

Augmenting Graphs to Minimize the Diameter

Fabrizio Frati¹, Serge Gaspers^{2,3},
Joachim Gudmundsson^{1,3}, and Luke Mathieson⁴

¹ University of Sydney, Australia
brillo@it.usyd.edu.au, 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 [1,7,11].

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.

PROBLEM: Bounded Cost Minimum Diameter Edge Addition (BCMD)

INPUT: An undirected graph $G = (V, E)$, a weight function $w : [V]^2 \rightarrow \mathbb{N}$, a cost function $c : [V]^2 \rightarrow \mathbb{N}^*$, 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 [14]. For background on parameterized complexity we refer to [6,8,15] and for background on approximation algorithms to [17].

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 [3,5,12,13], 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 $\beta \cdot D_{opt}^B$, where D_{opt}^B 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. [16]; see also the paper by Li et al. [13]. Bilò et al. [2] showed that, as a consequence of the results in [3,5,13], there exists no $(c \log n, \delta < 1 + 1/D_{opt}^B)$ -approximation algorithm for BCMD if $D_{opt}^B \geq 2$, unless P=NP. For the case in which $D_{opt}^B \geq 6$, they proved a stronger lower bound, namely that there exists no $(c \log n, \delta < \frac{5}{3} - \frac{7 - (D_{opt}^B + 1) \bmod 3}{3D_{opt}^B})$ -approximation algorithm, unless P=NP.

Dodis and Khanna [5] gave an $(O(\log n), 2 + 2/D_{opt}^B)$ -approximation algorithm (see also [12]). Li et al. [13] showed a $(1, 4 + 2/D_{opt}^B)$ -approximation algorithm. The analysis of the latter algorithm was later improved by Bilò et al. [2], who showed that it gives a $(1, 2 + 2/D_{opt}^B)$ -approximation. In the same paper they also gave a $(O(\log n), 1)$ -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 $(1, 2 + 2\omega/D_{opt}^B)$ -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 $D_{opt}^B \geq 2\omega$ and for some constant c , there is no $(c \log n, \delta < 2 - 3\omega/D_{opt}^B)$ -approximation algorithm for this restriction of the BCMD problem, unless P=NP.

Arbitrary Costs and Weights. To the best of our knowledge, only one theory paper considered the general BCMD problem. In 1999, Dodis and Khanna [5] presented an $O(n \log D_{opt}^B, 1)$ -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 $((k + 1)^2, 3)$ -, $(k, 4)$ -, and $(1, 3k + 2)$ -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 technical reasons, we add self-loops with weight 0 and cost 1 at each vertex of K .

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, \dots, 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' and F' , where $E' = \bigcup_{0 \leq i \leq k} E_i$, and

$$F' = \left\{ ((u, i), (v, i + c(\{u, v\}))) : 0 \leq i \leq B - c(\{u, v\}), \{u, v\} \in [V]^2 \setminus E \right\}.$$

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.

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 H contains a directed path P_H with weight W connecting vertices (u, i) and (v, j) , for some $j \geq i$. Then, there exists a $(j - i)$ -bounded-cost path P_K in K with weight W connecting u and v .*

Lemma 2. *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) .*

We have the following.

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

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 [9]. 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 total $O(Bn^2 + Bn \log(Bn))$ 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 total $O(Bn^3 + Bn^2 \log(Bn))$ 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 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 $T = (U \subseteq V, D)$, so that $C \subseteq U$, 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 $C = \{c_1, \dots, c_{B+1}\}$ 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 $\text{dist}_G(c_i, \{c_1, \dots, c_{i-1}\})$ 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 $\text{dist}_G(v, C) \leq D_{opt}^B$ 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 [9]. Let c_{B+2} denote a vertex maximizing $\text{dist}_G(c_{B+2}, C)$, and denote this distance by R . By definition, $\text{dist}_G(v, C) \leq R$ for every $v \in V$. To prove the lemma it remains to show that $R \leq D_{opt}^B$. For the sake of contradiction, assume $D_{opt}^B < R$. Then, $C \cup \{c_{B+2}\}$ is a set of $B + 2$ vertices with pairwise distance larger than D_{opt}^B in G . 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 $\text{dist}_{G'}(C') > D$. 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 $\text{dist}_{G''}(C'') > D$.*

Now, since $C \cup \{c_{B+2}\}$ is a set of $B + 2$ vertices with pairwise distance larger than D_{opt}^B in G , by iteratively using the claim we have that in any B -augmentation G_B of

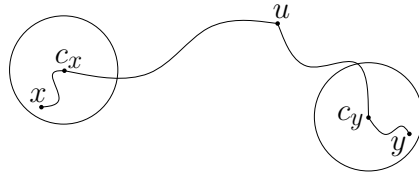


Fig. 1. Illustrating the path defined in the proof of Lemma 5

G , we have a set of $B + 2 - |F| \geq 2$ vertices with pairwise distance greater than D_{opt}^B , thus contradicting the definition of D_{opt}^B . This concludes the proof of the lemma. \square

3.2 A Minimum Height Tree

Let C be a set of $B + 1$ cluster centers such that the $B + 1$ clusters with centers at $C = \{c_0, \dots, c_B\}$ and radius D_{opt}^B 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 $w : [V]^2 \rightarrow \mathbb{N}$. Let $C \subseteq V$ and let u be a vertex in V . A Shortest Path Tree of G , C , and u , denoted by $\text{SPT}(G, C, u)$, 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 $h(T)$, 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 $w : [V]^2 \rightarrow \mathbb{N}$ and a cost function $c : [V]^2 \rightarrow \mathbb{N}^*$. Let $C \subseteq V$, let u be a vertex in V , and let $B \geq 0$ be an integer. A Minimum Height B SPT of G , C , and u , denoted by $\text{MH}_B\text{SPT}(G, c, u)$, is a $\text{SPT}(G_B, C, u)$ of minimum height over all B -augmentations G_B of G .

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

Lemma 4. The height of a $\text{MH}_B\text{SPT}(G, C, u)$ is at most D_{opt}^B .

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

We now present a relationship between the BCMD problem and the problem of computing a $\text{MH}_B\text{SPT}(G, C, u)$.

Lemma 5. Let G'_B be a B -augmentation of G such that it holds $h(\text{SPT}(G'_B, C, u)) = h(\text{MH}_B\text{SPT}(G, C, u))$, for any $u \in V$. Then, the diameter of G'_B is at most $4 \cdot D_{opt}^B$.

Proof. Consider two vertices x and y in V , see Fig. 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, $\text{dist}_G(x, c_x), \text{dist}_G(c_y, y) \leq D_{opt}^B$. Since $h(\text{SPT}(G'_B, C, u)) = h(\text{MH}_B\text{SPT}(G, C, u))$ and by Lemma 4, it holds $\text{dist}_{G'_B}(c_x, u), \text{dist}_{G'_B}(u, c_y) \leq D_{opt}^B$. Hence, $\text{dist}_{G'_B}(x, y) \leq 4 \cdot D_{opt}^B$. \square

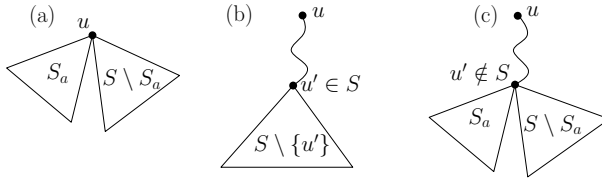


Fig. 2. Illustration for the proof of Lemma 6

3.3 Constructing a Minimum Height Tree

In this section, we show an algorithm to compute a $MH_B SPT(G, C, c_1)$.

We introduce some notation and terminology. Let $C' = C \setminus \{c_1\}$. Observe that a $MH_B SPT(G, C', c_1)$ is also a $MH_B SPT(G, C, c_1)$, given that a $MH_B SPT(G, C', c_1)$ contains c_1 as its root. Denote by $d_K^j(u, v)$ 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 $MH_j SPT(G, S, u)$. Hence, the height of a $MH_B SPT(G, C', c_1)$ 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 $\gamma(u, S, j) = d_K^j(u, c_i)$ where $S = \{c_i\}$. If $|S| > 1$, then*

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

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

If $|S| = m > 1$, then suppose that the lemma holds for each $\gamma(u', S', j')$ with $|S'| \leq m - 1$ by induction. 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 Fig. 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 Fig. 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 Fig. 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 v' is found such that $v' \in S$ or the degree of v' is at least 3. If $v' \in S$, then every internal vertex of $P(u, u')$ has degree 2 in T and does not belong to S , hence Case (b) applies. If $v' \notin S$, then the degree of v' 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 discuss the three cases.

In Case (a), T is composed of two subtrees $\text{MH}_x\text{SPT}(G, S_a, u)$ and $\text{MH}_y\text{SPT}(G, S \setminus S_a, u)$, only sharing vertex u , with $\emptyset \subsetneq S_a \subsetneq S$. The height of T is the maximum of the heights of $\text{MH}_x\text{SPT}(G, S_a, u)$ and $\text{MH}_y\text{SPT}(G, S \setminus S_a, u)$; also the cost of T is at most $x + y$. By induction, the heights of $\text{MH}_x\text{SPT}(G, S_a, u)$ and $\text{MH}_y\text{SPT}(G, S \setminus S_a, u)$ are $\gamma(u, S_a, x)$ and $\gamma(u, S \setminus S_a, y)$, respectively. Thus, the height of T is $\max\{\gamma(u, S_a, x), \gamma(u, S \setminus S_a, y)\}$ and hence $\gamma(u, S, j) = \max\{\gamma(u, S_a, x), \gamma(u, S \setminus S_a, y)\}$. 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 $j_3 = y$.

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 $\text{MH}_y\text{SPT}(G, S \setminus \{u'\}, u')$. The height of T is the sum of $d_K^x(u, u')$ and the height of $\text{MH}_y\text{SPT}(G, S \setminus \{u'\}, u')$; also the cost of T is at most $x + y$. By induction, the height of $\text{MH}_y\text{SPT}(G, S \setminus \{u'\}, u')$ is $\gamma(u', S \setminus \{u'\}, y)$. Thus, the height of T is $d_K^x(u, u') + \gamma(u', S \setminus \{u'\}, y)$ and hence $\gamma(u, S, j) = d_K^x(u, u') + \gamma(u', S \setminus \{u'\}, y)$. 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 $j_3 = 0$.

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 $\text{MH}_y\text{SPT}(G, S_a, u')$, and of a $\text{MH}_z\text{SPT}(G, S \setminus S_a, u')$ 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 $\text{MH}_y\text{SPT}(G, S_a, u')$ and $\text{MH}_z\text{SPT}(G, S \setminus S_a, u')$; also the cost of T is at most $x + y + z$. By induction, the heights of $\text{MH}_y\text{SPT}(G, S_a, u')$ and $\text{MH}_z\text{SPT}(G, S \setminus S_a, u')$ are $\gamma(u', S_a, y)$ and $\gamma(u', S \setminus S_a, z)$, respectively. Thus, the height of T is $d_K^x(u, u') + \max\{\gamma(u', S_a, y), \gamma(u', S \setminus S_a, z)\}$ and hence $\gamma(u, S, j) = d_K^x(u, u') + \max\{\gamma(u', S_a, y), \gamma(u', S \setminus S_a, z)\}$. 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 $j_3 = z$.

This concludes the induction and hence 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.*

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 $3 \cdot D_{opt}^k$. Denote by c_i and c_j the centers of the clusters v_i and v_j belong to, respectively. We have $\text{dist}_{G'}(v_i, v_j) \leq \text{dist}_{G'}(v_i, c_i) + \text{dist}_{G'}(c_i, c_j) + \text{dist}_{G'}(c_j, v_j)$. By Lemma 3, $\text{dist}_{G'}(v_i, c_i), \text{dist}_{G'}(v_j, c_j) \leq D_{opt}^k$; also, by construction, $\text{dist}_{G'}(c_i, c_j) \leq D_{opt}^k$, and the theorem follows. \square

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 [9] 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 $4 \cdot D_{opt}^k$. Denote by c_i and c_j the centers of the clusters v_i and v_j belong to, respectively. We have $\text{dist}_{G'}(v_i, v_j) \leq \text{dist}_{G'}(v_i, c_i) + \text{dist}_{G'}(c_i, c_1) + \text{dist}_{G'}(c_1, c_j) + \text{dist}_{G'}(c_j, v_j)$. By Lemma 3, $\text{dist}_{G'}(v_i, c_i), \text{dist}_{G'}(v_j, c_j) \leq D_{opt}^k$; by construction, $\text{dist}_{G'}(c_i, c_1), \text{dist}_{G'}(c_1, c_j) \leq D_{opt}^k$, and the theorem follows. \square

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 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 [10], 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 $(3k + 2)D_{opt}^k$. Denote by P_C the (unique) subset of F connecting the clusters v_i and v_j belong to. Let $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ be the edges of P_C in order from v_i to v_j . Then, $\text{dist}_{G'}(v_i, v_j) \leq \text{dist}_G(v_i, x_1) + w(x_1, y_1) +$

$\text{dist}_G(y_1, x_2) + \dots + w(x_m, y_m) + \text{dist}_G(y_m, v_j)$. By Lemma 3, $\text{dist}_G(y_i, x_{i+1}) \leq 2D_{opt}^k$, and $\text{dist}_G(v_i, x_1), \text{dist}_G(y_m, v_j) \leq D_{opt}^k$. Also, $w((x_i, y_i)) \leq D_{opt}^k$, and the theorem follows. \square

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. 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 3 with target diameter $d = 2$. 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 2 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 3 and target diameter 2.

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 of disjoint subsets T_i, T_j of T (where $i \neq j$),
- 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 $y, y' \in Y$,
- yt_i for each pair of vertices $y \in Y$ and $t_i \in T_i$ for each $i \in [m]$ where the corresponding element $s \in S$ is in the corresponding set $x \in X$ in the SET COVER instance,
- $t_i u_{jl}$ for each pair of vertices $t_i \in T_i$ and $u_{jl} \in U$ such that $i \in \{j, l\}$, and
- $u_{ij} u_{lp}$ for each pair of vertices $u_{ij}, u_{lp} \in U$.

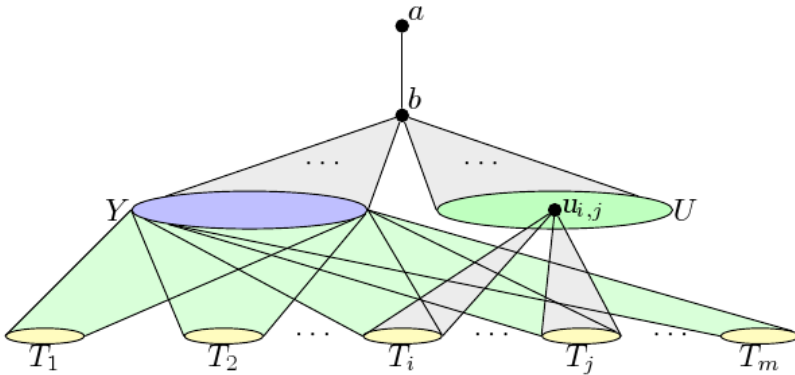


Fig. 3. Sketch of the construction for the SET COVER to U-BCMD reduction. The edge sets represented in gray are complete, the edge sets represented in light green correspond to the set membership from the SET COVER instance. The vertex sets Y and U are cliques. The vertex sets T_i are independent sets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

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

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

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.

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

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

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 $\varepsilon > 0$.*

Acknowledgments. SG acknowledges support from the Australian Research Council (grant DE120101761). JG is funded by the Australian Research Council 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., Gyarfas, A., Ruzsinko, M.: Decreasing the diameter of bounded degree graphs. *Journal of Graph Theory* 35, 161–172 (1999)
2. Bilo, D., Guala, L., Proietti, G.: Improved approximability and non-approximability results for graph diameter decreasing problems. *Theoretical Computer Science* 417, 12–22 (2012)
3. Chepoi, V., Vaxes, 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: Kaplan, H. (ed.) SWAT 2010. LNCS, vol. 6139, pp. 420–431. Springer, Heidelberg (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. Erdos, P., Renyi, A., Sos, 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. 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)
10. 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)
11. Grigorescu, E.: Decreasing the diameter of cycles. *Journal of Graph Theory* 43(4), 299–303 (2003)
12. Kapoor, S., Sarwat, M.: Bounded-diameter minimum-cost graph problems. *Theory of Computing Systems* 41(4), 779–794 (2007)
13. 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)
14. Marx, D.: Parameterized complexity and approximation algorithms. *The Computer Journal* 51(1), 60–78 (2008)
15. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford (2006)
16. Schoone, A.A., Bodlaender, H.L., van Leeuwen, J.: Diameter increase caused by edge deletion. *Journal of Graph Theory* 11, 409–427 (1997)
17. Vazirani, V.V.: Approximation algorithms. Springer (2001)