On the Multi-Service Center Problem

Takehiro Ito[1,a)]   Naonori Kakimura[2,b)]   Yusuke Kobayashi[3,c)]

Abstract: The multi-service center problem is a variant of facility location problems. In the problem, we consider locating p facilities on a graph, each of which provides distinct service required by all vertices. Each vertex incurs the cost determined by the sum of the weighted distances to the p facilities. The aim of the problem is to minimize the maximum cost among all vertices. This problem is known to be NP-hard for general graphs, while it is solvable in polynomial time when p is a fixed constant. In this paper, we give sharp analyses for the complexity of the problem from the viewpoint of graph classes and weights on vertices.

Keywords: facility location problem, graph algorithm

1. Introduction

Facility location is one of the most well-studied topics in combinatorial optimization. There are various kinds of settings depending on the situations. (See e.g., [4].) Generally, in facility location problems, we are given a set of clients and a set of facilities in a graph, and we aim to decide which facilities are open to satisfy the demand of the clients. For example, the well-known *k-center problem* is to place k facilities in a graph so that the maximum distance from each client to their closest facility is minimized [6], [7]. Note that this standard situation assumes that all k facilities can provide the same service so that each client meets their demand by only accessing one facility.

Yu and Li [10] recently proposed a new framework of facility location problems, called *multi-service location problems*, motivated by the situation where each facility provides different services and each client needs to access all facilities to meet their demand. As the first problem of this kind, they proposed the *p-service center* problem defined as follows. (The formal definition will be given in Section 2.) In the problem, we assume that clients are all vertices in a graph G, and facilities can be located on any place in G even on an edge. When we locate p facilities, each of which provides distinct service, the cost of each client v is determined by the sum of the weighted distances to the p facilities, where the weighted distance from v to a facility x is the shortest-path distance from v to x multiplied by a positive weight (representing the *demand*) of v to the service provided by x. The aim of the problem is to find a location of p facilities that minimizes the maximum cost among the clients.

Yu and Li [10] studied the computational complexity of *p-service center* for several cases. They designed a polynomial-

time algorithm for general graphs when p is a fixed constant, and an $O(n \log n)$-time algorithm for trees having n vertices when restricted to $p = 2$. On the negative side, they showed that the problem is NP-hard for general graphs when p is a part of input. Anzai et al. [1] showed that this case remains NP-hard even for split graphs with identical edge-length.

In this paper, we consider a simple generalization of *p-service center*, that is, each client can have zero or negative weights (demands) to a facility; recall that the weight must be positive in the original setting. This generalization is very simple, but enables us to express several natural situations: a zero demand means that the client does not need the service provided by the facility, while a negative demand means that the client refuses the service provided by the facility; furthermore, any vertex can be a non-client by setting all demands to be zero. In this paper, we sharply analyze the computational complexity of this generalized problem from the viewpoint of graph classes and weights of vertices. Our main contributions are summarized as follows:

(1) The problem with nonnegative weights is solvable in polynomial time for trees, even when the number p of facilities is a part of input.

(2) The problem with nonnegative weights is strongly NP-hard for cycles with identical edge-length. Thus, the problem cannot be solved in pseudo-polynomial time even for a cycle unless P = NP.

(3) When clients are allowed to have negative weights, the problem becomes NP-hard even for paths of only three vertices and strongly NP-hard for stars.

Thus, the problem is polynomially solvable only for trees with nonnegative weights, and is computationally intractable even for a bit larger graph class or negative weights. Let us remark that, while both of the algorithms by Yu and Li [10] require that the number p of facilities is a fixed constant, our algorithm in (1) allows to have p as a part of input.

The rest of the paper is organized as follows. In Section 2, we give a formal definition of the problem studied in this paper. In

1   Graduate School of Information Sciences, Tohoku University
2   Department of Mathematics, Keio University
3   Research Institute for Mathematical Sciences, Kyoto University
a)   takehiro@ecei.tohoku.ac.jp
b)   kakimura@math.keio.ac.jp
c)   yusuke@kurims.kyoto-u.ac.jp

Section 3, we present a polynomial-time algorithm on a tree. Section 4 is devoted to showing the hardness results. Some proofs are omitted from this extended abstract.

2. Problem Definition

In this section, we formally define the problem studied in this paper.

Let $G = (V, E)$ be an undirected connected graph. For a subgraph H of G, we sometimes denote by $V(H)$ and $E(H)$ the vertex set and edge set of H, respectively. Assume that each edge $e \in E$ has a length $\ell_e \in \mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ is the set of all nonnegative real numbers. We may assume that all vertices in G are clients, and each facility can be located on any place in G, even on an edge. We will refer to interior locations on an edge $e \in E$ by their distances along e from its two endpoints. Throughout the paper, a *point* on G indicates either a vertex in V or an interior location on an edge in E. For notational convenience, we sometimes denote simply by G the set of all points on the graph. For two points $x, y \in G$, let $\mathsf{dist}(x, y)$ denote the shortest-path length between x and y.

Let I be the set of facilities. Then, a *location* of I on a graph $G = (V, E)$ is a tuple X of $|I|$ points on G (which are not necessarily distinct). We denote by G^I the family of all the locations of I on G. Suppose that each vertex $v \in V$ has a weight $w_{v,i} \in \mathbb{R}$ for a facility $i \in I$, where $\mathbb{R}$ is the set of all real numbers; the weight $w_{v,i}$ represents the demand of v to the service provided by $i \in I$. For each vertex $v \in V$ and a location $X \in G^I$, the *cost* $\mathsf{cost}(v, X)$ of v to receive the service from X is defined as follows:

$$\mathsf{cost}(v, X) := \sum_{i \in I} w_{v,i} \cdot \mathsf{dist}(v, x_i),$$

where x_i denotes the point on G at which the facility $i \in I$ is placed by X. In this paper, we study the following problem:

The MULTI-SERVICE CENTER problem
Instance. A graph $G = (V, E)$, an edge length $\ell_e \in \mathbb{R}_{\geq 0}$ for $e \in E$, a set I of facilities, and a weight $w_{v,i} \in \mathbb{R}$ for $v \in V$ and $i \in I$.
Question. Find a location X of I on G that minimizes $\max_{v \in V} \mathsf{cost}(v, X)$.

We call the problem p-SERVICE CENTER if the number p of facilities is a fixed constant. In addition, we sometimes write the name of the problem with its restriction: For example, the problem is called MULTI-SERVICE CENTER *with nonnegative weights* if all weights $w_{v,i}$ are nonnegative for $v \in V$ and $i \in I$.

3. Polynomial-Time Algorithm for Trees with Nonnegative Weights

Recall that Yu and Li [10] showed that p-SERVICE CENTER with positive weights is solvable in polynomial time for general graphs, and 2-SERVICE CENTER with positive weights is solvable in $O(n \log n)$ time for trees having n vertices. Both of the algorithms require that the number p of facilities is fixed. In this section, we prove that MULTI-SERVICE CENTER with *nonnegative* weights is solvable in polynomial time for trees even when the number p of facilities is taken as a part of input, as in the following theorem.

Theorem 1 MULTI-SERVICE CENTER with nonnegative weights can be solved in polynomial time for trees.

In the remainder of this section, we prove Theorem 1. For notational convenience, we may assume that each edge of a given tree has a positive length; this assumption does not lose the generality because we simply regard each edge e with $\ell_e = 0$ as having a sufficiently small positive length.

3.1 Technical highlights

We first explain our main ideas and proof techniques briefly.

To describe a polynomial-time algorithm for trees, let us first consider the case when a graph is a path. In this case, it is not difficult to see that the problem can be reduced to a linear programming problem. In fact, we can identify a point on the path with a 1-dimensional coordinate x by taking one of the end of the path as the origin. Then, the distance from each client to x can be expressed by an absolute value function with respect to x. Therefore, MULTI-SERVICE CENTER for a path is equivalent to minimizing the maximum of the sum of absolute value functions with nonnegative coefficients, which can be formulated as a linear programming problem.

In order to extend the above observation to the tree case, we identify a point on a tree with a path from a specified vertex (a root). Then, we can represent a point on the tree by a vector in the m-dimensional space, where m is the number of edges in the tree. This representation gives us a linear programming problem to find p vectors in the m-dimensional space, as formulated in Problem (3) later. However, since not all m-dimensional vectors correspond to a (feasible) point on the tree, the linear programming problem is a relaxation of MULTI-SERVICE CENTER. The key ingredient of our algorithm is to prove that the linear programming problem has in fact an optimal solution corresponding to an optimal facility location (Lemma 2). Since our proof is constructive, we can find an optimal facility location in polynomial time by solving the linear programming problem.

3.2 Algorithm

Let $T = (V, E)$ be a tree. We choose an arbitrary vertex r in V as the root of T, and regard T as a rooted tree. For notational convenience, when we denote an edge e by $e = uv$, we may assume that u is the parent of v. For each vertex v on T, we denote by P_v the path in T from r to v. For each interior point x of an edge $e_x = u_x v_x$, we denote by P_x the path in T from r to v_x, that is, $P_x = P_{v_x}$. For each edge $e = uv$, let $T - e$ be the subgraph of T obtained by deleting e from T. Then, $T - e$ consists of exactly two trees that have u and v, respectively; we denote the two trees by T_u and T_v where $u \in V(T_u)$ and $v \in V(T_v)$, respectively.

Let x be any point on T, and assume that x is located on an edge $e_x = u_x v_x$; note that $x = u_x$ or $x = v_x$ may hold. Then, we can express the point x using a vector $\tilde{x}$ in $\mathbb{R}_{\geq 0}^E$, defined as follows:

$$\tilde{x}(e) = \begin{cases} \ell_e & \text{if } e \in E(P_{u_x}) = E(P_x) \setminus \{e_x\}, \\ \mathsf{dist}(u_x, x) & \text{if } e = e_x, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Conversely, we say that a vector $\tilde{x} \in \mathbb{R}_{\geq 0}^E$ is *admissible* if there

exist an edge $e_x = u_x v_x$ and d_x in $[0, \ell_{e_x}]$ such that $\tilde{x}$ has the form of (1) in which $\text{dist}(u_x, x)$ is replaced with d_x. Then, there exists a one-to-one correspondence between a point $x \in T$ and an admissible vector $\tilde{x}$, and hence any point on T can be represented as an admissible vector. When a vertex $v \in V$ and a point x on T are expressed by $\tilde{v} \in \mathbb{R}_{\geq 0}^E$ and $\tilde{x} \in \mathbb{R}_{\geq 0}^E$, respectively, it holds that

$$\text{dist}(v, x) = \sum_{e \in E} |\tilde{v}(e) - \tilde{x}(e)| \qquad (2)$$

because we have

$$|\tilde{v}(e) - \tilde{x}(e)| = \begin{cases} \ell_e & \text{if } e \in (E(P_x) \, \triangle \, E(P_v)) \setminus \{e_x\}, \\ \text{dist}(u_x, x) & \text{if } e = e_x \notin E(P_v), \\ \text{dist}(v_x, x) & \text{if } e = e_x \in E(P_v), \\ 0 & \text{otherwise.} \end{cases}$$

For a vertex $v \in V$ and any vector $\tilde{x} \in \mathbb{R}_{\geq 0}^E$ (which is not necessarily admissible), we define

$$d_e(v, \tilde{x}) = |\tilde{v}(e) - \tilde{x}(e)|,$$

where $\tilde{v}$ is a vector expressing v by (1). Consider the problem of finding $|I|$ vectors $\tilde{x}_i \in \mathbb{R}_{\geq 0}^E$ ($i \in I$) that minimizes

$$\max_{v \in V} \sum_{i \in I} \left(w_{v,i} \sum_{e \in E} d_e(v, \tilde{x}_i) \right) = \max_{v \in V} \sum_{i \in I} \left(w_{v,i} \sum_{e \in E} |\tilde{v}(e) - \tilde{x}_i(e)| \right) \qquad (3)$$

subject to $\tilde{x}_i(e) \in [0, \ell_e]$ for $i \in I$ and $e \in E$. Note that, by (2), we have $\sum_{e \in E} d_e(v, \tilde{x}) = \text{dist}(v, x)$ for any point x on T and its corresponding admissible vector $\tilde{x}$. Hence, if we have an additional constraint that each $\tilde{x}_i$ is admissible on the problem (3), then it is equivalent to MULTI-SERVICE CENTER. Thus the problem (3) can be seen as a relaxation of MULTI-SERVICE CENTER.

Lemma 1 The optimal value of the problem (3) is smaller than or equal to that of MULTI-SERVICE CENTER with nonnegative weights.

We say that a feasible solution of the problem (3) is *admissible* if each vector $\tilde{x}_i$ ($i \in I$) of the solution is admissible. Then, an admissible solution of the problem (3) gives a location of I on T. Lemma 1 and the following lemma ensure that solving the problem (3) is equivalent to solving MULTI-SERVICE CENTER with nonnegative weights.

Lemma 2 The problem (3) has an admissible optimal solution. Furthermore, given an optimal solution $\tilde{x}_i$ ($i \in I$) to the problem (3), we can construct an admissible optimal solution in polynomial time.

We are now ready to prove Theorem 1.

Proof of Theorem 1. It follows from Lemmas 1 and 2 that it suffices to solve the problem (3). Note that $\sum_{i \in I} (w_{v,i} \sum_{e \in E} |\tilde{v}(e) - \tilde{x}_i(e)|)$ is a separable-convex function. Since the maximum of convex functions is also convex, so is the objective function of (3). Therefore, the problem (3) is a convex programming problem, which can be solved in polynomial time (see e.g., [3]).

In fact, we can reduce the problem (3) to the following linear programming problem:

$$\begin{aligned} \text{minimize} \quad & c \\ \text{subject to} \quad & \sum_{i \in I} \left(w_{v,i} \sum_{e \in E} |\tilde{v}(e) - \tilde{x}_i(e)| \right) \leq c \quad (v \in V), \\ & \tilde{x}_i(e) \leq \ell_e \quad (i \in I, \, e \in E), \\ & \tilde{x}_i \in \mathbb{R}_{\geq 0}^E \quad (i \in I), \\ & c \in \mathbb{R}_{\geq 0}, \end{aligned}$$

where $\tilde{x}_i(e)$ ($i \in I$, $e \in E$) and c are variables. Note that the first constraint can be described by linear inequalities, since the left-hand side is

$$\begin{aligned} & \sum_{i \in I} \left(w_{v,i} \sum_{e \in E} |\tilde{v}(e) - \tilde{x}_i(e)| \right) \\ & = \sum_{i \in I} w_{v,i} \left(\sum_{e \in E(P_v)} (\ell_e - \tilde{x}_i(e)) + \sum_{e \in E \setminus E(P_v)} \tilde{x}_i(e) \right). \end{aligned}$$

Therefore, it is a linear programming problem with polynomial size, which can be solved in polynomial time (see e.g., [9]). □

4. Hardness Results

In this section, we show that MULTI-SERVICE CENTER is computationally intractable even for very restricted instances. We emphasize again that our analyses are sharp in contrast to Theorem 1.

4.1 Technical Highlights

Recall that, in MULTI-SERVICE CENTER, we are allowed to place each facility at any point on a graph (even on an edge), which makes a solution flexible. We design reductions so that reduced instances force all facilities to be placed at only vertices in any optimal solution. To ensure this condition, we need to analyze the structure of optimal solutions carefully. Interestingly, we will verify this condition for cycles (Theorem 3) by using the nonsingularity of a "distance matrix" [2], which has been studied in the area of algebraic graph theory.

4.2 NP-hardness for paths and stars with negative weights

In this subsection, we show that MULTI-SERVICE CENTER is intractable even for paths and stars if weights of vertices take negative integers. More specifically, the problem is NP-hard for paths of only three vertices, and is strongly NP-hard for stars. Indeed, a path of three vertices is a star, and hence we will construct a common reduction from the following problem:

> The EQUALLY PARTITION problem
> **Instance.** A set A of elements, a bound $b \in \mathbb{Z}_{\geq 0}$, and a size $s_i \in \mathbb{Z}_{\geq 0}$ for each $i \in A$ such that $\sum_{i \in A} s_i = mb$ for some positive integer m.
> **Question.** Can A be partitioned into m disjoint sets $A_1, A_2, \ldots, A_m$ such that $\sum_{i \in A_j} s_i = b$ for all $j \in \{1, 2, \ldots, m\}$?

Here, $\mathbb{Z}_{\geq 0}$ is the set of all nonnegative integers. We summarize our reduction from EQUALLY PARTITION to MULTI-SERVICE CENTER as in the following theorem.

Theorem 2 There is a polynomial-time reduction from EQUALLY PARTITION to MULTI-SERVICE CENTER for instances such that

(a) $G = (V, E)$ is a star $K_{1,m}$ with the center vertex r having m leaves;

(b) $\ell_e := 1$ for every $e \in E$;

(c) $I := A$; and

(d) for $v \in V$ and $i \in I \, (= A)$,

$$w_{v,i} := \begin{cases} -s_i & \text{if } v \in V \setminus \{r\}; \\ -s_i \cdot \frac{2(m-1)}{m} & \text{if } v = r. \end{cases}$$

Notice that EQUALLY PARTITION corresponds to an NP-hard problem PARTITION [5] if $m = 2$. In addition, for general m, EQUALLY PARTITION contains all instances of a strongly NP-hard problem 3-PARTITION [5]. Thus, the following corollary can be obtained from Theorem 2.

Corollary 1 The following (i) and (ii) hold.

(i) MULTI-SERVICE CENTER is NP-hard even when $G = (V, E)$ is a path of three vertices, $\ell_e = 1$ for every $e \in E$, and $w_{v,i} = w_{v',i}$ for any $v, v' \in V$ and $i \in I$.

(ii) MULTI-SERVICE CENTER is NP-hard in the strong sense even when $G = (V, E)$ is a star, and $\ell_e = 1$ for every $e \in E$.

As described in Theorem 2, our reduction from EQUALLY PARTITION to MULTI-SERVICE CENTER is as follows. Suppose that we are given an instance of EQUALLY PARTITION, that is, a set A of elements, a bound $b \in \mathbb{Z}_{\geq 0}$, and a size $s_i \in \mathbb{Z}_{\geq 0}$ for each $i \in A$ such that $\sum_{i \in A} s_i = mb$. Then, we construct a corresponding instance of MULTI-SERVICE CENTER as follows. Let $G = (V, E)$ be a star $K_{1,m}$ with the center vertex r and having m leaves $v_1, v_2, \ldots, v_m$. Set $\ell_e := 1$ for every $e \in E$, and define $I := A$. For $v \in V$ and $i \in I \, (= A)$, set

$$w_{v,i} := \begin{cases} -s_i & \text{if } v \in V \setminus \{r\}; \\ -s_i \cdot \frac{2(m-1)}{m} & \text{if } v = r. \end{cases}$$

This reduction can be done in polynomial time.

To show the correctness of our reduction above, it suffices to prove the following lemma.

Lemma 3 The original instance of EQUALLY PARTITION has a desired partition if and only if there is a location X of I for the corresponding instance of MULTI-SERVICE CENTER such that $\max_{v \in V} \mathsf{cost}(v, X) \leq -2(m-1)b$.

4.3 Strong NP-hardness for cycles with nonnegative weights

We show that the problem is strongly NP-hard even when restricted to cycles with identical edge-length and nonnegative integer weights.

Theorem 3 MULTI-SERVICE CENTER with nonnegative weights is NP-hard in the strong sense even when $G = (V, E)$ is a cycle, $\ell_e = 1$ for every $e \in E$, and $w_{v,i} = w_{v',i} \in \mathbb{Z}_{\geq 0}$ for any $v, v' \in V$ and $i \in I$.

Thus, MULTI-SERVICE CENTER cannot be solved in pseudo-polynomial time even for such restricted instances unless P = NP.

In the remainder of this subsection, we prove the theorem by giving a polynomial-time reduction from a strongly NP-hard problem 3-PARTITION to MULTI-SERVICE CENTER for such restricted instances. The 3-PARTITION problem is defined as follows [5]:

The 3-PARTITION problem
Instance. A set A of $3m$ elements, a bound $b \in \mathbb{Z}_{\geq 0}$, and a size $s_i \in \mathbb{Z}_{\geq 0}$ with $\frac{b}{4} < s_i < \frac{b}{2}$ for each $i \in A$ such that $\sum_{i \in A} s_i = mb$.
Question. Can A be partitioned into m disjoint sets $A_1, A_2, \ldots, A_m$ such that $\sum_{i \in A_j} s_i = b$ for all $j \in \{1, 2, \ldots, m\}$?

Note that since $\frac{b}{4} < s_i < \frac{b}{2}$ for each $i \in A$, we have $|A_j| = 3$ for all $j \in \{1, 2, \ldots, m\}$. It is known that 3-PARTITION remains NP-hard in the strong sense even if m is restricted to be odd [8].

Suppose that we are given a set A of $3m$ elements, a bound $b \in \mathbb{Z}_{\geq 0}$, and a size $s_i \in \mathbb{Z}_{\geq 0}$ for each $i \in A$ as an instance of 3-PARTITION, where m is an odd number. We construct a corresponding instance of MULTI-SERVICE CENTER as follows. Let $G = (V, E)$ be a cycle with m vertices such that $V = \{v_1, v_2, \ldots, v_m\}$, $E = \{v_1 v_2, v_2 v_3, \ldots, v_{m-1} v_m, v_m v_1\}$, and $\ell_e := 1$ for every $e \in E$. Define $I := A$, and set $w_{v,i} := s_i$ for $v \in V$ and $i \in I \, (= A)$. This reduction can be done in polynomial time.

To show the correctness of our reduction above, it suffices to prove the following lemma.

Lemma 4 The original instance of 3-PARTITION has a desired partition if and only if there is a location X of I for the corresponding instance of MULTI-SERVICE CENTER such that

$$\max_{v \in V} \mathsf{cost}(v, X) \leq \frac{(m^2 - 1)b}{4}.$$

References

[1] Anzai, T., Ito, T., Suzuki, A. and Zhou, X.: The multi-service center decision problem is NP-complete for split graphs, *the 6th World Congress on Engineering and Technology* (*CET 2016*) (2016).

[2] Bapat, R. B., Kirkland, S. J. and Neumann, M.: On distance matrices and Laplacians, *Linear Algebra and Its Applications*, Vol. 401, pp. 193–209 (2005).

[3] Boyd, S. and Vandenberghe, L.: *Convex Optimization*, Cambridge University Press, New York, NY, USA (2004).

[4] Drezner, Z. and Hamacher, H. W.(eds.): *Facility Location: Applications and Theory*, Springer-Verlag, Berlin Heidelberg (2002).

[5] Garey, M. R. and Johnson, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman & Co., New York, NY, USA (1990).

[6] Gonzalez, T. F.: Clustering to minimize the maximum intercluster distance, *Theoretical Computer Science*, Vol. 38, pp. 293–306 (1985).

[7] Hochbaum, D. S. and Shmoys, D. B.: A best possible heuristic for the k-center problem, *Mathematics of Operations Research*, Vol. 10, pp. 180–184 (1985).

[8] Sadasivam, S. and Zhang, H.: NP-Completeness of st-orientations for plane graphs, *Theoretical Computer Science*, Vol. 411, pp. 995–1003 (2010).

[9] Schrijver, A.: *Theory of Linear and Integer Programming*, John Wiley & Sons, Inc., New York, NY, USA (1986).

[10] Yu, H. and Li, C.: The multi-service center problem, *the 23rd International Symposium on Algorithms and Computation* (*ISAAC 2012*), Lecture Notes in Computer Science, Vol. 7676, pp. 578–587 (2012).