Operations Research Perspectives 3 (2016) 5–13

Contents lists available at ScienceDirect

Operations Research Perspectives

journal homepage: www.elsevier.com/locate/orp

Algorithms for the minimum spanning tree problem with resource allocation

Seiji Kataoka[*], Takeo Yamada

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

ARTICLE INFO

Article history:
Available online 13 January 2016

Keywords:
Minimum spanning tree problem
Resource allocation
Trade-off analysis
Branch-and-bound method

ABSTRACT

We formulate the *minimum spanning tree problem with resource allocation* (MSTRA) in two ways, as discrete and continuous optimization problems (d-MSTRA/c-MSTRA), prove these to be $\mathcal{NP}$-hard, and present algorithms to solve these problems to optimality. We reformulate d-MSTRA as the *knapsack constrained minimum spanning tree problem*, and solve this problem using a previously published branch-and-bound algorithm. By applying a 'peg test', the size of d-MSTRA is (significantly) reduced. To solve c-MSTRA, we introduce the concept of *f-fractionalsolution*, and prove that an optimal solution can be found within this class of solutions. Based on this fact, as well as conditions for 'pruning' subproblems, we develop an enumerative algorithm to solve c-MSTRA to optimality. We implement these algorithms in ANSI C programming language and, through extensive numerical tests, evaluate the performance of the developed codes on various types of instances.

1. Introduction

Numerous applications have been published on the *minimum spanning tree problem* (MST) [1,2] on an undirected graph [3], where each edge is associated with a non-negative *distance*. Here, 'distance' may be cost, time, toll or penalty of each edge in specific applications. In this article we are concerned with a variation of this problem, where each edge may have different 'modes' associated with different pair of distance and 'cost'. Or, we may have edges which can be 'strengthened' by increasing the amount of resources added to these edges. For example, in a city transportation network, we may take a bus or a train to go from one train station to the other, each with respective fare and traveling time. Or, while driving a car we may put more fuel to run faster. We formulate such a combination of the MST and the *resource allocation problem* (RA) [4] as discrete/continuous combinatorial optimization problems, prove these to be NP-hard, develop both of approximate and exact algorithms to solve these problems, and conduct a series of numerical experiments to evaluate the performance of the developed algorithms.

To describe the problem, let $G = (V, E)$ be a connected undirected graph, where V is a finite set of vertices and $E \subseteq V \times V$ is the set of edges. By n and m we denote the numbers of nodes and edges, i.e., $n = |V|$ and $m = |E|$. Associated with each edge $e \in E$ is a cost function $c_e(\cdot)$, which relates r_e, the amount of resource allocated to e, to the cost of this edge. Given a fixed amount R of total resources, our problem is to find a spanning tree of G and a resource allocation on that tree, such that the total cost incurred is minimized over all possible solutions. We formulate this as a discrete, as well as a continuous, optimization problem.

In discrete optimization framework, edges can be either one of 'normal' or 'priority' modes, and the resource requirement and cost of each edge take different values depending on the mode of that edge. If edge e is in normal mode, the amount of resource required is r_e^0, with the corresponding cost c_e^0, while in priority mode these are r_e^1 and c_e^1, respectively. These are related through a binary function $c(\cdot)$ as

$$c_e^0 = c_e(r_e^0), \qquad c_e^1 = c_e(r_e^1) \tag{1}$$

and, we assume

$$c_e^0 \geq c_e^1, \qquad r_e^0 \leq r_e^1, \quad \forall e \in E. \tag{2}$$

That is, in priority mode cost is smaller than in normal mode with an expense of increased resource allocation at that edge.

Let $\mathcal{T}$ denote the set of all the spanning trees in G, and by $T \in \mathcal{T}$ we mean a spanning tree as well as the set of edges constituting T. Resource allocation over T is represented by a binary vector $r = (r_e)_{e \in T}$ with $r_e \in \{r_e^0, r_e^1\}$, $\forall e \in T$. Then the problem is formulated as the following 'discrete' *minimum spanning tree problem with resource allocation*

[*] Corresponding author.
E-mail addresses: seiji@nda.ac.jp (S. Kataoka), yamada.144b@gmail.com (T. Yamada).

http://dx.doi.org/10.1016/j.orp.2015.12.001

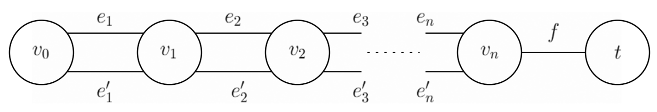

Fig. 1. Graph $G = (V, E)$ for the proof of $\mathcal{NP}$-hardness of c-MSTRA.

d-MSTRA:

$$\text{minimize} \quad z(T, r) := \sum_{e \in T} c_e(r_e) \tag{3}$$

$$\text{subject to} \quad \sum_{e \in T} r_e \le R, \tag{4}$$

$$r_e \in \{r_e^0, r_e^1\}, \quad \forall e \in T, \tag{5}$$

$$T \in \mathcal{T}. \tag{6}$$

Alternatively, the problem may be formulated as a continuous optimization problem. Here, the cost c_e of edge $e \in E$ is a non-increasing function of r_e defined on the continuous interval $[r_e^0, r_e^1]$ as $c_e = c_e(r_e)$. Specifically, we assume this to be linear

$$c_e(r_e) := s_e - \theta_e r_e, \tag{7}$$

and put $c_e^0 := c_e(r_e^0)$ and $c_e^1 := c_e(r_e^1)$ for simplicity. From (2), we have $\theta_e \ge 0$. Thus, the 'continuous' *minimum spanning tree problem with resource allocation* is as follows.

c-MSTRA:

$$\text{minimize} \quad z(T, r) = \sum_{e \in T} c_e(r_e) \tag{8}$$

$$\text{subject to} \quad \sum_{e \in T} r_e \le R, \tag{9}$$

$$r_e^0 \le r_e \le r_e^1, \quad \forall e \in T, \tag{10}$$

$$T \in \mathcal{T}. \tag{11}$$

To prove $\mathcal{NP}$*-hardness* of these problems, we first note that the standard 0–1 knapsack problem

KP:

$$\text{maximize} \quad \sum_{j=1}^{n} p_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_j \le R, \quad \forall x_j \in \{0, 1\},$$

is $\mathcal{NP}$-hard.

Theorem 1. *d-MSTRA and c-MSTRA are both* $\mathcal{NP}$*-hard.*

Proof ($\mathcal{NP}$*-Hardness of d-MSTRA*)**.** Corresponding to the knapsack problem let $G = (V, E)$ be with $V = \{v_0, v_1, \ldots, v_n\}$ and $E = \{(v_0, v_1), \ldots, (v_{n-1}, v_n)\}$. For each edge $e = (v_{j-1}, v_j) \in E$ we set $r_e^0 = 0$, $c_e^0 = p_j$, and $r_e^1 = w_j$, $c_e^1 = 0$. Then, d-MSTRA is identical to KP, and thus $\mathcal{NP}$-hard.

($\mathcal{NP}$-hardness of c-MSTRA) Given the knapsack problem above, let $G = (V, E)$ be a graph with $V = \{v_0, v_1, \ldots, v_n, t\}$ and $E = \{e_1, e'_1, e_2, e'_2, \ldots, e_n, e'_n, f\}$. Here, e_j and e'_j both connect v_{j-1} and v_j, and $f = (v_n, t)$ (see Fig. 1). The resource and cost are $r^0(e_j) = r^1(e_j) = w_j$ and $c^0(e_j) = c^1(e_j) = M - p_j$ for e_j, $r^0(e'_j) = r^1(e'_j) = 0$ and $c^0(e'_j) = c^1(e'_j) = M$ for e'_j, and $r_f^0 = 0$, $r_f^1 = R$, $c_f^0 = c_f^1 = 0$ for f, where M is a constant satisfying $M > \max_{1 \le j \le n} p_j$. For a solution (T, r) of c-MSTRA on this graph, we introduce a 0–1 variable x_j such that $x_j = 1$ if $e_j \in T$ and $x_j = 0$ otherwise ($j = 1, 2, \ldots, n$). Note that $x_j = 0$ implies $e'_j \in T$. Then, this particular c-MSTRA can be rewritten as

$$\text{minimize} \quad \sum_{j=1}^{n} (M - p_j) x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_j \le R, \quad \forall x_j \in \{0, 1\}.$$

This is equivalent to KP, and thus c-MSTRA is $\mathcal{NP}$-hard. $\square$

The problems formulated above may be regarded as a sort of trade-off analysis of spanning trees with respect to two criteria, the amount of resource consumed $r(T) = \sum_{e \in T} r_e$ and the cost $c(T) = \sum_{e \in T} c_e$ of the tree, provided that r_e and c_e are *a priori* given constants. Hassin and Levin [5] gave a polynomial time approximation scheme, and Yamada et al. [6] gave a branch-and-bound algorithm for such a problem. Trade-off analysis in general is quite standard in scheduling [7–9] and resource allocation [4] problems. If $r(T)$ is regarded as a second objective function, rather than a constraint, we have a multi-objective minimum spanning tree problem [10,11]. An important feature that distinguishes d- and c-MSTRAs from the previous researches is the fact that the coefficient c_e is a function of r_e, thus it can be enhanced by allocating larger amount of resources. To our knowledge, trade-off analysis in this framework is new in this paper.

In Section 2, we discuss d-MSTRA: how this can be reduced to the *knapsack constrained minimum spanning tree problem* (KCMST [6], see also [12]), and how computation can be speeded up by reducing the size of the problem. Sections 3 and 4 explore c-MSTRA and develop solution algorithm to solve this problem to optimality. Finally, in Section 5 a series of numerical experiments are done, both for d- and c-MSTRAs, to examine the behavior of the developed algorithms. Throughout theoretical development in these sections, KCMST plays a key role. Thus, KCMST and its solution algorithm are briefly reviewed in the Appendix for readers' convenience.

2. Solution algorithm for d-MSTRA with problem reduction

In this section we show that d-MSTRA can be reformulated as a KCMST on a 'doubly edged graph'. Furthermore, by applying the 'peg test' the problem is substantially reduced in size. The reduced problem can be solved by SOLVE_KCMST routine [6] much faster than solving the unreduced problem directly.

2.1. d-MSTRA as KCMST

We introduce $\bar{G} = (\bar{V}, \bar{E})$ as the graph which is obtained from G by *doubling* each edge $e \in E$ into edges e^0 and $e^1 \in \bar{E}$, corresponding to normal and priority modes, respectively. Thus, these edges are incident to the identical pair of nodes as $e \in E$, and the resource allocation and cost at e^0 (e^1, resp.) are r_e^0 and c_e^0 (r_e^1 and c_e^1, resp.). We have $\bar{V} = V$ and $|\bar{E}| = 2$ m. These graphs are illustrated in Fig. 2 for a planar example. Here we employ the following simplified notation for edges and trees in graph $\bar{G}$. Superscripts in the edges of $\bar{E}$ are usually omitted, unless otherwise needed. Therefore, by $e \in E$ we mean either e^0 or e^1, and c_e may refer to either c_e^0 or c_e^1. $\mathcal{T}$ denotes the set of all the spanning trees in $\bar{G}$. Thus, d-MSTRA can be rewritten as the following

KCMST:

$$\text{minimize} \quad z(T) := \sum_{e \in T} c_e \tag{12}$$

$$\text{subject to} \quad \sum_{e \in T} r_e \le R, \tag{13}$$

$$T \in \mathcal{T}. \tag{14}$$