The Constrained Minimum Spanning Tree Problem
(Extended Abstract)

R. Ravi* M. X. Goemans†

Abstract

Given an undirected graph with two different nonnegative costs associated with every edge e (say, $w_e$ for the weight and $l_e$ for the length of edge e) and a budget $L$, consider the problem of finding a spanning tree of total edge length at most $L$ and minimum total weight under this restriction. This constrained minimum spanning tree problem is weakly NP-hard. We present a polynomial-time approximation scheme for this problem. This algorithm always produces a spanning tree of total length at most $(1 + \epsilon)L$ and of total weight at most that of any spanning tree of total length at most $L$, for any fixed $\epsilon > 0$. The algorithm uses Lagrangean relaxation, and exploits adjacency relations for matroids.

Keywords: Approximation algorithm, minimum spanning trees, Lagrangean relaxation, adjacency relations.

1 Introduction

Given an undirected graph $G = (V, E)$ and nonnegative integers $l_e$ and $w_e$ for each edge $e \in E$, we consider the problem of finding a spanning tree that has low total cost with respect to both the cost functions $l$ and $w$. For convenience, we will refer to $l_e$ and $w_e$ of an edge $e$ as its length and weight respectively. Thus the problem we consider is that of finding a spanning tree with small total weight and small total length.

*GSIA, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213. Email: ravi@cmu.edu.
†MIT, Department of Mathematics, Room 2-382, Cambridge, MA 02139. Email: goemans@math.mit.edu. Research supported in part by NSF contract 9302476-CCR, ARPA Contract N00014-95-1-1246, and a Sloan fellowship.
This is a bicriteria problem. A natural way to formulate such problems is to specify a budget on one of the cost functions and minimize the other objective under this constraint. The problem therefore becomes a capacitated problem. In this case, we can specify a budget \( L \) on the total length of the spanning tree and require a tree of minimum weight under this budget restriction. We call this problem the \textit{Constrained Minimum Spanning Tree problem}.

**Lemma 1.1** [1] The constrained minimum spanning tree problem is (weakly) NP-hard.

Define an \((\alpha, \beta)\)-approximation for this problem as a polynomial-time algorithm that always outputs a spanning tree with total length at most \( \alpha L \) and of total weight at most \( \beta W \), where \( W \) is the minimum weight of any spanning tree of \( G \) of length at most \( L \). In other words, \( W \) is the answer to the constrained minimum spanning tree problem formulated in the previous paragraph. Observe that the definition is not completely symmetric in the two cost functions; the quantity \( L \) is given.

In this extended abstract, we first present a \((2, 1)\)-approximation algorithm for the constrained minimum spanning tree problem. The algorithm is based on Lagrangean relaxation, and the proof of the performance guarantee exploits the fact that two adjacent spanning trees on the spanning tree polytope differ by exactly two edges (one in each tree). Moreover, this algorithm can be implemented in almost linear time using an elegant technique of Meggido [6].

We then refine the algorithm to derive an approximation scheme. The precise result is given below.

**Theorem 1.2** For any fixed \( \epsilon > 0 \), there is a \((1 + \epsilon, 1)\)-approximation algorithm for the constrained minimum spanning tree problem that runs in polynomial time.

The same result holds if we replace the set of spanning trees by the bases of any matroid.

Note also that the above approximation can be used to derive a \((1, 1 + \epsilon)\)-approximation algorithm for the constrained minimum spanning tree problem that runs in pseudopolynomial time. This observation follows from more general arguments in [5]; we reproduce it here for completeness. In this latter problem, we must find a tree of length at most the budget \( L \) and of cost at most \((1 + \epsilon)\) times the minimum weight of any tree of length at most \( L \). The idea is to use the weights rather than the lengths