3 Steiner Tree and TSP

In this chapter, we will present constant factor algorithms for two fundamental problems, metric Steiner tree and metric TSP. The reasons for considering the metric case of these problems are quite different. For Steiner tree, this is the core of the problem – the rest of the problem reduces to this case. For TSP, without this restriction, the problem admits no approximation factor, assuming $\mathbf{P} \neq \mathbf{NP}$. The algorithms, and their analyses, are similar in spirit, which is the reason for presenting these problems together.

3.1 Metric Steiner tree

The Steiner tree problem was defined by Gauss in a letter he wrote to Schumacher (reproduced on the cover of this book). Today, this problem occupies a central place in the field of approximation algorithms. The problem has a wide range of applications, all the way from finding minimum length interconnection of terminals in VLSI design to constructing phylogeny trees in computational biology. This problem and its generalizations will be studied extensively in this book, see Chapters 22 and 23.

Problem 3.1 (Steiner tree) Given an undirected graph $G = (V, E)$ with nonnegative edge costs and whose vertices are partitioned into two sets, *required* and *Steiner*, find a minimum cost tree in G that contains all the required vertices and any subset of the Steiner vertices.

We will first show that the core of this problem lies in its restriction to instances in which the edge costs satisfy *the triangle inequality*, i.e., G is a complete undirected graph, and for any three vertices u, v, and w, $\text{cost}(u, v) \leq \text{cost}(u, w) + \text{cost}(v, w)$. Let us call this restriction the *metric Steiner tree problem.*

Theorem 3.2 *There is an approximation factor preserving reduction from the Steiner tree problem to the metric Steiner tree problem.*

Proof: We will transform, in polynomial time, an instance I of the Steiner

vertex set V. Define the cost of edge (u, v) in G' to be the cost of a shortest u–v path in G. G' is called the *metric closure* of G. The partition of V into required and Steiner vertices in I' is the same as in I.

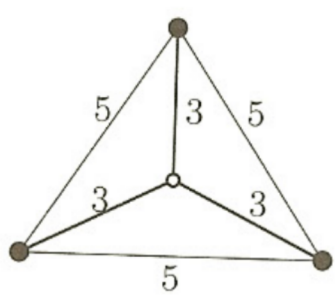
For any edge $(u, v) \in E$, its cost in G' is no more than its cost in G. Therefore, the cost of an optimal solution in I' does not exceed the cost of an optimal solution in I.

Next, given a Steiner tree T' in I', we will show how to obtain, in polynomial time, a Steiner tree T in I of at most the same cost. The cost of an edge (u, v) in G' corresponds to the cost of a path in G. Replace each edge of T' by the corresponding path to obtain a subgraph of G. Clearly, in this subgraph, all the required vertices are connected. However, this subgraph may, in general, contain cycles. If so, remove edges to obtain tree T. This completes the approximation factor preserving reduction. □

As a consequence of Theorem 3.2, any approximation factor established for the metric Steiner tree problem carries over to the entire Steiner tree problem.

3.1.1 MST-based algorithm

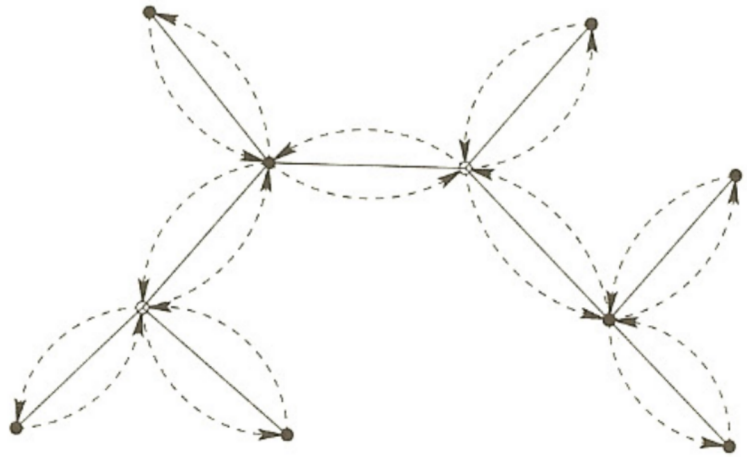
Let R denote the set of required vertices. Clearly, a minimum spanning tree (MST) on R is a feasible solution for this problem. Since the problem of finding an MST is in **P** and the metric Steiner tree problem is **NP**-hard, we cannot expect the MST on R to always give an optimal Steiner tree; below is an example in which the MST is strictly costlier.



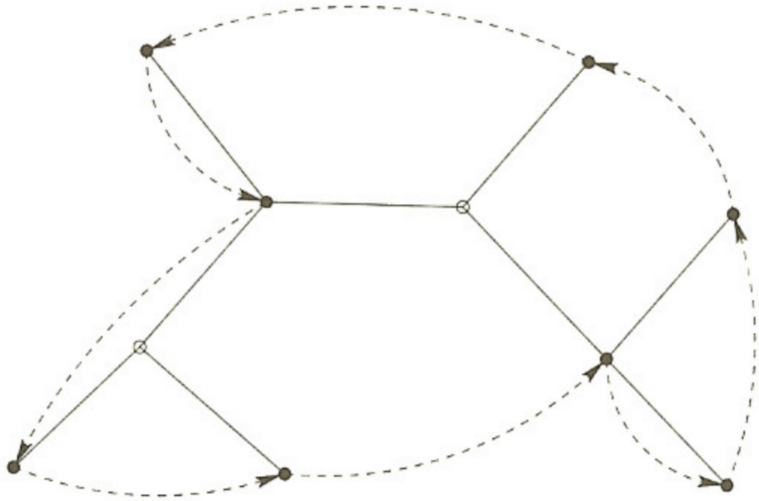
Even so, an MST on R is not much more costly than an optimal Steiner tree:

Theorem 3.3 *The cost of an MST on R is within* $2 \cdot \mathrm{OPT}$.

Proof: Consider a Steiner tree of cost OPT. By doubling its edges we obtain an Eulerian graph connecting all vertices of R and, possibly, some Steiner vertices. Find an Euler tour of this graph, for example by traversing the edges in DFS (depth first search) order:



The cost of this Euler tour is $2 \cdot \mathrm{OPT}$. Next obtain a Hamiltonian cycle on the vertices of R by traversing the Euler tour and "short-cutting" Steiner vertices and previously visited vertices of R:



Because of triangle inequality, the shortcuts do not increase the cost of the tour. If we delete one edge of this Hamiltonian cycle, we obtain a path that spans R and has cost at most $2 \cdot \mathrm{OPT}$. This path is also a spanning tree on R. Hence, the MST on R has cost at most $2 \cdot \mathrm{OPT}$. □

Theorem 3.3 gives a straightforward factor 2 algorithm for the metric Steiner tree problem: simply find an MST on the set of required vertices. As in the case of set cover, the "correct" way of viewing this algorithm is in the setting of LP-duality theory. In Chapters 22 and 23 we will see that LP-duality provides the lower bound on which this algorithm is based and also helps solve generalizations of this problem.

Example 3.4 For a tight example, consider a graph with n required vertices and one Steiner vertex. An edge between the Steiner vertex and a required vertex has cost 1, and an edge between two required vertices has cost 2 (not all edges of cost 2 are shown below). In this graph, any MST on R has cost $2(n-1)$, while $\mathrm{OPT} = n$.