Dual Heuristics on the Exact Solution of Large Steiner Problems

Marcus Poggi de Aragão [a] Eduardo Uchoa [a]
Renato F. Werneck [a]

[a] *Pontifícia Universidade Católica do Rio de Janeiro, Dept. de Informática*
{poggi,uchoa,rwerneck}@inf.puc-rio.br

Abstract

We present dual heuristics for the directed cut formulation of the Steiner problem in graphs. These heuristics usually give tight lower and upper bounds, and are enough to quickly solve two thirds of the instances from the literature. For harder instances, we propose two exact algorithms using those heuristics: branch-and-ascent, an implicit enumeration without LP solving; and a branch-and-cut that starts from bases provided by dual heuristics, which may be called afterwards to improve convergence. These algorithms have a good practical performance and solved several open instances, including the I320 series and very large and degenerated problems from VLSI layout.

Key words: Steiner problem, dual ascent, branch-and-ascent, branch-and-cut, VLSI layout.

1 Introduction

The *Steiner Problem in Graphs* (SPG) is: given a graph $G = (V, E)$, positive edge costs c and a set $T \subseteq V$ of *terminal vertices*, find a connected subgraph (V', E') of G with $T \subseteq V'$ minimizing $\sum_{e \in E'} c_e$. The SPG is one of the most widely studied NP-hard problems. The directed cut formulation (DCF) of SPG works over the directed graph $G_D = (V, A)$ obtained by replacing each edge in E by two opposite arcs and choosing any terminal r to be the *root*. Let W be the collection of all vertex-sets w containing some terminal but not the root and let $\delta^-(w)$ be the directed cut made up by the arcs entering w. Let x_a indicate whether arc a belongs to the solution (an arborescence rooted at r) or not. The linear relaxation of DCF, (P), and its dual, (D), are:

$$\begin{array}{llll} \text{Min} & \sum(c_a x_a \,:\, a \in A) & \text{Max} & \sum(y_w \,:\, w \in W) \\ \text{s.t.} & x(\delta^-(w)) \geq 1,\ \forall w \in W & \text{s.t.} & \sum(y_w \,:\, a \in \delta^-(w)) \leq c_a,\ \forall a \in A \\ & x_a \geq 0,\ \forall a \in A & & y_w \geq 0,\ \forall w \in W \end{array}$$

2 Dual Heuristics

Strong formulations for the SPG, like DCF, usually give lower bounds very close to the optimum IP value. The bottleneck of many exact algorithms is the expensive solution of these formulations. Wong [7] proposed a dual ascent algorithm for the approximate solution of (D). Let y be a feasible solution of (D). Denote the reduced cost w.r.t. y by $\bar{c}_a = c_a - \sum(y_w \,:\, a \in \delta^-(w))$. Initialize $y = 0$ and at each iteration increase y_w as much as possible, where w corresponds to a maximal set of vertices that can reach a terminal, but not r, by arcs of zero reduced cost. Good primal solutions are obtained within the final set of zero reduced cost arcs. This algorithm already yields quite good lower and upper bounds on most instances from the literature. We propose three additional dual heuristics to improve bounds and reduce instance sizes.

Dual Scaling: Given a feasible solution y of (D), let $y' = \alpha y$, where $\alpha \in [0, 1]$ (we used $.875$). Then apply dual ascent starting from y'. This simple procedure may improve lower bounds and also provide less degenerated solutions, which are convenient to initialize the first LP of the branch-and-cut algorithm.

Dual Adjustment: Let y be a feasible dual solution and $\bar{x}$ be the best known integral solution. Make y' equal to y, but with $y'_w = 0$ for every w such that $\delta^-(w)$ contains more than one arc in $\bar{x}$. Afterwards, apply dual ascent starting from y'. Apart from improving lower bounds, this procedure helps the branch-and-cut algorithm to escape from points of degeneracy.

Active Fixation by Reduced Costs: Let y be a dual solution with cost Z_d, and let Z_{INC} be the cost of the best known integral solution. Given an arc a such that $c_a > Z_{INC} - Z_d$, make y' equal to y, but with $y'_w = 0$ for every w such that $a \in \delta^-(w)$. Then apply a modified dual ascent forbidding the use of a, to get a new dual solution y''. The reduced cost $\bar{c_a}$ w.r.t. y'' is equal to c_a. If the cost of y'' plus c_a exceeds Z_{INC}, then arc a can be fixed to 0. This procedure can lead to significant fixations much earlier than the traditional passive procedure. To allow further reductions, we use the fact that, if an arc is selected, it must be connected to the rest of the Steiner tree (see [1] or [5]).

The following table summarizes the performance of dual heuristics applied to the three most usual sets of benchmark instances (available at [3]): OR-Library (series c, d, and e, after [1] reduction tests), VLSI (series alue, alut, diw, dmxa, diw, gap, msm ,and taq, after [6] tests) and incidence (series I80, I160, I320, and I640, but only the first of each group of five similar instances). Instances solved by preprocessing were discarded. The *gap* columns indicate the average difference between lower and upper bounds; *red.* is the average reduction in the number of arcs (by fixation). The tests were run on an 350MHz AMD machine. Note that two thirds of the instances were solved to optimality. Even some open instances [3] were solved in less than 3 minutes each: I320-223 (opt.: 6695), I320-224 (6694), I320-225 (6691), and I640-121 (4906).

	only dual ascent				**all dual heuristics**			
class	time (s)	gap (%)	red. (%)	total solved	time (s)	gap (%)	red. (%)	total solved
OR-Library	0.62	0.82	83.89	37/57	1.61	0.23	89.51	51/57
VLSI	14.74	1.19	47.93	15/77	24.52	0.58	74.70	49/77
Incidence	38.41	2.77	51.01	14/80	106.35	1.55	65.80	44/80
Overall	19.83	1.68	58.66	66/214	49.01	0.85	75.31	144/214

3 Exact Algorithms over Dual Heuristics

Branch-and-Ascent: When dual heuristics are not able to improve bounds anymore, we branch on a non-terminal vertex v. On the first branch, we oblige v to be part of the solution by making it a terminal. On the second branch we remove v and its adjacent arcs from the graph. Dual heuristics are applied again in the subproblems. We also use a GRASP heuristic [4] to obtain better initial upper bounds. The number of nodes visited by this method is usually larger than those visited by LP-based algorithms, but each node is solved much faster. The algorithm is particularly effective on incidence instances. It found the optima for all formerly open instances of I320 [3]: I320-215 (opt.: 8015), I320-241 (7027), I320-242 (7072), I320-243 (7044), I320-244 (7078), I320-245 (7046), I320-311 (17945), I320-312 (18122), I320-313 (17991), I320-314 (18088), I320-315 (17987), I320-321 (15648), I320-322 (15646), I320-323 (15654), I320-324 (15667), I320-325 (15649), I320-341 (16296), I320-342 (16228), I320-343 (16281), I320-344 (16295), and I320-345 (16289).

Branch-and-Cut: The first LP starts from a basis corresponding to the best dual heuristic solution. The algorithm proceeds by performing the usual rounds of separation, but dual solutions provided by the LP solver can be improved by dual heuristics in order to avoid degeneracy, accelerate convergence and reduce problem size. All open VLSI instances [2,5,3] were solved: alue7065 (opt.: 23881), alue7080 (62449), alut2610 (12239), and alut2625 (35459).

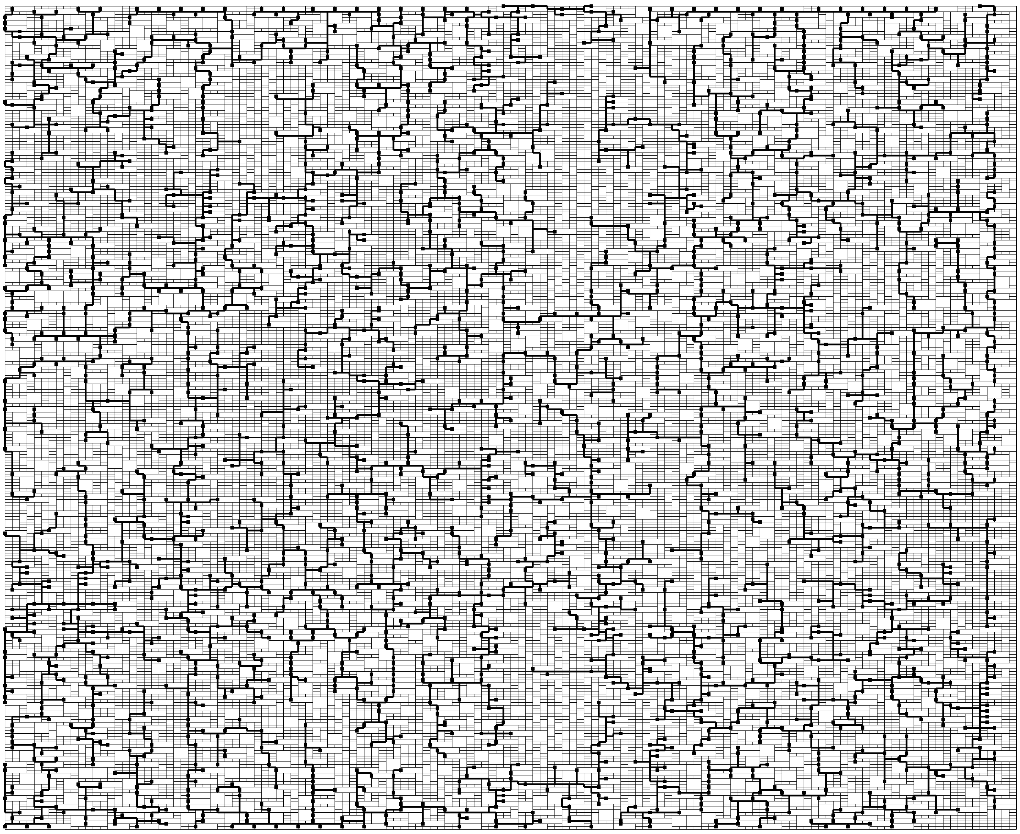

Fig. 1. Optimal solution of alue7080 ($|V| = 34479$, $|E| = 55494$, $|T| = 2344$)

References

[1] C. Duin – Steiner's problem in graphs, PhD thesis, Univ. of Amsterdam (1993).

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

[3] T. Koch, A. Martin, and S. Voß – SteinLib: An updated library on Steiner tree problems in graphs, ZIB-Report 00-37, Berlin (2000), URL: http://elib.zib.de/steinlib.

[4] C. C. Ribeiro, E. Uchoa, and R. F. Werneck – A hybrid GRASP with perturbations and adaptive path-relinking for the Steiner Problem in graphs, Proceedings of the Workshop on Algorithm Engineering as a New Paradigm, Kyoto University (2000).

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

[6] E. Uchoa, M. Poggi de Aragão, and C. C. Ribeiro – Preprocessing Steiner problems from VLSI Layout, Report MCC. 32/99 PUC - Rio de Janeiro (1999).

[7] R. Wong – A dual ascent approach for Steiner tree problems on a directed graph, *Mathematical Programming* **28** (1984) 271-287.