New Benchmark Instances for the Steiner Problem in Graphs

Isabel Rosseti
Marcus Poggi de Aragão
Celso C. Ribeiro
Eduardo Uchoa
Renato F. Werneck
{*rosseti, poggi, celso, uchoa, rwerneck*}*@inf.puc-rio.br*

PUC-RioInf.MCC07/01 April, 2001

Abstract
In this work, 50 Steiner problems in graphs's instances are proposed. They are based on otimization problems as set covering and code covering. These instances are interesting because, in most of them, a good gap between linear relaxation and primal solution was obtained.

Keywords
Steiner Problems's Instances; Steiner Problems in Graphs; Set Covering Problems; Code Covering Problems.

Resumo
Nesse trabalho, são propostas 50 instâncias para o problema de Steiner em grafos. Elas são baseadas em outros problemas de otimização combinatória, a saber: problema de recobrimento de código e problema de recobrimento de conjuntos. Essas instâncias são interessantes porque, na grande maioria delas, foi obtido um bom *gap* entre a solução primal e a relaxação linear.

Palavras-chave
Instâncias para o Problema de Steiner em Grafos; Problema de Steiner em Grafos; Problema de Recobrimento de Conjuntos; Problema de Recobrimento de Código.

1 Introduction

Let $G = (V, E)$ be a connected undirected graph, where V is the set of nodes and E denotes the set of edges. Given a non-negative weight function $w : E \to R_+$ associated with its edges and a subset $X \subseteq V$ of terminal nodes, the Steiner problem in graphs (SPG) consists in finding a minimum weighted connected subtree of G spanning all terminal nodes in X. The solution of SPG is a Steiner minimum tree. This is one of the most widely studied NP-hard problems, with many applications.

Three sets of benchmark instances are commonly used in the literature to assess the performance of algorithms for the SPG: instances from the OR-Library [2], the "incidence" instances of [3, 4], and the VLSI instances from [8], all of which are available from the SteinLib [19]. However, we believe that these instances are rapidly becoming inadequate for benchmarking, for the following reasons:

- They have already been solved to optimality, with the exception of a few of the larger incidence instances [3, 8, 9, 13, 14, 18]. In fact, optimality can be proven within seconds in most cases. They are not challenging enough to stimulate further development of exact algorithms.
- Metaheuristics can easily find optimal solutions to a large portion of these instances [1, 4, 5, 6, 10, 15, 16]. It is becoming increasingly hard to compare different metaheuristics properly. Deciding among different variants, parameter values, and implementation choices is also difficult, since they often lead to quite similar solutions, which are frequently optimal.

This paper proposes three new series of benchmark instances that will hopefully lead to a better assessment of exact and approximate algorithms for the SPG. Even though some of them are somewhat artificial, we believe they can play an important role in the development of algorithms whose ultimate goal is solving real-world instances. Algorithms that can cope with a variety of artificially hard instances tend to be very robust.

Our first goal was to design instances hard to be solved exactly. Current state-of-the-art exact algorithms are based on linear programming formulations that yield very tight bounds for most current benchmark instances. They can solve several of these instances to optimality without branching. Even when branching is necessary, it is often possible to reduce the problem size significantly using reduction tests and fixation by reduced costs. For this reason, we tried to create instances with large duality gaps. To make them even harder, we introduced various degrees of symmetry, both in the structure of the underlying graphs and in terminal placement. In practice, symmetry increases the number of nodes in the search trees of branch-and-bound algorithms. LP-based lower bounds tend to improve very slowly as branches are performed.

We also aimed at designing instances to challenge metaheuristics. For each instance originally created, we generated another one with the same structure, but with perturbed edge weights. Since this results in instances with a much smaller number of optimal solutions, finding one of them by usual heuristic search methods tends to be harder.

We note that the 50 instances proposed here are reasonably small. The number of nodes in each graph ranges from 64 to 4096, while the number of edges ranges from 128 to 28512. Therefore, although much harder, our instances are no bigger than those currently in use. Three

classes of instances are described in the next section. Section 3 presents some computational results and concluding remarks.

2 Description

2.1 Hypercubes (hc)

Graphs in this series are d-dimensional hypercubes, with $d \in \{6, \ldots, 12\}$. For each value of d, the corresponding graph has 2^d nodes and $d \cdot 2^{d-1}$ edges. These graphs are bipartite, and both partition subsets have the same cardinality. The vertices in one of such subsets become terminals ($|X| = 2^{d-1}$). Edge weights in the originally created instances are unitary. The perturbed instances have integer edge weights randomly chosen from a uniform distribution in the interval $[100, 110]$. These instances seem to be extremely difficult for existing algorithms. For example, using the branch-and-ascent algorithm in [13], we could not solve to optimality the instances with more than 128 nodes. Duality gaps are large and symmetry makes traditional branching schemes much less effective.

The naming convention is hc$d$[u|p], where u stands for "unperturbed" and p for "perturbed". For example, hc8u corresponds to an 8-dimensional hypercube with unperturbed weights.

2.2 Code covering (cc)

Let V_q^n be the set of all n-dimensional vectors whose components are integers in the interval $[0, q-1]$. The *Code Covering Problem* (CCP) is defined as follows: given V_q^n and a positive integer r, find a minimum cardinality subset C of V_q^n such that there exists a vector $x \in C$ with $d(x, y) \leq r$ for all $y \in V_q^n$ (where d denotes the Hamming distance). This NP-hard problem [7] is equivalent to finding a minimum dominating set in a graph $G = (V_q^n, E)$, where $(x, y) \in E$ whenever $d(x, y) \leq r$.

We created instances for the SPG using 13 of such graphs and taking as the set of terminals a solution (i.e., a dominating set) heuristically obtained by the algorithm in [12]. Edges have incidence weights so as to make preprocessing ineffective [3, 4]. In the unperturbed case, an edge has weight 1 if it connects non-terminals, 2 if it is incident to a single terminal, and 3 if it connects two terminals. The perturbed case follows the same principle, but with weights uniformly distributed in the intervals $[100, 110]$, $[200, 210]$, or $[300, 310]$, depending on the number of terminals incident to an edge. Although these instances do challenge current exact algorithms, they are not as hard as the hc instances. The graphs in this case are still very symmetric, but terminal placement is not.

The naming convention is cc$n$-q[u|p] (the value of r is omitted because we have used only $r = 1$). For example, cc3-4p corresponds to a perturbed problem derived from a CCP instance with $q = 3$ and $n = 4$ (and $r = 1$).

2.3 Bipartite graphs (bip)

This series contains ten instances defined on irregular bipartite graphs, with one of the partition subsets acting as the set of terminals. These graphs are derived from random instances of the

Set Covering Problem (SCP) from the OR-Library. Given an instance I with m rows and n columns of SCP, we build an instance I' for the SPG as follows. There is a terminal in I' associated with each row i in I. Similarly, each non-terminal node in I' is associated with a column j in I. Whenever row i covers column j, there is an edge linking terminal i to the non-terminal node j in I'. The resulting graph has $m + n$ vertices, m terminals, and as many edges as nonzero entries in the coefficient matrix of the SCP instance I. Edge weights are unitary in the unperturbed case and uniformly distributed in the interval $[100, 110]$ for the instances with perturbations.

The naming convention is bip$I$[u|p], where I is the name of the SCP instance. For example, bipe2u is the SPG instance with unperturbed weights associated with instance scpe2 of the OR-Library.

Our motivation in creating this series was to have instances with a certain structure (bipartite), but without the artificial symmetry found in the previous two classes. Using SCP instances is no better or worse than generating random bipartite graphs from scratch, but the OR-Library instances do have the advantage of already being publicly available.

3 Experimental results

This section presents some preliminary results on the new instances proposed. We have conducted some experiments to determine if they are indeed hard to be solved.

Tables 1 to 3 present the results obtained for each of the new classes of instances (without preprocessing, whose effectiveness is negligible for these instances). For each instance, we first report its dimensions. Next, we give the weights of the solutions found by five different heuristics: (SPH) shortest-path heuristic [17], (HGP) hybrid GRASP with perturbations [16], (HGP-PR) hybrid GRASP with perturbations and adaptive path-relinking [16], (RTS) reactive tabu search [1], and (RTS-PR) reactive tabu search with path-relinking [1]. These implementations of metaheuristics were selected only because they were readily available to us. The shortest-path heuristic is included in this study only to indicate to the reader how a simple and fast approximate algorithm performs for these instances.

We now comment on the relative effectiveness of these heuristics when applied to the new classes of instances. We stress that our goal here is not to try to establish the superiority of one heuristic over the others. Instead, we want to show that the new instances provide a test bed on which insightful conclusions can be easily drawn, which is not always the case for the instances currently available in the literature.

Results in italics emphasize the best solution value among those obtained by the five heuristics. HGP-PR is clearly the heuristic with the best performance, finding the best solution for all but three instances. Comparing the implementations of metaheuristics which do not make use of path-relinking, HGP is never less effective than RTS for the unperturbed instances, which indicates that it is more suited to instances with many global optima. On the other hand, RTS performs better than HGP for most perturbed instances.

We also give in these tables the linear programming lower bound (LP), the best upper bound we could find (GHLS3), and an estimate of the duality gap given by the percentual difference between these values (gap). The upper bound was obtained with the GHLS3 variant [11] of the hybrid GRASP with perturbations and adaptive path-relinking (HGP-PR). GHLS3 uses a

more powerful local search based on key-nodes and a larger pool of elite solutions. Results in bold face where proved to be optimal solutions by the algorithm described in [13].

Instance	$\|V\|$	$\|E\|$	$\|X\|$	SPH	HGP	HGP-PR	RTS	RTS-PR	LP	GHLS3	gap (%)
hc6u	64	192	32	41	*39*	*39*	*39*	*39*	37.1	**39**	≤2.63
hc7u	128	448	64	80	*77*	*77*	*77*	*77*	73.4	**77**	≤4.05
hc8u	256	1024	128	157	*149*	*149*	151	151	145.1	148	≤1.37
hc9u	512	2304	256	319	*296*	*296*	304	304	286.8	292	≤1.74
hc10u	1024	5120	512	627	*588*	*588*	606	606	567.7	582	≤2.46
hc11u	2048	11264	1024	1219	*1173*	*1173*	1200	1200	≥1125.2	1162	≤3.20
hc12u	4096	24576	2048	2427	*2336*	*2336*	2396	2396	≥2201.1	2303	≤4.59
hc6p	64	192	32	4302	4017	*4003*	*4003*	*4003*	3867.6	**4003**	≤3.49
hc7p	128	448	64	8384	7932	*7905*	7909	7909	7646.8	**7905**	≤3.37
hc8p	256	1024	128	16746	15637	*15376*	15573	15526	15115.7	15322	≤1.36
hc9p	512	2304	256	32509	31108	*30572*	30996	30920	29877.6	30258	≤1.27
hc10p	1024	5120	512	64472	61905	*61030*	61633	61605	59213.4	60494	≤2.16
hc11p	2048	11264	1024	128204	123129	*120804*	122741	122741	117388.7	120096	≤2.25
hc12p	4096	24576	2048	253825	244674	*243390*	244477	244477	≥232709.0	238673	≤2.56

The computation of the LP bound was interrupted for some instances due to limited computational resources.

Table 1: Hypercube instances

References

[1] M.P. Bastos and C.C. Ribeiro, "Reactive tabu search with path-relinking for the Steiner problem in graphs", *Essays and Surveys in Metaheuristics* (C.C. Ribeiro and P. Hansen, eds.), Kluwer, 2001, to appear.

[2] J.E. Beasley, "OR-Library: Distributing test problems by electronic mail", *Journal of the Operational Research Society* 41 (1990), 1069–1072.

[3] C.W. Duin, *Steiner's problem in graphs: Approximation, reduction, variation*, Doctorate Dissertation, Institute of Actuarial Science and Economics, University of Amsterdam, 1994.

[4] C.W. Duin and S. Voss, "Efficient path and vertex exchange in Steiner tree algorithms", *Networks* 29 (1997), 89–105.

[5] C.W. Duin and S. Voss, "The Pilot method: A strategy for heuristic repetition with application to the Steiner problem in graphs", *Networks* 34 (1999), 181–191.

[6] M. Gendreau, J.-F. Larochelle, and B. Sansó, "A tabu search heuristic for the Steiner tree problem", *Networks* 34 (1999), 163-172.

[7] M. Hall and J.H. van Lint, *Combinatorics*, NATO ASI Series C16, D. Reidel, Dordrecht, 1975.

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

[9] A.P. Lucena and J.E. Beasley, "A branch and cut algorithm for the Steiner problem in graphs", *Networks* 31 (1998), 39–59.

Instance	$\|V\|$	$\|E\|$	$\|X\|$	SPH	HGP	HGP-PR	RTS	RTS-PR	LP	GHLS3	gap (%)
cc6-2u	64	192	12	*32*	*32*	*32*	*32*	*32*	29.8	**32**	6.67
cc3-4u	64	288	8	*23*	*23*	*23*	*23*	*23*	21.0	**23**	9.52
cc3-5u	125	750	13	*36*	*36*	*36*	*36*	*36*	≥32.8	**36**	≤9.09
cc5-3u	243	1215	27	76	72	*71*	74	73	≥69.0	71	≤1.43
cc9-2u	512	2304	64	187	*171*	*171*	178	*171*	≥162.7	167	≤2.45
cc6-3u	729	4368	76	217	201	*198*	209	206	≥194.1	197	≤1.03
cc3-10u	1000	13500	50	132	*126*	*126*	130	129	≥123.8	**125**	≤0.81
cc10-2u	1024	5120	135	381	349	*346*	362	359	≥332.4	342	≤2.70
cc3-11u	1331	19965	61	163	*154*	*154*	158	157	≥151.0	153	≤1.32
cc3-12u	1728	28512	74	191	*186*	*186*	187	187	≥182.0	186	≤2.20
cc11-2u	2048	11263	244	687	624	*619*	669	651	≥600.5	614	≤2.16
cc7-3u	2187	15308	222	612	562	*554*	598	588	≥534.1	552	≤3.18
cc12-2u	4096	24574	473	1315	1201	*1184*	1287	1255	≥1141.0	1179	≤3.24
cc6-2p	64	192	12	3388	*3271*	*3271*	*3271*	*3271*	3078.3	**3271**	6.24
cc3-4p	64	288	8	2349	*2338*	*2338*	*2338*	*2338*	2194.0	**2338**	6.56
cc3-5p	125	750	13	3673	3667	*3661*	3664	3664	≥3384.6	**3661**	≤8.15
cc5-3p	243	1215	27	8266	7491	*7404*	7484	7484	≥7117.8	7299	≤2.54
cc9-2p	512	2304	64	18704	17836	*17376*	17946	17904	≥16766.0	17296	≤3.16
cc6-3p	729	4368	76	22680	20850	*20554*	21060	20657	≥20064.1	20458	≤1.96
cc3-10p	1000	13500	50	14149	13084	13061	13118	*13003*	≥12663.6	12860	≤1.55
cc10-2p	1024	5120	135	38608	36552	*35867*	37234	36545	≥34297.2	35466	≤3.41
cc3-11p	1331	19965	61	17111	15924	*15728*	15940	15917	≥15436.8	15609	≤1.11
cc3-12p	1728	28512	74	20626	19285	19167	19201	*19159*	≥18634.9	18838	≤1.09
cc11-2p	2048	11263	244	70666	66073	*64334*	68486	67328	≥61905.0	63841	≤3.13
cc7-3p	2187	15308	222	63339	59005	*57601*	61190	60303	≥55071.9	57339	≤4.12
cc12-2p	4096	24574	473	135953	126541	*122928*	131778	128015	≥117884.5	121772	≤3.30

The computation of the LP bound was interrupted for some instances due to limited computational resources.

Table 2: Code cover instances

[10] S.L. Martins, P. Pardalos, M.G. Resende, and C.C. Ribeiro, "A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy", *Journal of Global Optimization* 17 (2000), 267–283.

[11] M. Poggi de Aragão, C.C. Ribeiro, E. Uchoa, and R.F. Werneck, "Hybrid local search for the Steiner problem in graphs", *Extended Abstracts of the 4th Metaheuristics International Conference*, Porto, 2001.

[12] M. Poggi de Aragão and C.C. de Souza, "Upper bounds for minimum covering codes via tabu search", *Extended Abstracts of the 3rd Metaheuristics International Conference*, pages 359–364, Angra dos Reis, 1999.

[13] M. Poggi de Aragão, E. Uchoa, and R.F. Werneck, "Dual heuristics on the exact solution of large Steiner problems", *Electronic Notes in Discrete Mathematics* 7 (2001), to appear.

[14] T. Polzin and S. Vahdati, "Improved algorithms for the Steiner problem in networks", *Discrete Applied Mathematics*, 2001, to appear.

[15] C.C. Ribeiro and M.C. Souza, "Tabu search for the Steiner problem in graphs", *Networks* 36 (2000), 138–146.

[16] C.C. Ribeiro, E. Uchoa, and R.F. Werneck, "A hybrid GRASP with perturbations for the Steiner problem in graphs", presented at the *International Conference on Adaptive Memory and Evolution: Tabu Search and Scatter Search*, Oxford, 2001.

Instance	$\|V\|$	$\|E\|$	$\|X\|$	SPH	HGP	HGP-PR	RTS	RTS-PR	LP	GHLS3	gap (%)
bipe2u	550	5013	50	60	*55*	*55*	*55*	*55*	52.4	54	≤1.85
bip42u	990	3610	200	270	248	*239*	259	258	232.0	237	≤1.72
bip62u	1199	10000	200	247	229	*227*	233	231	213.3	221	≤3.27
bip52u	1819	7326	200	275	248	*247*	261	261	229.1	235	≤2.17
bipa2u	3140	17795	300	385	358	*356*	371	367	≥329.3	342	≤3.64
bipe2p	550	5013	50	6334	5721	5684	5687	*5666*	5515.4	5660	≤2.61
bip42p	990	3619	200	28758	25737	*25175*	25833	25285	24373.8	24818	≤1.82
bip62p	1199	10000	200	25267	23877	*23291*	23779	23500	22445.2	22944	≤2.22
bip52p	1819	7336	200	29616	26137	*25415*	26079	25884	24186.3	24936	≤3.10
bipa2p	3140	17797	300	40457	37368	*36439*	37415	37345	34685.7	35774	≤3.14

The computation of the LP bound was interrupted for some instances due to limited computational resources.

Table 3: Bipartite instances

[17] H. Takahashi and A. Matsuyama, "An approximate solution for the Steiner problem in graphs", *Math. Japonica* 24 (1980), 573–577.

[18] E. Uchoa, M. Poggi de Aragão, and C.C. Ribeiro, "Preprocessing Steiner problems from VLSI layout", submitted for publication, 1999.

[19] S. Voss, A. Martin, and T. Koch, "SteinLib Testdata Library", online document at http://elib.zib.de/steinlib/steinlib.html, last visited on March 28, 2001.