Augmenting Graphs to Minimize the Diameter

Fabrizio Frati[1], Serge Gaspers[2,3],
Joachim Gudmundsson[1,3], and Luke Mathieson[4]

[1] University of Sydney, Australia,
{fabrizio.frati,joachim.gudmundsson}@sydney.edu.au
[2] The University of New South Wales, Australia, sergeg@cse.unsw.edu.au
[3] NICTA, Australia
[4] Macquarie University, Australia, luke.mathieson@mq.edu.au

Abstract. We study the problem of augmenting a weighted graph by inserting edges of bounded total cost while minimizing the diameter of the augmented graph. Our main result is an FPT 4-approximation algorithm for the problem.

1 Introduction

We study the problem of minimizing the diameter of a weighted graph by the insertion of edges of bounded total cost. This problem arises in practical applications [2, 4] such as telecommunications networks, information networks, flight scheduling, protein interactions, and it has also received considerable attention from the graph theory community, see for example [1, 7, 13].

We introduce some terminology. Let $G = (V, E)$ be an undirected weighted graph. Let $[V]^2$ be the set of all possible edges on the vertex set V. A *non-edge* of G is an element of $[V]^2 \setminus E$. The *weight* of a path in G is the sum of its edge weights. For any $u, v \in V$, the *shortest* u-v *path* in G is the path connecting u and v in G with minimum weight. The weight of this path is said to be the *distance* between u and v in G. Finally, the *diameter* of G is the largest distance between any two vertices in G. The problem we study in this paper is formally defined as follows (denote by $\mathbb{N}$ the set of natural numbers including 0 and by $\mathbb{N}^*$ the set of natural numbers excluding 0).

PROBLEM: Bounded Cost Minimum Diameter Edge Addition (BCMD)
INPUT: An undirected graph $G = (V, E)$, a weight function $w : [V]^2 \to \mathbb{N}$, a cost function $c : [V]^2 \to \mathbb{N}^*$, and an integer B.
GOAL: A set F of non-edges with $\sum_{e \in F} c(e) \leq B$ such that the diameter of the graph $G_B = (V, E \cup F)$ with weight function w is minimized. We say that G_B is a B-*augmentation* of G.

The main result of this paper is a fixed parameter tractable (FPT) 4-approximation algorithm for BCMD with parameter B. FPT approximation algorithms are surveyed by Marx [16]. For background on parameterized complexity we refer to [6, 8, 17] and for background on approximation algorithms to [19].

Several papers in the literature already dealt with the BCMD problem. However, most of them focused on restricted versions of the problem, namely the one in which all costs and all weights are identical [3, 5, 14, 15], and the one in which all the edges have unit costs and the weights of the non-edges are all identical [2, 4].

The BCMD problem can be seen as a bicriteria optimization problem where the two optimization criteria are: (1) the cost of the edges added to the graph and (2) the diameter of the augmented graph. As is standard in the literature, we say that an algorithm is an (α, β)-approximation algorithm for the BCMD problem, with $\alpha, \beta \geq 1$, if it computes a set F of non-edges of G of total cost at most $\alpha \cdot B$ such that the diameter of $G' = (V, E \cup F)$ is at most $\beta \cdot D^B_{opt}$, where D^B_{opt} is the diameter of an optimal B-augmentation of G.

We survey some known results about the BCMD problem. Note that all the algorithms discussed below run in polynomial time.

Unit weights and unit costs. The restriction of BCMD to unit costs and unit weights was first shown to be NP-hard in 1987 by Schoone et al. [18]; see also the paper by Li et al. [15]. Bilò et al. [2] showed that, as a consequence of the results in [3, 5, 15], there exists no $(c \log n, \delta < 1 + 1/D^B_{opt})$-approximation algorithm for BCMD if $D^B_{opt} \geq 2$, unless P=NP. For the case in which $D^B_{opt} \geq 6$, they proved a stronger lower bound, namely that there exists no $(c \log n, \delta < \frac{5}{3} - \frac{7-(D^B_{opt}+1) \mod 3}{3D^B_{opt}})$-approximation algorithm, unless P=NP.

Dodis and Khanna [5] gave an $(O(\log n), 2{+}2/D^B_{opt})$-approximation algorithm (see also [14]). Li et al. [15] showed a $(1, 4 + 2/D^B_{opt})$-approximation algorithm. The analysis of the latter algorithm was later improved by Bilò et al. [2], who showed that it gives a $(1, 2 + 2/D^B_{opt})$-approximation. In the same paper they also gave an $(O(\log n), 1)$-approximation algorithm.

Unit costs and restricted weights. Some of the results from the unweighted setting have been extended to a restricted version of the weighted case, namely the one in which the edges of G have arbitrary non-negative integer weights, however all the non-edges of G have cost 1 and uniform weight $\omega \geq 0$.

Bilò et al. [2] showed how two of their algorithms can be adapted to this restricted weighted case. In fact, they gave a $(1, 2 + 2\omega/D^B_{opt})$-approximation algorithm and a $(2 - 1/B, 2)$-approximation algorithm. Similar results were obtained by Demaine and Zadimoghaddam in [4].

Bilò et al. [2] also showed that, for every $D^B_{opt} \geq 2\omega$ and for some constant c, there is no $(c \log n, \delta < 2 - 3\omega/D^B_{opt})$-approximation algorithm for this restriction of the BCMD problem, unless P=NP.

Arbitrary costs and weights. To the best of our knowledge, there is only one theory paper that has considered the general BCMD problem. In 1999, Dodis and Khanna [5] presented an $O(n \log D^B_{opt}, 1)$-approximation algorithm, assuming that all weights are polynomially bounded. Their result is based on a multicommodity flow formulation of the problem.

Our results. In this paper we study the BCMD problem with arbitrary integer costs and weights. Our main result is a $(1, 4)$-approximation algorithm with running time $O((3^B B^3 + n + \log(Bn))Bn^2)$. We also prove that, considering B as a parameter, it is $W[2]$-hard to compute a $(1 + c/B, 3/2 - \epsilon)$-approximation, for any constants c and

$\epsilon > 0$. Further, we present polynomial-time $((B+1)^2, 3)$-, $(B, 4)$-, and $(1, 3B+2)$-approximation algorithms for the unit-cost restriction of the BCMD problem.

2 Shortest Paths with Bounded Cost

Let $(G = (V, E), w, c, B)$ be an instance of the BCMD problem and let K denote the complete graph on the vertex set V. The edges of K have the same weights and costs as they have in G (observe that an edge e of K is either an edge or a non-edge of G).

For any $0 \leq \beta \leq B$, a path in K is said to be a *β-bounded-cost path* if it uses non-edges of G of total cost at most β. We consider the problem of computing, for *every* integer $0 \leq \beta \leq B$ and for every two vertices $u, v \in V$, a β-bounded-cost shortest path connecting u and v, if such a path exists. We call this problem the *All-Pairs B-Shortest Paths* (APSP$_B$) problem. We will prove the following.

Theorem 1. *The APSP$_B$ problem can be solved in $O(Bn^3 + Bn^2 \log(Bn))$ time using $O(Bn^2)$ space.*

In order to prove Theorem 1, we construct a directed graph $H = (U, F)$ as follows. First, consider G as a directed graph, i.e., replace every undirected edge $\{u, v\}$ with two arcs (u, v) and (v, u) with the same weight and cost as the edge $\{u, v\}$. Then, $H = (U, F)$ contains $B + 1$ copies of G, denoted by $G_0, \ldots, G_B$. For any $0 \leq i \leq B$, we denote by (v, i) the copy of vertex $v \in V$ in $G_i = (V_i, E_i)$. The arc set F contains the union of E', F', and M', where

$$E' = \bigcup_{0 \leq i \leq k} E_i,$$
$$F' = \Big\{ \big((u,i),(v, i + c(\{u, v\}))\big) \;:\; 0 \leq i \leq B - c(\{u, v\}),\; \{u, v\} \in [V]^2 \setminus E \Big\}, \text{ and}$$
$$M' = \Big\{ \big((u, i), (u, i + 1)\big) \;:\; 0 \leq i \leq B - 1,\; u \in V \Big\}.$$

For each $((u, i), (v, j)) \in F'$, the weight and the cost of $((u, i), (v, j))$ are $w(\{u, v\})$ and $c(\{u, v\}) = j - i$, respectively. For each $((u, i), (u, i + 1)) \in M'$, the weight and the cost of $((u, i), (u, i + 1))$ are 0 and 1, respectively.

Observation 1 *The number of vertices in U is $(B + 1)n$ and the number of arcs in F is $O(Bn^2)$.*

We will use directed graph H to efficiently compute β-bounded-cost shortest paths in K. This is possible due to the following two lemmata.

Lemma 1. *Suppose that there exists a β-bounded-cost path P_K in K with weight W connecting vertices u and v. Then, there exists a directed path P_H in H with weight W connecting vertices $(u, 0)$ and (v, β).*

Proof. Consider a path $P_K = \langle u = v_1, v_2, \ldots, v = v_m \rangle$ in K with weight W. Set $(v_1, 0)$ to be the first vertex of P_H. Suppose that path P_H has been defined until a vertex (v_h, j), corresponding to vertex v_h of P_K, for some $1 \leq h < m$. If edge (v_h, v_{h+1}) of P_K is an edge of G, then let (v_{h+1}, j) be the vertex corresponding to v_{h+1}. If edge

(v_h, v_{h+1}) of P_K is a non-edge of G, then let $(v_{h+1}, j + c(\{v_h, v_{h+1}\}))$ be the vertex corresponding to v_{h+1}. This defines path P_H up to a vertex (v, β'). Assuming that $\beta' \leq \beta$, path P_H terminates with a set of edges with weight 0 connecting (v, j) and $(v, j + 1)$, for every $\beta' \leq j \leq \beta - 1$; these edges are in M', and hence in H, by construction. It remains to prove that $\beta' \leq \beta$ and that P_H has weight W. Every edge (x, y) of P_K that is an edge of G corresponds to an edge $((x, a), (y, a))$ of H with the same weight. Moreover, every edge (x, y) of P_K that is a non-edge of G corresponds to an edge $((x, a), (y, b))$ of H, with $c\{x, y\} = b - a$ and with the same weight. By definition, P_K uses non-edges of G of total cost at most β. Hence, $\beta' \leq \beta$; also, P_H has weight exactly W and the lemma follows. □

Lemma 2. *Let P_H be a shortest directed path connecting two vertices (u, i) and (v, j) of H, with $j \geq i$. Let W be the weight of P_H. Then, there exists a $(j - i)$-bounded-cost path P_K in K with weight W connecting u and v.*

Proof. First, we construct a path P'_H connecting (u, i) and (v, j) in H such that the weight of P'_H is W and, for each vertex w in K, all the vertices of the form $(w, \cdot)$ appear consecutively in P'_H. Indeed, P'_H can be obtained from P_H by repeatedly performing the following operation. Consider any two vertices (w, p) and (w, r) such that there exists a vertex (z, q) between (w, p) and (w, r) in P_H, with $z \neq w$. Then, replace the subpath $P_H(w)$ of P_H between (w, p) and (w, r) with path $P'_H(w) = \langle (w, p), (w, p + 1), \ldots, (w, r) \rangle$. Observe that $P'_H(w)$ has weight zero; since P_H is a shortest directed path in H connecting (u, i) and (v, j), it follows that $P_H(w)$ also has weight zero, hence the weight of P_H is not altered by the replacement.

Second, we define a path P_K in K as follows. For each maximal sequence of vertices of the form $(w, \cdot)$ in P'_H, path P_K contains vertex w. If P'_H contains two adjacent vertices (w, p) and (z, q) with $w \neq z$, then P_K contains edge (w, z). By construction, P_K connects u and v. Since all the vertices of the form $(w, \cdot)$ appear consecutively in P'_H, it follows that P_K is a path. For every edge (w, z) of P_K there is a distinct edge of P'_H with the same weight and cost. Since every other edge of P'_H has weight zero and cost one, it follows that P_K has weight W and cost at most $j - i$. This proves the lemma. □

We have the following.

Corollary 1. *There is a β-bounded-cost shortest path connecting vertices u and v in K with weight W if and only if there is a shortest directed path in H connecting vertices $(u, 0)$ and (v, β) with weight W.*

Proof. We prove the necessity. If there is a β-bounded-cost shortest path P_K connecting vertices u and v in K with weight W, then by Lemma 1 there is a directed path P_H in H connecting vertices $(u, 0)$ and (v, β) with weight W. Suppose, for a contradiction, that P_H is not a shortest directed path connecting $(u, 0)$ and (v, β). Then, there exists a shortest directed path P'_H in H connecting $(u, 0)$ and (v, β) with weight $W' < W$. By Lemma 2, there exists a β-bounded-cost path P'_K in K with weight W' connecting u and v,contradicting the fact that P_K is a β-bounded-cost shortest path connecting vertices u and v.

We prove the sufficiency. If there is a shortest directed path P_H in H connecting vertices $(u, 0)$ and (v, β) with weight W, then by Lemma 2 there exists a β-bounded-cost path P_K in K with weight W connecting u and v. Suppose, for a contradiction, that P_K is not a shortest path. Then, there exists a β-bounded-cost shortest path P'_K in K with weight $W' < W$ connecting u and v. By Lemma 1, there exists a directed path P'_H in H connecting vertices $(u, 0)$ and (v, β) with weight W', contradicting the fact that P_H is a shortest directed path connecting vertices $(u, 0)$ and (v, β). □

We are now ready to prove Theorem 1. Consider any vertex u in K. We first mark every vertex that can be reached from $(u, 0)$ in H with the weight of its shortest path from $(u, 0)$. By Observation 1, H has $O(Bn)$ vertices and $O(Bn^2)$ edges, hence this can be done in $O(Bn^2 + Bn \log(Bn))$ time [10]. For every $0 \leq \beta \leq B$ and for every vertex $v \neq u$, by Corollary 1 the weight of a β-bounded cost shortest path in K is the same as the weight of a shortest directed path from $(u, 0)$ to (v, β) in H. Hence, for every $0 \leq \beta \leq B$ and for every vertex $v \neq u$, we can determine in $O(Bn^2 + Bn \log(Bn))$ total time the weight of a β-bounded cost shortest path in K connecting u and v. Thus, for every $0 \leq \beta \leq B$ and for every pair of vertices u and v in K, we can determine in $O(Bn^3 + Bn^2 \log(Bn))$ total time the weight of a β-bounded cost shortest path in K connecting u and v. This concludes the proof of Theorem 1.

3 Arbitrary Costs and Weights

Our algorithms, as with many afore-mentioned approximation algorithms for the BCMD problem, use a clustering approach as a first phase to find a set C of $B+1$ cluster centers. The idea of the algorithm is to create a minimum height rooted tree T with vertex set U, where $C \subseteq U$, by adding a set of edges of total cost at most B to G. We will prove that such a tree approximates an optimal B-augmentation.

3.1 Clustering

We start by defining the clustering approach used to generate the $B + 1$ cluster centers. Whereas a costly binary search is used in [4] to guess the radius of the clusters, we adapt the approach of [2] to our more general setting.

For two vertices u, v, we denote by $\text{dist}_G(u, v)$ the distance between u and v in G. For a vertex u and a set of vertices S, we denote by $\text{dist}_G(u, S)$ the minimum distance between u and any vertex from S in G, i.e., $\text{dist}_G(u, S) = \min_{v \in S}\{\text{dist}_G(u, v)\}$. For a set of vertices S, we denote by $\text{dist}_G(S)$ the minimum distance between any two distinct vertices from S in G, i.e., $\text{dist}_G(S) = \min_{u \in S}\{\text{dist}_G(u, S \setminus \{u\})\}$.

The clustering phase computes a set $C = \{c_1, \ldots, c_{B+1}\}$ of $B + 1$ cluster centers as follows. Vertex c_1 is an arbitrary vertex in V; for $2 \leq i \leq B + 1$, vertex c_i is chosen so that $\text{dist}_G(c_i, \{c_1, \ldots, c_{i-1}\})$ is maximized. Ties are broken arbitrarily.

Lemma 3. *The clustering phase computes in $O(Bn^2)$ time a set $C \subseteq V$ of size $B + 1$ such that $\text{dist}_G(v, C) \leq D^B_{opt}$ for every vertex $v \in V$.*

Proof. First, note that the above described algorithm can easily be implemented in $O(Bn^2)$ time using B iterations of Dijkstra's algorithm with Fibonacci heaps [10]. Let c_{B+2} denote a vertex maximizing $\text{dist}_G(c_{B+2}, C)$, and denote this distance by R. By definition, $\mathrm{dist}_G(v, C) \leq R$ for every $v \in V$. To prove the lemma it remains to show that $R \leq D^B_{opt}$. For the sake of contradiction, assume $D^B_{opt} < R$. Then, $C \cup \{c_{B+2}\}$ is a set of $B + 2$ vertices with pairwise distance larger than D^B_{opt} in G. Namely, for every $2 \leq i \leq B + 2$, we have $\mathrm{dist}_G(c_i, \{c_1, \ldots, c_{i-1}\}) \geq \mathrm{dist}_G(c_{B+2}, \{c_1, \ldots, c_{i-1}\}) \geq \mathrm{dist}_G(c_{B+2}, C) = R > D^B_{opt}$. We prove the following claim.

Claim 1 *Let G' be a weighted graph and let C' be a set of vertices in G' such that $\mathrm{dist}_{G'}(C') > D$ and such that $|C'| \geq 3$. Then, for every graph G'' obtained from G' by adding a single non-edge of G' with non-negative weight, there is a set $C'' \subset C'$ with $|C''| = |C'| - 1$ and with $\mathrm{dist}_{G''}(C'') > D$.*

Proof. Let (u, v) denote the edge that is added to G' to obtain G''. For the sake of contradiction, assume that there is no vertex $w \in C'$ such that $\mathrm{dist}_{G''}(C' \setminus \{w\}) > D$. That is, every set $C'' \subset C'$ with $|C''| = |C'| - 1$ contains two vertices whose distance is at most D. Then, there are four vertices $w_1, w_2, w_3, w_4 \in C$ such that $\mathrm{dist}_{G''}(w_1, w_2) \leq D$ and $\mathrm{dist}_{G''}(w_3, w_4) \leq D$ (Case 1), or there are three vertices $w_1, w_2, w_3 \in C$ such that $\mathrm{dist}_{G''}(w_1, w_2) \leq D$, $\mathrm{dist}_{G''}(w_1, w_3) \leq D$, and $\mathrm{dist}_{G''}(w_2, w_3) \leq D$ (Case 2). Indeed, assume that we are neither in Case 1 nor in Case 2. Construct a graph A whose vertices are those in C' and such that there is an edge (w_i, w_j) if and only if $\mathrm{dist}_{G''}(w_i, w_j) \leq D$. Since we are not in Case 1, we have that A does not contain two non-adjacent edges, hence it is either a star plus an independent set or a 3-cycle plus an independent set. Since we are not in Case 2, it follows that A is a star plus an independent set. Hence, there is a vertex $w \in C'$ such that removing w and its incident edges from A turns A into an empty graph. Thus, $\mathrm{dist}_{G''}(C' \setminus \{w\}) > D$, a contradiction which proves that we are either in Case 1 or in Case 2.

Suppose that we are in Case 1. By assumption, we have that $\mathrm{dist}_{G''}(w_1, w_2) < \mathrm{dist}_{G'}(w_1, w_2)$ and $\mathrm{dist}_{G''}(w_3, w_4) < \mathrm{dist}_{G'}(w_3, w_4)$. Hence, (u, v) is an edge of any shortest path $P_{1,2}$ from w_1 to w_2 and of any shortest path $P_{3,4}$ from w_3 to w_4. Assume, without loss of generality, that u is encountered before v when traversing $P_{1,2}$ starting at w_1 and when traversing $P_{3,4}$ starting at w_3 (otherwise swap w_1 and w_2 and/or w_3 and w_4). Therefore, we get

$$\text{(1A)}\ \mathrm{dist}_{G'}(w_1, u) + \mathrm{dist}_{G'}(v, w_2) \leq D, \text{ and}$$
$$\text{(1B)}\ \mathrm{dist}_{G'}(w_3, u) + \mathrm{dist}_{G'}(v, w_4) \leq D.$$

However, since $\mathrm{dist}_{G'}(C') > D$, we have

$$\text{(1C)}\ \mathrm{dist}_{G'}(w_1, u) + \mathrm{dist}_{G'}(u, w_3) > D, \text{ and}$$
$$\text{(1D)}\ \mathrm{dist}_{G'}(w_2, v) + \mathrm{dist}_{G'}(v, w_4) > D.$$

Denote $K := \mathrm{dist}_{G'}(w_1, u) + \mathrm{dist}_{G'}(v, w_2) + \mathrm{dist}_{G'}(w_3, u) + \mathrm{dist}_{G'}(v, w_4)$. Inequalities (1A) and (1B) give $K \leq 2D$, while inequalities (1C) and (1D) give $K > 2D$, a contradiction.

Suppose that we are in Case 2. Denote by $P_{1,2}$, $P_{1,3}$, and $P_{2,3}$ three paths in G'' with weight at most D connecting w_1 and w_2, connecting w_1 and w_3, and connecting w_2 and w_3, respectively. Since $\mathrm{dist}_{G'}(\{w_1, w_2, w_3\}) > D$, all these paths use edge

(u, v). Without loss of generality, assume $\mathrm{dist}_{G'}(w_1, u) \le \mathrm{dist}_{G'}(w_1, v)$. Hence, both $P_{1,2}$ and $P_{1,3}$ reach u before v when traversing such paths starting at w_1. Without loss of generality, assume that $P_{2,3}$ reaches u before v when traversing such path starting at w_2 (otherwise, swap w_2 and w_3). Therefore, we get

$$\text{(2A)}\ \mathrm{dist}_{G'}(w_1, u) + \mathrm{dist}_{G'}(v, w_2) \le D, \text{ and}$$
$$\text{(2B)}\ \mathrm{dist}_{G'}(w_2, u) + \mathrm{dist}_{G'}(v, w_3) \le D.$$

However, since $\mathrm{dist}_{G'}(\{w_1, w_2, w_3\}) > D$, we have

$$\text{(2C)}\ \mathrm{dist}_{G'}(w_2, v) + \mathrm{dist}_{G'}(v, w_3) > D, \text{ and}$$
$$\text{(2D)}\ \mathrm{dist}_{G'}(w_1, u) + \mathrm{dist}_{G'}(u, w_2) > D.$$

Denote $L := \mathrm{dist}_{G'}(w_1, u) + \mathrm{dist}_{G'}(v, w_2) + \mathrm{dist}_{G'}(w_2, u) + \mathrm{dist}_{G'}(v, w_3)$. Inequalities (2A) and (2B) give $L \le 2D$, while inequalities (2C) and (2D) give $L > 2D$, a contradiction. This concludes the proof of the claim. □

Now, since $C \cup \{c_{B+2}\}$ is a set of $B + 2$ vertices with pairwise distance larger than D^B_{opt} in G, by iteratively using the claim we have that in any B-augmentation G_B of G, we have a set of $B + 2 - |F| \ge 2$ vertices with pairwise distance greater than D^B_{opt}, thus contradicting the definition of D^B_{opt}. This concludes the proof of the lemma. □

3.2 A minimum height tree

Let $C = \{c_0, \ldots, c_B\}$ be a set of $B + 1$ cluster centers such that the $B + 1$ clusters with centers at C and radius D^B_{opt} cover the vertices of G. This set can be computed as described in the previous section.

Definition 1. *Let $G = (V, E)$ be a graph together with a weight function $w : [V]^2 \to \mathbb{N}$. Let $C \subseteq V$ and let u be a vertex in V. A Shortest Path Tree of G, C, and u, denoted by* SPT(G, C, u)*, is a tree T rooted at u, spanning C, whose vertices and edges belong to V and E, respectively, and such that, for every vertex $c \in C$, it holds $d_T(u, c) = d_G(u, c)$.*

The *height* of a weighted rooted tree T, which is denoted by $\hbar(T)$, is the maximum weight of a path from the root to a leaf.

Definition 2. *Let $G = (V, E)$ be a graph together with a weight function $w : [V]^2 \to \mathbb{N}$ and a cost function $c : [V]^2 \to \mathbb{N}^*$. Let $C \subseteq V$, let u be a vertex in V, and let $B \ge 0$ be an integer. A Minimum Height$_B$ SPT of G, C, and u, denoted by* MH$_B$SPT (G, c, u)*, is an* SPT(G_B, C, u) *of minimum height over all B-augmentations G_B of G.*

Let G_B be a B-augmentation of G with diameter D^B_{opt}.

Lemma 4. *The height of a* MH$_B$SPT (G, C, u) *is at most D^B_{opt}.*

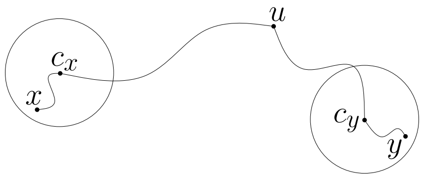

Fig. 1. Illustrating the path defined in the proof of Lemma 5.

Proof. By definition, we have (A) $\hbar(\text{MH}_B\text{SPT}(G, C, u)) \leq \hbar(\text{SPT}(G_B, C, u))$. Since G_B is a B-augmentation of G with diameter D^B_{opt}, we have (B) $\hbar(\text{SPT}(G_B, C, u)) \leq D^B_{opt}$. Inequalities (A) and (B) together prove the lemma. □

We now present a relationship between the BCMD problem and the problem of computing a MH_BSPT (G, C, u).

Lemma 5. *Let u be any vertex in V and let G'_B be a B-augmentation of G such that $\hbar(\text{SPT}(G'_B, C, u)) = \hbar(\text{MH}_B\text{SPT}(G, C, u))$. Then, the diameter of G'_B is at most $4 \cdot D^B_{opt}$.*

Proof. Consider two vertices x and y in V, see Figure 1. Let c_x and c_y be centers of the clusters x and y belong to, respectively. Then, we have $\text{dist}_{G'_B}(x, y) \leq \text{dist}_G(x, c_x) + \text{dist}_{G'_B}(c_x, u) + \text{dist}_{G'_B}(u, c_y) + \text{dist}_G(c_y, y)$. By Lemma 3, $\text{dist}_G(x, c_x), \text{dist}_G(c_y, y) \leq D^B_{opt}$. Since $\hbar(\text{SPT}(G'_B, C, u)) = \hbar(\text{MH}_B\text{SPT}(G, C, u))$ and by Lemma 4, it holds $\text{dist}_{G'_B}(c_x, u), \text{dist}_{G'_B}(u, c_y) \leq D^B_{opt}$. Hence, $\text{dist}_{G'_B}(x, y) \leq 4 \cdot D^B_{opt}$. □

3.3 Constructing a minimum height tree

In this section, we give an algorithm to compute a MH_BSPT (G, C, c_1).

We introduce some notation and terminology. Let $C' = C \setminus \{c_1\}$. Observe that a MH_BSPT (G, C', c_1) is also a MH_BSPT (G, C, c_1), given that a MH_BSPT (G, C', c_1) contains c_1 as its root. Denote by $d^j_K(u, v)$ the minimum weight of a j-bounded cost path connecting u and v in K. For any $u \in V$, for any $S \subseteq C'$, and for any $0 \leq j \leq B$, let $\gamma(u, S, j)$ denote the height of a MH_jSPT (G, S, u). Hence, the height of a MH_BSPT (G, C', c_1) is $\gamma(c_1, C', B)$. The following main lemma gives a dynamic programming recurrence for computing $\gamma(c_1, C', B)$.

Lemma 6. *For any $u \in V$, any $S \subseteq C'$, and any $0 \leq j \leq B$, the following hold: If $|S| = 1$, then $\gamma(u, S, j) = d^j_K(u, c_i)$ where $S = \{c_i\}$. If $|S| > 1$, then*

$$\gamma(u, S, j) = \min_{\substack{v \in V \\ S' \subsetneq S \\ j = j_1 + j_2 + j_3}} d^{j_1}_K(u, v) + \max\{\gamma(v, S', j_2), \gamma(v, S \setminus S', j_3)\}.$$

Proof. If $|S| = \{c_i\}$, then MH_jSPT $(G, \{c_i\}, u)$ is a minimum-weight path connecting u and c_i and having total cost at most j. Hence, $\gamma(u, S, j) = d^j_K(u, c_i)$. In particular, notice that, if $u = c_i$, then $\gamma(u, \{u\}, j) = d^j_K(u, u) = 0$.

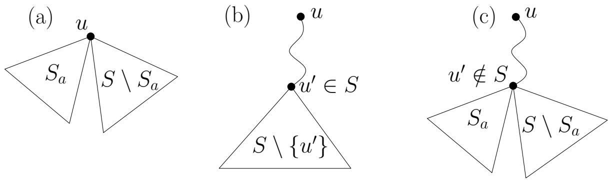

Fig. 2. Illustration for the proof of Lemma 6.

Assume that $|S| = m > 1$. Denote by T any MH$_j$SPT (G, S, u). Denote by $P(v, w)$ the unique path in T connecting two vertices v and w of T. We distinguish three cases, based on the structure of T. In Case (a), the degree of u in T is at least two (see Figure 2(a)). In Case (b), the degree of u in T is one and there exists a vertex $u' \in S$ such that every internal vertex of $P(u, u')$ has degree 2 in T and does not belong to S (see Figure 2(b)). Finally, in Case (c), the degree of u in T is one and there exists a vertex $u' \notin S$ such that every internal vertex of $P(u, u')$ has degree 2 in T and does not belong to S, and such that the degree of u' is greater than two (see Figure 2(c)).

First, we prove that one of the three cases always applies. If the degree of u in T is at least two, then Case (a) applies. Otherwise, the degree of u is 1. Traverse T from u until a vertex u' is found such that $u' \in S$ or the degree of u' is at least 3. If $u' \in S$, then every internal vertex of $P(u, u')$ has degree 2 in T and does not belong to S, hence Case (b) applies. If $u' \notin S$, then the degree of u' is at least 3, and every internal vertex of $P(u, u')$ has degree 2 in T and does not belong to S, hence Case (c) applies.

We now prove that, in each of the three cases, the recursive computation of $\gamma(u, S, j)$ is correct. That is, we show that the value $\gamma(u, S, j)$ computed by the recurrence in the statement of the lemma is at most $\hbar(T)$; observe that $\gamma(u, S, j)$ cannot be smaller than $\hbar(T)$, by the assumption that T is a MH$_j$SPT (G, S, u).

In Case (a), T is composed of two subtrees MH$_x$SPT (G, S_a, u) and MH$_y$SPT $(G, S \setminus S_a, u)$, only sharing vertex u, with $\emptyset \subsetneq S_a \subsetneq S$. The height of T is the maximum of the heights of MH$_x$SPT (G, S_a, u) and MH$_y$SPT $(G, S \setminus S_a, u)$; also, the cost of T is $x + y$. By definition, the heights of MH$_x$SPT (G, S_a, u) and MH$_y$SPT $(G, S \setminus S_a, u)$ are $\gamma(u, S_a, x)$ and $\gamma(u, S \setminus S_a, y))$, respectively. Thus, the height of T is $\hbar(T) = \max\{\gamma(u, S_a, x), \gamma(u, S \setminus S_a, y)\}$. Such a value is found by the recursive definition of $\gamma(u, S, j)$ with $v = u$, $S' = S_a$, $j_1 = 0$, $j_2 = x$, and $j_3 = y$, hence the value $\gamma(u, S, j)$ computed by the recurrence in the statement of the lemma is at most $\hbar(T)$.

In Case (b), T is composed of a path from u to u' with cost x and weight $d_K^x(u, u')$, and of a MH$_y$SPT $(G, S \setminus \{u'\}, u')$. The height of T is the sum of $d_K^x(u, u')$ and the height of MH$_y$SPT $(G, S \setminus \{u'\}, u')$; also the cost of T is $x + y$. By definition, the height of MH$_y$SPT $(G, S \setminus \{u'\}, u')$ is $\gamma(u', S \setminus \{u'\}, y)$. Thus, the height of T is $\hbar(T) = d_K^x(u, u') + \gamma(u', S \setminus \{u'\}, y)$. Such a value is found by the recursive definition of $\gamma(u, S, j)$ with $v = u'$, $S' = S \setminus \{u'\}$, $j_1 = x$, $j_2 = y$, and $j_3 = 0$, hence the value $\gamma(u, S, j)$ computed by the recurrence in the statement of the lemma is at most $\hbar(T)$.

In Case (c), T is composed of a path from u to u' with cost x and weight $d_K^x(u, u')$, of a MH$_y$SPT (G, S_a, u'), and of a MH$_z$SPT $(G, S \setminus S_a, u')$ with $\emptyset \subsetneq S_a \subsetneq S$. The height of T is the sum of $d_K^x(u, u')$ and the maximum between the heights of MH$_y$SPT (G, S_a, u') and MH$_z$SPT $(G, S \setminus S_a, u')$; also the cost of T is $x+y+z$. By definition, the heights of MH$_y$SPT (G, S_a, u') and MH$_z$SPT $(G, S \setminus S_a, u')$ are $\gamma(u', S_a, y)$ and $\gamma(u', S \setminus S_a, z)$, respectively. Thus, the height of T is $\hbar(T) = d_K^x(u, u') + \max\{\gamma(u', S_a, y), \gamma(u', S \setminus S_a, z)\}$. Such a value is found by the recursive definition of $\gamma(u, S, j)$ with $v = u'$, $S' = S_a$, $j_1 = x$, $j_2 = y$, and $j_3 = z$, hence the value $\gamma(u, S, j)$ computed by the recurrence in the statement of the lemma is at most $\hbar(T)$.

This concludes the proof of the lemma. □

Lemma 6 yields the following.

Theorem 2. *There exists a $(1, 4)$-approximation algorithm for the* BCMD *problem with $O((3^B B^3 + n + \log(Bn))Bn^2)$ running time.*

Proof. Given an instance (G, w, c, B) of the BCMD problem, by Theorem 1 we can determine, for every pair of vertices $u, v \in V$ and for every $1 \leq j \leq B$, the minimum weight of a j-bounded cost path connecting u and v in total $O((n+\log(Bn))Bn^2)$ time. By Lemma 3, a clustering of G can be computed in $O(Bn^2)$ time. Due to Lemma 6, the problem of computing a MH$_B$SPT $(G, C \setminus \{c_1\}, c_1)$ can be solved by dynamic programming over the triples (u, S, j) (there are $O\left(B\binom{B+1}{s}n\right)$ such triples with $|S| = s$); the computation of the value for any such triple requires to take a minimum over $j^3 2^{|S|} n$ values, hence the dynamic programming running time is $O\left(nB\sum_{s=0}^{B}\binom{B+1}{s}B^3 2^s n\right) = O(B^4 3^B n^2)$. Observe that the dynamic programming can be designed in such a way that a rooted tree with height equal to $\gamma(u, S, j)$ is computed together with the value of $\gamma(u, S, j)$. This is trivially done in the base case; moreover, in the inductive case it only requires, for each $v \in V$, each $S' \subsetneq S$, and each $j = j_1 + j_2 + j_3$, the computation of a shortest path tree. Finally, by Lemma 5, augmenting G with the non-edges that are present in a MH$_B$SPT $(G, C \setminus \{c_1\}, c_1)$ yields a B-augmentation G_B whose diameter is at most $4 \cdot D_{opt}^B$. □

4 Unit Costs and Arbitrary Weights

For the special case in which each edge has unit cost and arbitrary weight, our techniques lead to several results, that are described in the following. Observe that, in this case we are allowed to insert in G exactly k non-edges of G, where $k = B = O(n^2)$. We remark that Theorem 2 gives a $(1, 4)$-approximation algorithm running in $O((3^k k^3 + n)kn^2)$ time for this special case.

In the following, we denote by C a clustering with $k + 1$ clusters constructed as described in Subsection 3.1. We first show a $((k + 1)^2, 3)$-approximation algorithm.

Theorem 3. *Given an instance of the* BCMD *problem with unit costs, there exists a $((k + 1)^2, 3)$-approximation algorithm with $O(kn^3)$ running time.*

Proof. For every pair of cluster centers $c_i, c_j \in C$ compute a shortest path in K between c_i and c_j that contains at most k non-edges of G. Add those edges to F and let $G' =$

$(V, E \cup F)$. By Theorem 1 and since $k = O(n^2)$, G' can be constructed in $O(kn^3)$ time. Observe that, for each pair of cluster centers, the algorithm adds at most k non-edges of G to F, thus at most $k(k+1)^2$ non-edges in total. We prove that, for every $v_i, v_j \in V$, there exists a path in G' connecting v_i and v_j whose weight is at most $3 \cdot D^k_{opt}$. Denote by c_i and c_j the centers of the clusters v_i and v_j belong to, respectively. We have $\mathrm{dist}_{G'}(v_i, v_j) \leq \mathrm{dist}_G(v_i, c_i) + \mathrm{dist}_{G'}(c_i, c_j) + \mathrm{dist}_G(c_j, v_j)$. By Lemma 3, $\mathrm{dist}_G(v_i, c_i), \mathrm{dist}_G(v_j, c_j) \leq D^k_{opt}$; also, by construction, $\mathrm{dist}_{G'}(c_i, c_j) \leq D^k_{opt}$, and the theorem follows. □

Next, we give a $(k, 4)$-approximation algorithm.

Theorem 4. *Given an instance of the* BCMD *problem with unit costs, there exists a $(k, 4)$-approximation algorithm with $O(kn^2)$ running time.*

Proof. Pick an arbitrary cluster center, say c_1. For every cluster center $c_j \in C \setminus \{c_1\}$, compute a shortest path between c_1 and c_j in K containing at most k non-edges of G. Add those edges to F and let $G' = (V, E \cup F)$. By Corollary 1, a shortest path between c_1 and c_j in K containing at most k non-edges of G corresponds to a shortest path between $(c_1, 0)$ and (c_j, k) in digraph H. By Observation 1, H has $O(kn)$ vertices and $O(kn^2)$ edges. Hence, Dijkstra's algorithm with Fibonacci heaps [10] computes all the shortest paths between $(c_1, 0)$ and (c_j, k), for every $c_j \in C \setminus \{c_1\}$, in total $O(kn^2)$ time. Observe that, for each cluster different from c_1, the algorithm adds at most k non-edges of G to F, thus at most k^2 non-edges in total. We prove that, for every $v_i, v_j \in V$, there exists a path in G' connecting v_i and v_j whose weight is at most $4 \cdot D^k_{opt}$. Denote by c_i and c_j the centers of the clusters v_i and v_j belong to, respectively. We have $\mathrm{dist}_{G'}(v_i, v_j) \leq \mathrm{dist}_G(v_i, c_i) + \mathrm{dist}_{G'}(c_i, c_1) + \mathrm{dist}_{G'}(c_1, c_j) + \mathrm{dist}_G(c_j, v_j)$. By Lemma 3, $\mathrm{dist}_G(v_i, c_i), \mathrm{dist}_G(v_j, c_j) \leq D^k_{opt}$; by construction, $\mathrm{dist}_{G'}(c_i, c_1), \mathrm{dist}_{G'}(c_1, c_j) \leq D^k_{opt}$, and the theorem follows. □

Finally, we present a $(1, 3k+2)$-approximation algorithm.

Theorem 5. *Given an instance of the* BCMD *problem with unit costs, there exists a $(1, 3k+2)$-approximation algorithm with $O(n^2 + k^2)$ running time.*

Proof. For every pair of clusters C_i and C_j, with $1 \leq i < j \leq k+1$, let e_{ij} be the edge of K of minimum weight connecting a vertex in C_i with a vertex in C_j. We denote by F' the set of these edges. For a subset F of F', we say that F *spans* C if the graph representing the adjacencies between clusters via the edges of F is connected. Let F be a minimum-weight set of k edges from F' spanning C. Let $G' = (V, E \cup F)$. The set F', and hence the graph G', can be constructed in $O(n^2 + k^2)$ time as follows. Consider all the edges of K and keep, for each pair of clusters, the edge with smallest weight. This can be done in $O(n^2)$ time. Finally, compute in $O(k^2)$ time a minimum spanning tree of the resulting graph [11], that has $O(k)$ vertices and $O(k^2)$ edges. Observe that the algorithm adds at most k non-edges of G to F. We prove that, for every $v_i, v_j \in V$, there exists a path in G' connecting v_i and v_j whose weight is at most $(3k+2)D^k_{opt}$. Denote by P_C the (unique) subset of F connecting the clusters v_i and v_j belong to. Let $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ be the edges of P_C in order from v_i to v_j. Then, $\mathrm{dist}_{G'}(v_i, v_j) \leq \mathrm{dist}_G(v_i, x_1) + w(x_1, y_1) +$

$\mathrm{dist}_G(y_1, x_2) + \ldots + w(x_m, y_m) + \mathrm{dist}_G(y_m, v_j)$. By Lemma 3, $\mathrm{dist}_G(y_i, x_{i+1}) \leq 2D^k_{opt}$, and $\mathrm{dist}_G(v_i, x_1), \mathrm{dist}_G(y_m, v_j) \leq D^k_{opt}$. Also, $w(x_i, y_i) \leq D^k_{opt}$, and the theorem follows. □

5 Hardness Results

The main theorem of this section provides a parameterized intractability result for BCMD with unit weights and unit costs, and some related problems.[1] The U-BCMD problem has as input an unweighted graph $G = (V, E)$ and two integers k and d, and the question is whether there is a set $F \subseteq [V]^2 \setminus E$, with $|F| \leq k$, such that the graph $(V, E \cup F)$ has diameter at most d. The parameter is k. We will show that U-BCMD is $W[2]$-hard. We will also provide refinements to the minimum conditions required for intractability, namely U-BCMD remains NP-complete for graphs of diameter three with target diameter two. We note that although Dodis and Kanna [5] provide an inapproximability reduction from SET COVER, they begin with a disconnected graph, and expand the instance with a series of size-two sets, which does not preserve the size of the optimal solution, and therefore their reduction cannot be used to show parameterized complexity lower bounds.

Theorem 6. SET COVER *is polynomial-time reducible to* U-BCMD. *Moreover, the reduction is parameter preserving and creates an instance with diameter three and target diameter two.*

Proof. Let (X, S, k) be an instance of SET COVER where S is the base set and $X \subset \mathcal{P}(S)$ is the set from which we must pick the set cover of S with size at most k. We construct an instance $(G = (V, E), k, d)$ of U-BCMD as follows.

Let $m = |X| \cdot k$. The vertex set V is the disjoint union of 5 sets:

- a set Y corresponding to the set X where for each $x \in X$ we have a vertex $y \in Y$,
- a set $T = \biguplus_{i \in [m]} T_i$ corresponding to S where, for each $s \in S$ and $i \in [m]$, we have a vertex $t_i \in T_i$ (*i.e.*, we have m copies of a set of vertices corresponding to S),
- a set U with $\binom{m}{2}$ vertices u_{ij}, one for each pair (T_i, T_j) with $i \neq j$,
- the set $\{a\}$, and
- the set $\{b\}$.

The edge set E consists of the following edges:

- ab,
- by for each vertex $y \in Y$,
- bu_{ij} for each vertex $u_{ij} \in U$,
- yy' for each pair of vertices $y, y' \in Y$,

[1] After the publication of the conference version of this paper [9] and after the first submission of the present journal paper, James Nastos informed us about a paper whose results overlap with the intractability results of this paper. Namely, Gao, Hare, and Nastos [12] proved that the BCMD problem with unit weights and costs is $W[2]$-hard, for every target diameter $d \geq 2$. Their reduction is from "dominating set" and is similar to the one we present in this paper. We apologize to the authors of [12] for not being aware of their previous result.

– $u_{ij}u_{lp}$ for each pair of vertices $u_{ij}, u_{lp} \in U$,
– yt for each pair of vertices $y \in Y$ and $t \in T_i$ such that the element $s \in S$ corresponding to t is in the set $x \in X$ corresponding to y in the SET COVER instance, and
– tu_{jl} for each pair of vertices $t \in T_i$ and $u_{jl} \in U$ such that $i \in \{j, l\}$.

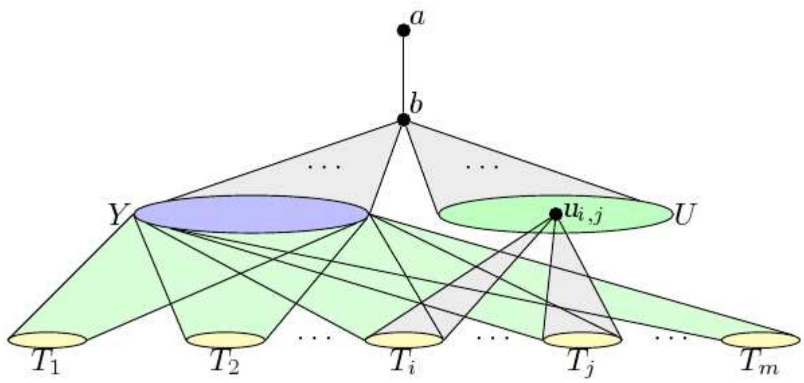

Fig. 3. Sketch of the construction for the SET COVER to U-BCMD reduction. The edge sets represented in gray are complete, the edge sets represented in light green correspond to the set membership from the SET COVER instance. The vertex sets Y and U are cliques. The vertex sets T_i are independent sets for all $i \in [m]$.

We set $d = 2$. Note that k in the U-BCMD instance is the same k as for the SET COVER instance. The construction is sketched in Figure 3.

Claim 2 *For all* $v, v' \in V \setminus \{a\}$ *we have* $\operatorname{dist}(v, v') \leq 2$.

Proof. The vertices of U are at distance one from each other. The vertices of Y are at distance one from each other. Vertex b is at distance 1 from the vertices of Y and U. Therefore the vertices of U and Y are at most distance 2 from each other via the path through b. Each vertex $t \in T$ is at distance one from some vertex $y \in Y$. As Y is a clique, t is at distance at most two from all the vertices in Y. Each vertex $t \in T$ is at distance one from some vertex $u \in U$. As U is a clique, t is at distance at most two from all the vertices in U. For each pair of vertices $t_i \in T_i$ and $t_j \in T_j$ there is a vertex $u_{ij} \in U$ such that $t_i u_{ij} \in E$ and $t_j u_{ij} \in E$. If $i = j$ then any vertex $u_{ik} \in U$ will suffice. Thus all the vertices of T are at most distance 2 from each other and from b. □

Claim 3 *For all* $v \in V$ *we have* $\operatorname{dist}(a, v) \leq 3$*. Moreover,* $\operatorname{dist}(a, v) = 3$ *if and only if* $v \in T$.

Proof. As the distance from b to all other vertices is at most 2, the distance from a to all other vertices is at most 3. Moreover, as the distance from b to the vertices of U and Y is one, the distance from a to these vertices is two. Therefore the only vertices at distance three from a are the vertices of T. □

Thus we are concerned only with reducing the distance between a and the vertices of T.

Claim 4 (X, S, k) *is a* YES-*instance of* SET COVER *if and only if* (G, k, d) *is a* YES-*instance of* U-BCMD.

Proof. Let $X' \subseteq X$ be the set cover that witnesses that (X, S, k) is a YES-instance of SET COVER. Let $Y' \subseteq Y$ be the set of vertices that corresponds to X'. We have $|Y'| = |X'| \leq k$. If we add the edges ay for all $y \in Y'$, then a is at distance at most 2 from all vertices $t \in T$. As X' is a set cover of S, for each $s \in S$ there is at least one set $x \in X'$ such that $s \in x$. Then there is an edge from a to the vertex y corresponding to x, and by the construction, y is adjacent to $t \in T$ if and only if the corresponding element s is in S, thus we have a path $a \rightsquigarrow y \rightsquigarrow t$.

Now, assume (G, k, d) is a YES-instance of U-BCMD. First consider the case where all the edges are added between a and the vertices of Y. Then the set $Y' \subseteq Y$ of vertices newly adjacent to a corresponds to a set cover $X' \subseteq X$ in the same way as before.

We must demonstrate that we may only (productively) add edges between a and Y. Observe that any edge in a path from a to a vertex t_i in T with length at most 2 must have a or t_i as an end-vertex. Hence, adding edges between two vertices in $\{b\} \cup U \cup Y$ does not help decreasing the diameter of G. Further, we cannot add the edge ab, as it already exists. Also, any edge (b, t_i) can be replaced by edge (a, t_i). Edge (t_i, t_j) is used in any length-2 path from a to the vertices in T only if (a, t_i) or (a, t_j) is in the solution. In the former case, (t_i, t_j) can be replaced by edge (a, t_j), in the latter case by (a, t_i). Hence, we can assume that any solution only uses edges connecting a with a vertex in Y or in U. Each edge from a to a vertex $u_{ij} \in U$ can only decrease the distance between a and the vertices in two sets T_i and T_j. Thus, as long as $|X| > 2$ there exists a set T_i none of whose adjacent vertices in U is adjacent to a. This implies that we must add edges from a to a subset Y' of Y such that T_i is dominated by Y'. Hence, Y' corresponds to a set cover of S. □

We note that the reduction is obviously polynomial-time computable, and the parameter k is preserved. The theorem now follows from the previous claims. □

Corollary 2. U-BCMD *is* NP-*complete even for graphs of diameter three with target diameter two.*

Proof. As it is already known that U-BCMD is in NP [5], the result for U-BCMD follows from Theorem 6. □

As SET COVER is $W[2]$-hard with parameter k, combined with Corollary 2 we also have the following result.

Corollary 3. U-BCMD *is* $W[2]$*-hard even for graphs of diameter three with target diameter two.*

We note additionally that as the initial graph has diameter 3 and the target diameter is 2, it is even NP-hard and W[2]-hard to decide if there is a set of k new edges that improves the diameter by one. Furthermore by taking a as source vertex, the results transfer immediately to the single-source version as discussed by Demaine & Zadimoghaddam [4].

The construction of Theorem 6 can even be extended to give a parameterized inapproximability result for U-BCMD.

Theorem 7. *It is $W[2]$-hard to compute a $(1 + \frac{c}{k}, \frac{3}{2} - \varepsilon)$-approximation for* U-BCMD *for any constants c and $\varepsilon > 0$.*

Proof. We repeat the construction of Theorem 6, except that we introduce $c + 1$ copies of the Y and T components and set $k' = k \cdot (c + 1)$, where k' is the parameter of U-BCMD. Let Y_i with $1 \leq i \leq c + 1$ be the copies of the Y components and let $T_{i,j}$ with $1 \leq i \leq c + 1$ and $1 \leq j \leq m$ be the copies of the T components. The edges are similar to the previous construction; we highlight the differing edges:

– by for all $y \in \bigcup_i Y_i$,
– yy' for each $y, y' \in \bigcup_i Y_i$,
– yt for each $y \in Y_i$ and $t \in T_{i,j}$ where the element $x \in X$ corresponding to y is in the set $s \in S$ corresponding to t, and
– tu_{ij} for each vertex $t \in \bigcup_h (T_{h,i} \cup T_{h,j})$ and each vertex $u_{ij} \in U$.

Then apart from a, all vertices remain at pairwise distance 2, with a at distance 3 from vertices in $\bigcup_{i,j} T_{i,j}$. To reduce the diameter to 2 we require the addition of edges from a to vertices of the Y components as before, furthermore we require edges to each copy, otherwise there is some $T_{i,j}$ that remains at distance 3 from a.

Thus, if the SET COVER instance has a solution of size k, then the U-BCMD instance has a solution of size $(c + 1)k = k'$. Conversely, let F be a set of at most $(1 + \frac{c}{k'})k' = k' + c$ edges such that the diameter of $G' = (V, E \cup F)$ is at most $(\frac{3}{2} - \varepsilon) \cdot 2$. Since the diameter of G' is integral, it is at most 2. Since there are $c + 1$ copies of Y, at least one of them has at most k' vertices adjacent to a, giving a set cover of size k'. □

Acknowledgments

A preliminary version of this paper was presented in [9]. FF acknowledges support from the Australian Research Council (grant DE140100708). SG acknowledges support from the Australian Research Council (grant DE120101761). JG acknowledges support from the Australian Research Council (grant FT100100755). NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

1. Alon, N., Gyárfás, A., Ruszinkó, M.: Decreasing the diameter of bounded degree graphs. Journal of Graph Theory 35, 161–172 (1999)
2. Bilò, D., Gualà, L., Proietti, G.: Improved approximability and non-approximability results for graph diameter decreasing problems. Theoretical Computer Science 417, 12–22 (2012)
3. Chepoi, V., Vaxès, Y.: Augmenting trees to meet biconnectivity and diameter constraints. Algorithmica 33(2), 243–262 (2002)
4. Demaine, E.D., Zadimoghaddam, M.: Minimizing the diameter of a network using shortcut edges. In: Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT). pp. 420–431 (2010)

5. Dodis, Y., Khanna, S.: Designing networks with bounded pairwise distance. In: Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC). pp. 750–759 (1999)
6. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science, Springer, New York (1999)
7. Erdős, P., Rényi, A., Sós, V.T.: On a problem of graph theory. Studia Sci. Math. Hungar. 1, 215–235 (1966)
8. Flum, J., Grohe, M.: Parameterized Complexity Theory, Texts in Theoretical Computer Science. An EATCS Series, vol. XIV. Springer, Berlin (2006)
9. Frati, F., Gaspers, S., Gudmundsson, J., Mathieson, L.: Augmenting graphs to minimize the diameter. In: Cai, L., Cheng, S.W., Lam, T.W. (eds.) International Symposium on Algorithms and Computation (ISAAC '13). LNCS, vol. 8283, pp. 383–393. Springer (2013)
10. Fredman, M.L., Tarjan, R.E.: Fibonacci heaps and their uses in improved network optimization algorithms. Journal of the ACM 34(3), 596–615 (1987)
11. Fredman, M.L., Willard, D.E.: Trans-dichotomous algorithms for minimum spanning trees and shortest paths. Journal of Computer and System Sciences 48(3), 533–551 (1994)
12. Gao, Y., Hare, D.R., Nastos, J.: The parametric complexity of graph diameter augmentation. Discrete Applied Mathematics 161(10-11), 1626–1631 (2013)
13. Grigorescu, E.: Decreasing the diameter of cycles. Journal of Graph Theory 43(4), 299–303 (2003)
14. Kapoor, S., Sarwat, M.: Bounded-diameter minimum-cost graph problems. Theory of Computing Systems 41(4), 779–794 (2007)
15. Li, C.L., McCormick, S.T., Simchi-Levi, D.: On the minimum-cardinality-bounded-diameter and the bounded-cardinality-minimum-diameter edge addition problems. Operations Research Letters 11(5), 303–308 (1992)
16. Marx, D.: Parameterized complexity and approximation algorithms. The Computer Journal 51(1), 60–78 (2008)
17. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford Lecture Series in Mathematics and its Applications, Oxford University Press, Oxford (2006)
18. Schoone, A.A., Bodlaender, H.L., van Leeuwen, J.: Diameter increase caused by edge deletion. Journal of Graph Theory 11, 409–427 (1997)
19. Vazirani, V.V.: Approximation algorithms. Springer (2001)