Approximability and Inapproximability of the Minimum Certificate Dispersal Problem[☆]

Tomoko IZUMI[*,a], Taisuke IZUMI[b], Hirotaka ONO[c], Koichi WADA[b]

[a] *College of Information Science and Engineering, Ritsumeikan University, Kusatsu, 525-8577 Japan.*
Tel/Fax: +81-77-561-3445
[b] *Graduate School of Engineering, Nagoya Institute of Technology, Nagoya, 466-8555, Japan.*
[c] *Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka, 819-0395, Japan.*

Abstract

Given an n-vertex directed graph $G = (V, E)$ and a set $R \subseteq V \times V$ of requests, we consider to assign a set of edges to each vertex in G so that for every request (u, v) in R the union of the edge sets assigned to u and v contains a path from u to v. *The Minimum Certificate Dispersal Problem* (MCD) is defined as one to find an assignment that minimizes the sum of the cardinalities of the edge sets assigned to each vertex. This problem has been shown to be NP-hard in general, though it is polynomially solvable for some restricted classes of graphs and restricted request structures, such as bidirectional trees with requests of all pairs of vertices. In this paper, we give an advanced investigation about the difficulty of MCD by focusing on the relationship between its (in)approximability and request structures. We first show that MCD with general R has $\Theta(\log n)$ lower and upper bounds on approximation ratio under the assumption $P \neq NP$. We then assume R forms a clique structure, called *Subset-Full*, which is a natural setting in the context of the application. Interestingly, under this natural setting, MCD becomes to be 2-approximable, though it has still no polynomial time approximation algorithm whose factor better than 677/676 unless $P = NP$. Finally, we show that this approximation ratio can be improved to 3/2 for undirected variant of MCD with Subset-Full.

Key words: minimum certificate dispersal problem, graph theory, approximation algorithms, combinatorial optimization

[☆] An extended abstract of this article was presented in *Proceedings of the 15th Annual International Conference,* (COCOON 2009), Lecture Notes in Computer Science, Vol. 5609, pp. 56–65, Springer, 2009.

[*] Corresponding author.
Email addresses: izumi-t@fc.ritsumei.ac.jp (Tomoko IZUMI), t-izumi@nitech.ac.jp (Taisuke IZUMI), ono@inf.kyushu-u.ac.jp (Hirotaka ONO), wada@nitech.ac.jp (Koichi WADA)

1. Introduction

Background and Motivation. Let $G = (V, E)$ be a directed graph and $R \subseteq V \times V$ be a set of ordered pairs of vertices, which represents requests about reachability between two vertices. For given G and R, we consider an assignment of a set of edges to each vertex in G. The assignment satisfies a request (u, v) if the union of the edge sets assigned to u and v contains a path from u to v. *The Minimum Certificate Dispersal Problem* (MCD) is the one to find the assignment satisfying all requests in R that minimizes the sum of the cardinalities of the edge sets assigned to each vertex.

This problem is motivated by a requirement in public-key based security systems, which are known as a major technique for supporting secure communication in a distributed system [1, 2, 3, 4, 5, 6, 7]. The main problem of the systems is to make each user's public key available to others in such a way that its authenticity is verifiable. One of well-known approaches to solve this problem is based on public-key certificates. A public-key certificate contains the public key of a user v encrypted by using the private key of a user u. If a user u knows the public key of another user v, user u can issue a certificate from u to v. Any user who knows the public key of u can use it to decrypt the certificate from u to v for obtaining the public key of v. All certificates issued by users in a network can be represented by a certificate graph: Each vertex corresponds to a user and each directed edge corresponds to a certificate. When a user w has communication request to send messages to a user v securely, w needs to know the public key of v to encrypt the messages with it. For satisfying a communication request from a vertex w to v, vertex w needs to get vertex v's public-key. To compute v's public-key, w uses a set of certificates stored in w and v in advance. Therefore, in a certificate graph, if a set of certificates stored in w and v contains a path from w to v, then the communication request from w to v is satisfied. In terms of cost to maintain certificates, the total number of certificates stored in all vertices must be minimized for satisfying all communication requests.

While, from the practical aspect, MCD should be handled in the context of distributed computing theory, its inherent difficulty as an optimization problem is not so clear even in centralized settings: Jung et al. discussed MCD with a restriction of available paths in [4] and proved that the problem is NP-hard. In their work, to assign edges to each vertex, only the restricted paths which are given for each request is allowed to be used. MCD, with no restriction of available paths, is first formulated in [7]. In [7], MCD, with no restriction of available paths, is proved to be also NP-hard even if the input graph is strongly connected. Known results about the complexity of MCD are actually only these NP-hardness. This fact yields a theoretical interest of revealing the (in)approximability of MCD. As for the positive side, MCD is polynomially solvable for bidirectional trees, rings and Cartesian products of graphs [7].

This paper also investigates how the request structures affect the difficulty of MCD. As seen above, MCD is doubly structured in a sense: One structure is the graph G itself and the other is the request structure R. We would like to

Table 1: Approximability / Inapproximability bounds shown in this paper

<table>
<tr><td rowspan="2"></td><td colspan="3">Restriction on request</td></tr>
<tr><td>Arbitrary</td><td>Subset-Full</td><td>Full</td></tr>
<tr><td rowspan="2">Inapproximability</td><td>$\Omega(\log n)$</td><td rowspan="2">677/676</td><td rowspan="2">open</td></tr>
<tr><td>261/260
(for bidirectional graphs)</td></tr>
<tr><td rowspan="2">Approximation ratio</td><td rowspan="2">$O(\log n)$</td><td>2</td><td>2 [7]</td></tr>
<tr><td colspan="2">3/2
(for undirected graphs)</td></tr>
</table>

n is the number of vertices.

investigate how the tractability of MCD changes as the topology of R changes. In passing, a typical doubly structured problem in this sense is the *H-coloring problem* [8]. The H-coloring problem is coloring problem with restrictions of adjacent colors, which are given by a graph H. That is, when the graph H is a complete graph, the H-coloring problem is equivalent to the traditional coloring problem. About H-coloring, so-called dichotomy theorem is well known: H-coloring is solvable in polynomial time if and only if H has a loop or is bipartite graph; otherwise the problem is NP-complete. On MCD, our interest here is to investigate whether the hardness (of approximation) of MCD depends on the restrictions about R. A similar structure is also found in the VPN design problem [14]. It is defined as a certain kind of connection-establishment problems, and allows the optimal solution computable within polynomial time when the request is all-to-all connections(i.e., in the context of MCD, R induces a complete subgraph)[15].

Revealing the relationship between tractability and request structures is a natural problem not only from the theoretical viewpoint but also from the practical viewpoint, because, in public-key based security systems, a set of requests should have a certain type of structures. For example, it is reasonable to consider the situation in which a set of vertices belonging to a certain community should have requests between each other in the community. This situation is interpreted that R forms a clique structure. Thus the following question arises: If R forms a clique, can the approximability of MCD be improved?

Our Contribution. In this paper, we investigate the approximability of MCD from the perspective how the structure of R affects the complexity of MCD. We classify the set R of requests according to the elements of R: R is *subset-full* if for a subset V' of V, R consists of all reachable pairs of vertices in V', and R is *full* if the subset V' is equal to V. Note that Subset-Full corresponds to the situation that R forms a clique. Table 1 summarizes the results in this paper.

Here we review our contribution. We first consider the general case: We show that if we have no restriction about R, a lower bound on approximation ratio for MCD is $\Omega(\log n)$ and an upper bound is $O(\log n)$, where n is the number

of vertices. Namely, the lower and upper bounds coincide as $\Theta(\log n)$ in terms of order. Moreover, it is proved that we can still obtain the inapproximability $\Omega(1)$ of MCD even when the graph class is restricted to bidirectional graphs.

As the second half of the contribution, for subset-full requests, we show that the lower bound of approximation ratio for MCD is 677/676 and the upper bound is 2. The lower bound is obtained by a gap-preserving reduction from VERTEX-COVER. The upper bound is proved by a detailed analysis of the algorithm MinPivot , which is proposed in [7]. While Zheng et al. have shown that MinPivot achieves approximation ratio 2 with full requests, we can obtain the same approximation ratio by a different approach even when the set of requests is subset-full. In addition, by extending the approach, it is also shown that MinPivot guarantees 3/2 approximation ratio for MCD of the undirected variant with subset-full requests.

The remainder of the paper is organized as follows. In Section 2, we define the Minimum Certificate Dispersal Problem (MCD). Section 3 presents inapproximability of MCD with general R and one with Subset-Full. The upper bound of MCD with general R and one with Subset-Full are shown in Sections 4 and 5 respectively. Section 6 concludes the paper.

2. Minimum Certificate Dispersal Problem

Let $G = (V, E)$ be a directed graph, where V and E are the sets of vertices and edges in G respectively. An edge in E connects two distinct vertices in V. The edge from vertex u to v is denoted by (u, v). The numbers of vertices and edges in G are denoted by n and m, respectively (i.e., $n = |V|, m = |E|$). A sequence of edges $p(v_0, v_k) = (v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$ is called a *path* from v_0 to v_k of length k. A path $p(v_0, v_k)$ can be represented by a sequence of vertices $p(v_0, v_k) = (v_0, v_1, \ldots, v_k)$. For a path $p(v_0, v_k)$, v_0 and v_k are called the source and destination of the path respectively. The length of a path $p(v_0, v_k)$ is denoted by $|p(v_0, v_k)|$. For simplicity, we treat a path as the set of edges on the path when no confusion occurs. A shortest path from u to v is the one whose length is the minimum of all paths from u to v, and the distance from u to v is the length of a shortest path from u to v, denoted by $d(u, v)$.

A *dispersal* D of a directed graph $G = (V, E)$ is a family of sets of edges indexed by V, that is, $D = \{D_v \subseteq E | v \in V\}$. We call D_v a local dispersal of v. A local dispersal D_v indicates the set of edges assigned to v. The *cost* of a dispersal D, denoted by $c(D)$, is the sum of the cardinalities of all local dispersals in D (i.e., $c(D) = \Sigma_{v \in V} |D_v|$). A request is a reachable ordered pair of vertices in G. For a request (u, v), u and v are called the source and destination of the request respectively. A set R of requests is *subset-full* if there exists a subset V' of V such that R consists of all reachable pairs of vertices in V' (*i.e.*, $R = \{(u, v) | u$ is reachable to v in G, $u, v \in V' \subseteq V\}$), and R is *full* if the subset V' is equal to V. We say a dispersal D of G *satisfies* a set R of requests if a path from u to v is included in $D_u \cup D_v$ for any request $(u, v) \in R$.

The *Minimum Certificate Dispersal Problem (MCD)* is defined as follows:

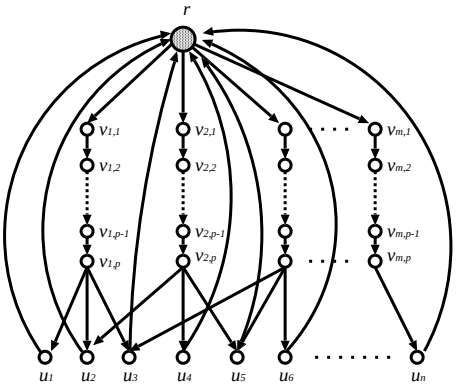


Figure 1: Reduction for general case (from SET-COVER)

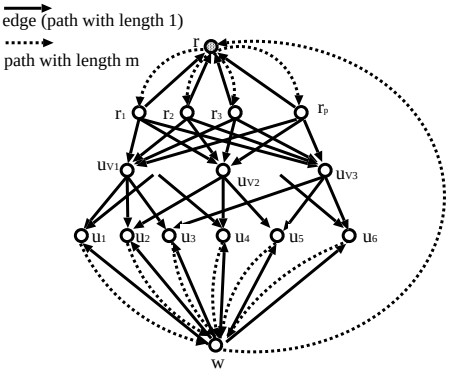


Figure 2: Reduction for Subset-Full (from VERTEX-COVER)

Definition 1 (Minimum Certificate Dispersal Problem (MCD)).
INPUT: A directed graph $G = (V, E)$ and a set R of requests.
OUTPUT: A dispersal D of G satisfying R with minimum cost.

The minimum among costs of dispersals of G that satisfy R is denoted by $c_{min}(G, R)$. For short, the cost $c_{min}(G, R)$ is also denoted by $c_{min}(G)$ when R is full. Let D^{Opt} be an optimal dispersal of G which satisfies R (i.e., D^{Opt} is one such that $c(D^{Opt}) = c_{min}(G, R)$).

In this paper, we deal with MCD for undirected graphs in Section 5.3. For an undirected graph G, the edge between vertices u and v is denoted by (u, v) or (v, u). When an edge (u, v) is included in a local dispersal D_v, the vertex v has two paths from u to v and from v to u.

3. Inapproximability

It was shown in [7] that MCD for strongly connected graphs is NP-hard by a reduction from the VERTEX-COVER problem. In this section, we provide another proof of NP-hardness of MCD for strongly connected graphs, which implies a stronger inapproximability. Here, we show a reduction from the SET-COVER problem. For a collection $\mathcal{C}$ of subsets of a finite universal set U, $\mathcal{C}'$ ($\subseteq \mathcal{C}$) is called a *set cover* of U if every element in U belongs to at least one member of $\mathcal{C}'$. Given $\mathcal{C}$ and a positive integer k, SET-COVER is the problem of deciding whether a set cover $\mathcal{C}' \subseteq \mathcal{C}$ of U with $|\mathcal{C}'| \leq k$ exists. By considering the graph where each element corresponds to an edge and each subset to a vertex, it becomes equivalent to VERTEX-COVER. Then, from the definition, each element is contained in exactly two subsets.

The reduction from SET-COVER to MCD is as follows: Given a universal set $U = \{1, 2, \ldots, n\}$ and its subsets $S_1, S_2, \ldots, S_m$ and a positive integer k as an instance $\mathcal{I}$ of SET-COVER, we construct a graph $G_{\mathcal{I}}$ including gadgets that

mimic (a) elements, (b) subsets, and (c) a special gadget: (a) For each element i of the universe set $U = \{1, 2, \ldots, n\}$, we prepare an element gadget u_i (it is just a vertex); let V_U be the set of element vertices, i.e., $V_U = \{u_i \mid i \in U\}$. (b) For each subset $S_j \in \mathcal{C}$, we prepare a directed path $(v_{j,1}, v_{j,2}, \ldots, v_{j,p})$ of length $p-1$, where p is a positive integer used as a parameter. The end vertex $v_{j,p}$ is connected to the element gadgets that correspond to elements belonging to S_j. For example, if $S_1 = \{2, 4, 5\}$, we have directed edges $(v_{1,p}, u_2)$, $(v_{1,p}, u_4)$ and $(v_{1,p}, u_5)$. (c) The special gadget just consists of a base vertex r. This r has directed edges to all $v_{j,1}$'s of $j = 1, 2, \ldots, m$. Also r has an incoming edge from each u_i. See Figure 1 as an example of the reduction, where $S_1 = \{1, 2, 3\}, S_2 = \{2, 4, 5\}$ and $S_3 = \{3, 5, 6\}$. We can see that $G_{\mathcal{I}}$ is strongly connected. The set $R_{\mathcal{I}}$ of requests contains the requests from the base vertex r to all element vertices u_i, i.e., $R = \{(r, u_i) \mid u_i \in V_U\}$.

We can show the following, although we omit the proof because it is straightforward: (i) If the answer of instance $\mathcal{I}$ of SET-COVER is yes, then $c_{min}(G_{\mathcal{I}}, R_{\mathcal{I}}) \leq pk + n$. (ii) Otherwise, $c_{min}(G_{\mathcal{I}}, R_{\mathcal{I}}) \geq p(k+1) + n$. About the inapproximability of SET-COVER, it is known that SET-COVER has no polynomial-time approximation algorithm with factor better than $0.2267 \ln n$, unless $P = NP$ [9]. More precisely, there exists g such that the following decision problem (SET-COVER GAP problem) is NP-hard: Given a SET-COVER instance, distinguishing between (a) there exists a set cover with at most size g, and (b) every set cover has size at least $0.2267 g \ln n$. By the above reduction, we obtain a *gap-preserving reduction* [10] as follows:

Lemma 1. *The above construction of $G_{\mathcal{I}}$ is a gap-preserving reduction from SET-COVER to MCD for strongly connected graphs such that*

(i) *if $OPT_{SC}(\mathcal{I}) \leq g$, then $c_{min}(G_{\mathcal{I}}, R_{\mathcal{I}}) \leq p \cdot g + n$,*

(ii) *if $OPT_{SC}(\mathcal{I}) \geq g \cdot c \ln n$, then $c_{min}(G_{\mathcal{I}}, R_{\mathcal{I}}) \geq (p \cdot g + n) \left(c \ln n - \frac{cn \ln n - n}{p \cdot g + n} \right)$,*

where $OPT_{SC}(\mathcal{I})$ and g denote the optimal value of SET-COVER and a gap parameter for $\mathcal{I}$ respectively, and $c = 0.2267$.

Note that for any positive constant $\alpha \leq 1$, there exists p of polynomial size with respect to n that satisfies $\left(c \ln n - \frac{cn \ln n - n}{p \cdot g + n} \right) \geq (p \cdot g + n) \cdot c \ln n \, (1 - \alpha)$. Thus, from the NP-hardness of SET-COVER GAP problem, for any positive constant $\alpha < 1$, there exists g' such that it is NP-hard to distinguish between (a) there exists a dispersal whose cost is at most size g', and (b) every dispersal has size at least $g' \cdot (c - \alpha) \ln n$. This implies the following theorem.

Theorem 2. *There exists no $((0.2267 - \alpha) \ln |V| - \varepsilon)$ factor approximation polynomial time algorithm of MCD for strongly connected graphs unless $P = NP$, where α and ε are arbitrarily small positive constants.*

It might be difficult to directly extend the result to more restricted classes of strongly connected graphs, e.g., bidirectional graphs, but we can still obtain

some inapproximability result for bidirectional graphs, by slightly modifying the graph $G_{\mathcal{I}}$, though we omit the details. We use a reduction not from SET-COVER but from VERTEX-COVER. The graph constructed from VERTEX-COVER is similar to $G_{\mathcal{I}}$, but we replace each (directed) edge by bidirectional edges, and also we delete edges between u_i's and r. Furthermore, we set $p = 1$. Then we obtain the following lemma:

Lemma 3. *There is a gap-preserving reduction from VERTEX-COVER for graphs with degree at most* 4 *to MCD for bidirectional graphs such that*

(i) *if* $OPT_{VC}(\mathcal{I}) = g$, *then* $c_{min}(G_{\mathcal{I}}, R_{\mathcal{I}}) \leq g + n$,

(ii) *if* $OPT_{VC}(\mathcal{I}) \geq c \cdot g$, *then* $c_{min}(G_{\mathcal{I}}, R_{\mathcal{I}}) \geq (g + n)\left(c - \frac{(c-1)n}{g+n}\right)$,

where $OPT_{VC}(\mathcal{I})$ *and* g *denote the optimal value of VERTEX-COVER and a gap parameter for* $\mathcal{I}$, *and* $c = 53/52$.

In this lemma, $c = 53/52$ represents an inapproximability of VERTEX-COVER for graphs with degree at most 4 under the assumption $P \neq NP$ [11]. Since we can assume $4 \cdot g \geq n$ (otherwise, the answer is clearly "no"), we obtain the following theorem.

Theorem 4. *There exists no* $(261/260 - \varepsilon)$ *factor approximation polynomial time algorithm of MCD for bidirectional graphs unless* $P = NP$, *where* ε *is an arbitrarily small positive constant.*

Again we consider another reduction from VERTEX-COVER for graphs with degree at most 4, in which we embed an instance to MCD problem with a subset-full request structure. As well as the reduction from SET-COVER, we prepare (a) edge gadgets, (b) vertex gadgets, and (c) special gadgets. The reduction from VERTEX-COVER to MCD with subset-full requests is as follows: Given $G = (V, E)$ with degree at most 4 and a positive integer k as an instance $\mathcal{I}$ of VERTEX-COVER, where $V = \{1, 2, \ldots, n\}$ is the vertex set and $E = \{e_1, e_2, \ldots, e_m\}$ is the edge set, we construct an MCD graph $G'_{\mathcal{I}}$. (a) For each edge e_i in E, we prepare an m-length directed path $(u_i, u_{i,1}, \ldots, u_{i,m-1}, w)$ and (w, u_i) as an edge gadget, where w is a common vertex among edge gadgets. (b) For each vertex $j \in V$, we prepare a vertex u_j^V as a vertex gadget. If j is connected with edge e_i, we add directed edges (u_j^V, u_i). For example, if $e_5 = \{2, 3\}$, we have directed edges (u_2^V, u_5), (u_3^V, u_5). Note that each u_i has exactly two incoming edges from vertex gadgets. (c) The special gadgets consist of p base vertices $r_1, r_2, \ldots, r_p$ and one root vertex r. Each r_j and r are connected by path $(r, r_{j,1}, \ldots, r_{j,m-1}, r_j)$ and edge (r_j, r). Also, each r_i has directed edges to all u_j^V's of $j = 1, 2, \ldots, m$. Furthermore, we prepare an m-length directed path from w to r, i.e., $(w, w_1, \ldots, w_{m-1}, r)$. See Figure 2 as an example of the reduction, in which we have $e_2 = \{1, 2\}, e_3 = \{1, 3\}$ and $e_5 = \{2, 3\}$. We can see that $G'_{\mathcal{I}}$ is strongly connected.

The set R' of requests are defined as $R' = R_{a,a} \cup R_{a,c} \cup R_{c,c}$, where $R_{a,a} = \{(u_i, u_j) \mid i, j = 1, 2, \ldots, m, \text{ and } i \neq j\}$, $R_{a,c} = \{(u_i, r_j), (r_j, u_i) \mid i = 1, \ldots, m\}$

and $R_{c,c} = \{(r_i, r_j) \mid i, j = 1, 2, \ldots, p, \text{ and } i \neq j\}$. Let $V^{(a)}$ and $V^{(c)}$ denote $\{u_i \mid i = 1, \ldots, m\}$ and $\{r_j \mid j = 1, 2, \ldots, p\}$, respectively.

Lemma 5. *Let $p = m$. The above construction of $G'_{\mathcal{I}}$ and R' is a gap-preserving reduction from VERTEX-COVER with degree at most 4 to MCD with subset-full requests for strongly connected graphs such that:*

(i) *If $OPT_{VC}(\mathcal{I}) = g(\mathcal{I})$, then $c_{min}(G'_{\mathcal{I}}, R') \leq m(g(\mathcal{I}) + 3m + 3)$.*

(ii) *If $OPT_{VC}(\mathcal{I}) > c \cdot g(\mathcal{I})$, then $c_{min}(G'_{\mathcal{I}}, R') > m(g(\mathcal{I}) + 3m + 3)(c - \frac{(3m+3)(c-1)}{g(\mathcal{I})+3m+3})$,*

where $OPT_{VC}(\mathcal{I})$ denotes the optimal value of VERTEX-COVER for $\mathcal{I}$ and $c = 53/52$.

Proof. In this proof, we define $k_1 := g(\mathcal{I})$ and $k_2 := c \cdot g(\mathcal{I})$. We first show (i). For a vertex cover C with size k_1, we construct a solution of MCD as follows: Assume edge e_i is covered by a vertex $c(i)$ in C, and let $D_{u_i} = \{(u_i, w), (w, u_i), (u^V_{c(i)}, u_i), (w, r)\}$ for $i = 1, 2, \ldots, m$, where $\{(u_i, w)\} = \{(u_i, u_{i,1}), (u_{i,1}, u_{i,2}), \ldots, (u_{i,m-2}, u_{i,m-1}), (u_{i,m-1}, w)\}$ and $\{(w, r)\} = \{(w, w_1), (w_1, w_2), \ldots, (w_{m-2}, w_{m-1}), (w_{m-1}, r)\}$. Also let $D_{r_j} = \{(r, r_j), (r_j, r)\} \cup \{(r_j, u^V_i) \mid i \in C\}$ for $j = 1, 2, \ldots, p(= m)$, where $\{(r, r_j)\} = \{(r, r_{j,1}), (r_{j,1}, r_{j,2}), \ldots, (r_{j,m-2}, r_{j,m-1}), (r_{j,m-1}, r_j)\}$. Then we have $c(D) = m(2m+2)+p(m+1+k_1) = m(3m + 3 + k_1)$, which shows (i).

We next show (ii) by contradiction. We assume that there exists an instance $\mathcal{I}$ of VERTEX-COVER whose optimal solution size is more than k_2, but $c_{min}(G'_{\mathcal{I}}, R') \leq m(3m + 3 + k_2)$. Suppose that D^{Opt} (for simplicity, we denote by D in this proof) is an optimal solution of MCD instance $G'_{\mathcal{I}}$ and R'. We can treat directed paths $p(u_i, w) = (u_i, u_{i,1}, \ldots, w)$, for $i = 1, \ldots, m$, $p(w, r) = (w, w_1, \ldots, w_{m-1}, r)$ and $p(r, r_j) = (r, r_{j,1}, \ldots, r_{j,m-1}, r_j)$ for $j = 1, \ldots, p$, as edges with length m, because these edges are used only to make w, r and r_j directly reachable from u_i, w and r, respectively; in an optimal solution, they are not chosen separately in D. Thus from now on, we denote $p(u_i, w), p(w, r)$ and $p(r, r_j)$ simply by (u_i, w), (w, r) and (r, r_j), for each i and j. In this notation, the costs of $(u_i, w), (w, r)$ and (r, r_j) are all m. We first claim that $(u_i, w) \in D_{u_i}$ and $(r, r_j) \in D_{r_j}$ for every $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, p$. Otherwise, $|\{i \mid (u_i, w) \notin D_{u_i}\}| + |\{j \mid (r, r_j) \notin D_{r_j}\}| \geq 1$ holds. Let $A = \{i \mid (u_i, w) \notin D_{u_i}\}$ and $B = \{j \mid (r, r_j) \notin D_{r_j}\}$. Since $(u_i, r_j) \in R'$ for any pair of i and j, we have $(u_i, w), (w, r), (r, r_j) \in D_{u_i} \cup D_{r_j}$ for any i and j. This implies that $\{(u_i, w) \mid i \in A\} \subseteq D_{u_j}$ for any j, and $\{(r, r_j) \mid j \in B\} \subseteq D_{u_i}$ for any i. Also if $(w, r) \notin D_{u_i}$ for some i, then $(w, r) \in D_{r_j}$ for every j, and if $(w, r) \notin D_{r_j}$ for some j, then $(w, r) \in D_{r_i}$ for every i. These imply that $c(D) = \sum_i |D_{u_i}| + \sum_j |D_{u_j}| \geq mp|A| + m^2|B| + m(m - |A|) + m(p - |B|) + \min\{m, p\}m = m^2(|A| + |B| + 3) - m(|A| + |B|)$. Then, if $|A| + |B| > 0$, we have $c(D) \geq 3m^2 + m(|A| + |B|)(m - 1) \geq 3m^2 + m(m - 1)$. Since we can assume $k_2 < m - 4$ (otherwise, we can solve the original vertex cover problem in polynomial time by an exhaustive search), we have $c(D) \geq 3m^2 + m(m - 1) =$

$3m^2+m(3+m-4) > 3m^2+m(3+k_2)$, which contradicts the assumption; we can assume $(u_i, w) \in D_{u_i}$ and $(r, r_j) \in D_{r_j}$ for every i and j, and either $(w, r) \in D_{u_i}$ for every i or $(w, r) \in D_{r_j}$ for every j. For these, we should allocate cost $3m^2$ in $c(D)$.

Let us now consider $R_{a,c}$. We first consider the reachability from r_i to u_j. In order to make u_j's reachable from r_i, we can have the following two strategies: One is that r_i takes a route via some $u_{i'}$ and w, and then reaches other u_j's. The other is that r_i takes a route to every u_j via a u^V vertex (not via another $u_{j'}$). We call the former strategy (s1) and the latter (s2). To realize (s1), D_{r_i} or D_{u_j}'s should contain $(u_{i'}, w)$. If D_{r_i} does not contain $(u_{i'}, w)$, then $(m-1)$ D_{u_j}'s contain $(u_{i'}, w)$, but it contradicts the size of $c(D)$. Thus, for any r_i in this strategy (s1), there exists i' such that $(u_{i'}, w) \in D_{r_i}$. If $p-1(= m-1)$ r_i's take (s1), we need extra costs $m(m-1)$ for $c(D)$, which contradicts the size of $c(D)$ again; there are at least two r_i's taking the other strategy (s2). From the above argument, we can assume that if r_i takes (s2), $(u_j, w) \notin D_i$ holds for any i. Paths between r_i's and u_j's form a directed acyclic graph that ends at u_j's; w is not reachable from u_i's in D_{r_a} and D_{r_b}, where both r_a and r_b take (s2). For any r_a taking (s2), there exists $C_a \subseteq V$ of G such that for any u_j some $i \in C_a$ satisfies $(u_i^V, u_j) \in D_{r_a} \cup D_{u_j}$ (this condition implies that C_a is a vertex cover of G), and for any $i \in C_a$ $(r_a, u_i^V) \in D_{r_a} \cup D_{u_j}$.

The cost allocated at this point is evaluated as follows. Let α be the ratio of r_i's taking strategy (s1). (Consecutively, the ratio of r_i's taking strategy (s2) is $1-\alpha$. The numbers of r_i's taking (s1) and (s2) are $p\alpha = m\alpha$ and $p(1-\alpha) = m(1-\alpha)$, respectively.) For each r_i taking strategy (s1), we should have $(u_{i'}, w) \in D_{r_i}$ and $(r_i, u_{j'}^V) \in D_{r_i} \cup \bigcup_j D_{u_j}$ for some i' and j', whose cost is at least $m+1$ for each; in total, $m\alpha(m+1)$. For $m(1-\alpha)$ r_i's taking strategy (s2), it costs at least $m(1-\alpha)k^* + m$, where k^* denotes the size of minimum vertex covers of G. Thus the total cost newly booked by the previous paragraph is $m\alpha(m+1) + m(1-\alpha)k^* + m$.

Next we consider $R_{a,a}$. By the above argument, we have $(r, r_i) \in D_{r_i}$ for every i. To make r_i reachable from r_j, there are two ways: one is $(r_j, r) \in D_{r_i} \cup D_{r_j}$, and the other is that $D_{r_i} \cup D_{r_j}$ includes a path from r_j to r via w. The former costs at least 1 per r_i. In the latter case, the cost may be absorbed by other paths. In fact, if r_j takes strategy (s2) stated above, r_j may have a path from r_j to w; the cost for connecting r_j and r can be 0 (in case strategy (s2), we cannot include any (u_a, w), it should take cost 1). Thus the total cost allocated here is at least $m(1-\alpha)$.

Finally, we consider $R_{c,c}$. Similarly as above, we have $(u_i, w) \in D_{u_i}$ for every i. To make u_i reachable from u_j, there are two ways: one is $(w, u_i) \in D_{u_i} \cup D_{u_j}$, and the other is that $D_{u_i} \cup D_{u_j}$ includes a path from w to u_i via r. The former costs at least 1 per u_i. In the latter case, the cost may be absorbed by other paths. However, in the previous argument, any (r, r_a) are not in D_{u_i} but some (r, r_a) should be included in D_{u_i}; the cost of (r, r_a), m, is newly added. That is, the total cost allocated here is at least m.

Summing them up, we have cost at least $(3+\alpha)m^2 + m(3 + k^*(1-\alpha)) \leq c_{min}(G'_{\mathcal{I}}, R') \leq m(3m+3+k_2)$. This yields $k_2 - k^* = \alpha(m-k^*) \geq 0$, which

contradicts the assumption that $k_2 < k^*$. □

The constant $c = 53/52$ represents an inapproximability bound for VERTEX-COVER with degree at most 4 under the assumption $P \neq NP$ [11]. From this lemma and $4g(\mathcal{I}) \geq m$, we obtain the following theorem:

Theorem 6. *There exists no* $(677/676 - \varepsilon)$ *factor approximation polynomial time algorithm of MCD with subset-full requests for strongly connected graphs unless* $P = NP$*, where* ε *is an arbitrarily small positive constant.*

Remark 1. Some readers may consider that it might be possible to get much stronger inapproximability bounds (e.g., $\Omega(\log n)$) from SET-COVER, by tuning the value of p. However, it is actually not possible. If we let p be larger value, e.g., n^2, then the structure of optimal solutions drastically changes; by letting each of u_i's have larger D_{u_j}, we can keep D_{r_i} with a smaller size, which is no longer a gap-preserving reduction. In fact, in the following section, we present a 2-approximation polynomial time algorithm, which implies that there does not exist any gap-preserving polynomial time reduction from SET-COVER.

4. Approximability

In the previous section, we show that it is difficult to design a polynomial time approximation algorithm of MCD whose factor is better than $(0.2267(1 + \alpha)^{-1} \ln n - \varepsilon)$, even if we require that the input graph is strongly connected. In this section, in contrast, we show that MCD has a polynomial time approximation algorithm whose factor is $O(\log n)$, which is applicable for general graphs. This implies that we clarify an optimal approximability / inapproximability bound in terms of order under the assumption $P \neq NP$.

The idea of $O(\log n)$-approximation algorithm is based on formulating MCD as a *submodular set cover problem* [12]: Let us consider a finite set N, a nonnegative cost function c_j associated with each element $j \in N$, and non-decreasing submodular function $f : 2^N \mapsto Z^+$. A function f is called *non-decreasing* if $f(S) \leq f(T)$ for $S \subseteq T \subseteq N$, and is called *submodular* if $f(S) + f(T) \geq f(S \cap T) + f(S \cup T)$ for $S, T \subseteq N$. For a subset $S \subseteq N$, the cost of S, say $c(S)$, is $\sum_{j \in S} c_j$.

By these f, c and N, the submodular set cover problem is formulated as follows: [**Minimum Submodular Set Cover (SSC)**]

$$\min \left\{ \sum_{j \in S} c_j : f(S) = f(N) \right\}.$$

It is known that the greedy algorithm of SSC has approximation ratio $H(\max_{j \in N} f(j))$ where $H(i)$ is the i-th harmonic number if f is integer-valued and $f(\emptyset) = 0$ [12]. Note that $H(i) < \ln i + 1$.

We here claim that our problem can be cast as a submodular set cover problem. Let $N = \bigcup_{u \in V} \{x_{e,u} \mid e \in E\}$. Intuitively, $x_{e,u} \in S \subseteq N$ represents

that the local dispersal of u contains $e \in E$ in S, i.e., $e \in D_u$. For $S \subseteq N$, we define $d_S(u,v)$ as the distance from u to v under the setting that each edge $e \in D_u \cup D_v$ of S has length 0 otherwise 1. That is, if all edges are included in $D_u \cup D_v$ of S, then $d_S(u,v) = 0$. If no edge is included in $D_u \cup D_v$ of S, then $d_S(u,v)$ is the length of a shortest path from u to v of G. Let $f(S) = \sum_{(u,v)\in R}(d_\emptyset(u,v) - d_S(u,v))$. This f is integer-valued and $f(\emptyset) = 0$. In the problem setting of MCD, we can assume that for any $(u,v) \in R$, G has a (directed) path from u to v. (Otherwise, we have no solution). Then the condition $f(N) = f(S)$ means that all the requests are satisfied. Also cost c reflects the cost of MCD.

Then we have the following lemma:

Lemma 7. *Function f defined as above is a non-decreasing submodular function.*

Proof. Since it is obvious that f is non-decreasing, we only show the submodularity of f. By the inductive property, it is sufficient to show that $f(S \cup \{x_{e,u}\}) + f(S \cup \{x_{e',v}\}) \geq f(S) + f(S \cup \{x_{e,u}, x_{e',v}\})$.

$$\begin{aligned} f(S \cup \{x_{e,u}\}) - f(S) &= \textstyle\sum_{(i,j)\in R}(d_S(i,j) - d_{S\cup\{x_{e,u}\}}(i,j)) \\ &= \textstyle\sum_{(u,j)\in R}(d_S(u,j) - d_{S\cup\{x_{e,u}\}}(u,j)) \\ &+ \textstyle\sum_{(i,u)\in R}(d_S(i,u) - d_{S\cup\{x_{e,u}\}}(i,u)) \end{aligned} \tag{1}$$

$$\begin{aligned} f(S \cup \{x_{e',v}\}) - f(S \cup \{x_{e,u}, x_{e',v}\}) &= \textstyle\sum_{(i,j)\in R}(d_{S\cup\{x_{e,u},x_{e',v}\}}(i,j) \\ &- d_{S\cup\{x_{e',v}\}}(i,j)) \\ &= \textstyle\sum_{\substack{(u,j)\in R \\ j\neq v}}(d_{S\cup\{x_{e,u}\}}(u,j) - d_S(u,j)) \\ &+ \textstyle\sum_{\substack{(i,u)\in R \\ i\neq v}}(d_{S\cup\{x_{e,u}\}}(i,u) - d_S(i,u)) \\ &+ d_{S\cup\{x_{e,u},x_{e',v}\}}(v,u) - d_{S\cup\{x_{e',v}\}}(v,u) \\ &+ d_{S\cup\{x_{e,u},x_{e',v}\}}(u,v) - d_{S\cup\{x_{e',v}\}}(u,v) \end{aligned} \tag{2}$$

By summing (1) and (2) up, we obtain $f(S \cup \{x_{e,u}\}) + f(S \cup \{x_{e',v}\}) - (f(S) + f(S \cup \{x_{e,u}, x_{e',v}\})) = \sum_{(i,j)=(u,v),(v,u)}(d_{S\cup\{x_{e,u},x_{e',v}\}}(i,j) - d_{S\cup\{x_{e',v}\}}(i,j) - d_{S\cup\{x_{e,u}\}}(i,j) + d_S(i,j))$. Since d_S's are defined by shortest path lengths, we can see that $d_S(u,v) - 2 \leq d_{S\cup\{x_{e,u},x_{e',v}\}}(u,v) \leq d_S(u,v)$ and $d_S(u,v) - 1 \leq d_{S\cup\{x_{e,u}\}}(u,v), d_{S\cup\{x_{e',v}\}}(u,v) \leq d_S(u,v)$. If $d_{S\cup\{x_{e,u},x_{e',v}\}}(u,v) = d_S(u,v) - 2$, then both $d_{S\cup\{x_{e,u}\}}(u,v)$ and $d_{S\cup\{x_{e,u}\}}(u,v)$ are $d_S(u,v)-1$. Also, if $d_{S\cup\{x_{e,u},x_{e',v}\}}(u,v) = d_S(u,v) - 1$, then $d_{S\cup\{x_{e,u}\}}(u,v)$ or $d_{S\cup\{x_{e',v}\}}(u,v)$ is $d_S(u,v) - 1$. In any case, we have $d_{S\cup\{x_{e,u},x_{e',v}\}}(u,v) - d_{S\cup\{x_{e',v}\}}(u,v) - d_{S\cup\{x_{e,u}\}}(u,v) + d_S(u,v) \geq 0$. Since we similarly have $d_{S\cup\{x_{e,u},x_{e',v}\}}(v,u) - d_{S\cup\{x_{e',v}\}}(v,u) - d_{S\cup\{x_{e,u}\}}(v,u) + d_S(v,u) \geq 0$, $f(S \cup \{x_{e,u}\}) + f(S \cup \{x_{e',v}\}) \geq f(S) + f(S \cup \{x_{e,u}, x_{e',v}\})$ holds. □

Notice that f can be computed in polynomial time.

By these, MCD is formulated as a submodular set cover problem. Since we have $\max_{x_{e,u}\in N} f(\{x_{e,u}\}) \leq |R| \max_{u,v} d_\emptyset(u,v) \leq n^3$, the approximation ratio of the greedy algorithm is $O(\log n)$. We obtain the following.

Theorem 8. *There is a polynomial time algorithm with approximation factor* $O(\log n)$ *for MCD.*

5. Approximation Algorithm for Subset-Full

Zheng et al. have proposed a polynomial-time algorithm for MCD, called MinPivot, which achieves approximation ratio 2 for strongly connected graphs when a set R of requests is full. In this section, we show that even when R is subset-full, MinPivot achieves approximation ratio 2 for strongly connected graphs. Moreover, we show that MinPivot is a 3/2-approximation algorithm for MCD of the undirected variant with subset-full requests.

5.1. Algorithm MinPivot

A pseudo-code of the algorithm MinPivot is shown in Algorithm1 [1]. For the explanation of the algorithm, we define $\mathcal{P}(u, v)$ as the minimum-cardinality set of edges that constitute a round-trip path between u and v on G.

In a dispersal returned by MinPivot , one vertex is selected as a *pivot*. Each request is satisfied by a path via the selected pivot. The algorithm works as follows: It picks up a vertex u as a candidate of the pivot. Then, for vertices v, w in each request $(v, w) \in R$, MinPivot stores a round-trip path between v and the pivot u in D_v such that the sum of edges included in the round-trip path is minimum. For the vertex w, the round-trip path between w and the pivot u is also stored in D_w in the same way. Since there is a path from v to w via the pivot u in $D_v \cup D_w$ for each request (v, w), the dispersal satisfies R. For every pivot candidate, the algorithm MinPivot computes the corresponding dispersal as stated above. Finally, the minimum-cost one among all computed dispersals is chosen and returned.

In [7], the following theorem is proved.

Theorem 9. *For a strongly connected graph* G*,* MinPivot *is a 2-approximation algorithm for MCD on* G *with the full request. It completes in* $O(n^7)$ *time* [2] *for a strongly connected graph and in* $O(nm)$ *time for an undirected graph.*

5.2. Proof of 2-approximation for Strongly Connected Graphs

In this subsection, we prove the following theorem.

Theorem 10. *For a strongly connected graph* G *and a subset-full request* R*,* MinPivot *is a 2-approximation algorithm.*

[1] Although the original MinPivot is designed to work for any set of requests, we here show a simplified one because we focus on the case when R is subset-full.

[2] Since for directed graphs, $|\mathcal{P}(u, v)| \leq d(u, v) + d(v, u)$ holds in general, it is insufficient to simply compute the shortest paths.

Algorithm 1 MinPivot $(G = (V, E), R)$

```
1: V_R := {v, w ∈ V | (v, w) ∈ R}
2: for all u ∈ V do
3:    for all v ∈ V do
4:       if v ∈ V_R then
5:          D_v := P(u, v)
6:       else
7:          D_v := ∅
8:       end if
9:    end for
10:   D(u) := {D_v | v ∈ V}
11: end for
12: output D(u) such that c(D(u)) = min_{u∈V}{c(D(u))}.
```

We first introduce several notations used in the proof: The set of vertices included in requests in R is denoted by V_R, that is, $V_R = \{u, v \mid (u, v) \in R\}$. Let x be a vertex in V_R with the minimum local dispersal in D^{Opt} (i.e., $|D_x^{Opt}| = \min\{|D_v^{Opt}| \mid v \in V_R\}$). When there is more than one vertex with the minimum local dispersal, x is defined as one of them chosen arbitrarily. In the following argument, it is sufficient to consider only the case of $|D_x^{Opt}| > 0$: If $|D_x^{Opt}|$ is zero, any vertex in V_R must have two paths from/to x in its local dispersal to satisfy the requests for x. Then, the optimal solution is equivalent to that computed by MinPivot whose pivot candidate is x, which implies that MinPivot returns an optimal solution. Let D^{MP} denote an output of the algorithm MinPivot. The following proposition clearly holds.

Proposition 11. *For a dispersal D, if there exists a vertex u such that the local dispersal D_v of any vertex v in V_R contains a round-trip path between v and u, then $c(D^{MP}) \leq c(D)$.*

The idea of the proof is that we construct a feasible dispersal D with cost at most $2 \cdot c(D^{Opt})$, which satisfies the condition shown in Proposition 11. It follows that the cost of the solution by MinPivot is bounded by $2 \cdot c(D^{Opt})$. We construct the dispersal D from D^{Opt} by additionally giving the minimum-size local dispersal to all vertices in V_R. More precisely, the local dispersal D_v of every vertex $v \in V_R$ is the union of D_v^{Opt} and D_x^{Opt} (i.e., $D_v = D_v^{Opt} \cup D_x^{Opt}$).

Theorem 10 is easily proved from the following lemma and Proposition 11.

Lemma 12. *In the dispersal D constructed in the above way, every vertex v in V_R has a round-trip path between v and x in its local dispersal D_v. In addition, $c(D) \leq 2 \cdot c(D^{Opt})$ is satisfied.*

Proof. Every local dispersal D_v contains paths from v to x and from x to v since $D_v^{Opt} \cup D_x^{Opt}$ contains the paths to satisfy the requests (x, v) and (v, x). From the construction of the dispersal D, we obtain $c(D) \leq c(D^{Opt}) + |D_x^{Opt}| \cdot$

$|V_R|$. Now, the size of the local dispersal D_x^{Opt} is the minimum of all local dispersals in D^{Opt}, and the local dispersal of the vertex not included in V_R is empty in D^{Opt}. Therefore, we obtain $|D_x^{Opt}| \cdot |V_R| \leq c(D^{Opt})$. It implies that $c(D) \leq 2 \cdot c(D^{Opt})$. □

5.3. Proof of 3/2-approximation for Undirected Graphs

In this subsection, we prove that the approximation ratio of MinPivot is improved for MCD of the undirected variant. That is, we prove the following theorem.

Theorem 13. *For an undirected graph G and a subset-full request R,* MinPivot *is a 3/2- approximation algorithm.*

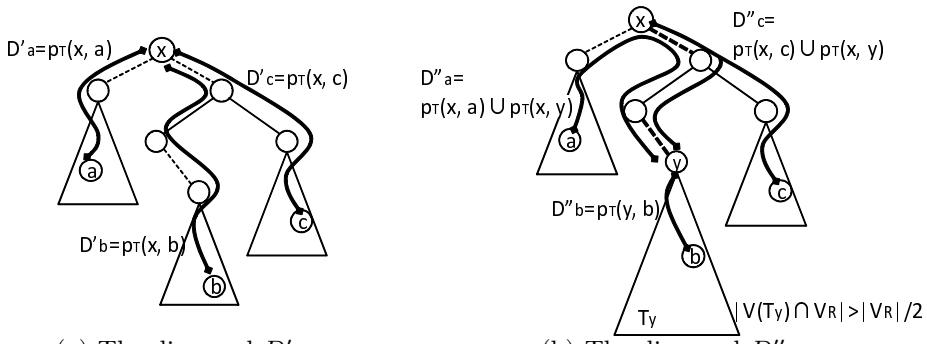
In the proof, we take the same approach as the one of Theorem 10: We construct a dispersal D with cost at most $\frac{3}{2} \cdot c(D^{Opt})$, which satisfies the condition in Proposition 11. Since Proposition 11 also clearly holds in undirected graphs, it follows that the cost of the solution by MinPivot is bounded by $\frac{3}{2} \cdot c(D^{Opt})$. In the proof of Theorem 10, we show that when all the edges in D_x^{Opt} are added to the local dispersal of every vertex in V_R, the cost of the dispersal D is at most twice as much as that of the optimal dispersal. Our proof of Theorem 13 is based on the idea that we construct a dispersal D by adding each edge in D_x^{Opt} to at most $|V_R|/2$ local dispersals.

In what follows, we show the construction of D. We define a rooted tree T from an optimal dispersal D^{Opt}. To define T, we first assign a *weight* to each edge: To any edge in D_x^{Opt}, the weight zero is assigned. All the other edges are assigned the weight one. A rooted tree $T = (V, E_T)$ $(E_T \subseteq E)$ is defined as a shortest path tree with root x (in terms of weighted graphs) that spans all the vertices in V_R. Let $p_T(u, v)$ be the shortest path from a vertex u to v on the tree T. The weight of a path $p(u, v)$ is defined by the total weight of the edges on the path and denoted by $w(p(u, v))$. For each vertex v, let $p_T(v, v) = \emptyset$ and $w(p_T(v, v)) = 0$.

Lemma 14. *On $T = (V, E_T)$ for an optimal dispersal D^{Opt}, $\sum_{v \in V_R} w(p_T(x, v)) < c(D^{Opt})$.*

Proof. For the vertex x, $w(p_T(x, x)) < |D_x^{Opt}|$ clearly holds, since $|D_x^{Opt}| > 0$. For any other vertex v in V_R, the set R of requests necessarily includes (x, v) (remind that R is subset-full). To satisfy (x, v), in the optimal dispersal, $D_x^{Opt} \cup D_v^{Opt}$ includes a path $p(x, v)$, and thus, $p(x, v) \setminus D_x^{Opt} \subseteq D_v^{Opt}$. This implies $|p(x, v) \setminus D_x^{Opt}| \leq |D_v^{Opt}|$. Since any edge in D_x^{Opt} has weight zero and all other edges have weight one, the weight of $p(x, v)$ is equal to $|p(x, v) \setminus D_x^{Opt}|$. From the definition of $p_T(x, v)$, we obtain $w(p_T(x, v)) \leq w(p(x, v)) \leq |D_v^{Opt}|$.

In an optimal dispersal D^{Opt}, the local dispersal D_v^{Opt} of each vertex v in $V \setminus V_R$ has no edges since there is no request for v in R. Thus, it follows $\sum_{v \in V_R} w(p_T(x, v)) < \sum_{v \in V_R} |D_v^{Opt}| = c(D^{Opt})$. □



(a) The dispersal D' (b) The dispersal D''

Figure 3: Examples of the proposed dispersals. The dotted edges represent edges included in D_x^{Opt} and the heavy dotted edges represent edges included in $\hat{D}_x^{Opt}$

For each edge e in D_x^{Opt}, let $C(e)$ be the number of vertices from which path to the vertex x on T includes the edge e: $C(e) = |\{v \in V_R \mid e \in p_T(x, v)\}|$. The construction of the desired dispersal depends on whether any edge e in D_x^{Opt} satisfies $C(e) \leq |V_R|/2$ or not.

In the case that $C(e) \leq |V_R|/2$ holds for any edge e in D_x^{Opt}, the dispersal D' is constructed in the following way: $D' = \{D'_v | v \in V\}$, where

- for the vertex v in V_R, $D'_v = p_T(x, v)$,
- for the vertex v in $V \setminus V_R$, $D'_v = \emptyset$.

Figure 3(a) shows one example of the dispersal D'. In the figure, the dotted edges represent edges included in D_x^{Opt} and the thick curves represent the local dispersal of each vertex.

Lemma 15. $c(D^{MP}) \leq c(D') \leq \frac{3}{2} \cdot c(D^{Opt})$

Proof. From the definitions of T and $C(e)$, we obtain $|p_T(x, v)| = w(p_T(x, v)) + |p_T(x, v) \cap D_x^{Opt}|$ and $\sum_{v \in V_R} |p_T(x, v) \cap D_x^{Opt}| = \sum_{e \in D_x^{Opt}} C(e)$. Thus, $c(D') = \sum_{v \in V_R} w(p_T(x, v)) + \sum_{e \in D_x^{Opt}} C(e)$. From Lemma 14 and the assumption that $C(e) \leq |V_R|/2$, it follows that $c(D') \leq c(D^{Opt}) + |D_x^{Opt}| \cdot \frac{|V_R|}{2}$. Since $|D_x^{Opt}| \cdot |V_R| \leq c(D^{Opt})$ holds, we obtain $c(D') \leq \frac{3}{2} \cdot c(D^{Opt})$. The local dispersal D'_v of v in V_R includes a path from x to v, thus, $c(D^{MP}) \leq c(D')$ holds by Proposition 11. □

We consider the case that there is an edge such that $C(e) > |V_R|/2$. Let T_v be the subtree of T induced by vertex v and all of v's descendants, and $V(T_v)$ be the set of vertices in T_v. The set of edges in D_x^{Opt} such that $C(e) > |V_R|/2$ is denoted by $\hat{D}_x^{Opt}$. Let y be the vertex farthest from x of those adjacent to some edge in $\hat{D}_x^{Opt}$.

Lemma 16. *All edges in $\hat{D}_x^{Opt}$ are on the path $p_T(x, y)$.*

PROOF. If a path $p_T(x, w)$ from x to a vertex $w \in V_R$ contains an edge (u, v), then vertex w is a descendant of u and v. That is, $w \in V(T_v) \cap V_R$ holds. Thus, from the definition of $C(e)$, we have $C((u, v)) = |V(T_v) \cap V_R|$ for each edge $(u, v) \in D_x^{Opt}$ where u is the parent of v. Therefore, the edge (u, v) satisfies $C((u, v)) > |V_R|/2$ iff $|V(T_v) \cap V_R| > |V_R|/2$.

We prove the lemma by contradiction. Suppose for contradiction that there is an edge (u, v) such that $(u, v) \in \hat{D}_x^{Opt}$ and $(u, v) \notin p_T(x, y)$. Let v be a child of u on T. From $(u, v) \notin p_T(x, y)$, it follows that vertex v is not an ancestor of the vertex y on T. Since vertex y is the farthest vertex from x, from which the edge to its parent is contained in $\hat{D}_x^{Opt}$, vertex v is not a descendant of y. Thus, we obtain $V(T_v) \cap V(T_y) = \emptyset$. In addition, $C((u, v)) = |V(T_v) \cap V_R| > |V_R|/2$ holds. From $V(T_v) \cap V(T_y) = \emptyset$ and $|V(T_v) \cap V_R| > |V_R|/2$, we obtain $|V(T_y) \cap V_R| \leq |V_R|/2$. It contradicts the definition of the vertex y. □

In the case that there is an edge such that $C(e) > |V_R|/2$, a dispersal D'' is constructed so that every vertex in V_R has the path from itself to vertex y on T: $D'' = \{D''_v | v \in V\}$, where

- for the vertex v in $V_R \cap V(T_y)$, $D''_v = p_T(y, v)$,
- for the vertex v in $V_R \setminus V(T_y)$, $D''_v = p_T(x, v) \cup p_T(x, y)$,
- for the vertex v in $V \setminus V_R$, $D''_v = \emptyset$.

Figure 3(b) shows one example of the dispersal D''. The heavy dotted edges represent edges included in $\hat{D}_x^{Opt}$. We can see that local dispersal of each vertex contains a path from itself to the vertex y.

Lemma 17. $c(D^{MP}) \leq c(D'') \leq \frac{3}{2} \cdot c(D^{Opt})$

PROOF. From the definition of the dispersal D'', we obtain $c(D'') \leq \sum_{v \in V_R \cap V(T_y)} |p_T(y, v)| + \sum_{v \in V_R \setminus V(T_y)} (|p_T(x, v)| + |p_T(x, y)|)$. Lemma 16 implies that the edge in $\hat{D}_x^{Opt}$ is contained by only vertices in $V_R \setminus V(T_y)$. Moreover, it implies that for each edge $e \in D_x^{Opt}$ that is not on $p_T(x, y)$, $e \in D_x^{Opt} \setminus \hat{D}_x^{Opt}$ and $C(e) \leq |V_R|/2$ hold. Since $|V_R \setminus V(T_y)| \leq |V_R|/2 < |V_R \cap V(T_y)|$, the following inequalities can be obtained in the same way as the proof of Lemma 15:

$$
\begin{aligned}
c(D'') &\leq \sum_{v \in V_R \cap V(T_y)} |p_T(y, v)| + \sum_{v \in V_R \setminus V(T_y)} (|p_T(x, v)| + |p_T(x, y)|) \\
&\leq \sum_{v \in V_R \cap V(T_y)} w(p_T(y, v)) + \sum_{v \in V_R \setminus V(T_y)} (w(p_T(x, v)) + w(p_T(x, y))) \\
&\quad + \sum_{e \in D_x^{Opt} \setminus \hat{D}_x^{Opt}} C(e) + \sum_{e \in \hat{D}_x^{Opt}} |V_R \setminus V(T_y)| \\
&\leq \sum_{v \in V_R \cap V(T_y)} w(p_T(y, v)) + |V_R \setminus V(T_y)| \cdot w(p_T(x, y))
\end{aligned}
$$

$$
\begin{aligned}
& + \sum_{v \in V_R \setminus V(T_y)} w(p_T(x,v)) + \frac{|V_R|}{2} \cdot |D_x^{Opt} \setminus \hat{D}_x^{Opt}| + \frac{|V_R|}{2} \cdot |\hat{D}_x^{Opt}| \\
\leq & \sum_{v \in V_R \cap V(T_y)} (w(p_T(y,v)) + w(p_T(x,y))) + \sum_{v \in V_R \setminus V(T_y)} w(p_T(x,v)) \\
& + \frac{|V_R|}{2} \cdot |D_x^{Opt}| \\
= & \sum_{v \in V_R} w(p_T(x,v)) + \frac{1}{2} \cdot c(D^{Opt}) \leq \frac{3}{2} \cdot c(D^{Opt})
\end{aligned}
$$

Since the local dispersal D''_v of every vertex v in V_R includes a path from v to y, $c(D^{MP}) \leq c(D'')$ holds by Proposition 11. □

From Lemmas 15 and 17, Theorem 13 is proved.

6. Concluding remarks

In this paper, we investigate the (in)approximability of MCD from a perspective of how topological structures of R affect the complexity of MCD. While the approximability bound of MCD for a general setting of R is evaluated as $\Theta(\log n)$ under the assumption $P \neq NP$, MCD for Subset-Full is 2-approximable though it is still inapproximable within a small constant factor unless $P = NP$. Moreover, in the undirected version of MCD, MCD for Subset-Full is 3/2-approximable.

The complexity of MCD for Full, which is a special case of Subset-Full, is still open. We have shown that MCD for Subset-Full is NP-hard, but it does not imply the hardness of MCD for Full. Recall that the Minimum Steiner Tree problem is NP-hard whereas the Minimum Spanning Tree has a polynomial time algorithm [13]. Since the relationship between the Minimum Steiner Tree and the Minimum Spanning Tree is similar to the one between MCD for Subset-Full and MCD for Full, it is not strange that MCD for Full is to be polynomially solvable. We actually conjecture that MinPivot returns an optimal solution for MCD with Full; if it is correct, we will obtain an interesting contrast similar to the relation between Minimum Steiner Tree and Minimum Spanning Tree.

Another open issue is the consideration of fault-tolerant property for MCD problem, which can be defined as the problem of establishing multipath connection between sources and destinations. This problem can be related to the minimum k-connected spanning subgraph problem, and several approaches can be imported from its previous literature[16].

Acknowledgments

This work is supported in part by KAKENHI no. 19700058, 21500013 and 21680001, Asahi-glass Foundation, Inamori Foundation and Hori Information Science Promotion Foundation.

References

[1] J. Hubaux, L. Buttyan, S. Capkun, The Quest for Security in Mobile Ad Hoc Networks, in: Proc. the 2nd ACM international symposium on Mobile ad hoc networking and computing, Mobihoc 2001, 2001, pp. 146-155.

[2] S. Capkun, L. Buttyan, J. P. Hubaux, Self-Organized Public-Key Management for Mobile Ad Hoc Networks, IEEE Transactions on Mobile Computing, 2 (1) (2003) 52-64.

[3] M. G. Gouda, E. Jung, Certificate Dispersal in Ad-Hoc Networks, in: Proc. the 24th International Conference on Distributed Computing Systems, ICDCS 2004, 2004, pp. 616-623.

[4] E. Jung, E. S. Elmallah, M. G. Gouda, Optimal dispersal of certificate chains, in: Proc. the 18th International Symposium on Distributed Computing, DISC 2004, 2004, pp. 435-449.

[5] M. G. Gouda, E. Jung, Stabilizing Certificate Dispersal, in: Proc. the 7th International Symposium on Self-Stabilizing Systems, SSS 2005, 2005, pp. 140-152.

[6] H. Zheng, S. Omura, J. Uchida, K. Wada, An Optimal Certificate Dispersal Algorithm for Mobile Ad Hoc Networks, IEICE Transactions on Fundamentals, E88-A (5) (2005) 1258-1266.

[7] H. Zheng, S. Omura, K. Wada, An Approximation Algorithm for Minimum Certificate Dispersal Problems, IEICE Transactions on Fundamentals, E89-A (2) (2006) 551-558.

[8] P. Hell, J. Nesetril, On the Complexity of H-Coloring, Combinatorial Theory Series B, 48 (1) (1990) 92-110.

[9] N. Alon, D. Moshkovitz, S. Safra, Algorithmic Construction of Sets for k-Restrictions, ACM Transactions on Algorithms, 2 (2) (2006) 153-177.

[10] S. Arora, C. Lund, Hardness of Approximation, in: D.S. Hochbaum (Ed.), Approximation Algorithms for NP-hard problems, PWS publishing company, 1995, pp. 399-446.

[11] M. Chlebík, J. Chlebíková, Complexity of approximating bounded variants of optimization problems, Theoretical Computer Science, 354(3) (2006) 320-338.

[12] L. A. Wolsey, An Analysis of the Greedy Algorithm for the Submodular Set Covering Problem, Combinatorica, 2 (4) (1982) 385-393.

[13] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

[14] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, B. Yener, Provisioning a virtual private network: a network design problem for multicommodity flow, in: Proc. ACM Symposium on Theory of Computing, 2001, pp. 389-398.

[15] N. Goyal, N. Olver, B. Shepherd, The VPN conjecture is true, in: Proc. ACM Symposium on Theory of Computing, 2009, pp. 443-450.

[16] J. Cheriyan, and R. Thurimella, Approximating Minimum-Size k-Connected Spanning Subgraphs via Matching, SIAM Journal on Computing, 30 (1998), 292-301.