Diameter Minimization by Shortcutting with Degree Constraints

1st Florian Adriaens
KTH Royal Institute of Technology
Stockholm, Sweden
adriaens@kth.se

2nd Aristides Gionis
KTH Royal Institute of Technology
Stockholm, Sweden
argioni@kth.se

Abstract—We consider the problem of adding a fixed number of new edges to an undirected graph in order to minimize the diameter of the augmented graph, under the constraint that the number of edges added for each vertex is bounded by an integer. The problem is motivated by network-design applications, where we want to minimize the worst case communication in the network without excessively increasing the degree of any single vertex, so as to avoid additional overload. We present three algorithms for this task, each with their own merits. The special case of a matching augmentation—when each vertex can be incident to at most one new edge—is of particular interest, for which we show an inapproximability result, and provide bounds on the smallest achievable diameter when these edges are added to a path. Finally, we empirically evaluate and compare our algorithms on several real-life networks of varying types.

Index Terms—approximation algorithms, network design, edge augmentation, diameter reduction.

I. INTRODUCTION

The diameter of a graph is defined as the greatest distance between any pair of vertices. It is a fundamental notion of a network, measuring the worst-case point-to-point distance in information networks, social networks, and communication networks. Ensuring a small diameter is a crucial property in network-design applications, e.g., minimizing latency in multiprocessor networks [1], or forming a small-world network to maximize the influence of a campaign [2].

There has been a considerable amount of research on the problem of augmenting undirected graphs with new edges in order to minimize the diameter of the resulting graph. This operation has been described in the literature as shortcutting [3]–[6]; and the newly-added edges are referred to as shortcut edges. Li, McCormick and Simchi-Levi [7] introduced the Bounded Cardinality Minimum Diameter (BCMD) problem, where the goal is to add at most $k$ shortcut edges so as to minimize the diameter of the augmented graph.

A potential downside of the BCMD problem formulation is that optimal solutions (and their approximations) might increase the degree of a single vertex by a substantial amount. In fact, this is a guaranteed side effect for solutions obtained by the known approximation algorithms for BCMD (see Section II), as a common theme in many of these algorithms is that they first partition the graph into $k + 1$ non-overlapping parts (the clustering step), and then connect the clusters by picking one cluster center and connecting it to the centers of all other clusters. The latter step is called the star-shortcutting step and adds at most $k$ shortcut edges to the graph, by increasing the degree of a single vertex by $k$. For certain applications, however, it is desirable to limit the increase of the degree of any single vertex. Due to physical, economical, or other limitations, many real-life network problems introduce such a degree constraint [5], [10]. For example, the work of Bokhari and Raza [11] was motivated by a question on how to decrease the diameter of a computer network, by adding additional links, under the constraint that no more than one I/O port is added to each processor. A second example is in the area of social networks: a service provider might be interested in recommending new friendships to users with the aim of increasing the overall connectivity of the network, so as to facilitate information diffusion and reducing polarization [12]–[14]. In [14] the largest distance between two members of different groups (i.e., the colored diameter) has been used as a measure of polarization. Recommending too many friendships to one individual user might not result in many actual links being materialized, due to the risk of overburdening that user. A better strategy might be to limit the number of recommendations per individual user.

Motivated by these application scenarios and the need for a different algorithmic approach than prior algorithms for BCMD, as star-shortcutting is not allowed anymore, we introduce and study a degree-constrained generalization of BCMD (see Problem 1). In this variant, denoted as BCMD-$\delta$, the increase of the degree of each vertex is limited to at most $\delta$. The original BCMD problem corresponds to BCMD-$\delta$ with $\delta = k$. Studying the BCMD problem with degree constraints was left as an open question in the recent work of Tan, E. J. Van Leeuwen and J. Van Leeuwen. [5]. Fig. 1 shows an illustrative example of the BCMD-$\delta$ problem, for different settings of $\delta$. In Fig. 1b we have an unlimited degree increase budget, corresponding to the original BCMD problem. In Fig. 1c we illustrate a solution to the BCMD-$\delta$ problem with $\delta = 1$: the vertex degrees are allowed to increase by at most one, meaning that the newly added edges have to form a matching in the augmented graph.

Preliminaries. All graphs $G = (V, E)$ in this paper are simple, undirected and unweighted. Let $n = |V|$, and $m = |E|$. The distance $d_G(u, v)$ between two vertices $u, v \in V$ is defined as the number of edges in a shortest path between...
Given a graph \( G \) it holds that the eccentricity of at most \( \Delta \) vertices is bounded by \( d \). The eccentricity of a vertex \( v \) is the largest distance between \( v \) and any other vertex. The diameter of the graph is the maximum eccentricity of any vertex. This distance between two sets of vertices is always defined as the shortest distance between any two vertices, one from each set. For \( X \subseteq V \), the induced subgraph \( G[X] \) of \( G \) by \( X \) is the graph whose vertex set is \( X \), and whose edge set consists of all of the edges in \( E \) that have both endpoints in \( X \). In case we omit the subscript in a notation, we always refer to the original graph \( G \).

**Results and outline.** Problem 1 introduces a generalization of the BCMD problem that restricts the maximum degree increase of each vertex. Specifically, given a budget \( \delta \) for each vertex, we ask to augment the graph by adding at most \( k \) new edges so as to minimize the diameter of the augmented graph, and the degree of each vertex increases by at most \( \delta \). All proofs can be found in the extended arXiv version of this paper [15].

**Problem 1 (Bounded Cardinality Minimum Diameter with degree constraints (BCMD-\(\delta\)).** Given a graph \( G = (V, E) \) and integers \( k, \delta \geq 1 \), find a set \( M \) of at most \( k \) non-edges in \( G \) that minimize the diameter of the augmented graph \( G' = (V, E \cup M) \), and for all \( v \in V \) it holds that \( \text{deg}_{G'}(v) \leq \text{deg}_G(v) + \delta \).

In the most restrictive setting, which corresponds to \( \delta = 1 \), each vertex can be incident to at most one shortcut edge, so the augmentation has to form a matching. Since this problem is of particular interest, we denote it as \( \text{BCMD-M} \) (Bounded Cardinality Minimum Diameter with Matching constraint).

Our results are summarized as follows:

1) In Section III we present lower and upper bounds on the optimum value of the \( \text{BCMD-M} \) problem when the input graph is an \( n \)-vertex path, thereby extending a result of Chung and Garey [10] to the setting of a matching augmentation.

2) We give three algorithms for the \( \text{BCMD-}\delta \) problem in Section IV. Section IV-A details a \( O(\log_{k+1} n) \)-approximation algorithm for connected graphs. In Section IV-B we give a constant-factor approximation in the case that \( k \leq \sqrt{n} - 1 \). Section IV-C details an intuitive heuristic without guarantees. All algorithms need \( O(km) \) time, and hence scale quite well for limited values of \( k \).

3) In Section V we show that there exists no \( \left(\frac{4}{3} - \epsilon\right) \)-approximation for \( \text{BCMD-M} \), assuming \( P \neq \text{NP} \).

4) In Section VI we empirically evaluate and compare the performance of our algorithms proposed in Section IV on real-life networks of varying types and sizes.

### II. RELATED WORK

Several constant-factor approximation algorithms for the \( \text{BCMD} \) problem, and weighted variants thereof, are known. Besides proving \( \text{NP} \)-completeness, [7] proposed an elegant \( (4 + \frac{2}{D^*}) \)-approximation algorithm for \( \text{BCMD} \), where \( D^* \) denotes the smallest achievable diameter. [16] refined the performance analysis of [7], and showed that the algorithm in fact guarantees a \( (2 + \frac{2}{D^*}) \)-approximation. [3] considered weighted graphs, with the requirement that all shortcut edges have an equal nonnegative weight, and they provided a \( (4 + \epsilon) \)-approximation for arbitrary \( \epsilon > 0 \). [17] generalized the problem by assigning costs to the shortcut edges, and requiring that the sum of the costs of all shortcut edges does not exceed a given budget parameter. They showed that the general variant of \( \text{BCMD} \), with arbitrary positive edge weights and arbitrary nonnegative shortcut edge costs, admits a \( (2 + \frac{2}{D^*}) \)-approximation if one is allowed to exceed the budget by a factor of \( O(\log k) \). [18] showed that the general variant of \( \text{BCMD} \) admits a fixed-parameter tractable 4-approximation algorithm.

### Prior work on shortcutting with degree constraints.

The problem of shortcutting a graph while respecting degree constraints has been studied before. [10] provided lower and upper bounds on the smallest achievable diameter when edges are added to a path. They posed as an open question to extend their results when there is a constraint on the maximum degree increase of each vertex. In Section III we extend their bounds to the case of a matching augmentation (\( \delta = 1 \)). [19] showed that adding a random matching of maximum possible size to a cycle gives a graph with diameter close to the optimum value, which is about \( \log_2 n \). [11] proved that by adding at most \( \left\lceil \frac{n}{2} \right\rceil \) matching edges, any connected graph can be shortcut to a diameter of \( O(\log n) \). More recently, [5] improved the results of [11] and showed that \( O(\frac{n}{\log n}) \) matching edges are sufficient.
to achieve a diameter of $O(\log n)$. Moreover, they provided several complexity results on the dual problem (see also [18], [20]), i.e., minimizing the number of edges that need to be added to achieve a target diameter, while respecting degree constraints.

III. WARM UP: SHORTCUTTING A PATH WITH MATCHING EDGES

We present several results on the BCMD–M problem when the input graph is an $n$-vertex path. The ideas presented here are intuitive and they serve as a building block for our shortcutting algorithms on general graphs in Section IV.

[10] provided lower and upper bounds on the smallest achievable diameter when $k$ shortcut edges are added to an $n$-vertex path, without any constraints on the maximum increase of the degrees. They proved that it is not possible to achieve a diameter smaller than $\frac{n}{k+1} - 1$, and gave an algorithm that reduces the diameter to $\frac{n}{k+1} + 3$. In the following sections we extend their bounds on the smallest achievable diameter when the vertex-degree increase is limited to one, i.e., when $k$ matching edges are added to a path.

A. Upper and lower bounds

Let $M(n,k)$ denote the smallest achievable diameter after adding $k$ matching edges to an $n$-vertex path. The lower bound of [10] also applies to our setting, and we immediately find that $M(n,k) \geq \frac{n}{k+1} - 1$. On the other hand, it is known that any graph with maximum degree three has diameter at least $\log_2(n) - 2$ [19]. Theorem 1 unifies both results into one lower bound. The upper bound follows from a method that achieves this lower bound, up to constant factors. The main idea is to first divide the path into small connected segments, and then connecting the segments by embedding them into a full 3-tree (see Definition 1 and Figure 2).

**Theorem 1.** For all $1 \leq k \leq n/2$, the smallest achievable diameter after adding $k$ matching edges to an $n$-vertex path satisfies

$$M(n,k) \geq \frac{n}{k+1} + \log_2(k+1) - 2,$$  \hfill (1)

$$M(n,k) \leq \frac{n}{k+1} + 4 \log_2(k+1) + 1.$$  \hfill (2)

**Definition 1.** A full (or Moore) d-tree is an (undirected) rooted tree in which every vertex has degree $d$, and all levels are filled to maximum size except possibly the leaf level [5, Section 2.1.2].

IV. SHORTCUTTING GENERAL GRAPHS

We present three algorithms for shortcutting general graphs with degree constraints. In Section IV-A we present an algorithm that achieves a $O(\log_2+1 k)$-approximation for connected graphs. In Section IV-B we present a constant-factor approximation when $k \leq \sqrt{\delta n} - 1$. In Section IV-C we detail a fast heuristic without approximation guarantees. All three algorithms need $O(km)$ time, making them scalable on most real-life networks when $k$ is not too large.

![Diagram](image)

**Algorithm 1.** Log. approximation BCMD–$\delta$.

**Input:** graph $G$, $k \geq 1$, $\delta \geq 1$ and $3 \leq \beta \leq n$.

1. $X \leftarrow \emptyset$
2. $C = \{ \text{a maximal family of disjoint } \beta\text{-segments} \}$
3. for $\min\{k + 1, |C|\}$ times do
4. Pick segment $C^\ast \in C \setminus X$ with $d_G(C^\ast, X) = \max_{C \subseteq C \setminus X} d_G(C, X)$,
5. $X \leftarrow X \cup \{ C^\ast \}$
6. end for
7. Connect the segments in $X$ into a full $d$-tree with $d = \beta, \delta$, such that every vertex has at most $\delta$ incident shortcut edges. See Fig. 2b for an example with $\delta = 1$ and $\beta = 3$.

A. A $O(\log_2+1 k)$-approximation for connected graphs.

The high-level strategy is reminiscent of the path case. After finding at most $k + 1$ small connected segments in the graph, we embed them into a full tree by connecting the segments using at most $k$ shortcut edges. The branching factor of the tree will depend on the degree budget $\delta$ and the size of the segments. The main difference with the path case is that we will require the segments to be far apart. Our strategy consists of first finding a maximal family of vertex-disjoint segments. From this family, we select a set of at most $k + 1$ segments that are far apart, using a similar idea as the $k$-center heuristic of Dyer and Frieze [21], [22].

We start by formally defining a segment.

**Definition 2.** A $\beta$-segment $C$, for $1 \leq \beta \leq n$, is a set of vertices $C \subseteq V$ with $|C| = \beta$, such that the induced subgraph $G[C]$ is connected.

Theorem 2 provides the approximation guarantee of Algorithm 1.

**Theorem 2.** Assume $G$ is connected. Algorithm 1 is a $O(\log_2+1 k)$-approximation for the BCMD–$\delta$ problem, for the choice $\beta = 3$.

**Running time.** Algorithm 1 needs $O(km)$ time for connected graphs when $\beta = 3$. Finding a maximal family of $3$-segments (line 2) can be done in linear time $O(m)$ (see e.g., [5, Section 3.1]). Lines 3–5 can be implemented in $O(km)$ time (similarly as in the $k$-center heuristic [21]); in every iteration, keep a dictionary of the distances from all segments to the current set $X$. After adding $C^\ast$ to $X$, we update the dictionary efficiently by first computing the distances from $C^\ast$ to all other segments (by doing a BFS in $O(m)$ time), and then for each segment.
Algorithm 2 Constant approximation BCMD–δ.

Input: graph \( G \), \( 1 \leq k \leq \sqrt{\delta n} - 1 \), \( \delta \geq 1 \).
1: \( \{c_i\} = k+1 \) cluster centers computed according to [21].
2: \( C_{\text{max}} \) \( \leftarrow \) largest cluster (center).
3: \( \forall c_i \neq c_{\text{max}}: \) add edge \( \{c_i, v\} \) for some \( v \in C_{\text{max}} \) s.t. \( v \) is incident to less than \( \delta \) new shortcut edges.

Taking the minimum of the old dictionary value and this newly computed distance.

B. Constant-factor approximation for small \( k \)

In case when \( k \leq \sqrt{\delta n} - 1 \), we obtain a constant-factor approximation for the BCMD–δ problem, even for disconnected graphs, using a slightly different approach. Algorithm 2 first computes \( k+1 \) non-necessarily distinct cluster centers (line 1), again according to the \( k \)-center heuristic of Dyer and Frieze [21], [22]. In particular, the clustering phase (line 1) computes a set of \( k+1 \) non-necessarily distinct cluster centers \( C = \{c_1, \ldots, c_{k+1}\} \subseteq V \) as follows: the first center \( c_1 \) is an arbitrary vertex. For \( 2 \leq i \leq k+1 \), the center \( c_i \) is chosen as a vertex that maximizes the distance \( d_{c_i}(c_1, \{c_1, \ldots, c_{i-1}\}) \). The cluster partitions are defined by assigning every non-center vertex to the nearest center, breaking ties arbitrarily. Let \( D_{k,\delta} \) denote the optimum value of BCMD–δ. It is straightforward to show that the radius of each cluster is a lower bound of \( D_{k,\delta} \). Moreover, if \( k \) is small (or \( \delta \) is large), we can prove that the largest cluster contains enough vertices to shortcut to all other centers, leading to a constant-factor approximation. Lemma 1 formalizes this.

Lemma 1. If \( k \leq \sqrt{\delta n} - 1 \), Algorithm 2 is a \((4 + 2/D_{k,\delta})\)-approximation algorithm for BCMD–δ.

Running time. The running time of Algorithm 2 is \( O(km) \), since the bottleneck is the \( k \)-center heuristic from [21].

C. Greedy 2-Sweep heuristic

Lastly, we propose a third intuitive heuristic that works well in practice, but comes without any approximation guarantees. It is inspired by the fast heuristic lower bound called 2-Sweep [23], which is used to estimate the diameter in large graphs. Algorithm 3 picks a vertex \( u \) that is furthest away from a randomly chosen vertex, such that \( u \) has less than \( \delta \) shortcut edges incident to it. Then it picks a vertex \( v \) that is furthest away from \( u \), also incident to less than \( \delta \) shortcut edges. Add the shortcut edge \( \{u, v\} \). Repeat \( k \) times. Running time of this heuristic is \( O(km) \), similar to the previous algorithms.

V. INAPPROXIMABILITY OF BCMD–M

Using a reduction from the NP-complete SetCover problem [24] to BCMD–M, we show that one cannot get an approximation ratio better than 4/3. Clearly the result also holds for the BCMD–δ problem, as it includes BCMD–M as a special instance.
Fig. 3. The performance of the proposed algorithms for various settings of $k$ and $\delta$. The interruption of the blue colored plot (constant-factor approximation algorithm) indicates that the criterion for the algorithm to run is not fulfilled, as discussed in Section IV-B.
on the running time roughly leads to doubling the running time. This does not seem to be that large, but the time complexity of our methods is 2^5 \cdot k = 1024 on the two largest datasets. Doubling the number of shortcuts and δ = 1 does not meet the criteria to run. This happens whenever the largest cluster does not have enough vertices (with enough budget) to shortcut to the remaining clusters. In Section IV-B we showed that the algorithm is guaranteed to work if k ≤ \sqrt{\delta n} – 1, but in practice the algorithm did run for larger values of k.

Running time. Fig. 4 shows the running time as a function of k on the two largest datasets. Doubling the number of edges k roughly leads to doubling the running time. This corresponds to the O(kn^2) time complexity of our methods and the baseline. The influence of δ on the running time is negligible.

Acknowledgment

This research is supported by the Academy of Finland projects AIDA (317085) and MLDB (325117), the ERC Advanced Grant REBOUND (834862), the EC H2020 RIA project SoBigData++ (871042), and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[1] L. Benini and G. De Micheli, “Networks on chips: A new SoC paradigm,” Computer, vol. 35, no. 1, pp. 70–78, 2002.
[2] N. Laoutaris, L. J. Poplawski, R. Rajaraman, R. Sundaram, and S.-H. Teng, “Bounded budget connection (BBC) games or how to make friends and influence people, on a budget,” in Proceedings of the 27th ACM symposium on Principles of distributed computing (PODC), 2008.
[3] E. Demaine and M. Zadimoghadam, “Minimizing the diameter of a network using shortcut edges,” in SWoAT, 2010.
[4] A. Meyerson and B. Tagiku, “Minimizing average shortest path distances via shortcut edge addition,” in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 2009, pp. 272–285.
[5] R. B. Tan, E. J. Van Leeuwen, and J. Van Leeuwen, “Shortcutting directed and undirected networks with a degree constraint,” Discrete Applied Mathematics, vol. 220, pp. 91–117, 2017.
[6] N. Chepoi and Y. Vaxes, “Augmenting trees to meet biconnectivity and diameter constraints,” Algorithmica, vol. 33, no. 2, pp. 243–262, 2002.
[7] C. Li, S. McCormick, and D. Simchi-Levi, “On the minimum-cardinality-bounded-diameter and the bounded-cardinality-minimum-diameter edge addition problems,” Operations Research Letters, 1992.
[8] Y. Ma, A., W. W. D., H. W. J., and W. C., “Cluster of cases of the acquired immune deficiency syndrome. patients linked by sexual contact,” The American Journal of Medicine, 1984.
[9] J. Kunegis, “Konec: The koblenz network collection,” in 22nd Int. Conf World Wide Web, 2013, pp. 1343–1350.
[10] F. R. K. Chung and M. D. Gardner, “Diameter bounds for altered graphs,” Journal of Graph Theory, vol. 8, no. 4, pp. 511–534, 1984.
[11] S. H. Bokhari and A. D. Raza, “Reducing the diameters of computer networks,” IEEE Trans. Comput., vol. 35, no. 8, pp. 757–761, 1986.
[12] K. Garamella, G. De Franciscis Morales, A. Gionis, and M. Mathioudakis, “Reducing controversy by connecting opposing views,” in Web Search and Data Mining, 2017, pp. 81–90.
[13] S. Haddadan, C. Menghini, M. Riondato, and E. Upfal, “Repubhlik: Reducing polarized bubble radius with link insertions,” in 14th ACM International Conference on Web Search and Data Mining, 2021.
[14] R. Interian, J. Moreno, and C. Ribeiro, “Polarization reduction by minimum-cardinality edge additions: Complexity and integer programming approaches,” Int. Transactions in OR, vol. 28, 2020.
[15] F. Adriaens and A. Gionis, “Diameter minimization by shortcutting with degree constraints,” 2022. [Online]. Available: https://arxiv.org/abs/2209.00370
[16] D. Bili, L. Günl, and G. Piovetto, “Improved approximability and non-approximability results for graph diameter decreasing problems,” Theoretical Computer Science, vol. 417, pp. 12–22, 2012.
[17] D. Libra, D. Chudamani, and D. Upfal, “Reducing polarized bubble radius with link insertions,” in 14th ACM International Conference on Web Search and Data Mining, 2021.
[18] R. Interian, J. Moreno, and C. Ribeiro, “Polarization reduction by minimum-cardinality edge additions: Complexity and integer programming approaches,” Int. Transactions in OR, vol. 28, 2020.
[19] F. Adriaens and A. Gionis, “Diameter minimization by shortcutting with degree constraints,” 2022. [Online]. Available: https://arxiv.org/abs/2209.00370
[20] D. Bili, L. Günl, and G. Piovetto, “Improved approximability and non-approximability results for graph diameter decreasing problems,” Theoretical Computer Science, vol. 417, pp. 12–22, 2012.
[21] D. Dodis and S. Khandla, “Designing networks with bounded pairwise distance,” in 31st annual ACM STOC, 1999, pp. 750–759.
[22] F. Fraui, S. Gaspars, J. Gudmundsson, and L. Mathieson, “Augmenting graphs to minimize the diameter,” Algorithmica, 2015.
[23] B. Bollobás and F. R. K. Chung, “The diameter of a cycle plus a random matching,” SIAM Journal on Discrete Mathematics, 1988.
[24] Y. Gao, D. R. Hare, and J. Nastos, “The parametric complexity of graph diameter augmentation,” Discrete Applied Mathematics, vol. 161, no. 10, pp. 1626–1631, 2013.
[25] M. Dyer and A. Frieze, “A simple heuristic for the p-centre problem,” Operations Research Letters, vol. 3, no. 6, pp. 285–288, 1985.
[26] T. F. Gonzalez, “Clustering to minimize the maximum intercluster distance,” Theoretical Computer Science, vol. 38, pp. 293–306, 1985.
[27] C. Magnien, M. Latapy, and M. Habib, “Fast computation of empirically tight bounds for the diameter of massive graphs,” Journal of Experimental Algorithmics, vol. 13, pp. 1–10, 2009.
[28] R. M. Karp, Reducibility among Combinatorial Problems. Boston, MA: Springer US, 1972, pp. 85–103.
[29] P. Crescenzi, R. Grossi, L. Lanzi, and A. Marino, “On computing the diameter of real-world directed (weighted) graphs,” in International Symposium on Experimental Algorithms, 2012, pp. 99–110.
[30] E. Cho, S. A. Myers, and J. Leskovec, “Friendship and mobility: User movement in location-based social networks,” in Proceedings of the 17th ACM SIGKDD KDD, 2011.
[31] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D’Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, L. Matthews et al., “Reactome: a knowledgebase of biological pathways,” Nucleic Acids Research, vol. 33, pp. D428–D432, 2005.
[32] D. J. Watts and S. H. Strogatz, “Collective dynamics of small-world networks,” Nature, vol. 393, no. 6684, pp. 440–442, 1998.
[33] J. Leskovec and A. Krevl, “SNAP Datasets,” http://snap.stanford.edu/