Discrete Optimization 16 (2015) 51–61

Contents lists available at ScienceDirect

Discrete Optimization

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

Min–max cover of a graph with a small number of parts

Boaz Farbstein, Asaf Levin *

Faculty of Industrial Engineering and Management, The Technion, 32000 Haifa, Israel

ARTICLE INFO

Article history:
Received 4 April 2013
Received in revised form 27 October 2014
Accepted 6 January 2015
Available online 3 February 2015

Keywords:
Approximation algorithms
Routing
Vehicle routing problems

ABSTRACT

We consider a variety of vehicle routing problems. The input consists of an undirected graph and edge lengths. Customers located at the nodes have to be visited by a set of vehicles. Two important parameters are k the number of vehicles, and the longest distance traveled by a vehicle denoted by λ. Here, we consider k to be a given bound on the maximum number of vehicles, and thus the decision maker cannot increase its value. Therefore, the goal will be to minimize λ. We study different variations of this problem, where for instance instead of servicing the customers using paths, we can serve them using spanning trees or cycles. For all these variations, we present new approximation algorithms with FPT time (where k is the parameter) which improve the known approximation guarantees for these problems.

1. Introduction

We consider a variety of vehicle routing problems. The input consists of an undirected graph $G = (N, E)$ and non-negative edge lengths $l(e)$ for $e \in E$. Customers located at the nodes of the graph have to be visited by a set of vehicles. Two important parameters are k the number of vehicles, and the longest distance traveled by a vehicle denoted by λ. Here, we consider k to be a given bound on the maximum number of vehicles, and thus the decision maker cannot increase its value, i.e., this is a hard constraint. Therefore, the goal will be to minimize λ. We study different variations of this problem. That is, we study various bottleneck optimization problems. In these problems we would like to cover the input graph whose edges are associated with lengths using at most k (connected) subgraphs, such that the maximum length of a subgraph in the solution is minimized. Our optimization problems differ in the constraints imposed on the topology of the subgraphs (e.g., being trees or paths) and in the definition of covering the input graph.

More formally, given an undirected graph $G = (N, E)$ and a non-negative edge length function $l : E \rightarrow \mathbb{R}_+$, we denote the total length of a subset of edges $E' \subseteq E$ by $l(E')$, that is, $l(E') = \sum_{e \in E'} l(e)$. If $G' = (N', E')$ is a subgraph of G, then we define $l(G') = l(E')$. In what follows we always assume that l is a non-negative symmetric length function. When we add the assumption that l satisfies the triangle inequality we say that l is a metric, otherwise we say that l is a length function. Given a collection $F_i = (N_i, E_i)$ for $i = 1, \ldots, t$ of t subgraphs of G, we denote by $\bigcup_{i=1}^t F_i = (\bigcup_{i=1}^t N_i, \bigcup_{i=1}^t E_i)$ (and $F_1 \bigcup F_2 = \bigcup_{i=1}^2 F_i$). We write $N' \subseteq \bigcup_{i=1}^t F_i$ if $N' \subseteq \bigcup_{i=1}^t N_i$, and similarly $E' \subseteq \bigcup_{i=1}^t F_i$ if $E' \subseteq \bigcup_{i=1}^t E_i$.

Min–max covering problems are well studied in the field of operations research. A 3-approximation algorithm for the min–max tree cover problem (the version where l is a length function, G_i needs to be a tree for all i, and covering G is covering the nodes of G) was presented by [1] improving the approximation ratio of 4 presented by both [2,3]. This problem is known to be APX-hard with a lower bound of $\frac{3}{2}$ on the possible approximation ratio of any polynomial time algorithm [4]. A problem similar to the Min–max tree cover is the Min–max R-rooted tree cover where the input also includes a set of roots

* Corresponding author.
E-mail addresses: boazf@tx.technion.ac.il (B. Farbstein), levinas@ie.technion.ac.il (A. Levin).

http://dx.doi.org/10.1016/j.disopt.2015.01.001

and each output tree must contain a distinct root from this set. Even et al. [3] presented a 4-approximation algorithm for this problem. Another variant of min–max cover by trees is the min–max rooted subtree cover problem where the input for the Min–max tree cover problem is augmented by a root $r \in N$ that must appear in every output tree. Nagamochi [5] presented a $(3 - \frac{2}{k+1})$-approximation algorithm for this problem followed by a $(2 + \varepsilon)$-approximation algorithm of Nagamochi and Okada [6]. Variants with other topologies of the subgraphs that are used to cover the nodes of the graph (besides trees) are also studied in [2,7–9]. The above problems have also a Min-Sum variant where the objective function is to minimize the sum of the lengths of the subgraphs (or the average length of a subgraph) instead of minimizing the maximum length of a subgraph. There is usually a trade off between the two variants, where minimizing the sum of the lengths of the subgraphs may result in a very large individual subgraph and minimizing the maximum subgraph may result in a larger sum of the lengths of the subgraphs. Minimizing the maximum length of a subgraph is usually harder than minimizing the total length of all subgraphs. For example, the min–max tree cover that we study is an NP-hard problem (even for $k = 2$) whereas finding k trees such that their total length is minimized can be done in polynomial time (using the greedy algorithm for finding a minimum cost basis of a matroid). A comparison between min–max vehicle routing models and min-sum vehicle routing models was done by Bertazzi et al. [10]. This comparison also motivates by practical applications the study of the min–max versions. We refer to [10] for details.

In this paper, we consider k to be a parameter. The time complexity of the approximation algorithms may be exponential in k and our main results will have time complexity of the form $f(\frac{k}{\varepsilon}) \cdot poly(n)$ where f is an exponential function and $\varepsilon > 0$ being a parameter that defines the quality of approximation. This type of time complexity is motivated by the FPT (fixed parameter tractable) literature (see [11–13]). That is, our approximation algorithms can be seen as parameterized approximation algorithms running in FPT-time. In terms of the directions for classifying such results which combine approximations and FPT as suggested by Marx [13], our algorithms have a parameter which is a combination of an instance parameter (k) and the parameter which determines the quality of approximation (ε). The study of algorithms with FPT time where k is the parameter is justified both because in the practical use of the vehicle routing problem the number of vehicles is significantly smaller than the number of nodes in the graph and since the problems are NP-hard even for small constant value of k. When we consider previous works on the problems, k is seen as part of the input and thus these works do not allow FPT time complexity. However, in this paper we allow FPT time complexity and this enables our improvements.

An $\mathcal{R}$-approximation algorithm for a minimization problem is a polynomial time algorithm which always finds a feasible solution of cost at most $\mathcal{R}$ times the cost of an optimal solution. The infimum value of $\mathcal{R}$ for which an algorithm is an $\mathcal{R}$-approximation is called the approximation ratio or the performance guarantee of the algorithm. A polynomial time approximation scheme (PTAS) is a family of approximation algorithms such that the family has a $(1 + \varepsilon)$-approximation algorithm for any $\varepsilon > 0$. A fully polynomial time approximation scheme (FPTAS) is a PTAS whose time complexity is a polynomial of the length of the (binary) encoding of the input and of $\frac{1}{\varepsilon}$.

Paper outline. In Section 2 we formally define the variants of the problem which we consider in this paper. Afterward, in Section 3 we develop the basic algorithm which is used throughout the paper. In a nutshell, we show that we are able to find a set of trees which cover the nodes of a graph, such that in an optimal solution there is no edge connecting nodes from different trees, and the total length of the trees is at most $k \cdot (1+\varepsilon) \cdot \lambda$. Afterward, in the remaining of the paper we construct an approximate solution for each problem, based on this collection of trees. We conclude the paper by showing that these problems that we study are NP-hard for any fixed value of k (sometimes for all $k \neq 1$).

2. Problems definition and statement of the results

The problems considered in this paper are defined as follows.

Min–Max tree cover
Input: A graph $G = (N, E)$, a length function $l : E \to \mathbb{R}_+$, and an integer $k > 0$.
Output: Trees $T_1, \ldots, T_k$, such that $N \subseteq \bigcup T_i$.
Objective function: $\min \max_i l(T_i)$.
Our approximation ratio: $2 + \varepsilon - \frac{2}{k+1}$.
Previous best known approximation ratio [1]: 3.

Min–Max R-rooted tree cover
Input: A graph $G = (N, E)$, a length function $l : E \to \mathbb{R}_+$, and a set of roots $R \subseteq N$ such that $|R| = k$.
Output: Trees $T_1, \ldots, T_k$, such that $N \subseteq \bigcup T_i$ and each T_i has a distinct root $r_i \in R$. That is, there exists a one-to-one function $h : \{1, 2, \ldots, k\} \to R$ such that the tree T_i contains $h(i)$ which is called the root of T_i.
Note that as before, different trees may share edges and nodes. In particular, the root of tree T_i may appear as a node in tree T_j but for $i \neq j$ the roots of T_i and T_j must be distinct.
Objective function: $\min \max_i l(T_i)$.
Our approximation ratio: $3 + \varepsilon - \frac{2}{k+1}$.
Previous best known approximation ratio [3]: 4.

Min–Max path cover
Input: A complete graph $G = (N, E)$, a metric $l : E \to \mathbb{R}_+$, and an integer $k > 0$.