



Master's Thesis

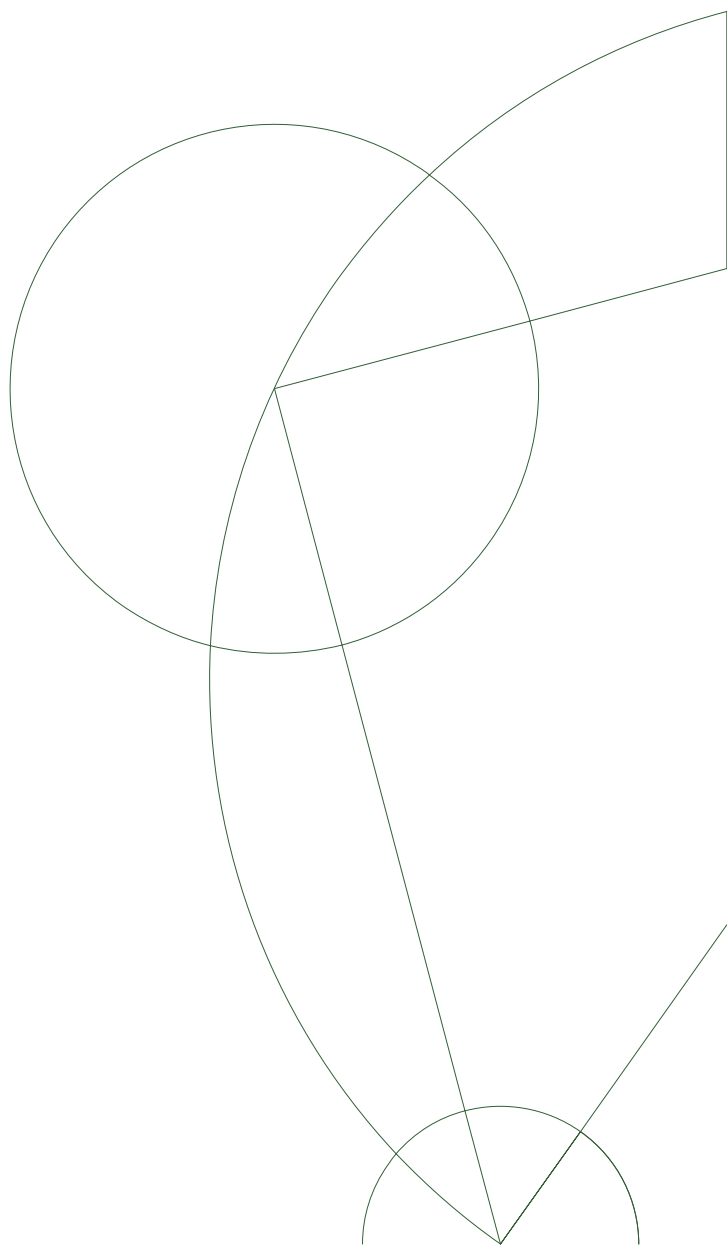
William Sprent – bsprent@gmail.com

Prize-Collecting Steiner Trees

From the Perspective of the Prize-Collecting Travelling Salesman Problem

Supervisors: Pawel Winter

Aug 2018



Abstract

bla bla bla

Contents

1	Introduction	2
2	Background	3
2.1	Combinatorial Optimisation Problems	3
2.2	The Traveling Salesman Problem	3
3	Steiner Trees in Graphs	4
3.1	The Steiner Tree Problem	4
3.1.1	ILP Formulations	5
3.2	Steiner Aborescence Problem	6
3.2.1	Reduction from Other Variants	7
3.2.2	ILP Formulations	7
3.3	Prize-Collecting Steiner Tree Problem	7
3.4	Current Methods For Solving the PCSTP	7
3.4.1	Preprocessing	7
3.4.2	Primal Heuristics	7
3.4.3	Exact Algorithms	7

Chapter 1

Introduction

Chapter 2

Background

2.1 Combinatorial Optimisation Problems

2.2 The Traveling Salesman Problem

bla bla

Chapter 3

Steiner Trees in Graphs

3.1 The Steiner Tree Problem

The Steiner Tree Problem (STP) is defined as follows. Given an undirected graph $G = (V, E, c)$ with vertices V , terminal set $N \subseteq V$, edges E , and edge weights $c : E \rightarrow \mathbb{R}$, a Steiner Tree is a tree, $T \in E$, which spans N with minimal cost. We denote vertices which must be part of a feasible solution, that is $v \in N$, as *terminals*, and vertices which are not required to be part of a solution but are so anyway, that is $v \in T \cap (V \setminus N)$, as *Steiner points* or *Steiner vertices*.

Furthermore, let $S \subset V$ define a cut in G . If we have $S \cap N \neq \emptyset$ and $(V \setminus S) \cap \emptyset$ then we call S a *Steiner cut*.

When $N = V$, the STP is equivalent to the Minimum Spanning Tree Problem, and when $|N| = 2$ the STP is equivalent to finding the shortest path between two vertices. However, while these problems both have polynomial-time solutions, the STP is an NP-Hard problem, the decision variant being part of Edmund Karp's original 21 NP-Complete problems (Karp, 1972).

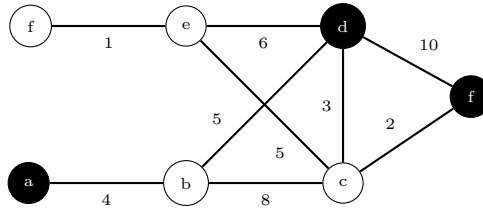


Figure 3.1: Instance of the Steiner Tree Problem. Terminals are coloured black and Steiner Points coloured white.

Figure (3.1) shows an instance of the STP with three terminals and four Steiner points. Since vertices A, C, and D are terminals, they must be spanned by any feasible solution. Figure (3.2a) shows a feasible solution

$$T = \{(a, b), (b, d), (d, g)\}$$

with cost

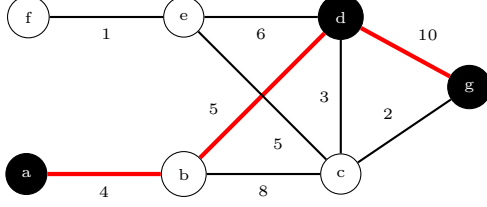
$$c(T) = 4 + 5 + 10 = 19.$$

However, T is not a Steiner tree as there exists at least one feasible solution with lower cost, i.e. the solution

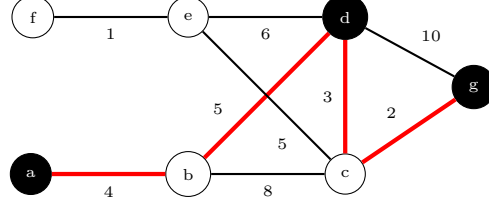
$$T' = \{(a, b), (b, d), (d, c), (c, g)\}$$

in Figure (3.2b) which has cost

$$c(T') = 4 + 5 + 3 + 2 = 14.$$



(a) Feasible but not optimal.



(b) Minimal weight, optimal.

Figure 3.2: Solutions to the STP in (3.1). Red edges are part of the solution.

3.1.1 ILP Formulations

Cut Formulation Let x be a decision vector of length $|E|$ where $x_{ij} = 1$ implies that $(i, j) \in T$ and $x_{ij} = 0$ implies that $(i, j) \notin T$, and let c be a vector of node-weight s.t. $c_{ij} = c((i, j))$. Then define the function $x : E \rightarrow \mathbb{Z}$ as

$$x(E') = \sum_{i,j \in E'} x_{ij}$$

that is, $x(E)$ is equal to the number of selected edges in E . Finally let,

$$\delta(S) = \{(i, j) \mid i \in S \wedge j \in (E \setminus S)\}$$

be all edges which span the cut defined by S . Then we can formulate the STP as in ILP in terms of cuts (see Formulation (3.1)).

$$\underset{x}{\text{minimize}} \quad c^T x \tag{3.1a}$$

$$\text{subject to} \quad x(\delta(S)) \geq 1 \quad \forall S \subset V \tag{3.1b}$$

$$S \cap N \neq \emptyset$$

$$S \cap (V \setminus N) \neq \emptyset$$

$$x \in \mathbb{B}. \tag{3.1c}$$

Formulation 3.1: The *Cut Formulation* of the STP (Koch and Martin, 1998).

Constraints (3.1b) ensures that any feasible solution, T , must span all terminals in G , by requiring that every Steiner cut in G must be crossed by an edge in T . This, combined with objective function ensures that any optimal solution to (3.1) must be a Steiner tree in G and vice versa.

3.2 Steiner Aborescence Problem

The Steiner Aborescence Problem (SAP) is the directed version of the Steiner Tree Problem. Given a *directed* graph, $G = (V, A, c)$, with vertices V , arcs A , a non-empty terminal set $N \subseteq V$, a root terminal $r \in N$, and arc-weights $c : A \rightarrow \mathbb{R}$, then a Steiner Aborescence, $T \subseteq A$, is an aborescence in G , rooted in r , which spans N and has minimal cost.

As with the STP, we denote vertices in N as *terminals*, and any non-terminals spanned by a Steiner Aborescence as *Steiner vertices* or *Steiner Points*.

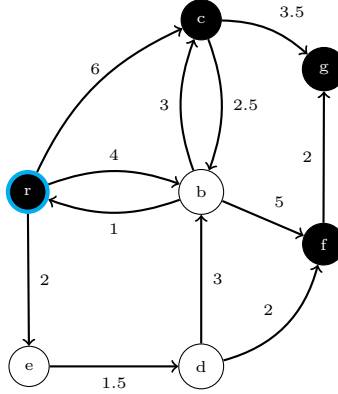


Figure 3.3: Instance of the Steiner Aborescence Problem. Terminals are coloured black, Steiner Points are coloured white, and the root terminal has a blue outline.

Figure (3.3) shows an instance of the SAP with four terminals. Figure (3.4) shows a Steiner Aborescence in (3.3) with cost $c(T) = 13.5$. It is worth noting that since all arcs must point away from the root in an aborescence that while the path c, b, r has lower cost than the arc (r, c) , it cannot be part of a solution. Similarly, solutions to the SAP in the same graph but with root c have lower cost.

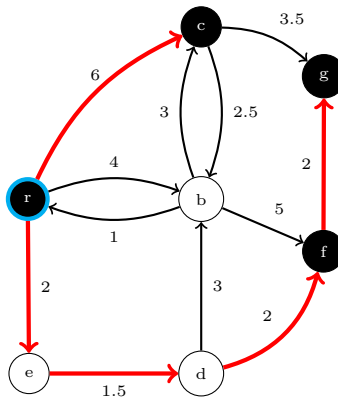


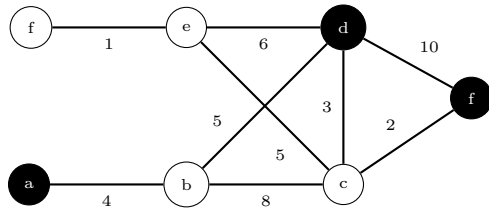
Figure 3.4: Steiner Aborescence in (3.3) with cost 13.5.

The SAP is important due to a tendency of reformulating the undirected STP into the SAP (Koch and Martin, 1998), as well as the PCSTP into the SAP (Gamrath et al., 2017; Klau et al., 2004) and prize-collecting variants of the SAP (Leitner et al., 2016; Ljubic et al., 2005) before stating them as integer programs. This is due to indications that

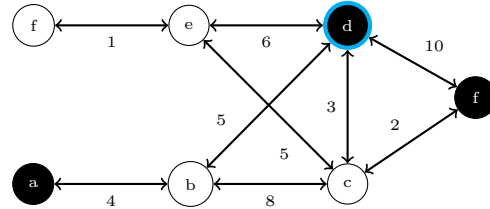
LP relaxations of directed Steiner Tree Problem variants behave better than undirected variants (Chopra and Rao, 1994).

3.2.1 Reduction from Other Variants

Steiner Tree Problem



(a) Problem Instance (3.1)



(b) Instance (3.1) as a SAP instance. Root node is coloured blue.

Figure 3.5: Reduction from STP to SAP.

Prize-Collecting Steiner Tree Problem

3.2.2 ILP Formulations

3.3 Prize-Collecting Steiner Tree Problem

3.4 Current Methods For Solving the PCSTP

3.4.1 Preprocessing

3.4.2 Primal Heuristics

3.4.3 Exact Algorithms

Bibliography

- Sunil Chopra and M. R. Rao. The steiner tree problem i: Formulations, compositions and extension of facets. *Mathematical Programming*, 64(1-3):209–229, 3 1994. ISSN 0025-5610. doi: 10.1007/BF01582573.
- Gerald Gamrath, Thorsten Koch, Stephen J Maher, Daniel Rehfeldt, and Yuji Shinano. Scip-jack—a solver for stp and variants with parallelization extensions. *Mathematical Programming Computation*, 9(2):231–296, 2017.
- Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.
- Gunnar W. Klau, Ivana Ljubić, Andreas Moser, Petra Mutzel, Philipp Neuner, Ulrich Pferschy, Günther Raidl, and René Weiskircher. Combining a memetic algorithm with integer programming to solve the prize-collecting steiner tree problem. In Kalyanmoy Deb, editor, *Genetic and Evolutionary Computation – GECCO 2004*, pages 1304–1315, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-24854-5.
- Thorsten Koch and Alexander Martin. Solving steiner tree problems in graphs to optimality. *Networks*, 32(3):207–232, 1998.
- Markus Leitner, Ivana Ljubic, Martin Luipersbeck, and Markus Sinnl. A dual-ascent-based branch-and-bound framework for the prize-collecting steiner tree and related problems. In *Technical Report*. University of Vienna, 2016.
- Ivana Ljubic, René Weiskircher, Ulrich Pferschy, Gunnar W Klau, Petra Mutzel, and Matteo Fischetti. Solving the prize-collecting steiner tree problem to optimality. In *ALENEX/ANALCO*, pages 68–76, 2005.