Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computers & Operations Research 32 (2005) 239–255
computers & operations research
www.elsevier.com/locate/dsw

Heuristic and exact algorithms for the spanning tree detection problem

Takeo Yamada*, Seiji Kataoka, Kohtaro Watanabe

Department of Computer Science, The National Defense Academy, Yokosuka, Kanagawa 239-8686, Japan

Abstract

Given an integer α and an undirected graph with edges associated with integer weights, the spanning tree detection problem (STDP) is to find, if one exists, a spanning tree of weight α. STDP is $\mathcal{NP}$-hard. In this paper we develop both approximate and exact algorithms to solve STDP. Approximate algorithm consists of a greedy method to construct an initial spanning tree quickly, and a local search method that improves the weight of the tree successively toward α. To solve STDP completely, we take a 'divide and conquer' approach and develop an exact algorithm. These algorithms are implemented in C language and we conduct some numerical tests to evaluate the performance of the developed algorithms for various types and sizes of instances. In most cases, we are able to solve STDPs with up to 1000 nodes in less than a few seconds. Moreover, to solve harder instances we try tabu search as well, and mention how the developed algorithm can be modified to list up all the spanning trees of weight α.

1. Introduction

Let $G=(V,E)$ be an undirected graph [1] with vertex set $V=\{v_1,\dots,v_n\}$ and edge set $E=\{e_1,\dots,e_m\}\subseteq V\times V$. Each edge $e\in E$ is associated with an integer *weight* $w(e)$. We assume G is *connected* and *simple*, i.e., there exist neither self-loops nor multiple edges. A *tree* is a connected acyclic subgraph of G. For a tree T, its weight $w(T)$ is defined as the sum of the weights of constituent edges. A tree is a *spanning tree* if it covers all the nodes of G. The minimum spanning tree problem (MSTP, [2]) has been well studied in this framework, and efficient algorithms are widely known to solve this problem [3,4]. With minor changes, the same algorithms can be used to find the *maximum* spanning tree as well.

* Corresponding author.
E-mail address: yamada@nda.ac.jp (T. Yamada).

doi:10.1016/S0305-0548(03)00215-6

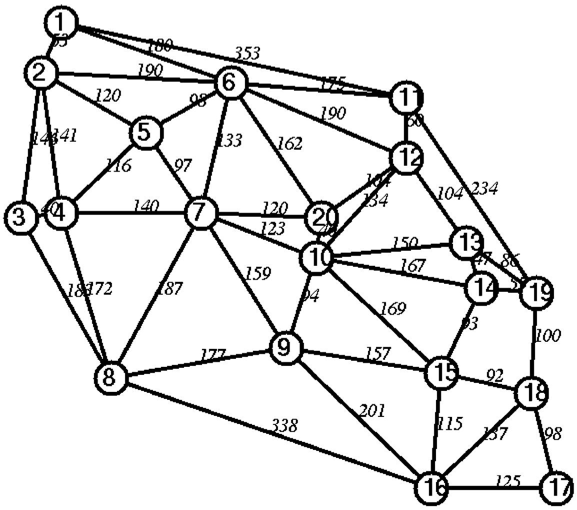

Fig. 1. A planar graph.

In this paper, we are concerned with the following variation of the problem, which is termed the spanning tree detection problem (STDP).

STDP: Given an arbitrary integer α and an undirected graph G, find, if one exists, a spanning tree T of G such that $w(T)=\alpha$.

Here, the integer α is referred to as the 'target value', and we call a spanning tree with this weight an α-*spanning tree*, or α-ST for short. Closely related to this problem is the following *exact spanning tree problem* [5,6].

ESTP: Determine if there exists an α-ST in G.

STDP is $\mathcal{NP}$-hard [7], since ESTP is solved whenever STDP is solved, and ESTP is known to be $\mathcal{NP}$-complete [5]. Although ESTP can be solved by a pseudo-polynomial time algorithm due to Barahona et al. [8], this does not help solving STDP, since the algorithm in [8] determines the existence problem, but does not actually yield an α-ST as an 'evidence'.

Although no published algorithms are known to us to solve STDP, some naive approaches are possible. First, we may list up all the spanning trees included in G and see if there exists an α-ST among them. For an algorithm to list up all the spanning trees readers are referred to, e.g., Read and Tarjan [9]. Alternatively, we may list up the K smallest (or largest) spanning trees for sufficiently large K using the algorithm found in, e.g., [10]. However, except for very small instances, usually the number of all the spanning trees in a graph is too large to allow such a primitive approach.

Example 1. By Kirchhoff's method [11] we know that there exists more than 2×10^9 spanning trees in the graph of Fig. 1. Thus, complete enumeration is almost infeasible in practice.

STDP may be encountered in the following applications.