600	14160	60	5.91373357	5.99359	1.4	293	1	>7200.5
estein60-12	3600	14160	60	5.95716839	6.141861	3.1	320	1	>7204.6
estein60-13	3600	14160	60	5.59642276	5.603556	0.1	312	1	>7200.3
estein60-14	3600	14160	60	5.66210571	5.662257	0.0	383	1	>7200.1

cont. next page

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Dual	Primal	Gap %	C	N	t [s]
estein60-1	3600	14160	60	5.51154626	5.548722	0.7	281	1	>7200.1
estein60-2	3600	14160	60	5.65366654	5.656678	0.1	309	1	>7200.2
estein60-3	3600	14160	60	5.44595966	5.561215	2.1	353	1	>7200.0
estein60-4	3600	14160	60	5.45561303	5.470499	0.3	305	1	>7200.0
estein60-5	3600	14160	60	6.03356772	6.042196	0.1	258	1	>7200.0
estein60-6	3600	14160	60	5.83580351	5.897848	1.1	266	1	>7200.5
estein60-7	3600	14160	60	5.80358472	5.816953	0.2	266	1	>7200.3
estein60-8	3600	14160	60	5.54060717	5.594983	1.0	327	1	>7200.0
estein60-9	3600	14160	60	5.76131581	5.762446	0.0	317	1	>7200.3

Table 29. Detailed computational results for the RSMTP, test set solids.

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Optimum	C	N	t [s]
cube	8	24	8	**7**	1	1	0.0
dodecahedron	343	1764	20	**7.69398**	138	14249	6269.7
icosahedron	125	600	12	**20.944264**	42	7	5.2
octahedron	27	108	6	**6**	1	1	0.0
tetrahedron	18	66	4	**2.682521**	5	1	0.0

Table 30. Detailed computational results for the RSMTP, test set cancer.

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Dual	Primal	Gap %	C	N	t [s]
cancer1_4D	600	3820	20	**28**			86	1	0.9
cancer2_4D	256	1536	20	**21**			8	1	0.0
cancer3_6D	20580	197078	110	**146**			218	1	191.2
cancer4_6D	34560	340416	93	**136**			1291	1	51344.7
cancer5_6D	8000	74400	48	**69**			745	1	7309.8
cancer6_6D	5120	46592	50	**55**			406	1	17.8
cancer7_6D	21000	203300	109	**140**			516	1	2834.5
cancer8_6D	8640	80064	77	**89**			226	1	55.2
cancer9_6D	6000	54800	46	**59**			133	1	14.2
cancer10_6D	10000	94000	82	**92**			127	1	21.6
cancer11_8D	4762800	64777860	75	–	–	–	0	1	memout
cancer12_8D	918750	12031250	58	87.5882353	113	29.0	189	1	>136341.2
cancer13_8D	86400	1039680	70	**88**			618	1	3800.1
cancer14_8D	27648	308736	54	**63**			131	1	106.0

Table 31. Detailed computational results for the PCSTP, test set JMP.

	Original			Presolved							
Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	t [s]	Optimum	C	N	t [s]
K100.10	115	722	15	112	716	15	0.0	**133567**	8	1	0.0
K100.1	112	762	12	109	750	12	0.0	**124108**	10	1	0.0
K100.2	114	756	14	106	724	14	0.0	**200262**	20	1	0.8
K100.3	111	874	11	102	848	11	0.0	**115953**	19	1	0.3
K100.4	111	788	11	107	774	11	0.0	**87498**	10	1	0.1
K100.5	117	812	17	111	796	17	0.0	**119078**	12	1	0.1
K100.6	112	680	12	108	664	12	0.0	**132886**	11	1	0.1
K100.7	114	708	14	110	694	14	0.0	**172457**	14	1	0.4
K100.8	116	776	16	107	744	16	0.0	**210869**	13	1	0.2
K100.9	112	732	12	105	716	12	0.0	**122917**	12	1	0.1
K100	115	786	15	103	736	15	0.0	**135511**	10	1	0.1
K200	234	1580	34	225	1558	34	0.0	**329211**	17	1	1.4
K400.10	450	3308	50	438	3270	50	0.0	**394191**	53	1	14.9
K400.1	465	3324	65	459	3290	65	0.0	**490771**	36	1	10.2
K400.2	462	3420	62	448	3360	62	0.0	**477073**	35	1	12.9
K400.3	456	3314	56	445	3258	56	0.0	**415328**	29	1	7.2
K400.4	456	3182	56	448	3148	56	0.0	**389451**	32	1	6.5
K400.5	477	3368	77	467	3340	77	0.0	**519526**	38	1	15.4
K400.6	456	3482	56	436	3398	56	0.0	**374849**	30	1	4.9
K400.7	468	3286	68	456	3248	68	0.0	**474466**	39	1	13.6
K400.8	461	3392	61	453	3366	61	0.0	**418614**	33	1	5.5

cont. next page

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	t [s]	Optimum	C	N	t [s]
K400.9	454	3318	54	444	3278	54	0.0	**383105**	29	1	6.6
K400	463	3402	63	452	3362	63	0.0	**350093**	32	1	5.1
P100.1	133	760	33	131	654	33	0.0	**926238**	17	1	0.3
P100.2	127	750	27	121	620	27	0.0	**401641**	39	1	0.2
P100.3	125	776	25	124	662	25	0.0	**659644**	12	1	0.1
P100.4	133	760	33	122	654	33	0.0	**827419**	10	1	0.1
P100	134	832	34	131	686	34	0.0	**803300**	15	1	0.1
P200	249	1462	49	231	1232	49	0.0	**1317874**	31	1	1.3
P400.1	521	3144	121	496	2892	121	0.1	**2808440**	42	1	6.9
P400.2	508	3034	108	482	2766	108	0.1	**2518577**	32	1	3.1
P400.3	514	3028	114	485	2768	114	0.1	**2951725**	57	1	7.0
P400.4	495	2852	95	469	2602	95	0.1	**2852956**	23	1	3.5
P400	495	2964	95	472	2692	95	0.1	**2459904**	36	1	4.0

Table 32. Detailed computational results for the PCSTP, test set CRR.

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	t [s]	Optimum	C	N	t [s]
C01-A	506	1280	6	156	568	6	0.0	**18**	5	1	0.0
C01-B	506	1280	6	156	568	6	0.0	**85**	33	1	0.1
C02-A	511	1310	11	144	544	11	0.0	**50**	6	1	0.0
C02-B	511	1310	11	144	544	11	0.0	**141**	26	1	0.1
C03-A	584	1748	84	292	1156	84	0.0	**414**	16	1	1.3
C03-B	584	1748	84	292	1156	84	0.0	**737**	27	1	1.5
C04-A	626	2000	126	376	1498	126	0.0	**618**	52	1	4.6
C04-B	626	2000	126	376	1498	126	0.0	**1063**	18	1	2.6
C05-A	751	2750	251	587	2414	251	0.1	**1080**	32	1	39.1
C05-B	751	2750	251	587	2414	251	0.1	**1528**	16	1	13.7
C06-A	506	2030	6	375	1726	6	0.0	**18**	11	1	0.0
C06-B	506	2030	6	375	1726	6	0.0	**55**	53	1	0.2
C07-A	511	2060	11	394	1800	11	0.0	**50**	13	1	0.1
C07-B	511	2060	11	394	1800	11	0.0	**102**	32	1	0.5
C08-A	584	2498	84	479	2262	84	0.1	**361**	46	1	3.7
C08-B	584	2498	84	479	2262	84	0.0	**500**	24	1	2.1
C09-A	626	2750	126	550	2582	126	0.0	**533**	50	1	9.2
C09-B	626	2750	126	550	2582	126	0.0	**694**	33	1	5.7
C10-A	751	3500	251	694	3368	251	0.1	**859**	23	1	27.6
C10-B	751	3500	251	694	3368	251	0.1	**1069**	114	1	45.5
C11-A	506	5030	6	506	4410	6	0.1	**18**	13	1	0.1
C11-B	506	5030	6	506	4410	6	0.1	**32**	132	1	1.0
C12-A	511	5060	11	510	4548	11	0.1	**38**	21	1	0.4
C12-B	511	5060	11	510	4548	11	0.1	**46**	87	1	0.8
C13-A	584	5498	84	582	4932	84	0.2	**236**	54	1	7.0
C13-B	584	5498	84	582	4932	84	0.1	**258**	32	1	4.4
C14-A	626	5750	126	626	5142	126	0.1	**293**	22	1	4.4
C14-B	626	5750	126	626	5142	126	0.1	**318**	15	1	3.6
C15-A	751	6500	251	751	5856	251	0.3	**501**	17	1	17.9
C15-B	751	6500	251	751	5856	251	0.2	**551**	10	1	10.7
C16-A	506	25030	6	506	9510	6	0.3	**11**	49	1	1.2
C16-B	506	25030	6	506	9510	6	0.6	**11**	49	1	1.2
C17-A	511	25060	11	511	9468	11	0.3	**18**	90	1	1.2
C17-B	511	25060	11	511	9468	11	0.3	**18**	90	1	1.4
C18-A	584	25498	84	584	10060	84	0.5	**111**	34	1	5.7
C18-B	584	25498	84	584	10060	84	0.6	**113**	38	1	7.1
C19-A	626	25750	126	626	10210	126	0.4	**146**	15	1	3.9
C19-B	626	25750	126	626	10210	126	0.6	**146**	30	1	5.9
C20-A	751	26500	251	751	11040	251	0.5	**266**	14	1	14.8
C20-B	751	26500	251	751	11040	251	0.7	**267**	9	1	10.0
D01-A	1006	2530	6	280	1050	6	0.0	**18**	4	1	0.0
D01-B	1006	2530	6	280	1050	6	0.0	**106**	40	1	0.4
D02-A	1011	2560	11	300	1114	11	0.0	**50**	4	1	0.1
D02-B	1011	2560	11	300	1114	11	0.0	**218**	28	1	0.2
D03-A	1168	3502	168	596	2342	168	0.0	**807**	34	1	6.4

cont. next page

Instance	Original $\lvert V\rvert$	Original $\lvert A\rvert$	Original $\lvert T\rvert$	Presolved $\lvert V\rvert$	Presolved $\lvert A\rvert$	Presolved $\lvert T\rvert$	Presolved t [s]	Optimum	C	N	t [s]
D03-B	1168	3502	168	596	2342	168	0.1	**1509**	27	1	6.0
D04-A	1251	4000	251	738	2962	251	0.1	**1203**	79	1	21.1
D04-B	1251	4000	251	738	2962	251	0.1	**1881**	55	1	25.9
D05-A	1501	5500	501	1181	4856	501	0.2	**2157**	494	1	746.6
D05-B	1501	5500	501	1181	4856	501	0.2	**3135**	35	1	176.7
D06-A	1006	4030	6	767	3512	6	0.1	**18**	9	1	0.1
D06-B	1006	4030	6	767	3512	6	0.1	**67**	128	1	1.3
D07-A	1011	4060	11	766	3532	11	0.0	**50**	34	1	0.1
D07-B	1011	4060	11	766	3532	11	0.0	**103**	47	1	0.7
D08-A	1168	5002	168	977	4586	168	0.1	**755**	82	1	27.1
D08-B	1168	5002	168	977	4586	168	0.1	**1036**	30	1	12.2
D09-A	1251	5500	251	1076	5108	251	0.2	**1070**	79	3	106.9
D09-B	1251	5500	251	1076	5108	251	0.4	**1420**	38	1	33.2
D10-A	1501	7000	501	1367	6706	501	1.0	**1671**	265	1	491.0
D10-B	1501	7000	501	1367	6706	501	0.7	**2079**	297	1	935.1
D11-A	1006	10030	6	999	9394	6	0.2	**18**	21	1	0.6
D11-B	1006	10030	6	999	9394	6	0.4	**29**	172	1	2.3
D12-A	1011	10060	11	1011	9416	11	0.5	**42**	54	1	3.2
D12-B	1011	10060	11	1011	9416	11	0.2	**42**	57	1	1.9
D13-A	1168	11002	168	1166	10292	168	0.5	**445**	107	1	69.6
D13-B	1168	11002	168	1166	10292	168	0.4	**486**	21	1	11.0
D14-A	1251	11500	251	1250	10832	251	0.5	**602**	83	1	102.7
D14-B	1251	11500	251	1250	10832	251	0.6	**665**	33	1	34.9
D15-A	1501	13000	501	1500	12294	501	1.1	**1042**	29	1	185.7
D15-B	1501	13000	501	1500	12294	501	0.9	**1108**	18	1	163.4
D16-A	1006	50030	6	1006	21224	6	1.7	**13**	89	1	4.1
D16-B	1006	50030	6	1006	21224	6	1.7	**13**	79	1	4.8
D17-A	1011	50060	11	1011	21144	11	2.4	**23**	109	1	7.2
D17-B	1011	50060	11	1011	21144	11	2.3	**23**	121	1	7.0
D18-A	1168	51002	168	1168	21626	168	1.8	**218**	31	1	27.0
D18-B	1168	51002	168	1168	21626	168	1.8	**223**	28	1	24.4
D19-A	1251	51500	251	1251	21986	251	2.2	**306**	28	1	43.3
D19-B	1251	51500	251	1251	21986	251	2.0	**310**	36	1	55.1
D20-A	1501	53000	501	1501	23946	501	2.5	**536**	14	1	133.5
D20-B	1501	53000	501	1501	23946	501	2.4	**537**	11	1	125.5

Table 33. Detailed computational results for the PCSTP, test set PUCNU.

Instance	Original $\lvert V\rvert$	Original $\lvert A\rvert$	Original $\lvert T\rvert$	Presolved $\lvert V\rvert$	Presolved $\lvert A\rvert$	Presolved $\lvert T\rvert$	Presolved t [s]	Dual	Primal	Gap %	C	N	t [s]
bip42nu	1401	9164	201	1191	8744	201	0.2	224.342633	227	1.2	205	1607	>7200.6
bip52nu	2401	17194	201	2020	16426	201	0.4	220.217617	223	1.3	139	617	>7200.5
bip62nu	1401	21204	201	1400	21202	201	0.4	210.260966	215	2.3	200	8	>7200.7
bipa2nu	3601	37946	301	3441	37626	301	1.1	320.365297	329	2.7	168	1	>7201.4
bipe2nu	601	10326	51	601	10326	51	0.1	**53**			266	9	184.1
cc10-2nu	1160	11050	136	1160	11050	136	0.1	165.575422	168	1.5	212	79	>7200.1
cc11-2nu	2293	23990	245	2293	23990	245	0.5	300.298142	309	2.9	167	1	>7201.3
cc12-2nu	4570	51986	474	4570	51986	474	1.5	557.508916	571	2.4	125	1	>7212.0
cc3-10nu	1051	27300	51	1051	27300	51	0.1	58.4811788	61	4.3	275	519	>7200.2
cc3-11nu	1393	40296	62	1393	40296	62	0.2	75.2496405	85	13.0	429	23	>7200.8
cc3-12nu	1803	57468	75	1803	57468	75	0.4	90.163976	98	8.7	233	1	>7200.8
cc3-4nu	73	624	9	73	624	9	0.0	**10**			41	1	0.1
cc3-5nu	139	1578	14	139	1578	14	0.0	**17**			38	1	1.1
cc5-3nu	271	2592	28	271	2592	28	0.0	**36**			104	1	25.7
cc6-2nu	77	456	13	77	456	13	0.0	**15**			23	1	0.3
cc6-3nu	806	9192	77	806	9192	77	0.1	**95**			264	8	833.9
cc7-3nu	2410	31948	223	2410	31948	223	0.6	267.418586	275	2.8	169	1	>7200.8
cc9-2nu	577	4992	65	577	4992	65	0.0	**83**			276	50	704.9

Table 34. Detailed computational results for the RPCSTP, test set cologne1.

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	t [s]	Optimum	C	N	t [s]
i101M1	758	12704	11	664	11568	11	0.4	**109271.503**	5	1	0.5
i101M2	758	12704	11	664	11568	11	0.4	**315925.31**	177	1	10.6
i101M3	758	12704	11	664	11568	11	0.4	**355625.409**	139	1	16.5
i102M1	760	12730	12	667	11606	12	0.4	**104065.801**	2	1	0.5
i102M2	760	12730	12	667	11606	12	0.4	**352538.819**	138	1	16.5
i102M3	760	12730	12	667	11606	12	0.4	**454365.927**	145	1	22.1
i103M1	764	12738	14	672	11618	14	0.4	**139749.407**	22	1	0.8
i103M2	764	12738	14	672	11618	14	0.4	**407834.228**	118	1	10.8
i103M3	764	12738	14	672	11618	14	0.4	**456125.488**	127	1	22.0
i104M2	744	12598	4	650	11474	4	0.3	**89920.8353**	159	1	2.6
i104M3	744	12598	4	650	11474	4	0.2	**97148.789**	196	1	3.9
i105M1	744	12604	4	650	11480	4	0.2	**26717.2025**	3	1	0.3
i105M2	744	12604	4	650	11480	4	0.2	**100269.619**	178	1	5.7
i105M3	744	12604	4	650	11480	4	0.2	**110351.163**	209	1	10.1

Table 35. Detailed computational results for the RPCSTP, test set cologne2.

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	t [s]	Optimum	C	N	t [s]
i201M2	1812	33522	10	1764	32412	10	1.1	**355467.684**	422	1	17.8
i201M3	1812	33522	10	1764	32412	10	1.2	**628833.614**	470	1	150.8
i201M4	1812	33522	10	1764	32412	10	1.2	**773398.303**	507	1	214.8
i202M2	1814	33520	11	1767	32414	11	1.1	**288946.832**	311	1	23.9
i202M3	1814	33520	11	1767	32414	11	1.0	**419184.159**	653	1	101.7
i202M4	1814	33520	11	1767	32414	11	1.0	**430034.264**	410	1	132.9
i203M2	1824	33584	16	1780	32480	16	1.0	**459894.776**	371	1	30.3
i203M3	1824	33584	16	1780	32480	16	1.1	**643062.02**	517	1	323.1
i203M4	1824	33584	16	1780	32480	16	1.4	**677733.067**	459	1	341.9
i204M2	1805	33454	5	1757	32356	5	1.0	**161700.545**	217	1	10.4
i204M3	1805	33454	5	1757	32356	5	1.2	**245287.203**	374	1	28.2
i204M4	1805	33454	5	1757	32356	5	1.2	**245287.203**	441	1	29.3
i205M2	1823	33640	14	1775	32534	14	1.4	**571031.415**	231	1	19.7
i205M3	1823	33640	14	1775	32534	14	1.1	**672403.143**	239	1	30.2
i205M4	1823	33640	14	1775	32534	14	1.0	**713973.623**	361	1	42.0

Table 36. Detailed computational results for the MWCSP, test set ACTMOD. The number of terminals was not changed during preprocessing.

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	t [s]	Optimum	C	N	t [s]
drosophila001	5298	187214	72	3977	183910	5.2	**24.3855064**	1626	3188	4287.2
drosophila005	5421	187952	195	4135	184720	12.3	**178.663952**	249	1	1393.9
drosophila0075	5477	188288	251	4207	185092	15.2	**260.523557**	335	1	1011.0
HCMV	3919	58916	56	2818	55814	1.6	**7.55431486**	255	1	53.5
lymphoma	2102	15914	68	1321	13960	0.7	**70.1663087**	91	1	9.8
metabol_expr_mice_1	3674	9590	151	772	3248	0.3	**544.94837**	219	1	35.8
metabol_expr_mice_2	3600	9174	86	653	2736	0.2	**241.077524**	74	1	3.2
metabol_expr_mice_3	2968	7354	115	536	2282	0.2	**508.260877**	69	1	6.0

Table 37. Detailed computational results for the MWCSP, test set JMPALMK. The number of terminals was not changed during preprocessing.

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	t [s]	Dual	Primal	Gap %	C	N	t [s]
1000-a-0.6-d-0.25-e-0.25	1443	12524	443	1438	12512	0.9	**931.538552**			39	1	172.1
1000-a-0.6-d-0.25-e-0.5	1638	13694	638	1636	13690	1.2	**1872.2754**			14	1	304.0
1000-a-0.6-d-0.25-e-0.75	1814	14750	814	1813	14748	1.3	**2789.57911**			0	1	179.9
1000-a-0.6-d-0.5-e-0.25	1621	13592	621	1618	13584	1.1	**522.525615**			66	1	2485.6
1000-a-0.6-d-0.5-e-0.5	1757	14408	757	1754	14400	1.2	**1197.85102**			2	1	111.8
1000-a-0.6-d-0.5-e-0.75	1881	15152	881	1878	15144	1.6	**1762.70747**			2	1	189.9
1000-a-0.6-d-0.75-e-0.25	1815	14756	815	1814	14754	1.6	**332.791924**			3	1	126.3
1000-a-0.6-d-0.75-e-0.5	1894	15230	894	1894	15230	1.5	**754.300601**			5	1	178.4

cont. next page

Instance	Original			Presolved								
	$\|V\|$	$\|A\|$	$\|T\|$	$\|V\|$	$\|A\|$	t [s]	Dual	Primal	Gap %	C	N	t [s]
1000-a-0.6-d-0.75-e-0.75	1949	15560	949	1949	15560	1.8	**998.215414**			5	1	271.4
1000-a-1-d-0.25-e-0.25	1443	29210	443	1443	29210	1.2	**939.39337**			0	1	27.5
1000-a-1-d-0.25-e-0.5	1638	30380	638	1638	30380	1.6	**1883.21361**			0	1	80.6
1000-a-1-d-0.25-e-0.75	1814	31436	814	1814	31436	2.3	**2789.57911**			0	1	176.5
1000-a-1-d-0.5-e-0.25	1621	30278	621	1621	30278	1.6	**533.4294**			0	1	59.1
1000-a-1-d-0.5-e-0.5	1757	31094	757	1757	31094	1.9	**1205.42131**			0	1	111.2
1000-a-1-d-0.5-e-0.75	1881	31838	881	1881	31838	2.7	**1770.27776**			0	1	176.6
1000-a-1-d-0.75-e-0.25	1815	31442	815	1815	31442	2.1	**336.829944**			0	1	118.1
1000-a-1-d-0.75-e-0.5	1894	31916	894	1894	31916	2.6	**760.284581**			0	1	165.9
1000-a-1-d-0.75-e-0.75	1949	32246	949	1949	32246	2.6	**1004.19939**			0	1	194.0
1500-a-0.6-d-0.25-e-0.25	2164	19302	664	2159	19290	1.5	1335.37039	1333.47643	0.1	5602	1	>129601.8
1500-a-0.6-d-0.25-e-0.5	2457	21060	957	2456	21058	2.0	**2799.67722**			103	1	16013.3
1500-a-0.6-d-0.25-e-0.75	2732	22710	1232	2732	22710	2.9	**4230.25112**			0	1	763.6
1500-a-0.6-d-0.5-e-0.25	2432	20910	932	2430	20904	2.1	**847.452011**			4	1	212.4
1500-a-0.6-d-0.5-e-0.5	2633	22116	1133	2632	22114	2.6	**1858.0926**			2	1	442.8
1500-a-0.6-d-0.5-e-0.75	2812	23190	1312	2811	23188	3.1	**2697.45876**			7	1	1546.6
1500-a-0.6-d-0.75-e-0.25	2739	22752	1239	2738	22750	2.9	**502.17599**			0	1	538.0
1500-a-0.6-d-0.75-e-0.5	2850	23418	1350	2850	23418	3.2	**1089.77117**			0	1	639.8
1500-a-0.6-d-0.75-e-0.75	2924	23862	1424	2924	23862	3.5	**1423.61063**			0	1	845.9
1500-a-1-d-0.25-e-0.25	2164	45032	664	2164	45032	2.6	**1377.0144**			0	1	88.9
1500-a-1-d-0.25-e-0.5	2457	46790	957	2457	46790	3.6	**2820.05174**			0	1	321.2
1500-a-1-d-0.25-e-0.75	2732	48440	1232	2732	48440	5.5	**4230.25112**			0	1	747.8
1500-a-1-d-0.5-e-0.25	2432	46640	932	2432	46640	3.5	**860.618961**			0	1	213.6
1500-a-1-d-0.5-e-0.5	2633	47846	1133	2633	47846	4.3	**1865.66289**			0	1	441.2
1500-a-1-d-0.5-e-0.75	2812	48920	1312	2812	48920	5.2	**2707.70001**			0	1	688.9
1500-a-1-d-0.75-e-0.25	2739	48482	1239	2739	48482	4.9	**502.17599**			0	1	517.5
1500-a-1-d-0.75-e-0.5	2850	49148	1350	2850	49148	5.4	**1089.77117**			0	1	615.9
1500-a-1-d-0.75-e-0.75	2924	49592	1424	2924	49592	5.9	**1423.61063**			0	1	795.9
500-a-0.62-d-0.25-e-0.25	712	6460	212	705	6436	0.2	**460.577357**			66	1	28.0
500-a-0.62-d-0.25-e-0.5	818	7096	318	813	7080	0.4	**992.967111**			5	1	11.9
500-a-0.62-d-0.25-e-0.75	910	7648	410	908	7642	0.5	**1447.54452**			0	1	23.1
500-a-0.62-d-0.5-e-0.25	805	7018	305	803	7010	0.3	**280.832378**			7	1	7.5
500-a-0.62-d-0.5-e-0.5	878	7456	378	876	7448	0.4	**655.623217**			7	1	18.2
500-a-0.62-d-0.5-e-0.75	945	7858	445	943	7850	0.6	**965.554694**			0	1	24.6
500-a-0.62-d-0.75-e-0.25	910	7648	410	908	7642	0.5	**171.628785**			0	1	15.6
500-a-0.62-d-0.75-e-0.5	945	7858	445	944	7854	0.6	**362.188212**			0	1	18.8
500-a-0.62-d-0.75-e-0.75	972	8020	472	972	8020	0.6	**490.623986**			0	1	24.0
500-a-1-d-0.25-e-0.25	712	14304	212	712	14304	0.4	**471.393285**			0	1	3.5
500-a-1-d-0.25-e-0.5	818	14940	318	818	14940	0.6	**995.313181**			0	1	10.8
500-a-1-d-0.25-e-0.75	910	15492	410	910	15492	0.7	**1447.54452**			0	1	22.3
500-a-1-d-0.5-e-0.25	805	14862	305	805	14862	0.6	**286.920868**			0	1	7.7
500-a-1-d-0.5-e-0.5	878	15300	378	878	15300	0.7	**661.711707**			0	1	14.0
500-a-1-d-0.5-e-0.75	945	15702	445	945	15702	0.8	**965.554694**			0	1	23.6
500-a-1-d-0.75-e-0.25	910	15492	410	910	15492	0.7	**171.628785**			0	1	15.7
500-a-1-d-0.75-e-0.5	945	15702	445	945	15702	0.8	**362.188212**			0	1	20.8
500-a-1-d-0.75-e-0.75	972	15864	472	972	15864	0.8	**490.623986**			0	1	22.8
750-a-0.647-d-0.25-e-0.25	1079	10406	329	1075	10394	0.6	**702.644057**			13	1	23.3
750-a-0.647-d-0.25-e-0.5	1229	11306	479	1227	11302	0.8	**1419.77986**			7	1	64.9
750-a-0.647-d-0.25-e-0.75	1364	12116	614	1363	12114	0.9	**2116.58233**			0	1	74.7
750-a-0.647-d-0.5-e-0.25	1206	11168	456	1204	11162	0.7	**403.177763**			0	1	23.0
750-a-0.647-d-0.5-e-0.5	1315	11822	565	1313	11816	0.9	**946.129495**			0	1	46.6
750-a-0.647-d-0.5-e-0.75	1412	12404	662	1410	12398	1.2	**1382.77203**			0	1	82.1
750-a-0.647-d-0.75-e-0.25	1366	12128	616	1365	12126	0.9	**266.983922**			0	1	55.4
750-a-0.647-d-0.75-e-0.5	1423	12470	673	1423	12470	1.0	**580.407832**			0	1	65.7
750-a-0.647-d-0.75-e-0.75	1462	12704	712	1462	12704	1.1	**764.156726**			0	1	85.8
750-a-1-d-0.25-e-0.25	1079	21612	329	1079	21612	0.8	**708.143835**			0	1	11.3
750-a-1-d-0.25-e-0.5	1229	22512	479	1229	22512	1.0	**1426.44904**			0	1	34.1
750-a-1-d-0.25-e-0.75	1364	23322	614	1364	23322	1.3	**2116.58233**			0	1	74.3
750-a-1-d-0.5-e-0.25	1206	22374	456	1206	22374	1.0	**403.177763**			0	1	23.2
750-a-1-d-0.5-e-0.5	1315	23028	565	1315	23028	1.2	**946.129495**			0	1	46.6
750-a-1-d-0.5-e-0.75	1412	23610	662	1412	23610	1.4	**1382.77203**			0	1	70.9
750-a-1-d-0.75-e-0.25	1366	23334	616	1366	23334	1.3	**266.983922**			0	1	49.4

cont. next page

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	t [s]	Dual	Primal	Gap %	C	N	t [s]
750-a-1-d-0.75-e-0.5	1423	23676	673	1423	23676	1.4	**580.407832**			0	1	62.7
750-a-1-d-0.75-e-0.75	1462	23910	712	1462	23910	1.6	**764.156726**			0	1	79.6

Table 38. Detailed computational results for the GSTP, test set GSTP1.

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	t [s]	Dual	Primal	Gap %	C	N	t [s]
gstp30f2	474	1828	30	465	1806	30	0.2	**569**			54	1	5.7
gstp31f2	349	1284	31	345	1274	31	0.1	**635**			88	1	7.9
gstp33f2	452	1746	33	450	1742	33	0.2	**513**			41	1	4.4
gstp34f2	1253	5000	34	1249	4990	34	1.0	635.746445	647	1.8	102	94	>7201.1
gstp36f2	442	1672	36	437	1662	36	0.2	**610**			90	1	11.6
gstp37f2	1054	4216	37	1052	4210	37	0.9	**485**			180	1	863.9
gstp38f2	618	2504	38	615	2496	38	0.3	**656**			62	403	4064.5
gstp39f2	707	3310	39	705	3304	39	0.6	412.61754	450	9.1	44	690	>7200.6

Table 39. Detailed computational results for the GSTP, test set GSTP2.

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	$\lvert T\rvert$	t [s]	Dual	Primal	Gap %	C	N	t [s]
gstp50f2	1142	4622	50	1140	4618	50	0.8	660.935008	674	2.0	113	119	>7200.8
gstp55f2	1751	6804	55	1749	6800	55	1.4	862.654988	891	3.3	152	11	>7201.4
gstp60f2	838	3528	60	837	3526	60	0.6	1154.87643	1164	0.8	162	791	>7200.7
gstp64f2	1860	7380	64	1855	7366	64	1.7	899.574265	938	4.3	117	13	>7201.7
gstp66f2	2623	10100	66	2619	10092	66	2.7	914.61628	920	0.6	247	1	>7203.2
gstp73f2	1911	7308	73	1899	7276	73	1.8	**1207**			284	1	6427.8
gstp76f2	1818	6990	76	1812	6972	76	1.7	**1026**			484	3	6967.6
gstp78f2	2355	9384	78	2348	9364	78	2.4	1057.95665	1100	4.0	113	19	>7202.5
gstp83f2	3177	12530	83	3171	12516	83	4.1	876.302444	908	3.6	199	1	>7204.4
gstp84f2	2358	9134	84	2351	9120	84	2.5	1006.91729	1095	8.7	77	12	>7202.6

Table 40. Detailed computational results for the HCDSTP, test set gr12. All instances have 10 terminals (before and after preprocessing).

Instance	Original $\lvert V\rvert$	$\lvert A\rvert$	Presolved $\lvert V\rvert$	$\lvert A\rvert$	t [s]	Optimum	C	N	t [s]
wo10-cr100-se0	809	14396	809	14396	0.0	**171486**	257	3	173.1
wo10-cr100-se10	809	14428	801	14232	0.1	**117081**	225	1	8.9
wo10-cr100-se11	809	14386	809	14386	0.0	**125785**	199	1	26.1
wo10-cr200-se7	809	44696	809	44678	0.1	**46306**	230	3	89.5
wo10-cr200-se8	809	44654	809	44636	0.1	**61177**	319	101	722.0
wo10-cr200-se9	809	44670	809	44652	0.1	**51737**	245	141	454.2
wo11-cr100-se10	809	7432	549	5718	0.3	**136516**	107	1	3.9
wo11-cr100-se11	809	7430	683	7352	0.0	**145251**	127	1	4.2
wo11-cr100-se1	809	7444	689	7440	0.0	**182082**	129	1	4.6
wo11-cr100-se2	809	7394	689	7390	0.0	**163872**	220	1	2.9
wo11-cr200-se10	809	15262	590	12550	0.4	**59523**	130	1	5.1
wo11-cr200-se11	809	15260	689	15244	0.0	**66786**	156	1	12.8
wo11-cr200-se1	809	15274	689	15258	0.0	**76353**	152	1	9.9
wo11-cr200-se2	809	15224	689	15208	0.0	**75434**	274	1	11.1
wo12-cr100-se10	809	9360	684	9278	0.0	**167223**	138	1	6.8
wo12-cr100-se11	809	9852	708	9846	0.0	**199679**	127	1	11.6
wo12-cr100-se1	809	9446	695	9420	0.0	**164198**	110	1	3.7
wo12-cr100-se7	809	9702	594	7968	0.3	**136232**	89	1	2.1
wo12-cr200-se9	809	28346	611	24362	0.6	**46408**	123	1	7.6

Table 41. Detailed computational results for the HCDSTP, test set gr14. All instances have 10 terminals (before and after preprocessing).

	Original		Presolved								
Instance	$\|V\|$	$\|A\|$	$\|V\|$	$\|A\|$	t [s]	Dual	Primal	Gap %	C	N	t [s]
wo10-cr100-se0	3209	215940	3209	215922	0.7	147594.343	174545	18.3	637	4	>7200.8
wo10-cr100-se11	3209	215932	3209	215914	0.7	114381.31	125394	9.6	640	12	>7200.9
wo10-cr200-se3	3209	643552	3209	643330	1.7	44787.6614	55497	23.9	960	69	>7201.9
wo10-cr200-se4	3209	643414	3209	643186	1.7	39526.3491	54475	37.8	673	341	>7202.0
wo11-cr100-se6	3209	115502	2773	115494	0.4	199930.546	220015	10.0	516	8	>7200.4
wo11-cr200-se2	3209	232858	2773	232844	0.8	68756.618	76436	11.2	645	20	>7200.8
wo11-cr200-se3	3209	233104	2732	228878	1.8	**57930**			724	83	2728.5
wo11-cr200-se4	3209	233038	2773	233024	0.8	62838.967	69220	10.2	1028	29	>7200.9
wo12-cr100-se0	3209	153366	1862	100468	10.1	**118617**			504	3	430.3
wo12-cr100-se5	3209	156578	2643	149328	3.3	**131631**			533	1	914.1
wo12-cr100-se6	3209	157214	2765	155536	0.5	140490.954	155919	11.0	325	21	>7200.5
wo12-cr100-se7	3209	158984	2394	133792	7.2	**122306**			386	18	1623.5
wo12-cr100-se8	3209	157912	2662	149786	3.3	**116077**			446	42	2622.0
wo12-cr100-se9	3209	156658	2161	121488	10.4	**100813**			392	1	302.1
wo12-cr200-se0	3209	445774	2173	340992	27.2	46329.6121	56249	21.4	932	247	>7227.2
wo12-cr200-se10	3209	446040	2765	445864	1.3	50635.8186	69874	38.0	1216	231	>7201.4
wo12-cr200-se11	3209	457496	2782	457456	1.2	54753.5749	71694	30.9	774	198	>7204.9
wo12-cr200-se4	3209	460250	2764	452090	1.4	59815.3406	79384	32.7	892	146	>7202.5
wo12-cr200-se5	3209	456998	2778	456974	1.3	51059.3851	59212	16.0	901	172	>7201.4
wo12-cr200-se6	3209	460500	2780	459786	1.3	54617.8695	66538	21.8	445	55	>7202.1
wo12-cr200-se7	3209	464090	2516	408220	15.1	54283.1768	62502	15.1	689	110	>7216.0

Table 42. Detailed computational results for the HCDSTP, test set gr16. All instances have 10 terminals (before and after preprocessing).

	Original		Presolved								
Instance	$\|V\|$	$\|A\|$	$\|V\|$	$\|A\|$	t [s]	Dual	Primal	Gap %	C	N	t [s]
wo10-cr100-se0	12509	2843882	11604	2843678	6.8	67934.3155	178781	163.2	1649	1	>7208.4
wo10-cr100-se10	12509	2844058	11319	2772610	129.4	68639.4849	122284	78.2	1603	1	>7331.9
wo10-cr100-se6	12509	2843894	11604	2843690	6.9	69686.5234	199237	185.9	1417	3	>7207.2
wo10-cr200-se0	12509	8741560	11604	8738884	29.7	36160	68834	90.4	311	1	>7231.1
wo10-cr200-se3	12509	8741850	11604	8739162	29.8	32976	59383	80.1	247	1	>7235.9
wo10-cr200-se4	12509	8741234	11604	8738558	29.7	34218.3333	66166	93.4	272	1	>7240.3
wo10-cr200-se5	12509	8740874	11604	8738198	29.7	35158	68277	94.2	240	1	>7244.3
wo10-cr200-se7	12509	8741906	9692	7159770	2939.8	32432.3125	46438	43.2	329	1	>10143.0
wo11-cr100-se0	12509	1634066	10654	1634018	3.9	92733.2864	204001	120.0	1237	1	>7204.4
wo11-cr100-se10	12509	1633968	8811	1319422	383.4	85769.1902	124389	45.0	1120	1	>7583.5
wo11-cr200-se2	12509	3416158	10654	3415928	8.9	45476.5249	76168	67.5	1312	1	>7211.3
wo11-cr200-se3	12509	3416916	10449	3341260	117.3	45394.1664	57820	27.4	1296	1	>7319.8
wo12-cr100-se2	12509	2172502	10486	2145056	5.3	96880.9782	194788	101.1	1509	1	>7207.4
wo12-cr100-se3	12509	2173508	10426	2122636	7.9	90073.8988	151797	68.5	1404	1	>7209.1
wo12-cr200-se2	12509	6560440	10543	6530350	22.8	43813.1724	81064	85.0	439	1	>7230.0
wo12-cr200-se3	12509	6557828	10494	6465210	22.8	40141.5455	62201	55.0	405	1	>7224.9
wo12-cr200-se4	12509	6420904	10422	6281784	19.9	43269.8722	83053	91.9	438	1	>7224.6
wo12-cr200-se7	12509	6766046	9903	6190724	1016.1	41470.7083	64796	56.2	400	1	>8231.3
wo12-cr200-se8	12509	6207724	10434	6178476	111.6	38677.7129	54757	41.6	427	1	>7313.8
wo12-cr200-se9	12509	6571406	9928	6168132	924.0	36254.0664	50364	38.9	462	1	>8124.9