Dijkstra meets Steiner: computational results of a fast exact Steiner tree algorithm

Stefan Hougardy, Jannik Silvanus, and Jens Vygen

Research Institute for Discrete Mathematics, University of Bonn

Abstract. We report detailed computational results of a new exact algorithm for the Steiner tree problem in graphs on DIMACS instances. In constrast to previous practical Steiner tree solvers which rely primarily on reductions, heuristics and branching, our algorithm combines a dynamic programming approach with future cost estimates and pruning. We achieve a best-known worst-case runtime. Computational results indicate that our algorithm performs particularly well on large-scale instances arising from VLSI design and on instances with few terminals. On the latter instance type, our strong worst-case runtime guarantee excludes excessive runtimes, which may occur with branch & bound techniques.

1 Introduction

The purpose of this paper is to present computational results of the algorithm described in [1] on instances of the 11th DIMACS implementation challenge. In Section 2, we give a very brief description of that algorithm, while Section 3 contains an analysis of our results. Detailed data can be found in Appendix A.

2 The Algorithm

We consider the well-known Steiner tree problem in graphs: Given an undirected graph G, costs $c : E(G) \to \mathbb{R}_{\geq 0}$ and a terminal set $D \subseteq V(G)$, find a tree Y in G such that $D \subseteq V(Y)$ and $c(E(Y))$ is minimum.

For a set $X \subseteq V(G)$, we denote by $\mathrm{smt}(X)$ the length of a shortest Steiner tree for the terminal set X. Our algorithm chooses an arbitrary root terminal $t \in D$. We call the remaining terminals $D' := D \setminus \{t\}$ source terminals. Now consider the function $l : V(G) \times 2^{D'} \to \mathbb{R}_{\geq 0}$ with $l(v, I) := \mathrm{smt}(\{v\} \cup I)$. Then, $l(t, D')$ gives the cost of an optimum Steiner tree.

Our algorithm uses dynamic programming to compute values of l. We maintain labels for pairs $(v, I) \in V(G) \times 2^{D'}$. Initially, we set $l(v, \{v\}) := 0$ for all $v \in D'$ and $l(v, I) := \infty$ for all other elements. In each iteration, one element (v, I) becomes active, updates other elements and becomes permanently labeled. More precisely, when such an element (v, I) becomes active, we proceed like Dijkstra's algorithm, updating $l(w, I)$ for all neighbors w of v by

$$l(w, I) := \min(l(w, I), l(v, I) + c(\{v, w\})).$$

Moreover, we perform a merge operation. For all labels (v, I') with $I' \subseteq D' \setminus I$, we update

$$l(v, I \cup I') := \min(l(v, I \cup I'), l(v, I) + l(v, I')).$$

We continue labeling until (t, D') becomes active.

This approach yields a theoretical worst-case runtime of $\mathcal{O}(3^k n + 2^k(n \log n + m))$, where $n = |V(G)|$, $m = |E(G)|$ and $k = |D|$.

Practical performance is significantly improved by using future cost estimates and pruning partial solutions, resulting in a competetive algorithm for instances with small terminal sets. See [1] for details and proofs.

3 Analysis of Results

We compare our results with those obtained by the state-of-the-art algorithm by Polzin and Vahdati [2], which produces the best results we are aware of. This algorithm successively performs various optimality-preserving reductions combined with a branch & bound approach. Note that our implementation is limited to instances with less than 64 terminals.

In Table 1, we give results on multiple instance classes (a) - (e). For each instance, we give its name, the number of vertices, edges and terminals. Then, we state the cost of an optimum solution as reported by our algorithm and the runtime in seconds. Detailed technical information about the runs can be found in the appendix. Moreover, for each instance, we give the runtime reported by Polzin and Vahdati. The computers we used for our experiments finished the DIMACS challenge benchmark roughly 25 % faster than the computer used by Polzin and Vahdati. For the lin testset, Polzin and Vahdati improved runtimes by modifying their algorithm to use stronger reductions. With default settings, their algorithm did not solve lin36 within a time limit of 24 hours.

Clearly, on VLSI-derived instances (a), the worst-case runtime of our algorithm is not reached, which is primarily caused by the high impact of our pruning method. In particular, on instances with large underlying graphs, our algorithm performs very well, beating the reduction-based solver. On rectilinear (b) and Euclidean instances (cf. Table 25 in Appendix A), pruning also is very effective.

In contrast, on group Steiner tree instances (c), our pruning implementation has no effect. Although these instances are based on VLSI-derived grid graphs with holes as well, they have been modified to model the groups as terminals: For each group of the group Steiner tree instance, a new terminal is added to the graph and connected to the elements of the group by edges of very high cost. By choosing the cost of these new edges sufficiently large, one can guarantee that a shortest Steiner tree in the new instance corresponds to a shortest group Steiner tree in the original instance and vice versa, since each terminal will be a leaf of the Steiner tree. To prune a label $(v, I) \in V(G) \times 2^{D'}$, our implementation requires a terminal $s \in D \setminus I$ such that $l(s, I) < l(v, I)$. On these instances, such a terminal s cannot exist, since any edge incident to s is much more expensive than any path in the original graph.

On incidence cost instances (d), where edges incident to terminals are assigned larger costs, a similar effect can be observed.

Although neither pruning nor future cost estimates do have a noticable effect on instances from the hard PUC testset (e), our algorithm performs very well on instances with few terminals. This is caused by the strong worst-case runtime guarantee, which, albeit exponential in the number of terminals, is quasilinear in the size of the graph.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]	Time PV [s]
(a) VLSI-derived grid graphs with holes						
diw0779	11821	22516	50	4440	1.58	1.26
diw0819	10553	20066	32	3399	0.29	0.52
diw0820	11749	22384	37	4167	1.46	1.06
lin23	3716	6750	52	17560	14.05	0.54
lin24	7998	14734	16	15076	0.09	1.73
lin30	19091	35644	31	27684	0.71	14.74
lin32	19112	35665	53	39832	142.60	816.51
lin34	38282	71521	34	45018	10.03	1848.24
lin35	38294	71533	45	50559	24.92	1911.09
lin36	38307	71546	58	55608	46.16	39931.77
(b) Rectilinear obstacle-avoiding instances preprocessed by ObSteiner						
ind5	114	228	33	1341	0.01	0.01
rc03	109	202	50	54160	0.12	0.00
rt02	788	1938	50	45852	0.59	1.99
(c) Group Steiner tree instances						
wrp3-14	128	247	14	1400250	3.65	0.01
wrp3-15	138	257	15	1500422	54.73	0.01
wrp3-16	204	374	16	1600208	12.02	0.03
wrp3-17	177	354	17	1700442	431.07	0.02
wrp3-19	189	353	19	1900439	1777.17	0.03
(d) Random graphs with so-called incidence costs						
i160-141	160	2544	12	2549	3.38	0.01
i320-111	320	1845	17	4273	1746.63	0.03
i640-022	640	204480	9	1756	4.41	0.52
i640-031	640	1280	9	3278	0.05	0.00
i640-043	640	40896	9	1931	1.11	0.13
(e) Artificial instances from the hard PUC testset						
cc3-4p	64	288	8	2338	0.01	1.99
cc3-4u	64	288	8	23	0.01	1.37
cc3-5p	125	750	13	3661	3.24	87.98
cc3-5u	125	750	13	36	4.97	115.83
cc6-2p	64	192	12	3271	0.10	0.40
cc6-2u	64	192	12	32	0.26	0.90

Table 1: Results on various instance types.

Note that our approach is very general and not limited to the future cost estimates and pruning strategies proposed in [1], which are tuned for instances from VLSI design. For example, one could easily achieve better results on the group Steiner tree instances: future cost estimates could incorporate the fixed costs induced by each remaining terminal, while pruning could exploit the fact that each Steiner tree has to include at least one element of each group.

We finally remark that our algorithm can easily be modified to compute all near-optimal Steiner trees efficiently, a feature that we used for packing Steiner trees in VLSI layout (see [1]). Algorithms based on reductions and branching typically do not have this property.

References

[1] Hougardy, S., Silvanus, J., Vygen, J.: Dijkstra meets Steiner: a fast exact goal-oriented Steiner tree algorithm. CoRR **abs/1406.0492** (2014). URL `http://arxiv.org/abs/1406.0492`

[2] Polzin, T., Vahdati, S.: The Steiner tree challenge: An updated study (2014). URL `http://dimacs11.cs.princeton.edu/papers/PolzinVahdatiDIMACS.pdf`

A Results on DIMACS Instances

We present detailed computational results on DIMACS testsets. Our implementation is limited to instances with less than 64 terminals, so we exclude instances with more terminals.

The implementation of our algorithm is written in the C++ programming language. We compiled our code using the gcc 4.4 compiler.

The experiments were performed single-threaded on four machines with identical hardware (3.33 GHz Intel Xeon W5590 CPUs, 144 GB main memory). These machines produced scores between 383.572438 and 395.399006 using the DIMACS benchmark code.

For these experiments, we set the time limit to 1800 s. On 95 % of the instances, the algorithm either took less than 180 s to terminate or exceeded the run time limit. The reported runtimes do not include the time to read the instance file from disk.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
alue2087	1244	1971	34	1049	0.972
alue2105	1220	1858	34	1032	0.198
alue7066	6405	10454	16	2256	0.053
alue7229	940	1474	34	824	0.028

Table 2: Results on the testset ALUE. Type: VLSI-derived grid graphs with holes.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
alut0787	1160	2089	34	982	1.601
alut0805	966	1666	34	958	0.198
alut2764	387	626	34	640	0.036

Table 3: Results on the testset ALUT. Type: VLSI-derived grid graphs with holes.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
diw0234	5349	10086	25	1996	0.070
diw0250	353	608	11	350	0.003
diw0260	539	985	12	468	0.002
diw0313	468	822	14	397	0.002
diw0393	212	381	11	302	0.001
diw0445	1804	3311	33	1363	0.054
diw0459	3636	6789	25	1362	0.028
diw0460	339	579	13	345	0.003
diw0473	2213	4135	25	1098	0.055
diw0487	2414	4386	25	1424	0.316
diw0495	938	1655	10	616	0.006
diw0513	918	1684	10	604	0.006
diw0523	1080	2015	10	561	0.006
diw0540	286	465	10	374	0.003
diw0559	3738	7013	18	1570	0.052

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
diw0778	7231	13727	24	2173	0.101
diw0779	11821	22516	50	4440	1.579
diw0795	3221	5938	10	1550	0.019
diw0801	3023	5575	10	1587	0.019
diw0819	10553	20066	32	3399	0.287
diw0820	11749	22384	37	4167	1.464

Table 4: Results on the testset DIW. Type: VLSI-derived grid graphs with holes.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
dmxa0296	233	386	12	344	0.002
dmxa0368	2050	3676	18	1017	0.015
dmxa0454	1848	3286	16	914	0.009
dmxa0628	169	280	10	275	0.001
dmxa0734	663	1154	11	506	0.004
dmxa0848	499	861	16	594	0.020
dmxa0903	632	1087	10	580	0.006
dmxa1010	3983	7108	23	1488	0.083
dmxa1109	343	559	17	454	0.010
dmxa1200	770	1383	21	750	0.059
dmxa1304	298	503	10	311	0.002
dmxa1516	720	1269	11	508	0.003
dmxa1721	1005	1731	18	780	0.013
dmxa1801	2333	4137	17	1365	0.046

Table 5: Results on the testset DMXA. Type: VLSI-derived grid graphs with holes.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
gap1307	342	552	17	549	0.029
gap1413	541	906	10	457	0.004
gap1500	220	374	17	254	0.006
gap1810	429	702	17	482	0.006
gap1904	735	1256	21	763	0.026
gap2007	2039	3548	17	1104	0.026
gap2119	1724	2975	29	1244	0.417
gap2740	1196	2084	14	745	0.010
gap2800	386	653	12	386	0.002
gap2975	179	293	10	245	0.001
gap3036	346	583	13	457	0.011
gap3100	921	1558	11	640	0.005

Table 6: Results on the testset GAP. Type: VLSI-derived grid graphs with holes.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
lin01	53	80	4	503	0.000
lin02	55	82	6	557	0.000
lin03	57	84	8	926	0.000
lin04	157	266	6	1239	0.000
lin05	160	269	9	1703	0.001
lin06	165	274	14	1348	0.004
lin07	307	526	6	1885	0.001
lin08	311	530	10	2248	0.001
lin09	313	532	12	2752	0.004
lin10	321	540	20	4132	0.022
lin11	816	1460	10	4280	0.006
lin12	818	1462	12	5250	0.008
lin13	822	1466	16	4609	0.012
lin14	828	1472	22	5824	0.011
lin15	840	1484	34	7145	0.048
lin16	1981	3633	12	6618	0.029
lin17	1989	3641	20	8405	0.041
lin18	1994	3646	25	9714	0.433
lin19	2010	3662	41	13268	9.970
lin20	3675	6709	11	6673	0.017
lin21	3683	6717	20	9143	0.058
lin22	3692	6726	28	10519	0.080
lin23	3716	6750	52	17560	14.048
lin24	7998	14734	16	15076	0.090
lin25	8007	14743	24	17803	0.289
lin26	8013	14749	30	21757	0.266
lin27	8017	14753	36	20678	1.484
lin29	19083	35636	24	23765	1.803
lin30	19091	35644	31	27684	0.707
lin31	19100	35653	40	31696	34.379
lin32	19112	35665	53	39832	142.601
lin34	38282	71521	34	45018	10.026
lin35	38294	71533	45	50559	24.925
lin36	38307	71546	58	55608	46.159

Table 7: Results on the testset LIN. Type: VLSI-derived grid graphs with holes.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
msm0580	338	541	11	467	0.002
msm0654	1290	2270	10	823	0.004
msm0709	1442	2403	16	884	0.010
msm0920	752	1264	26	806	0.022
msm1008	402	695	11	494	0.007
msm1234	933	1632	13	550	0.003
msm1477	1199	2078	31	1068	0.185
msm1707	278	478	11	564	0.002

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
msm1844	90	135	10	188	0.002
msm1931	875	1522	10	604	0.002
msm2000	898	1562	10	594	0.002
msm2152	2132	3702	37	1590	0.228
msm2326	418	723	14	399	0.002
msm2492	4045	7094	12	1459	0.030
msm2525	3031	5239	12	1290	0.011
msm2601	2961	5100	16	1440	0.036
msm2705	1359	2458	13	714	0.017
msm2802	1709	2963	18	926	0.025
msm3277	1704	2991	12	869	0.009
msm3676	957	1554	10	607	0.004
msm3727	4640	8255	21	1376	0.044
msm3829	4221	7255	12	1571	0.026
msm4038	237	390	11	353	0.002
msm4114	402	690	16	393	0.002
msm4190	391	666	16	381	0.008
msm4224	191	302	11	311	0.001
msm4312	5181	8893	10	2016	0.032
msm4414	317	476	11	408	0.001
msm4515	777	1358	13	630	0.007

Table 8: Results on the testset MSM. Type: VLSI-derived grid graphs with holes.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
taq0023	572	963	11	621	0.004
taq0365	4186	7074	22	1914	0.042
taq0431	1128	1905	13	897	0.008
taq0631	609	932	10	581	0.009
taq0739	837	1438	16	848	0.017
taq0741	712	1217	16	847	0.012
taq0751	1051	1791	16	939	0.017
taq0891	331	560	10	319	0.003
taq0910	310	514	17	370	0.008
taq0920	122	194	17	210	0.001
taq0978	777	1239	10	566	0.003

Table 9: Results on the testset TAQ. Type: VLSI-derived grid graphs with holes.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
1r111	1250	4704	6	28000	0.002
1r112	1250	4704	6	28000	0.004
1r113	1250	4704	6	26000	0.002
1r121	1250	4704	6	36000	0.002
1r122	1250	4704	6	45000	0.007

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
1r123	1250	4704	6	40000	0.005
1r131	1250	4704	6	43000	0.004
1r132	1250	4704	6	37000	0.005
1r133	1250	4704	6	36000	0.002
1r211	1250	4704	31	77000	0.345
1r212	1250	4704	30	81000	0.063
1r213	1250	4704	29	70000	0.694
1r221	1250	4704	31	79000	0.132
1r222	1250	4704	31	68000	0.061
1r223	1250	4704	31	77000	0.110
1r231	1250	4704	30	80000	0.128
1r232	1250	4704	29	86000	0.309
1r233	1250	4704	31	71000	1.515
1r311	1250	4704	56	–	timeout
1r312	1250	4704	60	–	timeout
1r313	1250	4704	58	106000	477.125
1r321	1250	4704	59	–	timeout
1r322	1250	4704	60	113000	1557.419
1r323	1250	4704	60	–	timeout
1r331	1250	4704	58	103000	1.174
1r332	1250	4704	58	109000	47.785
1r333	1250	4704	58	113000	1659.351

Table 10: Results on the testset 1R. Type: 2D grid graphs.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
2r111	2000	11600	9	28000	0.006
2r112	2000	11600	9	32000	0.008
2r113	2000	11600	9	28000	0.006
2r121	2000	11600	9	28000	0.009
2r122	2000	11600	9	29000	0.008
2r123	2000	11600	9	25000	0.008
2r131	2000	11600	9	27000	0.010
2r132	2000	11600	9	33000	0.015
2r133	2000	11600	9	29000	0.007
2r211	2000	11600	50	–	timeout
2r212	2000	11600	49	–	timeout
2r213	2000	11600	48	–	timeout
2r221	2000	11600	50	–	timeout
2r222	2000	11600	50	–	timeout
2r223	2000	11600	49	–	timeout
2r231	2000	11600	50	–	timeout
2r232	2000	11600	49	–	timeout
2r233	2000	11600	47	–	timeout

Table 11: Results on the testset 2R. Type: 3D grid graphs.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
es10fst01	18	20	10	22920745	0.000
es10fst02	14	13	10	19134104	0.000
es10fst03	17	20	10	26003678	0.000
es10fst04	18	20	10	20461116	0.000
es10fst05	12	11	10	18818916	0.000
es10fst06	17	20	10	26540768	0.000
es10fst07	14	13	10	26025072	0.000
es10fst08	21	28	10	25056214	0.000
es10fst09	21	29	10	22062355	0.000
es10fst10	18	21	10	23936095	0.000
es10fst11	14	13	10	22239535	0.000
es10fst12	13	12	10	19626318	0.000
es10fst13	18	21	10	19483914	0.000
es10fst14	24	32	10	21856128	0.000
es10fst15	16	18	10	18641924	0.000

Table 12: Results on the testset ES10FST. Type: Rectilinear instances after FST-preprocessing by GeoSteiner.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
es20fst01	29	28	20	33703886	0.004
es20fst02	29	28	20	32639486	0.001
es20fst03	27	26	20	27847417	0.002
es20fst04	57	83	20	27624394	0.003
es20fst05	54	77	20	34033163	0.002
es20fst06	29	28	20	36014241	0.001
es20fst07	45	59	20	34934874	0.001
es20fst08	52	74	20	38016346	0.006
es20fst09	36	42	20	36739939	0.003
es20fst10	49	67	20	34024740	0.002
es20fst11	33	36	20	27123908	0.001
es20fst12	33	36	20	30451397	0.002
es20fst13	35	40	20	34438673	0.002
es20fst14	36	44	20	34062374	0.005
es20fst15	37	43	20	32303746	0.001

Table 13: Results on the testset ES20FST. Type: Rectilinear instances after FST-preprocessing by GeoSteiner.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
es30fst01	79	115	30	40692993	0.019
es30fst02	71	97	30	40900061	0.015
es30fst03	83	120	30	43120444	0.010
es30fst04	80	115	30	42150958	0.008

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
es30fst05	58	71	30	41739748	0.004
es30fst06	83	119	30	39955139	0.029
es30fst07	53	64	30	43761391	0.004
es30fst08	69	93	30	41691217	0.005
es30fst09	43	44	30	37133658	0.012
es30fst10	48	52	30	42686610	0.008
es30fst11	79	112	30	41647993	0.005
es30fst12	46	48	30	38416720	0.011
es30fst13	65	84	30	37406646	0.005
es30fst14	53	58	30	42897025	0.023
es30fst15	118	188	30	43035576	0.069

Table 14: Results on the testset ES30FST. Type: Rectilinear instances after FST-preprocessing by GeoSteiner.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
es40fst01	93	127	40	44841522	0.019
es40fst02	82	105	40	46811310	0.008
es40fst03	87	116	40	49974157	0.050
es40fst04	55	55	40	45289864	0.011
es40fst05	121	180	40	51940413	0.082
es40fst06	92	123	40	49753385	0.024
es40fst07	77	95	40	45639009	0.063
es40fst08	98	137	40	48745996	0.014
es40fst09	107	153	40	51761789	0.024
es40fst10	107	152	40	57136852	0.069
es40fst11	97	135	40	46734214	0.017
es40fst12	67	75	40	43843378	0.020
es40fst13	78	95	40	51884545	0.017
es40fst14	98	134	40	49166952	0.021
es40fst15	93	129	40	50828067	0.037

Table 15: Results on the testset ES40FST. Type: Rectilinear instances after FST-preprocessing by GeoSteiner.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
es50fst01	118	160	50	54948660	0.028
es50fst02	125	177	50	55484245	0.446
es50fst03	128	182	50	54691035	0.041
es50fst04	106	138	50	51535766	0.270
es50fst05	104	135	50	55186015	0.241
es50fst06	126	182	50	55804287	0.447
es50fst07	143	211	50	49961178	0.044

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
es50fst08	83	96	50	53754708	0.085
es50fst09	139	202	50	53456773	0.492
es50fst10	139	207	50	54037963	2.946
es50fst11	100	131	50	52532923	0.018
es50fst12	110	149	50	53409291	0.172
es50fst13	92	116	50	53891019	0.045
es50fst14	120	167	50	53551419	0.089
es50fst15	112	147	50	52180862	0.108

Table 16: Results on the testset ES50FST. Type: Rectilinear instances after FST-preprocessing by GeoSteiner.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
es60fst01	123	159	60	53761423	0.417
es60fst02	186	280	60	55367804	3.037
es60fst03	113	142	60	56566797	0.067
es60fst04	162	238	60	55371042	0.149
es60fst05	119	148	60	54704991	0.070
es60fst06	130	174	60	60421961	1.100
es60fst07	188	280	60	58978041	0.071
es60fst08	109	133	60	58138178	0.365
es60fst09	151	216	60	55877112	0.471
es60fst10	133	177	60	57624488	0.040
es60fst11	121	154	60	56141666	0.562
es60fst12	176	257	60	59791362	3.279
es60fst13	157	226	60	61213533	0.082
es60fst14	118	149	60	56035528	0.057
es60fst15	117	151	60	56622581	0.099

Table 17: Results on the testset ES60FST. Type: Rectilinear instances after FST-preprocessing by GeoSteiner.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
att48fst	139	202	48	30236	0.383
berlin52fst	89	104	52	6760	32.304
eil51fst	181	289	51	409	513.936

Table 18: Results on the testset TSPFST. Type: Rectilinear instances after FST-preprocessing by GeoSteiner.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
ind1	18	31	10	604	0.001
ind2	31	57	10	9500	0.000
ind3	16	23	10	600	0.000
ind4	74	146	25	1086	0.012
ind5	114	228	33	1341	0.012
rc01	21	35	10	25980	0.000
rc02	87	176	30	41350	0.016
rc03	109	202	50	54160	0.117
rt01	262	740	10	2146	0.002
rt02	788	1938	50	45852	0.594

Table 19: Results on the testset Copenhagen14. Type: Instances of the Obstacle-avoiding rectilinear Steiner tree problem after FST-preprocessing by ObSteiner and merging the FSTs into a single graph.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
wrp3-11	128	227	11	1100361	0.059
wrp3-12	84	149	12	1200237	0.074
wrp3-13	311	613	13	1300497	19.264
wrp3-14	128	247	14	1400250	3.649
wrp3-15	138	257	15	1500422	54.730
wrp3-16	204	374	16	1600208	12.021
wrp3-17	177	354	17	1700442	431.069
wrp3-19	189	353	19	1900439	1777.170
wrp3-20	245	454	20	–	timeout
wrp3-21	237	444	21	–	timeout
wrp3-22	233	431	22	–	timeout
wrp3-23	132	230	23	–	timeout
wrp3-24	262	487	24	–	timeout
wrp3-25	246	468	25	–	timeout
wrp3-26	402	780	26	–	timeout
wrp3-27	370	721	27	–	timeout
wrp3-28	307	559	28	–	timeout
wrp3-29	245	436	29	–	timeout
wrp3-30	467	896	30	–	timeout
wrp3-31	323	592	31	–	timeout
wrp3-33	437	838	33	–	timeout
wrp3-34	1244	2474	34	–	timeout
wrp3-36	435	818	36	–	timeout
wrp3-37	1011	2010	37	–	timeout
wrp3-38	603	1207	38	–	timeout
wrp3-39	703	1616	39	–	timeout
wrp3-41	178	307	41	–	timeout
wrp3-42	705	1373	42	–	timeout
wrp3-43	173	298	43	–	timeout

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
wrp3-45	1414	2813	45	–	timeout
wrp3-48	925	1738	48	–	timeout
wrp3-49	886	1800	49	–	timeout
wrp3-50	1119	2251	50	–	timeout
wrp3-52	701	1352	52	–	timeout
wrp3-53	775	1471	53	–	timeout
wrp3-55	1645	3186	55	–	timeout
wrp3-56	853	1590	56	–	timeout
wrp3-60	838	1763	60	–	timeout
wrp3-62	670	1316	62	–	timeout

Table 20: Results on the testset WRP3. Type: Group Steiner tree instances arising from VLSI design modeled as Steiner tree instances by connecting each terminal to the vertices if its group by edges of very high cost.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
wrp4-11	123	233	11	1100179	0.172
wrp4-13	110	188	13	1300798	0.020
wrp4-14	145	283	14	1400290	2.598
wrp4-15	193	369	15	1500405	10.091
wrp4-16	311	579	16	1601190	137.867
wrp4-17	223	404	17	1700525	66.646
wrp4-18	211	380	18	1801464	1366.034
wrp4-19	119	206	19	1901446	0.915
wrp4-21	529	1032	21	–	timeout
wrp4-22	294	568	22	–	timeout
wrp4-23	257	515	23	–	timeout
wrp4-24	493	963	24	–	timeout
wrp4-25	422	808	25	–	timeout
wrp4-26	396	781	26	–	timeout
wrp4-27	243	497	27	–	timeout
wrp4-28	272	545	28	–	timeout
wrp4-29	247	505	29	–	timeout
wrp4-30	361	724	30	–	timeout
wrp4-31	390	786	31	–	timeout
wrp4-32	311	632	32	–	timeout
wrp4-33	304	571	33	–	timeout
wrp4-34	314	650	34	–	timeout
wrp4-35	471	954	35	–	timeout
wrp4-36	363	750	36	–	timeout
wrp4-37	522	1054	37	–	timeout
wrp4-38	294	618	38	–	timeout
wrp4-39	802	1553	39	–	timeout
wrp4-40	538	1088	40	–	timeout
wrp4-41	465	955	41	–	timeout

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
wrp4-42	552	1131	42	–	timeout
wrp4-43	596	1148	43	–	timeout
wrp4-44	398	788	44	–	timeout
wrp4-45	388	815	45	–	timeout
wrp4-46	632	1287	46	–	timeout
wrp4-47	555	1098	47	–	timeout
wrp4-48	451	825	48	–	timeout
wrp4-49	557	1080	49	–	timeout
wrp4-50	564	1112	50	–	timeout
wrp4-51	668	1306	51	–	timeout
wrp4-52	547	1115	52	–	timeout
wrp4-53	615	1232	53	–	timeout
wrp4-54	688	1388	54	–	timeout
wrp4-55	610	1201	55	–	timeout
wrp4-56	839	1617	56	–	timeout
wrp4-58	757	1493	58	–	timeout
wrp4-59	904	1806	59	–	timeout
wrp4-60	693	1370	60	–	timeout
wrp4-61	775	1538	61	–	timeout
wrp4-62	1283	2493	62	–	timeout
wrp4-63	1121	2227	63	–	timeout

Table 21: Results on the testset WRP4. Type: Group Steiner tree instances arising from VLSI design modeled as Steiner tree instances by connecting each terminal to the vertices if its group by edges of very high cost.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
I052a	160	237	23	13309487	0.011
I054a	540	817	25	15841596	0.016
I056a	290	439	34	14171206	0.079

Table 22: Results on the testset vienna-i-advanced. Type: Real-world telecommunication networks after an “advanced” preprocessing routine. We report the cost of an optimum solution in the original instance, computed as the sum of an optimum solution in the reduced instance and the fixed cost induced by the reductions.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
I052	2363	3761	40	–	timeout
I054	3803	6213	38	–	timeout
I056	1991	3176	51	–	timeout

Table 23: Results on the testset vienna-i-simple. Type: Real-world telecommunication networks after a “simple” preprocessing routine.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
berlin52	52	1326	16	1044	0.006
brasil58	58	1653	25	13655	0.003

Table 24: Results on the testset X. Type: Complete graphs with Euclidean costs.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
p455	100	4950	5	1138	0.001
p456	100	4950	5	1228	0.001
p457	100	4950	10	1609	0.001
p458	100	4950	10	1868	0.002
p459	100	4950	20	2345	0.002
p460	100	4950	20	2959	0.003
p461	100	4950	50	4474	0.031
p463	200	19900	10	1510	0.003
p464	200	19900	20	2545	0.009
p465	200	19900	40	3853	0.040

Table 25: Results on the testset P4E. Type: Complete graphs with Euclidean costs.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
p401	100	4950	5	155	0.001
p402	100	4950	5	116	0.001
p403	100	4950	5	179	0.001
p404	100	4950	10	270	0.001
p405	100	4950	10	270	0.004
p406	100	4950	10	290	0.002
p407	100	4950	20	590	0.216
p408	100	4950	20	542	0.159
p409	100	4950	50	–	timeout
p410	100	4950	50	1010	236.326

Table 26: Results on the testset P4Z. Type: Complete graphs with random costs.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
p619	100	180	5	7485	0.000
p620	100	180	5	8746	0.000
p621	100	180	5	8688	0.000
p622	100	180	10	15972	0.001
p623	100	180	10	19496	0.001
p624	100	180	20	20246	0.002
p625	100	180	20	23078	0.003

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
p626	100	180	20	22346	0.007
p627	100	180	50	40647	0.020
p628	100	180	50	40008	0.029
p629	100	180	50	43287	0.026
p630	200	370	10	26125	0.001
p631	200	370	20	39067	0.005
p632	200	370	40	56217	0.032

Table 27: Results on the testset P6E. Type: Sparse graphs with Euclidean costs.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
p602	100	180	5	8083	0.000
p603	100	180	5	5022	0.000
p604	100	180	10	11397	0.000
p605	100	180	10	10355	0.001
p606	100	180	11	13048	0.001
p607	100	180	21	15358	0.002
p608	100	180	21	14439	0.001
p609	100	180	20	18263	0.002
p610	100	180	50	30161	0.018
p611	100	180	50	26903	0.055
p612	100	180	50	30258	0.089
p613	200	370	10	18429	0.001
p614	200	370	20	27276	0.015
p615	200	370	40	42474	0.024

Table 28: Results on the testset P6Z. Type: Sparse graphs with random costs.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
b01	50	63	9	82	0.002
b02	50	63	13	83	0.001
b03	50	63	25	138	0.080
b04	50	100	9	59	0.000
b05	50	100	13	61	0.005
b06	50	100	25	122	0.223
b07	75	94	13	111	0.004
b08	75	94	19	104	0.004
b09	75	94	38	–	timeout
b10	75	150	13	86	0.006
b11	75	150	19	88	0.015
b12	75	150	38	174	94.694
b13	100	125	17	165	0.068
b14	100	125	25	235	1.531

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
b15	100	125	50	318	169.826
b16	100	200	17	127	0.004
b17	100	200	25	131	262.821
b18	100	200	50	–	timeout

Table 29: Results on the testset B. Type: Random sparse graphs with random costs.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
c01	500	625	5	85	0.001
c02	500	625	10	144	0.002
c06	500	1000	5	55	0.001
c07	500	1000	10	102	0.013
c11	500	2500	5	32	0.002
c12	500	2500	10	46	0.010
c16	500	12500	5	11	0.002
c17	500	12500	10	18	0.011

Table 30: Results on the testset C. Type: Random sparse graphs with random costs.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
d01	1000	1250	5	106	0.004
d02	1000	1250	10	220	0.038
d06	1000	2000	5	67	0.002
d07	1000	2000	10	103	0.009
d11	1000	5000	5	29	0.005
d12	1000	5000	10	42	0.008
d16	1000	25000	5	13	0.005
d17	1000	25000	10	23	0.045

Table 31: Results on the testset D. Type: Random sparse graphs with random costs.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
e01	2500	3125	5	111	0.008
e02	2500	3125	10	214	0.042
e06	2500	5000	5	73	0.005
e07	2500	5000	10	145	0.208
e11	2500	12500	5	34	0.007
e12	2500	12500	10	67	0.104
e16	2500	62500	5	15	0.013
e17	2500	62500	10	25	0.122

Table 32: Results on the testset E. Type: Random sparse graphs with random costs.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
i080-001	80	120	6	1787	0.001
i080-002	80	120	6	1607	0.000
i080-003	80	120	6	1713	0.000
i080-004	80	120	6	1866	0.000
i080-005	80	120	6	1790	0.000
i080-011	80	350	6	1479	0.001
i080-012	80	350	6	1484	0.001
i080-013	80	350	6	1381	0.001
i080-014	80	350	6	1397	0.002
i080-015	80	350	6	1495	0.001
i080-021	80	3160	6	1175	0.004
i080-022	80	3160	6	1178	0.004
i080-023	80	3160	6	1174	0.004
i080-024	80	3160	6	1161	0.006
i080-025	80	3160	6	1162	0.004
i080-031	80	160	6	1570	0.000
i080-032	80	160	6	2088	0.001
i080-033	80	160	6	1794	0.001
i080-034	80	160	6	1688	0.000
i080-035	80	160	6	1862	0.001
i080-041	80	632	6	1276	0.002
i080-042	80	632	6	1287	0.001
i080-043	80	632	6	1295	0.001
i080-044	80	632	6	1366	0.001
i080-045	80	632	6	1310	0.002
i080-101	80	120	8	2608	0.001
i080-102	80	120	8	2403	0.003
i080-103	80	120	8	2603	0.004
i080-104	80	120	8	2486	0.004
i080-105	80	120	8	2203	0.001
i080-111	80	350	8	2051	0.010
i080-112	80	350	8	1885	0.006
i080-113	80	350	8	1884	0.004
i080-114	80	350	8	1895	0.005
i080-115	80	350	8	1868	0.004
i080-121	80	3160	8	1561	0.022
i080-122	80	3160	8	1561	0.021
i080-123	80	3160	8	1569	0.023
i080-124	80	3160	8	1555	0.022
i080-125	80	3160	8	1572	0.023
i080-131	80	160	8	2284	0.001
i080-132	80	160	8	2180	0.002
i080-133	80	160	8	2261	0.002
i080-134	80	160	8	2070	0.002
i080-135	80	160	8	2102	0.001
i080-141	80	632	8	1788	0.013

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
i080-142	80	632	8	1708	0.012
i080-143	80	632	8	1767	0.008
i080-144	80	632	8	1772	0.009
i080-145	80	632	8	1762	0.008
i080-201	80	120	16	4760	0.088
i080-202	80	120	16	4650	0.119
i080-203	80	120	16	4599	0.897
i080-204	80	120	16	4492	8.227
i080-205	80	120	16	4564	0.504
i080-211	80	350	16	3631	119.087
i080-212	80	350	16	3677	109.470
i080-213	80	350	16	3678	118.177
i080-214	80	350	16	3734	102.434
i080-215	80	350	16	3681	133.969
i080-221	80	3160	16	3158	121.216
i080-222	80	3160	16	3141	114.031
i080-223	80	3160	16	3156	110.582
i080-224	80	3160	16	3159	112.269
i080-225	80	3160	16	3150	116.494
i080-231	80	160	16	4354	32.643
i080-232	80	160	16	4199	25.447
i080-233	80	160	16	4118	16.917
i080-234	80	160	16	4274	1.001
i080-235	80	160	16	4487	2.161
i080-241	80	632	16	3538	153.968
i080-242	80	632	16	3458	136.701
i080-243	80	632	16	3474	148.245
i080-244	80	632	16	3466	138.343
i080-245	80	632	16	3467	138.420
i080-301	80	120	20	5519	35.110
i080-302	80	120	20	5944	10.728
i080-303	80	120	20	5777	19.188
i080-304	80	120	20	5586	6.126
i080-305	80	120	20	5932	188.639
i080-311	80	350	20	–	timeout
i080-312	80	350	20	–	timeout
i080-313	80	350	20	–	timeout
i080-314	80	350	20	–	timeout
i080-315	80	350	20	–	timeout
i080-321	80	3160	20	–	timeout
i080-322	80	3160	20	–	timeout
i080-323	80	3160	20	–	timeout
i080-324	80	3160	20	–	timeout
i080-325	80	3160	20	–	timeout
i080-331	80	160	20	5226	801.603
i080-332	80	160	20	5362	870.465

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
i080-333	80	160	20	5381	519.693
i080-334	80	160	20	5264	974.638
i080-335	80	160	20	4953	1011.775
i080-341	80	632	20	–	timeout
i080-342	80	632	20	–	timeout
i080-343	80	632	20	–	timeout
i080-344	80	632	20	–	timeout
i080-345	80	632	20	–	timeout

Table 33: Results on the testset I080. Type: Random graphs with so-called incidence costs, designed to defy preprocessing.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
i160-001	160	240	7	2490	0.002
i160-002	160	240	7	2158	0.001
i160-003	160	240	7	2297	0.002
i160-004	160	240	7	2370	0.002
i160-005	160	240	7	2495	0.001
i160-011	160	812	7	1677	0.006
i160-012	160	812	7	1750	0.005
i160-013	160	812	7	1661	0.002
i160-014	160	812	7	1778	0.004
i160-015	160	812	7	1768	0.010
i160-021	160	12720	7	1352	0.033
i160-022	160	12720	7	1365	0.035
i160-023	160	12720	7	1351	0.033
i160-024	160	12720	7	1371	0.034
i160-025	160	12720	7	1366	0.032
i160-031	160	320	7	2170	0.002
i160-032	160	320	7	2330	0.002
i160-033	160	320	7	2101	0.002
i160-034	160	320	7	2083	0.001
i160-035	160	320	7	2103	0.002
i160-041	160	2544	7	1494	0.008
i160-042	160	2544	7	1486	0.007
i160-043	160	2544	7	1549	0.011
i160-044	160	2544	7	1478	0.010
i160-045	160	2544	7	1554	0.009
i160-101	160	240	12	3859	0.055
i160-102	160	240	12	3747	0.126
i160-103	160	240	12	3837	0.084
i160-104	160	240	12	4063	0.013
i160-105	160	240	12	3563	0.032
i160-111	160	812	12	2869	1.442
i160-112	160	812	12	2924	2.175
i160-113	160	812	12	2866	1.445

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
i160-114	160	812	12	2989	1.946
i160-115	160	812	12	2937	1.783
i160-121	160	12720	12	2363	4.782
i160-122	160	12720	12	2348	4.718
i160-123	160	12720	12	2355	5.195
i160-124	160	12720	12	2352	5.432
i160-125	160	12720	12	2351	4.770
i160-131	160	320	12	3356	0.240
i160-132	160	320	12	3450	0.148
i160-133	160	320	12	3585	0.331
i160-134	160	320	12	3470	0.084
i160-135	160	320	12	3716	0.279
i160-141	160	2544	12	2549	3.375
i160-142	160	2544	12	2562	3.084
i160-143	160	2544	12	2557	2.665
i160-144	160	2544	12	2607	2.891
i160-145	160	2544	12	2578	4.449
i160-201	160	240	24	–	timeout
i160-202	160	240	24	–	timeout
i160-203	160	240	24	–	timeout
i160-204	160	240	24	–	timeout
i160-205	160	240	24	–	timeout
i160-211	160	812	24	–	timeout
i160-212	160	812	24	–	timeout
i160-213	160	812	24	–	timeout
i160-214	160	812	24	–	timeout
i160-215	160	812	24	–	timeout
i160-221	160	12720	24	–	timeout
i160-222	160	12720	24	–	timeout
i160-223	160	12720	24	–	timeout
i160-224	160	12720	24	–	timeout
i160-225	160	12720	24	–	timeout
i160-231	160	320	24	–	timeout
i160-232	160	320	24	–	timeout
i160-233	160	320	24	–	timeout
i160-234	160	320	24	–	timeout
i160-235	160	320	24	–	timeout
i160-241	160	2544	24	–	timeout
i160-242	160	2544	24	–	timeout
i160-243	160	2544	24	–	timeout
i160-244	160	2544	24	–	timeout
i160-245	160	2544	24	–	timeout
i160-301	160	240	40	–	timeout
i160-302	160	240	40	–	timeout
i160-303	160	240	40	–	timeout
i160-304	160	240	40	–	timeout

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
i160-305	160	240	40	–	timeout
i160-311	160	812	40	–	timeout
i160-312	160	812	40	–	timeout
i160-313	160	812	40	–	timeout
i160-314	160	812	40	–	timeout
i160-315	160	812	40	–	timeout
i160-321	160	12720	40	–	timeout
i160-322	160	12720	40	–	timeout
i160-323	160	12720	40	–	timeout
i160-324	160	12720	40	–	timeout
i160-325	160	12720	40	–	timeout
i160-331	160	320	40	–	timeout
i160-332	160	320	40	–	timeout
i160-333	160	320	40	–	timeout
i160-334	160	320	40	–	timeout
i160-335	160	320	40	–	timeout
i160-341	160	2544	40	–	timeout
i160-342	160	2544	40	–	timeout
i160-343	160	2544	40	–	timeout
i160-344	160	2544	40	–	timeout
i160-345	160	2544	40	–	timeout

Table 34: Results on the testset I160. Type: Random graphs with so-called incidence costs, designed to defy preprocessing.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
i320-001	320	480	8	2672	0.004
i320-002	320	480	8	2847	0.005
i320-003	320	480	8	2972	0.003
i320-004	320	480	8	2905	0.005
i320-005	320	480	8	2991	0.004
i320-011	320	1845	8	2053	0.023
i320-012	320	1845	8	1997	0.016
i320-013	320	1845	8	2072	0.028
i320-014	320	1845	8	2061	0.026
i320-015	320	1845	8	2059	0.032
i320-021	320	51040	8	1553	0.378
i320-022	320	51040	8	1565	0.371
i320-023	320	51040	8	1549	0.297
i320-024	320	51040	8	1553	0.335
i320-025	320	51040	8	1550	0.295
i320-031	320	640	8	2673	0.009
i320-032	320	640	8	2770	0.009
i320-033	320	640	8	2769	0.010
i320-034	320	640	8	2521	0.005
i320-035	320	640	8	2385	0.006

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
i320-041	320	10208	8	1707	0.046
i320-042	320	10208	8	1682	0.052
i320-043	320	10208	8	1723	0.053
i320-044	320	10208	8	1681	0.078
i320-045	320	10208	8	1686	0.052
i320-101	320	480	17	5548	17.007
i320-102	320	480	17	5556	10.003
i320-103	320	480	17	6239	180.136
i320-104	320	480	17	5703	119.533
i320-105	320	480	17	5928	90.994
i320-111	320	1845	17	4273	1746.631
i320-112	320	1845	17	–	timeout
i320-113	320	1845	17	4205	1432.921
i320-114	320	1845	17	–	timeout
i320-115	320	1845	17	4238	1703.699
i320-121	320	51040	17	–	timeout
i320-122	320	51040	17	–	timeout
i320-123	320	51040	17	–	timeout
i320-124	320	51040	17	–	timeout
i320-125	320	51040	17	–	timeout
i320-131	320	640	17	5255	355.407
i320-132	320	640	17	5052	30.216
i320-133	320	640	17	5125	47.136
i320-134	320	640	17	5272	370.241
i320-135	320	640	17	5342	237.738
i320-141	320	10208	17	–	timeout
i320-142	320	10208	17	–	timeout
i320-143	320	10208	17	–	timeout
i320-144	320	10208	17	–	timeout
i320-145	320	10208	17	–	timeout
i320-201	320	480	34	–	timeout
i320-202	320	480	34	–	timeout
i320-203	320	480	34	–	timeout
i320-204	320	480	34	–	timeout
i320-205	320	480	34	–	timeout
i320-211	320	1845	34	–	timeout
i320-212	320	1845	34	–	timeout
i320-213	320	1845	34	–	timeout
i320-214	320	1845	34	–	timeout
i320-215	320	1845	34	–	timeout
i320-221	320	51040	34	–	timeout
i320-222	320	51040	34	–	timeout
i320-223	320	51040	34	–	timeout
i320-224	320	51040	34	–	timeout
i320-225	320	51040	34	–	timeout
i320-231	320	640	34	–	timeout

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
i320-232	320	640	34	–	timeout
i320-233	320	640	34	–	timeout
i320-234	320	640	34	–	timeout
i320-235	320	640	34	–	timeout
i320-241	320	10208	34	–	timeout
i320-242	320	10208	34	–	timeout
i320-243	320	10208	34	–	timeout
i320-244	320	10208	34	–	timeout
i320-245	320	10208	34	–	timeout

Table 35: Results on the testset I320. Type: Random graphs with so-called incidence costs, designed to defy preprocessing.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
i640-001	640	960	9	4033	0.023
i640-002	640	960	9	3588	0.023
i640-003	640	960	9	3438	0.022
i640-004	640	960	9	4000	0.040
i640-005	640	960	9	4006	0.042
i640-011	640	4135	9	2392	0.150
i640-012	640	4135	9	2465	0.277
i640-013	640	4135	9	2399	0.180
i640-014	640	4135	9	2171	0.036
i640-015	640	4135	9	2347	0.115
i640-021	640	204480	9	1749	4.476
i640-022	640	204480	9	1756	4.411
i640-023	640	204480	9	1754	3.427
i640-024	640	204480	9	1751	4.023
i640-025	640	204480	9	1745	4.594
i640-031	640	1280	9	3278	0.047
i640-032	640	1280	9	3187	0.033
i640-033	640	1280	9	3260	0.047
i640-034	640	1280	9	2953	0.020
i640-035	640	1280	9	3292	0.032
i640-041	640	40896	9	1897	0.920
i640-042	640	40896	9	1934	0.690
i640-043	640	40896	9	1931	1.106
i640-044	640	40896	9	1938	0.842
i640-045	640	40896	9	1866	0.414
i640-101	640	960	25	–	timeout
i640-102	640	960	25	–	timeout
i640-103	640	960	25	–	timeout
i640-104	640	960	25	–	timeout
i640-105	640	960	25	–	timeout
i640-111	640	4135	25	–	timeout
i640-112	640	4135	25	–	timeout

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
i640-113	640	4135	25	–	timeout
i640-114	640	4135	25	–	timeout
i640-115	640	4135	25	–	timeout
i640-121	640	204480	25	–	timeout
i640-122	640	204480	25	–	timeout
i640-123	640	204480	25	–	timeout
i640-124	640	204480	25	–	timeout
i640-125	640	204480	25	–	timeout
i640-131	640	1280	25	–	timeout
i640-132	640	1280	25	–	timeout
i640-133	640	1280	25	–	timeout
i640-134	640	1280	25	–	timeout
i640-135	640	1280	25	–	timeout
i640-141	640	40896	25	–	timeout
i640-142	640	40896	25	–	timeout
i640-143	640	40896	25	–	timeout
i640-144	640	40896	25	–	timeout
i640-145	640	40896	25	–	timeout
i640-201	640	960	50	–	timeout
i640-202	640	960	50	–	timeout
i640-203	640	960	50	–	timeout
i640-204	640	960	50	–	timeout
i640-205	640	960	50	–	timeout
i640-211	640	4135	50	–	timeout
i640-212	640	4135	50	–	timeout
i640-213	640	4135	50	–	timeout
i640-214	640	4135	50	–	timeout
i640-215	640	4135	50	–	timeout
i640-221	640	204480	50	–	timeout
i640-222	640	204480	50	–	timeout
i640-223	640	204480	50	–	timeout
i640-224	640	204480	50	–	timeout
i640-225	640	204480	50	–	timeout
i640-231	640	1280	50	–	timeout
i640-232	640	1280	50	–	timeout
i640-233	640	1280	50	–	timeout
i640-234	640	1280	50	–	timeout
i640-235	640	1280	50	–	timeout
i640-241	640	40896	50	–	timeout
i640-242	640	40896	50	–	timeout
i640-243	640	40896	50	–	timeout
i640-244	640	40896	50	–	timeout
i640-245	640	40896	50	–	timeout

Table 36: Results on the testset I640. Type: Random graphs with so-called incidence costs, designed to defy preprocessing.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
bipe2p	550	5013	50	–	timeout
bipe2u	550	5013	50	–	timeout
cc3-10p	1000	13500	50	–	timeout
cc3-10u	1000	13500	50	–	timeout
cc3-11p	1331	19965	61	–	timeout
cc3-11u	1331	19965	61	–	timeout
cc3-4p	64	288	8	2338	0.006
cc3-4u	64	288	8	23	0.008
cc3-5p	125	750	13	3661	3.244
cc3-5u	125	750	13	36	4.971
cc5-3p	243	1215	27	–	timeout
cc5-3u	243	1215	27	–	timeout
cc6-2p	64	192	12	3271	0.098
cc6-2u	64	192	12	32	0.259
hc6p	64	192	32	–	timeout
hc6u	64	192	32	–	timeout

Table 37: Results on the testset PUC. Type: Artificial instances designed to be hard for existing solvers.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
cc3-10n	1000	13500	50	–	timeout
cc3-11n	1331	19965	61	–	timeout
cc3-4n	64	288	8	13	0.002
cc3-5n	125	750	13	20	0.498
cc5-3n	243	1215	27	–	timeout
cc6-2n	64	192	12	18	0.029

Table 38: Results on the testset SPG-PUCN. Type: Unweighted instances of the PUC testset, which contains artificial instances designed to be hard for existing solvers.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
antiwheel5	10	15	5	7	0.000
design432	8	20	4	9	0.000
oddcycle3	6	9	3	4	0.000
oddwheel3	7	9	4	5	0.000
se03	13	21	4	12	0.000

Table 39: Results on the testset SP. Type: Artificial instances.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
mc2	120	7140	60	–	timeout
mc3	97	4656	45	–	timeout

Table 40: Results on the testset MC. Type: Artificial instances.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
csd02	3	2	1	0	0.000
csd03	6	6	3	4	0.000
csd04	10	12	6	8	0.000
csd05	15	20	10	13	0.008
csd06	21	30	15	19	2.716
csd07	28	42	21	–	timeout
csd08	36	56	28	–	timeout
csd09	45	72	36	–	timeout
csd10	55	90	45	–	timeout
csd11	66	110	55	–	timeout

Table 41: Results on the testset csd. Type: Artificial instances arising from generalizations of Steiner tree LP gap examples.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
g01-00	7	9	3	8	0.000
g01-01	8	10	4	9	0.000
g01-02	9	11	5	10	0.000
g01-03	10	12	6	11	0.000
g01-04	11	13	7	12	0.000
g01-05	12	14	8	13	0.001
g01-06	13	15	9	14	0.002
g01-07	14	16	10	15	0.007
g01-08	15	17	11	16	0.009
g01-09	16	18	12	17	0.041
g01-10	17	19	13	18	0.170
g01-11	18	20	14	19	0.439
g01-12	19	21	15	20	1.982
g01-13	20	22	16	21	17.874
g01-14	21	23	17	22	43.402
g01-15	22	24	18	23	39.768
g02-00	9	16	4	14	0.000
g02-01	10	17	5	15	0.000
g02-02	11	18	6	16	0.000
g02-03	12	19	7	17	0.000
g02-04	13	20	8	18	0.001
g02-05	14	21	9	19	0.003
g02-06	15	22	10	20	0.009

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
g02-07	16	23	11	21	0.024
g02-08	17	24	12	22	0.110
g02-09	18	25	13	23	0.329
g02-10	19	26	14	24	1.364
g02-11	20	27	15	25	6.448
g02-12	21	28	16	26	24.978
g02-13	22	29	17	27	110.338
g02-14	23	30	18	28	383.143
g02-15	24	31	19	29	1167.363
g03-00	11	25	5	22	0.000
g03-01	12	26	6	23	0.000
g03-02	13	27	7	24	0.000
g03-03	14	28	8	25	0.001
g03-04	15	29	9	26	0.003
g03-05	16	30	10	27	0.022
g03-06	17	31	11	28	0.034
g03-07	18	32	12	29	0.111
g03-08	19	33	13	30	0.429
g03-09	20	34	14	31	1.547
g03-10	21	35	15	32	5.858
g03-11	22	36	16	33	27.003
g03-12	23	37	17	34	105.134
g03-13	24	38	18	35	450.582
g03-14	25	39	19	36	1647.116
g03-15	26	40	20	–	timeout
g04-00	13	36	6	32	0.000
g04-01	14	37	7	33	0.001
g04-02	15	38	8	34	0.001
g04-03	16	39	9	35	0.008
g04-04	17	40	10	36	0.011
g04-05	18	41	11	37	0.037
g04-06	19	42	12	38	0.118
g04-07	20	43	13	39	0.427
g04-08	21	44	14	40	1.429
g04-09	22	45	15	41	6.070
g04-10	23	46	16	42	27.234
g04-11	24	47	17	43	100.749
g04-12	25	48	18	44	421.287
g04-13	26	49	19	45	1394.767
g04-14	27	50	20	–	timeout
g04-15	28	51	21	–	timeout
g05-00	15	49	7	44	0.001
g05-01	16	50	8	45	0.001
g05-02	17	51	9	46	0.004
g05-03	18	52	10	47	0.011
g05-04	19	53	11	48	0.035

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	Opt	Time [s]
g05-05	20	54	12	49	0.123
g05-06	21	55	13	50	0.398
g05-07	22	56	14	51	1.524
g05-08	23	57	15	52	5.431
g05-09	24	58	16	53	21.325
g05-10	25	59	17	54	87.963
g05-11	26	60	18	55	341.771
g05-12	27	61	19	56	1184.734
g05-13	28	62	20	–	timeout
g05-14	29	63	21	–	timeout
g05-15	30	64	22	–	timeout
g06-00	17	64	8	58	0.002
g06-01	18	65	9	59	0.003
g06-02	19	66	10	60	0.011
g06-03	20	67	11	61	0.046
g06-04	21	68	12	62	0.113
g06-05	22	69	13	63	0.405
g06-06	23	70	14	64	1.452
g06-07	24	71	15	65	4.836
g06-08	25	72	16	66	19.348
g06-09	26	73	17	67	85.610
g06-10	27	74	18	68	305.491
g06-11	28	75	19	69	1104.198
g06-12	29	76	20	–	timeout
g06-13	30	77	21	–	timeout
g06-14	31	78	22	–	timeout
g06-15	32	79	23	–	timeout
g07-00	19	81	9	74	0.006
g07-01	20	82	10	75	0.010
g07-02	21	83	11	76	0.033
g07-03	22	84	12	77	0.117
g07-04	23	85	13	78	0.379
g07-05	24	86	14	79	1.466
g07-06	25	87	15	80	5.068
g07-07	26	88	16	81	19.532
g07-08	27	89	17	82	74.725
g07-09	28	90	18	83	278.158
g07-10	29	91	19	84	1055.338
g07-11	30	92	20	–	timeout
g07-12	31	93	21	–	timeout
g07-13	32	94	22	–	timeout
g07-14	33	95	23	–	timeout
g07-15	34	96	24	–	timeout
g08-00	21	100	10	92	0.008
g08-01	22	101	11	93	0.032
g08-02	23	102	12	94	0.114

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
g08-03	24	103	13	95	0.375
g08-04	25	104	14	96	1.310
g08-05	26	105	15	97	4.939
g08-06	27	106	16	98	19.766
g08-07	28	107	17	99	75.585
g08-08	29	108	18	100	274.968
g08-09	30	109	19	101	1150.217
g08-10	31	110	20	–	timeout
g08-11	32	111	21	–	timeout
g08-12	33	112	22	–	timeout
g08-13	34	113	23	–	timeout
g08-14	35	114	24	–	timeout
g08-15	36	115	25	–	timeout
g09-00	23	121	11	112	0.027
g09-01	24	122	12	113	0.104
g09-02	25	123	13	114	0.369
g09-03	26	124	14	115	1.312
g09-04	27	125	15	116	4.730
g09-05	28	126	16	117	18.854
g09-06	29	127	17	118	73.276
g09-07	30	128	18	119	270.732
g09-08	31	129	19	120	1136.499
g09-09	32	130	20	–	timeout
g09-10	33	131	21	–	timeout
g09-11	34	132	22	–	timeout
g09-12	35	133	23	–	timeout
g09-13	36	134	24	–	timeout
g09-14	37	135	25	–	timeout
g09-15	38	136	26	–	timeout
g10-00	25	144	12	134	0.087
g10-01	26	145	13	135	0.336
g10-02	27	146	14	136	1.254
g10-03	28	147	15	137	4.672
g10-04	29	148	16	138	16.468
g10-05	30	149	17	139	62.038
g10-06	31	150	18	140	219.883
g10-07	32	151	19	141	873.472
g10-08	33	152	20	–	timeout
g10-09	34	153	21	–	timeout
g10-10	35	154	22	–	timeout
g10-11	36	155	23	–	timeout
g10-12	37	156	24	–	timeout
g10-13	38	157	25	–	timeout
g10-14	39	158	26	–	timeout
g10-15	40	159	27	–	timeout
g11-00	27	169	13	158	0.335

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
g11-01	28	170	14	159	1.194
g11-02	29	171	15	160	4.496
g11-03	30	172	16	161	17.489
g11-04	31	173	17	162	59.407
g11-05	32	174	18	163	221.688
g11-06	33	175	19	164	848.617
g11-07	34	176	20	–	timeout
g11-08	35	177	21	–	timeout
g11-09	36	178	22	–	timeout
g11-10	37	179	23	–	timeout
g11-11	38	180	24	–	timeout
g11-12	39	181	25	–	timeout
g11-13	40	182	26	–	timeout
g11-14	41	183	27	–	timeout
g11-15	42	184	28	–	timeout
g12-00	29	196	14	184	0.567
g12-01	30	197	15	185	4.443
g12-02	31	198	16	186	16.490
g12-03	32	199	17	187	60.509
g12-04	33	200	18	188	205.166
g12-05	34	201	19	189	799.017
g12-06	35	202	20	–	timeout
g12-07	36	203	21	–	timeout
g12-08	37	204	22	–	timeout
g12-09	38	205	23	–	timeout
g12-10	39	206	24	–	timeout
g12-11	40	207	25	–	timeout
g12-12	41	208	26	–	timeout
g12-13	42	209	27	–	timeout
g12-14	43	210	28	–	timeout
g12-15	44	211	29	–	timeout
g13-00	31	225	15	212	1.934
g13-01	32	226	16	213	14.746
g13-02	33	227	17	214	55.254
g13-03	34	228	18	215	201.517
g13-04	35	229	19	216	732.902
g13-05	36	230	20	–	timeout
g13-06	37	231	21	–	timeout
g13-07	38	232	22	–	timeout
g13-08	39	233	23	–	timeout
g13-09	40	234	24	–	timeout
g13-10	41	235	25	–	timeout
g13-11	42	236	26	–	timeout
g13-12	43	237	27	–	timeout
g13-13	44	238	28	–	timeout
g13-14	45	239	29	–	timeout

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
g13-15	46	240	30	–	timeout
g14-00	33	256	16	242	6.469
g14-01	34	257	17	243	55.458
g14-02	35	258	18	244	213.095
g14-03	36	259	19	245	812.034
g14-04	37	260	20	–	timeout
g14-05	38	261	21	–	timeout
g14-06	39	262	22	–	timeout
g14-07	40	263	23	–	timeout
g14-08	41	264	24	–	timeout
g14-09	42	265	25	–	timeout
g14-10	43	266	26	–	timeout
g14-11	44	267	27	–	timeout
g14-12	45	268	28	–	timeout
g14-13	46	269	29	–	timeout
g14-14	47	270	30	–	timeout
g14-15	48	271	31	–	timeout
g15-00	35	289	17	274	21.613
g15-01	36	290	18	275	181.282
g15-02	37	291	19	276	705.130
g15-03	38	292	20	–	timeout
g15-04	39	293	21	–	timeout
g15-05	40	294	22	–	timeout
g15-06	41	295	23	–	timeout
g15-07	42	296	24	–	timeout
g15-08	43	297	25	–	timeout
g15-09	44	298	26	–	timeout
g15-10	45	299	27	–	timeout
g15-11	46	300	28	–	timeout
g15-12	47	301	29	–	timeout
g15-13	48	302	30	–	timeout
g15-14	49	303	31	–	timeout
g15-15	50	304	32	–	timeout

Table 42: Results on the testset goemans. Type: Artificial instances arising from generalizations of Steiner tree LP gap examples.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
s1	15	35	8	10	0.003
s2	106	399	50	–	timeout

Table 43: Results on the testset skutella. Type: Artificial instances arising from generalizations of Steiner tree LP gap examples.

Instance	$\|V\|$	$\|E\|$	$\|D\|$	Opt	Time [s]
smc01	2	1	1	0	0.000
smc02	3	3	2	2	0.000
smc03	4	6	3	3	0.000
smc04	5	10	4	4	0.000
smc05	6	15	5	5	0.000
smc06	7	21	6	6	0.000
smc07	8	28	7	7	0.000
smc08	9	36	8	8	0.001
smc09	10	45	9	9	0.002
smc10	11	55	10	10	0.004
smc11	12	66	11	11	0.003
smc12	13	78	12	12	0.016
smc13	14	91	13	13	0.040
smc14	15	105	14	14	0.115
smc15	16	120	15	15	0.300

Table 44: Results on the testset smc. Type: Artificial instances arising from generalizations of Steiner tree LP gap examples.