M.O. Ball et al., Eds., *Handbooks in OR & MS, Vol. 7*

Chapter 9

Optimal Trees

Thomas L. Magnanti
Sloan School of Management and Operations Research Center, MIT, Cambridge, MA 02139, U.S.A.

Laurence A. Wolsey
C.O.R.E., Université Catholique de Louvain, 1348 Louvain-la-Neuve, Belgium

1. Introduction

Trees are particular types of graphs that on the surface appear to be quite specialized, so much so that they might not seem to merit in-depth investigation. Perhaps, surprisingly, just the opposite is true. As we will see in this chapter, tree optimization problems arise in many applications, pose significant modeling and algorithmic challenges, are building blocks for constructing many complex models, and provide a concrete setting for illustrating many key ideas from the field of combinatorial optimization.

A tree[1] is a connected graph containing no cycles. A tree (or subtree) of a general undirected graph $G = (V, E)$ with a node (or vertex) set V and edge set E is a connected subgraph $T = (V', E')$ containing no cycles. We say that the tree spans the nodes V'. For convenience, we sometimes refer to a tree by its set of edges with the understanding that the tree also contains the nodes incident to these edges. We say that T is a spanning tree (of G) if T spans all the nodes V of G, that is, $V' = V$. Recall that adding an edge $\{i, j\}$ joining two nodes in a tree T creates a unique cycle with the edges already in the tree. Moreover, a graph with n nodes is a spanning tree if and only if it is connected and contains $n - 1$ edges.

Trees are important for several reasons:

(i) Trees are the minimal graphs that connect any set of nodes, thereby permitting all the nodes to communicate with each other without any redundancies (that is, no extra arcs are needed to ensure connectivity). As a result, if the arcs of a network have positive costs, the minimum cost subgraph connecting all the

[1] Throughout this chapter, we assume familiarity with the basic definitions of graphs including such concepts as paths and cycles, cuts, edges incident to a node, node degrees, and connected graphs. We also assume familiarity with the max-flow min-cut theorem of network flows and with the elements of linear programming. The final few sections require some basic concepts from integer programming.

nodes is a tree that spans all of the nodes, that is, it is a spanning tree of the network.

(ii) Many tree optimization problems are quite easy to solve; for example, efficient types of greedy, or single pass, algorithms are able to find the least cost spanning tree of a network (we define and analyze this problem in Section 2). In this setting, we are given a general network and wish to find an optimal tree within this network. In another class of models, we wish to solve an optimization problem defined on a tree, for example, find an optimal set of facility locations on a tree. In this setting, dynamic programming algorithms typically are efficient methods for finding optimal solutions.

(iii) Tree optimization problems arise in a surprisingly large number of applications in such fields as computer networking, energy distribution, facility location, manufacturing, and telecommunications.

(iv) Trees provide optimal solutions to many network optimization problems.

Indeed, any network flow problem with a concave objective function always has an optimal tree solution (in a sense that we will define later). In particular, because (spanning) tree solutions correspond to basic solutions of linear programs, linear programming network problems always have (spanning) tree solutions.

(v) A tree is a core combinatorial object that embodies key structural properties that other, more general, combinatorial models share. For example, spanning trees are the maximal independent sets of one of the simplest types of matroids, and so the study of trees provides considerable insight about both the structure and solution methods for matroids (for example, the greedy algorithm for solving these problems, or linear programming representations of the problems). Because trees are the simplest type of network design model, the study of trees also provides valuable lessons concerning the analysis of more general network design problems.

(vi) Many optimization models, such as the ubiquitous traveling salesman problem, have embedded tree structure; algorithms for solving these models can often exploit the embedded tree structure.

Coverage

This paper has two broad objectives. First, it describes a number of core results concerning tree optimization problems. These results show that even though trees are rather simple combinatorial objects, their analysis raises a number of fascinating issues that require fairly deep insight to resolve. Second, because the analysis of optimal trees poses many of the same issues that arise in more general settings of combinatorial optimization and integer programming, the study of optimal trees provides an accessible and yet fertile arena for introducing many key ideas from the branch of combinatorial optimization known as polyhedral combinatorics (the study of integer polyhedra).

In addressing these issues, we will consider the following questions:

- Can we devise computationally efficient algorithms for solving tree optimization problems?
- What is the relationship between various (integer programming) formulations of tree optimization problems?