Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 34 (2007) 2147–2167
computers & operations research
www.elsevier.com/locate/cor

New primal–dual algorithms for Steiner tree problems

Vardges Melkonian

Department of Mathematics, Ohio University, Athens, Ohio 45701, USA

Available online 5 October 2005

Abstract

We present new primal–dual algorithms for several network design problems. The problems considered are the generalized Steiner tree problem (GST), the directed Steiner tree problem (DST), and the set cover problem (SC) which is a subcase of DST. All our problems are NP-hard; so we are interested in their approximation algorithms. First, we give an algorithm for DST which is based on the traditional approach of designing primal–dual approximation algorithms. We show that the approximation factor of the algorithm is k, where k is the number of terminals, in the case when the problem is restricted to quasi-bipartite graphs. We also give pathologically bad examples for the algorithm performance. To overcome the problems exposed by the bad examples, we design a new framework for primal–dual algorithms which can be applied to all of our problems. The main feature of the new approach is that, unlike the traditional primal–dual algorithms, it keeps the dual solution in the interior of the dual feasible region. The new approach allows us to avoid including too many arcs in the solution, and thus achieves a smaller-cost solution. Our computational results show that the interior-point version of the primal–dual most of the time performs better than the original primal–dual method.

Keywords: Steiner tree; Integer programming; Approximation algorithm; Primal–dual algorithm

1. Introduction

The primal–dual method is one of the main techniques for designing approximation algorithms for network design problems. In the last 10 years there has been significant progress in designing primal–dual algorithms for *undirected* network design problems [1–5]. *Directed* network design problems are usually much harder to solve, and little progress has been made for these problems. In this paper, we give a primal–dual algorithm for a class of directed network design problems. We also present a new approach

E-mail address: vardges@math.ohiou.edu.

0305-0548/$ - see front matter © 2005 Published by Elsevier Ltd.
doi:10.1016/j.cor.2005.08.009

of designing primal–dual algorithms. This new approach is universal, and can be applied to both directed and undirected problems.

The network design problems considered in this paper can be represented by the following general integer program. Given a graph (directed or undirected) with non-negative costs on the arcs we find a minimum cost subgraph where the number of arcs entering set S is at least one for all subsets $S \in \rho$, where ρ is a collection of node subsets. Formally, given a graph $G=(V, E)$, the network design problem is the following integer program:

$$(IP) \qquad \min \sum_{e \in E} c_e x_e \tag{1}$$

$$\text{s.t.} \sum_{e \in \delta^-(S)} x_e \geqslant 1, \quad \text{for each } S \in \rho, \tag{2}$$

$$x_e \in \{0, 1\}, \quad \text{for each } e \in E, \tag{3}$$

where $\delta^-(S)$ denotes the set of arcs entering S, and $\rho \subseteq 2^V$.

In the next three paragraphs, we introduce our network design problems as special cases of this integer program.

In *directed Steiner tree problem* (DST), we are given a directed graph $G=(V, E)$, a root node $r \in V$, and a set of terminals $T \subseteq V$. All the other nodes are called non-terminal (or Steiner) nodes. The problem is to find the minimum cost directed tree rooted at r that contains all the terminals and any subset of the non-terminal nodes. This problem is the special case of (IP) when $\rho = \{S \subseteq V \mid r \notin S \textit{ and } S \cap T \neq \emptyset\}$.

In the *set cover problem* (SC), we are given a universe $U = \{u_1, \dots, u_n\}$ of n elements, a collection of subsets of U, $\Sigma = \{S_1, \dots, S_k\}$ with a non-negative cost assigned to each of the subsets. The problem is to find a minimum cost subcollection of Σ that covers all elements of U. The following reduction from SC to DST shows that SC is a subcase of DST. For any instance of SC, an instance of DST is created the following way. Build a terminal t_i for each $u_i \in U$, a non-terminal node p_i for each subset $S_i \in \Sigma$. Have an arc $r \to p_i$ from the root to every non-terminal node with cost equal to the cost of the corresponding subset $S_i \in \Sigma$. Have an arc from a non-terminal node p_i to a terminal node t_j with cost zero if node u_j (corresponding to t_j) belongs to subset S_i (corresponding to p_i). It is easy to see that solving this instance of DST solves also the original instance of SC.

In *generalized Steiner tree problem* (GST), we are given an undirected graph $G=(V, E)$ and l pairs of vertices (s_i, t_i), $i = 1, \dots, l$; it is required to find a minimum-cost subset of edges $E' \subseteq E$ such that for all i, s_i and t_i are in the same connected component of (V, E'). The nodes to be connected are called terminals; all the other nodes are called non-terminal (or Steiner) nodes. This problem is the special case of (IP) when $\rho = \{S \subseteq V : |S \cap \{s_j, t_j\}| = 1 \text{ for some j}\}$.

Steiner tree and SC problems occupy an important place in the theory of approximation algorithms. They also have a wide range of applications, from VLSI design to computational biology. For a survey on the applications of Steiner tree problems see [6].

All our problems are NP-hard; so we are interested in approximation algorithms for them. For SC, no polynomial time algorithm can achieve an approximation better than $O(\log n)$ unless P = NP ([7]). Since DST contains SC as a special case, the same approximation hardness result applies to DST ($O(\log |T|)$ in this case). For SC, the lower bound $O(\log n)$ on the approximation factor is achieved by the greedy algorithm (Johnson [8], Lovász [9], Chvátal [10]). A simple randomized algorithm which achieves the same factor is given in Vazirani [11]. A primal–dual algorithm for SC was given by Bar-Yehuda and