Technische Universität Berlin

Fakultät IV
Institut für Algorithmik und Komplexitätstheorie

Master Thesis

Algorithms and Complexity for Centrality Improvement in Networks

Clemens Hoffmann

Supervisor:
Prof. Rolf Niedermeier

Co-Advisors:
Manuel Sorge and Hendrik Molter

April 24, 2017

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Speyer, den 23.8.2017

____

Clemens Hoffmann

Abstract

The task of improving the centrality of a node in a network has many applications, as a higher centrality often implies a larger impact on the network or less transportation or administration cost. This work studies the parameterized complexity of the NP-hard problem of improving a node's closeness and betweenness centrality by adding a certain number of edges to the network, denoted as CLOSENESS IMPROVEMENT and BETWEENNESS IMPROVEMENT. We show similarities between CLOSENESS IMPROVEMENT and MINIMUM DOMINATING SET. On the negative side, we show that in general, both problems are W[2]-hard if parameterized by the number of edge additions, even on graphs with constant diameter; for CLOSENESS IMPROVEMENT, we show W[2]-hardness even on split graphs. Furthermore, we show W[2]-hardness for the problem variants on directed graphs. On the positive side, we show fixed-parameter tractability for the parameters vertex cover size, distance to clique, and distance to cluster graph. Finally, we show W[1]-hardness for the problem of improving the betweenness centrality by deleting edges from the graph, and provide an outlook on problem variants such as improving a node's relative closeness centrality by removing a certain number of edges.

Abstrakt

Die Zentralität eines Knotens in einem Netzwerk kann auf vielfältige Art und Weise gemessen werden - zu den bekanntesten Verfahren zählen die Betweenness und die Closeness Centrality. Diese Zentralität durch das Einfügen von Kanten zu erhöhen, ist in vielerlei Anwendungen von Nutzen. Beispielsweise haben Knoten mit hoher Zentralität in sozialen Netzwerken einen hohen Einfluss innerhalb des Netzwerks. Für Transport- und Logistikunternehmen ist es von Interesse, Lager möglichst zentral zu legen, um Kosten zu sparen. Diese Abschlussarbeit untersucht die parametrisierte Komplexität der NP-harten Probleme der Maximierung der Betweenness und Closeness Centrality durch das Hinzufügen von einer begrenzten Anzahl an Kanten im Netzwerk, sowohl auf gerichteten als auch auf ungerichteten Graphen. Darüber hinaus werden Ähnlichkeiten der Probleme Dominating Set und Closeness Improvement aufgezeigt. Zu den zentralen Resultaten dieser Arbeit zählt der Beweis der W[2]-Härte beider Probleme sowohl auf gerichteten als auch auf ungerichteten Graphen, selbst mit konstantem Durchmesser und auf einigen eingeschränkten Graphklassen. Darüber hinaus wird gezeigt, dass das Problem der Maximierung der Closeness Centrality mit gewissen Parametern in FPT oder sogar in Polynomialzeit lösbar ist. Des Weiteren wird das Problem der Maximierung der Betweenness Centrality durch das Löschen von Kanten eingeführt und W[1]-Härte mit dem Parameter Anzahl der zu löschenden Kanten gezeigt. Als Ausblick werden eine Reihe weiterer Problemvarianten sowie offene Fragestellungen aufgeführt.

Contents

Chapter 1

Introduction/Motivation

Measuring the centrality of nodes in a network has attracted the interest of researchers since the second half of the 20th century. There are various interpretations of what makes a node more central than another node in a network (Freeman [Fre78]). Three popular centrality measurements are betweenness, closeness and node degree (Opsahl et al. [OAS10]). They have in common that centrality reflects the importance of a node based on structural properties of a node's location and embedding in the graph: For example, a node which has short distances to other nodes can be considered as central, while a node with very few neighbors located at a network's boundary is rather peripheral.

Analyzing network centrality has been studied intensively (e.g. Freeman [Fre78], Okamoto et al. [OCL08], Newman [New05], Brandes [Bra08]) and comprises a huge set of applications, e.g. network analysis (Newman and Girvan [NG04]) and the field of bioinformatics (Rubinov and Sporns [RS10]). Besides the academic interest, the model of centrality also fits many economical scenarios. For instance, a transport company might be interested in parking its vehicles and goods in a central depot such that the transportation costs are rather low. Hence, the company's task is to select central depots. The value of an airport might be impacted by the number of connections to other airports. Furthermore, analyzing the network centrality is also important in the field of computer science. For instance, determining the most central nodes in a computer network is useful for determining attractive data centers where the routes are rather short and peering costs are rather low. In social networks, influencers are somehow more central than other users. As there is no natural answer how to measure centrality, several indicators have been used in order to determine a node's centrality.

Two of these indicators are subject of the research within this work: The first one, referred to as *closeness centrality*, takes the sum of all distances to other nodes into account. By summing up the multiplicative inverses of these distances, a higher value means that on average the distances to other nodes are rather low. The usage of the multiplicative inverses of the distances is a solution for measuring the centrality of a node in a disconnected graph, as by convention, the distance between disconnected nodes is ∞. Using the multiplicative inverses, each node increases the closeness centrality by at most one. For a node z, it is formally defined as follows:

$$c_z = \sum_{\substack{u \in V \\ d(u,z) < \infty \\ u \neq z}} \frac{1}{d(u,z)} \tag{1.1}$$

However, the closeness centrality of a node does not necessarily determine its importance: Assume that a node has short distances to many isolated nodes, but its distances to other central nodes with a high degree are comparatively large. Its closeness centrality value might still be high, although from a structural perspective, the centrality of the node would be low. This deficit leads to another approach for measuring centrality which takes the proportional number of shortest paths containing z into account. This centrality is referred to as the *betweenness centrality*, where σ_{st} is the number of shortest path from s to t and σ_{stz} is the number of shortest paths from s to t containing z. For an undirected graph, the betweenness centrality is defined as follows, where z is a node of the graph:

$$b_z = \sum_{\substack{s,t \in V \\ s \neq t; s,t \neq z \\ \sigma_{st} \neq 0}} \frac{\sigma_{stz}}{\sigma_{st}} \tag{1.2}$$

Polynomial-time algorithms for measuring the centrality have been intensively studied since the 1970s. Modern approaches like scalable heuristics for very large networks have also been subject of research [Kan+11]. However, besides measuring a node's centrality, there are a lot of scenarios where the improvement of a node's centrality is desirable. E.g., a social network member might want to increase her impact on other users by increasing her own centrality. Another motivation for this problem is an airport operator who wants to increase the number of flights from and to her airport: The most natural way to increase this number is by increasing the number of shortest flights with intermediate landings at the airport.

In this work, we address the problem of *improving* the centrality of a node by performing a limited number of edge operations. Hence, we introduce and mainly focus on the following decision problems.

For the *closeness* centrality, we ask whether the centrality of a node z can be increased to a specified value r by adding k edges:

CLOSENESS IMPROVEMENT

Input: An undirected, unweighted graph $G = (V, E)$, a node $z \in V$, an integer k and a rational number r.

Question: Is there an edge set S, $S \cap E = \emptyset$, of size at most k such that $c_z \geq r$ in $G' = (V, E \cup S)$?

In the same way, we ask whether the *betweenness* centrality of a node z can be increased to a specified value r by adding k edges:

BETWEENNESS IMPROVEMENT

Input: An undirected, unweighted graph $G = (V, E)$, a node $z \in V$, an integer k and a rational number r.

Question: Is there an edge set S, $S \cap E = \emptyset$, of size at most k such that $b_z \geq r$ in $G' = (V, E \cup S)$?

By removing edges from a network, the closeness centrality of a node can only be decreased. The same does not hold for betweenness centrality: Assume that there are several shortest paths from s to t, some of which contain z and some do not. Then b_z can be increased by removing edges which are part of a shortest path not containing z. The model of removing edges in order to increase a node's centrality may be desirable in some situations: For instance, a network administrator might want some network traffic to pass a specific, monitored node by shutting down as few links as necessary. Hence, we additionally introduce and analyze the DESTRUCTIVE BETWEENNESS IMPROVEMENT problem, where we ask whether the betweenness centrality of a node z can be increased to a specified value r by removing at most k edges:

DESTRUCTIVE BETWEENNESS IMPROVEMENT

Input: An undirected, unweighted graph $G = (V, E)$, a node $z \in V$, an integer k and a real number r.

Question: Is there an edge set $S \subseteq E$ of size at most k such that $b_z \geq r$ in $G' = (V, E \setminus S)$?

In this work, we analyze the parameterized complexity of each of the introduced decision problems: First, we show that all problems are W[2]-hard with respect to the perhaps most natural parameter, i.e. the number of edges that are added or removed respectively. Moreover, we show W[1]-hardness or membership in the class FPT for other parameters and some restricted graph classes. For the natural parameter, we also show W[2]-hardness for the problem of maximally increasing the closeness or betweenness on directed, unweighted graphs. Moreover, for each of the introduced decision problems, we analyze the parametrized complexity of the variants where the input graph is a directed graph, and the task is to increase a node's centrality by adding or removing arcs to or from the input graph respectively. Therefore, we also define centrality in directed graphs in the corresponding sections.

1.1 Related Work

The field of centrality measurement has been widely studied in the past. The first work describing the ideas of betweenness centrality measurement is due to Bavelas [Bav48]. Three decades later, the current formal model of betweenness centrality has been defined by Freeman [Fre77]. The same work also provides a polynomial-time algorithm for measuring the betweenness centrality running in $\mathcal{O}(n^3)$ time. In the past years, the problem of efficiently measuring the betweenness centrality has been actively researched. For instance Brandes [Bra01] presents an algorithm for undirected graphs running in $\mathcal{O}(nm)$ time. A recent work discussing problem variants is due to Brandes [Bra08], such as *bounded-distance* betweenness and *edge* betweenness. Recently, randomized algorithms for approximating the betweenness centrality and some variants are due to Riondato and Kornaropoulos [RK16] and .

The problem of improving the betweenness or closeness centrality of a node by adding a limited number of edges was initiated by Crescenzi et al. [Cre+16] and D'Angelo et al. [DSV16]. They show that both problems, formulated as maximization problems where the task is to maximally improve a node's closeness or betweenness centrality by a specific number of edges, do not admit a polynomial-time approximation scheme. Furthermore, they present a factor $1 - \frac{1}{e}$ approximation for closeness improvement on directed graphs, and show that the problem cannot be approximated by a factor greater than $1 - \frac{1}{3e}$ (factor $1 - \frac{1}{2e}$ for betweenness improvement on directed, unweighted and undirected, weighted graphs, respectively), unless P $\neq$ NP. Furthermore, they introduce polynomial-time greedy strategies and present experimental results.

1.2 Our contribution

In this work, we present results for the parameterized complexity of centrality improvement problems. The highlights of this work are found in Chapter 2 and Chapter 3: In Chapter 2, the subject of research is the parameterized complexity of CLOSENESS IMPROVEMENT: We show that the problem is W[2]-hard with respect to the number of added edges, even on split graphs. For other parameters such as vertex cover size and distance to cluster graph, we show membership in the complexity class FPT. Chapter 3 covers results and observations for BETWEENNESS IMPROVEMENT: Analogously to the previous chapter, we show W[2]-hardness for the parameter number of

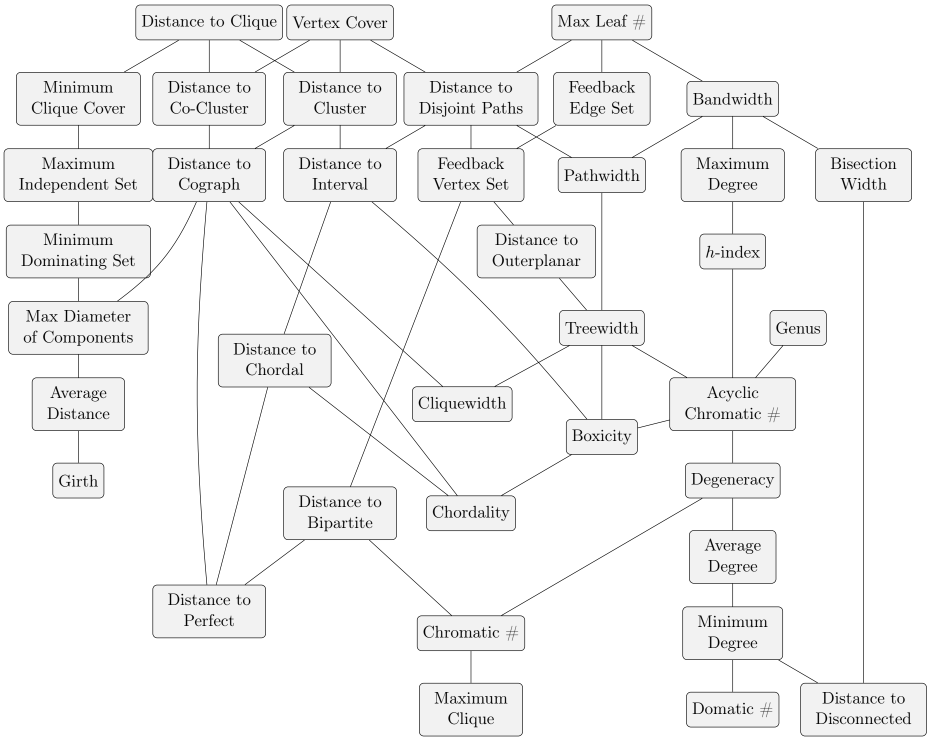


Figure 1: Hasse diagram of the known part of the boundedness relation between graph parameters. For two parameters that a connected by a line, the lower parameter is upper bounded by the upper parameter (that is, the upper parameter is larger). "Distance to X" is the number of nodes that have to be deleted in order to transform the input graph into a graph from the graph class X. Refer to Sorge and Weller [SW17] for the formal definitions of the parameters.

added edges, and membership in FPT for the parameter vertex cover size and the combined parameter distance to cluster graph and number of added edges.

For both problems, we introduce a variant where the input graph is a directed, unweighted graph. We also show W[2]-hardness for these problems with respect to the number of added arcs in Section 2.3 respectively Section 3.3. In Chapter 4, we introduce some variants of Betweenness Improvement such as Directed Betweenness Improvement, where the task is to remove edges from a graph in order to improve the betweenness centrality. We show that this problem is W[1]-hard on directed graphs with respect to the number of removed arcs.

Problem	Parameter	Results
Closeness Improvement	number of edge additions	W[2]-hard in general (Theorem 1)
		XP in general Corollary 1
		W[2]-hard with diameter 4 (Corollary 4)
		W[2]-hard for split graphs (Theorem 3)
		NP-hard on planar graphs with max degree 3 (Corollary 3)
		P on cluster graphs (Lemma 6)
		P with diameter at most 2 (Lemma 7)
	vertex cover size	FPT in general (Theorem 5)
	distance to clique	FPT in general (Theorem 4)
	distance to cluster	FPT in general (Theorem 6)
Directed Closeness Improvement	number of arc additions	W[2]-hard in general (Theorem 7)
		XP in general (Lemma 12)
		W[2]-hard on DAGs Corollary 5
		W[2]-hard with diameter 4 (Theorem 8)
		P with diameter at most 2 (Lemma 13)
Betweenness Improvement	number of edge additions	W[2]-hard in general (Theorem 9)
		XP in general (Corollary 6)
		NP-hard with max. h-index 4 (Corollary 7)
	vertex cover and number of edge additions	FPT in general (Theorem 10)
Directed Betweenness Improvement	number of arc additions	W[2]-hard in general (Theorem 7)
		XP in general (Corollary 8)
		W[2]-hard on DAGs (Corollary 9)
Directed Destructive Betweenness Improvement	number of arc subtractions	W[1]-hard (Theorem 12)

Table 1: Overview of our classification results.

An overview of all results of our complexity analysis can be found in Table 1. Herein, in general denotes that the result is valid for unrestricted graphs. As Closeness Improvement and Betweenness Improvement are W[2]-hard with respect to the number of edge additions, we analyze the parameterized complexity with other parameters, such as distance to clique, vertex cover size and distance to cluster graph. The main purpose is to locate each problem in the parameter hierarchy visualized in Figure 1 in order to be able to compare the centrality improvement problems with other problems in terms of parameterized complexity. As Dominating Set is fixed-parameter tractable with respect to distance to clique and vertex cover size, there was a strong motivation in comparing Closeness Improvement to this problem due to the relationship between these two problems as shown in Chapter 2. Moreover, we analyze the hardness of Closeness Improvement on split graphs, as this graph class is the perhaps most simple one with a core-periphery structure, which can be found in social and transportation networks (Rombach et al. [Rom+14]).

1.3 Preliminaries

This section covers the notation and basics of this work. However, we presume that the reader is familiar with the theoretical foundations of computer science, such as NP-completeness and graph theory.

1.3.1 Graph theory and basics

Graphs. An undirected graph $G = (V, E)$ is a tuple where V is a set of nodes and $E \subseteq \binom{V}{2}$ is a set of undirected edges. A directed graph $G' = (V, A)$ consists of a set of nodes V and a set of directed arcs $A \subseteq V \times V$. The number of nodes is referred to as n while the number of edges respectively arcs is referred to as m. For a node u of an undirected graph, $\deg(u)$ denotes the degree of u. For a graph $G = (V, E)$ and $S \subseteq V$, we refer to the induced subgraph $G' = (V \setminus S, E \setminus \{\{u, v\} \mid u \in S \vee v \in S)\}$ as $G - S$.

Distance. For two nodes $u, v, d(u, v)$ denotes the distance between u and v, i.e. the length of a shortest path from u to v. If u and v are not connected by a path, then $d(u, v) = \infty$.

Neighborhood. For a graph $G = (V, E)$, the *neighborhood* of a node u is the set of nodes adjacent the u. The *open* neighborhood of a node $u \in V$, that is the set of nodes adjacent to u not including u, is denoted by $N(u)$. The *closed* neighborhood, that is the set of neighbors of u which includes u, is denoted by $N[u]$. For any subset $V' \subseteq V$, we define $N_{V'}(u) := N(u) \cap V'$, and $N_{V'}[u] := N[v] \cap V'$.

Isolation. A node is isolated if and only if its degree in the graph is 0. Otherwise.

Cluster Graphs. A *cluster graph* is a graph which consists of a disjoint union of cliques, referred to as *clusters*.

Split Graphs. A *split graph* is a graph which can be partitioned into a clique and an independent set.

Independent Set. For an undirected graph $G = (V, E)$, an *independent set* is a set of nodes $V' \subseteq V$ such that $\{v, v'\} \notin E$ for all $v, v' \in V'$.

H-index. A graph has *h-index* k if there are at least k nodes with degree at least k.

Edge additions. For *Betweenness Improvement* and CLOSENESS IMPROVEMENT, a *solution* for an instance $I = (G = (V, E), z, k, r)$ is a set of edges S of size at most k such that the betweenness resp. closeness centrality of z in $G' = (V, E \cup S)$ is at least r. Sometimes we denote the number of edge additions k as the *natural parameter* of these problems. These terms also apply to the problem variants on directed graphs.

1.3.2 Parameterized complexity

Parameterized complexity. The motivation behind parameterized complexity (Niedermeier [Nie06], Downey and Fellows [DF13], Cygan et al. [Cyg+15], Flum and Grohe [FG06]) is to solve NP-hard decision problems more efficiently by restricting the exponential running time of an algorithm to one or more *parameters* of an instance, which are independent of the input size. Formally, in parameterized complexity theory (Downey and Fellows [DF13]), a language $L \subseteq \Sigma^* \times \Sigma^*$ is a *parameterized language*, where Σ is a finite alphabet. For an instance $(x, k) \in L$, the second component k is the *parameter*.

Parameterized languages. A parameterized language L is *fixed-parameter tractable* if and only if there is a computable function f that only depends on k such that it can be determined in $f(k) \cdot n^{\mathcal{O}(1)}$ time whether $(x, k) \in L$, where n is the input size. The corresponding complexity class of fixed-parameter tractable languages is FPT.

Some prominent fixed-parameter tractable languages are VERTEX COVER or FEEDBACK VERTEX SET, both with respect to the parameter solution size (see Section 1.3.3) for the problem definitions). Analogously to the assumption that P $\neq$ NP, we assume that many problems with natural parameters are not fixed-parameter tractable, such as DOMINATING SET and SET COVER. These problems are members of the complexity classes W[t] for a $t \geq 1$. In particular, DOMINATING SET and SET COVER are W[2]-complete, while for instance MAXIMUM CLIQUE and MAXIMUM INDEPENDENT SET are W[1]-complete.

W-Hierarchy. The *W-Hierarchy* is defined as follows ([DF13]):
FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \ldots \subseteq$ W[t] $\subseteq \ldots \subseteq$ W[Sat] $\subseteq$ W[P] $\subseteq$ XP.

Parameterized reductions. In order to classify parameterized languages in the parameterized complexity hierarchy, the framework of *parameterized reductions* is a useful tool:

Let $L, L' \subseteq \Sigma^* \times \Sigma^*$. A parameterized (many-to-one) reduction from L to L' is a function $f : \Sigma^* \times \Sigma^* \to \Sigma^* \times \Sigma^* : (I, k) \to (I', k')$ such that:

(i) $f(I, k)$ can be computed in $g(k) \cdot |I|^{\mathcal{O}(1)}$ time for some computable function g,

(ii) $k' \leq h(k)$ for some computable function h and
(iii) $(I, k) \in L \Leftrightarrow (I', k') \in L'$.

For instance, we can show that a parameterized language L is W[1]-hard by giving a parameterized reduction from a W[1]-hard problem, for instance MAXIMUM CLIQUE, to L. In order to show that $L \in$ W[1], we can give a parameterized reduction from MAXIMUM CLIQUE to L. If both reductions exist, then L is *W[1]-complete*.

XP. Finally, a parameterized problem L is in XP if it can be determined in $f(k) \cdot |I|^{\mathcal{O}(g(k)))}$ time whether $(I, k) \in L$, where f and g are computable functions only depending on k.

1.3.3 Decision problems

We introduce some decision problems, which are used in reductions within this work.

DOMINATING SET is a graph problem known to be W[2]-hard with respect to the parameter solution size. We use parameterized reductions from DOMINATING SET in order to prove W[2]-hardness for some centrality improvement problems.

DOMINATING SET

Input: An unweighted graph $G = (V, E)$ and an integer k.

Question: Is there a set of nodes $V' \subseteq V$ of size at most k such that each node not in V' has at least one neighbor in V'?

Furthermore we introduce SET COVER which is W[2]-hard with respect to the parameter solution size k:

SET COVER

Input: A family of subsets $S = \{S_1, \ldots, S_m\}$ over a universe $X = \{x_1, \ldots, x_n\}$ and an integer k.

Question: Is there a set $S' \subseteq S, |S'| \leq k$ such that $\bigcup_{R \in S'} R = X$?

By a slight modification where we ask if at least p elements can be covered by k sets, we obtain the following problem:

MAX k-SET COVER

Input: A family of subsets $S = \{S_1, \ldots, S_m\}$ over a universe of elements $X = \{x_1, \ldots, x_n\}$, integers k and p.

Question: Is there a set $S' \subseteq S$ of families of size at most k such that $|\bigcup_{R \in S'} R| = p$?

The W[2]-hardness of Set Cover with the parameter size k of subsets directly implies the W[2]-hardness of Max k-Set Cover by setting $p := n$.

Vertex Cover is fixed-parameter tractable with respect to the parameter k and the fastest algorithm at the time of writing runs in $\mathcal{O}(1.2738^k + kn)$ time [CKX10].

Vertex Cover

Input: An undirected graph $G = (V, E)$ and a positive integer k.

Question: Is there a $V' \subseteq V$ of size at most k such that for all $\{u, v\} \in E$ either $u \in V'$ or $v \in V'$?

Chapter 2

Closeness Centrality Improvement

This chapter presents algorithmic and hardness results for Closeness Improvement in general and for different graph classes. First, we show that the closeness centrality improvement of a node by a certain number of edge additions is maximal if each added edge contains z. We use this basic observation for showing the correctness of the hardness reductions within this chapter.

On the negative side, we use parameterized reductions from Dominating Set, Set Cover, and Max k-Set Cover to show W[2]-hardness for the parameters number of edge additions on undirected, unweighted graphs and number of arc additions on directed, unweighted graphs. Moreover, we show that Closeness Improvement is W[2]-hard even on split graphs.

On the positive side, we provide algorithms to show that Closeness Improvement is fixed-parameter tractable with respect to the vertex cover size and the distance to cluster, that is the number of nodes to remove such that the resulting graph is a cluster graph. As the running time of these algorithms circumvents practical usability, the main purpose is to give a classification in the parameterized complexity hierarchy.

Furthermore, we show that there is a relation between Closeness Improvement and Dominating Set. Let G be an undirected graph and k be a positive integer. If a dominating set for G of size at most k is computable in polynomial time, then we can solve a Closeness Improvement instance (G, z, k, r) in polynomial time. We analyze this observation and its implications; furthermore, we show similarities and differences between these two problems in terms of parameterized complexity.

We first introduce some general propositions and theorems that are used throughout the following proofs.

This first proposition is used to determine the running time of the algorithms presented in Section 2.2:

Proposition 1. *The closeness centrality of a node z in a connected, undirected, unweighted graph can be computed in $\mathcal{O}(n+m)$ time.*

Proof. By summing up the multiplicative inverses of all distances between z and all other nodes, we get the closeness centrality of z. The problem of finding the shortest paths between one source and all other nodes is the Single-Source-Shortest-Path problem which can be solved in $\mathcal{O}(n+m)$ time [Tho99]. □

The next observation is that the closeness centrality of a node is maximal if it is connected by a edge to all other nodes, independent of the structure of the graph.

Lemma 1. *The closeness centrality c_z of a node z in an undirected, unweighted graph is maximal if and only if z is adjacent to all other nodes in a graph.*

Proof. If z is adjacent to all other nodes, then the distance between z and each other node is 1 and therefore minimal, hence $c_z = n-1$. If there is at least one node which z is not adjacent to, then its distance is at least 2 and therefore $c_z \leq n-2+\frac{1}{2} < n-1$. □

We use the following theorem to proof the correctness of the hardness reductions in Section 2.1.

Lemma 2. *Let $I = (G = (V,E), z, k, r)$ be a* Closeness Improvement *instance. If I is a* Yes*-instance, then c_z can be increased to r by adding at most k edges, all of which contain z.*

Proof. Let $u_i, u_j \in V, u_i, u_j \neq z$ be two nodes of the input graph such that $e := \{u_i, u_j\} \notin E$. We show that, if a solution S contains $\{u_i, u_j\}$, then there is a solution S' of size at most k such that $e \notin S'$, but $\{z, u_p\} \in S'$ for some $u_p \in V, \{u_p, z\} \notin E \cup S$. Such a node exists as we restrict ourselves to non-trivial instances.

Let $e \in S$ such that e introduces a shortest path p of length l between z and some node $u_q \neq u_i$ such that each other path not containing e has length $l' > l$. Hence, e reduces the distance between z and u_q by at least 1. Without loss of generality, assume u_i precedes u_j in p. Also, it follows that $\{z, u_j\} \notin S$: Otherwise, $\{u_i, u_j\}$ does not reduce the distance between z and u_j. We obtain a solution which reduces the distance between z and u_q

by at least the same value if we replace e by $\{z, u_j\}$, as this reduces the distance between z and u_j to 1.

If the the edge $\{z, u_j\}$ already exists in the input graph, then there is no shortest path between z and any other node u_t containing e. Hence, e has no impact on c_z at all and replacing e by any other edge containing z increases c_z by at least $\frac{1}{2}$. As the instance is non-trivial, we can find such an edge. □

Lemma 2 directly implies that CLOSENESS IMPROVEMENT is in XP with respect to the number k of edge additions:

Corollary 1. CLOSENESS IMPROVEMENT *is solvable in* $\mathcal{O}(n^k)$ *time where* k *is the number of edge additions, and thus is in XP with respect to the parameter number of edge additions.*

Proof. As shown in Lemma 2, an optimal solution for a CLOSENESS IMPROVEMENT instance $I = (G, z, k, r)$ contains only edges where one endpoint is z. Hence, a brute-force approach can be used which iterates over all subsets of $V' \subseteq V$ of size at most k such that $(z, v) \notin E$ for all $v \in V'$. For each V', the closeness centrality of z after adding edges between z and each node in V' is computed and the subset which increases c_z at most forms the solution. The total running time of this algorithm is $\mathcal{O}(n^k \cdot (n + m))$. □

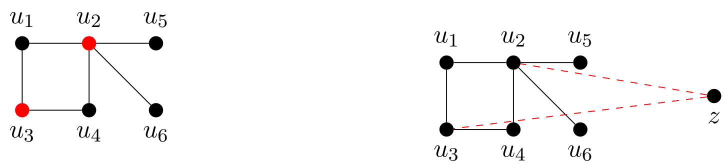
Next, we show that there is a relation between DOMINATING SET and CLOSENESS IMPROVEMENT. We make use of this relationship in the proofs of the following corollaries and theorems.

Lemma 3. *Let* $I = (G = (V, E), z, k, r)$ *be a* CLOSENESS IMPROVEMENT *instance. Assume* G *admits a dominating set* $D \subseteq V$ *of size at most* k*. If* I *is a* YES*-instance, then there is a solution* S *for* I *such that* $(z, u) \in S \cup E$ *for each* $u \in D$*.*

Proof. Let d be the degree of z in G. An optimal solution for a CLOSENESS IMPROVEMENT instance contains edges where one endpoint is z (Lemma 2). Then after adding k such edges, there are exactly $k + d$ nodes which have distance 1 to z. The distance between z and the other nodes in the graph is at least 2. Hence, a solution is optimal if the distance to each other node is exactly 2, which is the case if we connect z to each node forming the dominating set by an edge. □

Lemma 3 directly implies the following corollary:

Corollary 2. *Let* $I = (G, z, k, r)$ *be a* CLOSENESS IMPROVEMENT *instance. From Lemma 3 it follows directly that given a dominating set of size at most* k*, a solution for* I *is computable in polynomial time. In particular, if a dominating set of size at most* k *is computable in polynomial time, then* I *is decidable in polynomial time.*



(a) A Dominating Set instance $I = (G, k = 2)$. The red colored nodes form a solution for I.

(b) The constructed Closeness Improvement instance $(I' = G', z, k' = 2, r = k + \frac{n-k}{2})$. The red dashed edges form a solution for I'.

Figure 2: Parameterized reduction from Dominating Set to Closeness Improvement.

The next corollary is used to show that Closeness Improvement is fixed-parameter tractable with respect to the size of a vertex cover in Theorem 5.

Lemma 4. *Let $I = (G = (V, E), z, k, r)$ be a* Closeness Improvement *instance, and let d_0 be the number of isolated nodes in G, and $V' \subseteq V$ be a vertex cover of G of size ℓ. If $k \geq d_0 + \ell$, then I can be solved in $\mathcal{O}(1.2738^k + kn)$ time.*

Proof. From Corollary 2 we know that we obtain an optimal solution S for I if the neighborhood of z obtained by adding S to G is a dominating set. As a vertex cover union the set of isolated nodes is a dominating set, we also obtain an optimal solution if $k \geq \ell + d_0$. However, there does not necessarily need to exist a vertex cover of size k in G; we may also determine whether there is a vertex cover $V' \subseteq V$ of size at most $\deg(z) + k - d_0$, where $N(z) \subseteq V'$. If such a set V' exists, then we can optimally solve I by adding at most k edges to the input graph such that $\{z, u\} \in E \cup S$ for all $u \in V'$, and d_0 edges to between z and the isolated nodes of G. In this case, we can compute a solution S of size at most $k - d_0$ by solving the Vertex Cover instance $(G - N(z), k - d_0)$ in $\mathcal{O}(1.2738^k + kn)$ time [CKX10]. □

2.1 Hardness results

This section presents hardness proofs for Closeness Improvement. First, we show that the problem is W[2]-hard with respect of the probably most natural parameter, that is the number k of edge additions.

Theorem 1. Closeness Improvement *is W[2]-hard with respect to the parameter number k of edge additions.*

Proof. The proof is by a parameterized reduction from Dominating Set with parameter solution size k:

Let $I = (G = (V, E), k \in \mathbb{N})$ where $V = \{u_1, \ldots, u_n\}$ be a DOMINATING SET instance. We construct a CLOSENESS IMPROVEMENT instance $I' = (G' = (V', E'), z, k, k + \frac{1}{2}(n-k)$ as follows (see Figure 2): Given the input graph G, we simply add an isolated node z to the graph, that is $G' = (V \cup \{z\}, E)$.

We now show that the reduction is correct, i.e. I is a YES-instance if and only if I' is a YES-instance:

$\Rightarrow$: Let I be a YES-instance. Then there is a dominating set $V_{DS} \subseteq V$ of size k in G. After adding k edges between z and each node in V_{DS} in G', these k nodes have distance 1 to z and the $n-k$ neighbors of the nodes V_{DS} have distance 2 to z.

Hence, $c_z = k + \frac{n-k}{2}$. That is, I' is a YES-instance.

$\Leftarrow$: We prove the reverse direction by contraposition. That is, we show that I' is a NO-instance if I is a NO-instance. Let I be a NO-instance, that is there is no dominating set of size k in G. From Lemma 2 we know that we can maximally increase c_z by adding edges where one endpoint is z. However, after adding k edges between z and k nodes in G', there are $l \geq 1$ nodes in G' whose distances to z is $d \geq 3$.

Hence, $c_z \leq k + \frac{n-k-l}{2} + \frac{l}{d} < k + \frac{n-k}{2}$. That is, I' is a NO-instance. □

The next two results directly follow from the proof of Theorem 1:

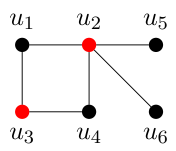
Corollary 3. CLOSENESS IMPROVEMENT *is NP-hard even on planar graphs with maximum degree 3.*

Proof. Let $I = (G, k)$ be a DOMINATING SET instance, where G is a planar graph with maximum degree 3. Let G' be the graph constructed by the reduction used in the proof of Theorem 1. Then G' is also a planar graph with maximum degree 3. As DOMINATING SET is NP-hard even on planar graphs with maximum degree 3 [GJ90], CLOSENESS IMPROVEMENT remains NP-hard on graphs of degree 3. □

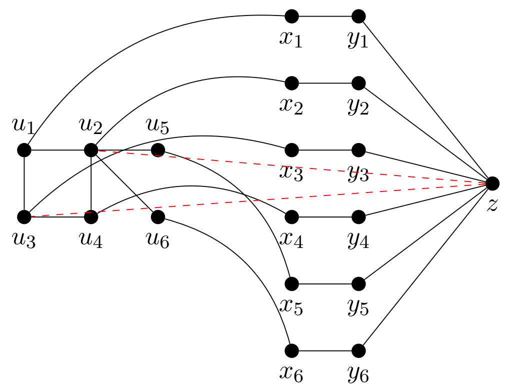
As many network have a slowly increasing or even decreasing diameter (Leskovec et al. [LKF05]), we analyze the parameterized complexity of CLOSENESS IMPROVEMENT with respect to the parameter graph diameter. By modifying the reduction in the proof of Theorem 1, we show that CLOSENESS IMPROVEMENT remains W[2]-hard on graphs with diameter 6.

Theorem 2. CLOSENESS IMPROVEMENT *is W[2]-hard with respect to the parameter number of edge additions k even on graphs with diameter 6.*

Proof. The proof is by a parameterized reduction from DOMINATING SET with parameter solution size k:



(a) A Dominating Set instance $I = (G, k = 2)$. The red colored nodes form a solution for I.



(b) The constructed Closeness Improvement instance $(I' = G', z, k' = 2, 2n + \frac{k}{2})$. The red dashed edges form a solution for I'.

Figure 3: Parameterized reduction from Dominating Set to Closeness Improvement on graphs with diameter six.

Let $I = (G = (V, E), k \in \mathbb{N})$ where $V = \{u_1, \ldots, u_n\}$ be a Dominating Set instance. We construct a Closeness Improvement instance $I' = (G' = (V', E'), z, k, 2n + \frac{k}{2})$ as follows (see Figure 3): Given the input graph G, we add $2n$ nodes $x_1, \ldots, x_n, y_1, \ldots, y_n$ such that each node x_i is adjacent to u_i and y_i. Furthermore, we add z and add edges between z and each $y_1, \ldots, y_n$. Formally, V' and E' are defined as follows:

$$\begin{aligned} V' &= V \cup \{x_i, y_i \mid 1 \le i \le n\} \cup \{z\}, \\ E' &= E \cup \{\{u_i, x_i\} \mid 1 \le i \le n\} \cup \{\{x_i, y_i\} \mid 1 \le i \le n\} \\ &\quad \cup \{\{z, y_i\} \mid 1 \le i \le n\}. \end{aligned}$$

We partition V' into the subsets $Y' := \{y_1, \ldots, y_n\}, X' := \{x_1, \ldots, x_n\}$ and $U' := \{u_1, \ldots, u_n\}$. The nodes in Y' have distance 1 to z, the nodes in X' have distance 2 to z and the nodes in U' all have distance 3 to z. First, we show that adding edges between z and nodes in U' is optimal:

Assume an edge $\{z, x_i\}, x_i \in X'$, is added. Then the distance between z and x_i is 1 and the distance between z and u_i is 2. If we instead add the edge $\{z, u_i\}$, then the distance between z and u_i is 1 and the distance between z and x_i remains 2. Furthermore, the edge $\{z, u_i\}$ may introduce shorter distances to the neighbors of u_i, which the edge $\{z, x_i\}$ does not. Hence, if a solution for I' contains $\{z, x_i\}$, we can replace that edge by $\{z, u_i\}$. Last, it remains to show that the reduction is correct, i.e. I is a Yes-instance if and only if I' is a Yes-instance:

$\Rightarrow$: Let I be a Yes-instance. Then there is a dominating set $U'_{DS} \subseteq U'$ of size k for $G' - (X' \cup Y')$. After adding k edges between z and each node in U'_{DS}, these k nodes have distance 1 to z and the $n-k$ neighbors in $U' \setminus U'_{DS}$ have distance 2 to z. Furthermore, each node in Y' has distance 2 to z and each node in X has distance 1 to z.

Hence, $c_z = k + \frac{n-k}{2} + \frac{n}{2} + n = 2n + \frac{k}{2}$. That is, I' is a Yes-instance.

$\Leftarrow$: We prove the other way by contraposition. That is, we show that I' is a No-instance if I is a No-instance. Let I be a No-instance. If there is no dominating set of size k in $G' - (X' \cup Y')$, then after adding k edges between z and nodes in U', there are $l \geq 1$ nodes in U' whose distances to z is still 3.

Hence, $c_z = k + \frac{n-k-l}{2} + \frac{l}{3} + \frac{n}{2} + n = 2n + \frac{k}{2} - \frac{l}{6}$, for $l \geq 1$. That is, I' is a No-instance.

□

The W[2]-hardness of Dominating Set on graphs with diameter 2 (Lokshtanov et al. [Lok+13]) and the reduction in the proof of Theorem 2 directly imply the following corollary:

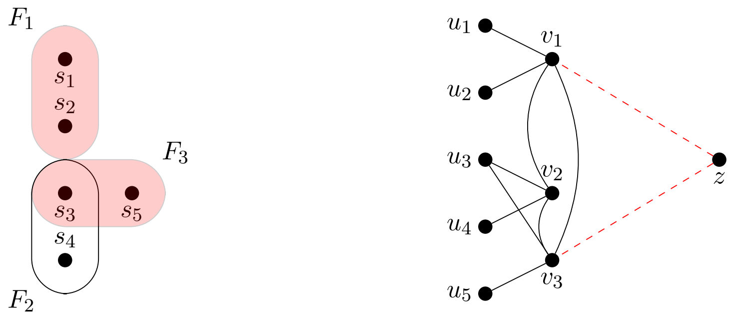
Corollary 4. Closeness Improvement *remains W[2]-hard on graphs with diameter 4.*

Proof. Let $I = (G, k)$ be a Dominating Set instance, where G has diameter 2. The graph of the instance $I' = (G, z, k, r)$ constructed by the reduction in the proof of Theorem 2 has diameter at most 4: The largest shortest paths in G' are the ones between x_i and x_j, if the distance between u_i and u_j is maximal, that is 2. Then these shortest paths have length 4 and either contain y_i, z and y_j, or u_i, an intermediate node u' and u_j. □

However, we show in Lemma 7 that Closeness Improvement is polynomial-time solvable on graphs of diameter 1 and 2.

Next, we show that Closeness Improvement remains W[2]-hard with respect to the number of edge additions k on split graphs by a parameterized reduction from Max k-Set Cover. The proof makes use of the following lemma, which provides information on the structure of a solution on split graphs.

Lemma 5. *Let (G, z, k, r) be a* Closeness Improvement *instance where $G = (V, E)$ is a split graph and $V = C \cup I$ such that the nodes in C induce a clique and the nodes in I induce an independent set. Furthermore, let $I = I_I \cup I_N$, where I_I is the set of isolated nodes and I_N is the set of non-isolated nodes. Let $u \in I_N, u' \in C$ and $e := \{z, u\}, e' := \{z, u'\} \notin E$. Assume $z \notin C$; otherwise, the instance is polynomial-time solvable (Lemma 8). If a solution $S \subseteq \binom{V}{2}$ contains e, then replacing e by e' does not decrease c_z.*



(a) A Max k-Set Cover instance $I = (U, \mathcal{F}, k = 2, r = 4)$ with solution $\{F_1, F_3\}$.

(b) The constructed Closeness Improvement instance $I' = (G, z, k = 2, r = 4\frac{5}{6})$. The red dashed lines form a solution for I'.

Figure 4: Parameterized reduction from Max k-Set Cover to Closeness Improvement on split graphs.

Proof. Depending on whether z is adjacent to a node in C, we analyze the closeness centrality gain of z by adding an edge $e \in S$ to the graph. For each case, we show that replacing e by e' does not decrease c_z.

Case 1: Assume z has a neighbor in C. Then the distance between z and each node in C is at most 2 and the distance between z and each node in I_N is at most 3. After adding the edge e to the graph, the distance between z and u decreases from 2 to 1. Each path between z and nodes in C containing e has length at least 2, and each path between z and nodes in $I_N \setminus \{u\}$ containing e has length at least 3. Hence, e does not decrease the distances between z and any node except to u. Consequently, replacing $\{z, u\}$ by an arbitrary edge $\{z, u'\}, u' \in V_C, \{z, u'\} \notin E \cup S$ does not decrease c_z.

Case 2: Assume z has no neighbor in C. After adding the edge e to the graph, the distance between z and u is 1 and the distance between z and each neighbor of u in C is 2. Moreover, the distance between z and each other node in C containing e is 3 and the distance to each node in $I_N \setminus \{u\}$ containing e is at least 3.

If we replace e by e', then the distance between z and u' is 1 and the distance between z and u is 2. The distances between z and each node in $C \setminus \{u'\}$ containing e' is 2 and the distance between z and each other node in I_N is at most 3.

Hence, the closeness centrality of z does not decrease if e is replaced by e'. □

Theorem 3. Closeness Improvement *is W[2]-hard on split graphs with respect to the parameter number of edge additions.*

Proof. We show that CLOSENESS IMPROVEMENT is W[2]-hard on split graphs by a parameterized reduction from MAX k-SET COVER. Let $I = (U = \{s_1, \ldots, s_n\}, \mathcal{F} = \{F_1, \ldots, F_m\}, k, r)$ be a MAX k-SET COVER instance. We construct a CLOSENESS IMPROVEMENT instance $I' = (G = (V, E), z, k, k + \frac{r+(m-k)}{2} + \frac{n-r}{3})$ where G is a split graph. The construction is as follows (Figure 4 provides an example):

First, we set $V := \{v_1, \ldots, v_m\} \cup \{u_1, \ldots, u_n\} \cup \{z\}$. In the next step, we add edges such that $C := \{v_1, \ldots, v_m\}$ induces a clique. Furthermore, we add an edge $\{u_i, v_j\}$ if $s_i \in F_j$. The constructed graph is a split graph: The set C induces a clique and the nodes z and $I := \{u_1, \ldots, u_n\}$ induce an independent set. Obviously, the CLOSENESS IMPROVEMENT instance can be constructed in polynomial time. Without loss of generality, we assume that each element is contained in at least one subset of the family $\mathcal{F}$. Hence, the induced independent set of G does not contain isolated nodes. We also assume that $k < n$, as $n \leq m$ and if $k \geq m$ then we have a trivial YES-instance.

Let S be a solution for the constructed CLOSENESS IMPROVEMENT instance. The subsets in $\mathcal{F}$ corresponding to the nodes of the solution S for the CLOSENESS IMPROVEMENT instance form a solution for the MAX k-SET COVER instance. It remains to show that the reduction is correct, i.e. that I is a YES-instance if and only if I' is a YES-instance:

$\Rightarrow$ If I is a YES-instance, then there is a set $\mathcal{F}' \subseteq \mathcal{F}$ of size k such that $|\bigcup_{F_j \in \mathcal{F}'}| = r$. Set $S := \{v_i \mid s_i \in \mathcal{F}'\}$. By adding k edges between z and the nodes in S, the closeness centrality of z is as follows: The distance between z and the k nodes in S is 1. Furthermore, the distance to each neighbor of a node in S is 2: These are the remaining $m - k$ nodes of the induced clique and, as the input instance is a YES-instance, there must also be at least r nodes of the induced clique which have neighbors in S. The distance to the remaining nodes which have no neighbor in S is 3. Hence, $c_z \geq k + \frac{r+(m-k)}{2} + \frac{n-r}{3}$.

$\Leftarrow$ We prove the reverse direction by contraposition. That is, we show that I' is a NO-instance if I is a NO-instance. Let I be a NO-instance. Then for each $\mathcal{F}' \subseteq \mathcal{F}$ of size at most k, there are at most $r - l, l \geq 1$ covered element.

As the constructed graph G is a split graph, we know from Lemma 5 that we can maximally increase c_z by adding edges between z and nodes in C. After adding k edges between z and nodes in C, we have the following distances:

- For k nodes in C, the distance to z is 1.

- For $m - k$ nodes in C, the distance to z is 2.
- For $r - l$ nodes in I, the distance to z is 2.
- For $n - (r - l)$ nodes in I, the distance to z is 3.

Hence, the maximal value for c_z in I' after adding k edges is $k + \frac{(r-l)+(m-k)}{2} + \frac{n-(r-l)}{3}$ for some $l \geq 1$. It follows that I' is a NO-instance. □

In the next section, we classify CLOSENESS IMPROVEMENT as fixed-parameter tractable with other graph parameters. Moreover, we show that CLOSENESS IMPROVEMENT is polynomial-time solvable on certain graph classes, such as cluster graphs.

2.2 Algorithmic results

The first result of this section covers the graph parameter distance to clique, that is the number of nodes to remove such that the graph is a clique. Due to its running time, the presented algorithm is useful for a classification of CLOSENESS IMPROVEMENT in terms of parameterized complexity, but rather useless in terms of practical application. However, for a sufficiently large parameter k, we can solve CLOSENESS IMPROVEMENT in linear time:

Proposition 2. *Let $I = (G, z, k, r)$ be a* CLOSENESS IMPROVEMENT *instance, where G has distance to clique $\ell < \deg(z) + k$. Then I can be solved in $\mathcal{O}(n + m)$ time.*

Proof. Let $V_R \subseteq V$ be a set of nodes of size at most ℓ such that $G - V_R$ is a clique. The set V_R and one node of $V \setminus V_R$ forms a dominating set. Hence, by adding an edge between z and each node in V_R that is not adjacent to z, and by adding the remaining edges between z and arbitrary nodes of $V \setminus V_R$, we maximally increase c_z. □

However, in general we can only show that CLOSENESS IMPROVEMENT is fixed-parameter tractable with respect to the parameter distance to clique.

Theorem 4. CLOSENESS IMPROVEMENT *can be solved in $2^\ell \cdot 2^{2^\ell} \cdot (m + n)$ time, where ℓ is the distance to clique.*

Proof. Let $I = (G, z, k, r)$ be a CLOSENESS IMPROVEMENT instance, where ℓ is G's distance to clique. Without loss of generality, assume $\ell \geq \deg(z) + k$: Otherwise, I is solvable in polynomial time (Proposition 2).

Let $V_R \subseteq V$ be a set of size ℓ such that $V_C := V \setminus V_R$ is a clique. As V_R can be computed in fpt-time with respect to its size ℓ (Hüffner et al. [Hüf+10]), we assume that V_R is given. Moreover, we introduce a partition $\mathcal{P}$ of V_C such that all nodes within a subset $P \in \mathcal{P}$ have the same neighborhood. We say that a subset P is *covered* if z is adjacent to at least one node $u \in P$; otherwise, we say P is *uncovered.*

We now introduce an algorithm which guesses an optimal solution S for Closeness Improvement in two steps in $\mathcal{O}(2^\ell \cdot 2^{2^\ell} \cdot (n+m))$ time, where ℓ is the distance to clique. The algorithm first guesses a subset $V'_R \subseteq V_R$ of size at most k such that $\{z, u\}$ is part of S for each $u \in V'_R$. As $k \leq \ell$ and V_R has size ℓ, there are $\mathcal{O}(2^\ell)$ combinations for V'_R.

Next, if $|V'_R| < k$, then the algorithm guesses a set of $k - |V'_R|$ nodes of V_C such that edges between z and these nodes are part of S. We now show that it is adequate to guess a family of subsets $\mathcal{P}' \subseteq \mathcal{P}$ of size $k - |V'_R|$ such that $\{z, u\} \in S$, where u is an arbitrary node of P. The correctness of this part of the algorithm requires two properties: First, as all nodes in a subset P are equivalent in terms of their neighborhood, for two nodes $u, v \in P$ the closeness centrality of z increases by the same value if we add $\{z, u\}$ or $\{z, v\}$ to the graph. Second, if z is adjacent to a node $u \in P$, then adding an edge to another node $v \in P$ only decreases the distance between z and v from 2 to 1; hence, c_z can be increased by at least the same value by replacing $\{z, v\}$ by any other edge $\{z, v'\}$ where v' is part of an uncovered subset. As $|\mathcal{P}| \leq 2^\ell$, there are $\mathcal{O}(2^{2^\ell})$ combinations for $\mathcal{P}'$.

In total, an optimal solution can be computed in $2^\ell \cdot 2^{2^\ell} \cdot (m+n)$ time. □

This result is interesting for classification purposes in terms of the parameter hierarchy. Next, we analyze the complexity of Closeness Improvement on cluster graphs and the parameterized complexity of Closeness Improvement with parameter distance to cluster graphs.

Lemma 6. Closeness Improvement *is solvable in $\mathcal{O}(n)$ time on cluster graphs.*

Proof. Let (G, z, k, r) be a Closeness Improvement instance and let c be the number of disjoint cliques. We refer to a cluster as *covered* if z is adjacent to at least one node in that cluster. As z is adjacent to all nodes in its own cluster, each solution may only add edges to nodes in clusters not containing z. Initially, all clusters except the one containing z are uncovered. For each uncovered cluster C, adding an edge between z and a node $u \in C$ yields distance 1 between z and u and distance 2 between z and all other nodes in that cluster. Each further edge between z and any $u' \in C$ only decreases the distance between z and u' by 1. Hence, the closeness centrality maximally increases by subsequently adding edges between z and nodes in the largest uncovered cluster. A list of all connected components along with their sizes can be computed in $\mathcal{O}(n)$ time. □

Furthermore, we show that Closeness Improvement is polynomial-time solvable on graph with diameter at most 2.

Lemma 7. Closeness Improvement *is solvable in* $\mathcal{O}(1)$ *time on graphs with diameter 1 and solvable in* $\mathcal{O}(n)$ *time on graphs with diameter 2.*

Proof. Let $I = (G, z, k, r)$ be a Closeness Improvement instance. If the diameter of G is a 1, then G is a clique, and the closeness centrality of z cannot be improved as no edges can be added to the graph. If the diameter of G is 2, then edges can be added between z and nodes which have distance 2 to z. However, adding such an edge between z and a node u with distance 2 to z does not decrease the distances between z and other nodes in the graph, as the distance is at most 2 and each path between z and nodes $u' \neq u$ containing u has length at least 2. □

For the class of split graphs, we show that Closeness Improvement can be solved in polynomial time in at least two cases:

Lemma 8. *Let* (G, z, k, r) *be a* Closeness Improvement*, where* G *is a split graph. If* z *is in the clique part of the split graph, then the instance is linear-time solvable.*

Proof. As z is already adjacent to all other nodes in the clique part, edges can be added to the independent set part only. The distance to nodes in the independent set is either 2 if G is connected, or ∞ for isolated nodes. By adding an edge to a node u of the independent set, the distance between z and u is decreased to 1 and the distances to all other nodes remains the same. Hence, an optimal solution can be computed as follows: First, if there are isolated nodes, then we add edges between z and these nodes in arbitrary order. Otherwise, or if there are less than k isolated nodes, we add edges to arbitrary nodes in the independent set. □

Lemma 9. *Let* (G, z, k, r) *be a* Closeness Improvement *instance, where* G *is a split graph. If the number of nodes inducing an independent set in* G *is less than* k*, then the instance is solvable in* $\mathcal{O}(n)$ *time.*

Proof. One node of the clique part plus all nodes from the independent set part form a dominating set. Hence, due to Lemma 3 the instance can be solved in polynomial-time by adding edges to all nodes of the independent set and at least one node of the clique. □

Next, we show that Closeness Improvement is fixed-parameter tractable with respect to the vertex cover size of a graph. The proof makes use of the following lemma:

Lemma 10. *Let* $I = (G = (V, E), z, k, r)$ *be a* Closeness Improvement *instance, where* G *has no vertex cover* $V' \subseteq V$ *of size* $deg(z) + k$ *such that* $N(z) \subseteq V'$*. Let* $V_{VC} \subseteq V$ *be a vertex cover of size* ℓ*, and* $V_{IS} := V \setminus V_{VC}$ *induce an independent set. We say that* $V_N \subseteq V_{IS}$ *is the set of non-isolated nodes of the induced independent set. Furthermore, let* $\mathcal{P}$ *be a partition of*

V_N such that all nodes in a subset $P \in \mathcal{P}$ have the same neighbors in V_{VC}. Then for each $P \in \mathcal{P}$, it is optimal to add at most one edge between z and a node in P if z is not adjacent to a node in P. If z is adjacent to a node in P, then we obtain an optimal solution by adding no additional edges between z and other nodes in P.

Proof. We distinguish whether z is adjacent to a neighbor of the nodes in P or not. Let S be an optimal solution for I.

- Assume z is adjacent to a neighbor $t \in V_{VC}$ of the nodes in P, and there are at least two nodes $u, v \in P$ such that $\{z, u\} \in E \cup S$ and $\{z, v\} \in S$: Then the closeness centrality increases by exactly $\frac{1}{2}$ by adding the edge $\{z, v\}$, as it decreases the distance between z and v from 2 (path via t) to 1. The distances to all the neighbors of v are not reduced, as the neighborhood of v is the same as the neighborhood of u, and $\{z, u\} \in E \cup S$. Hence, the solution obtained by replacing $\{z, v\}$ by some edge $\{z, v'\}$, $t' \in V_{VC}$ decreases c_z by at least the same value. Due to our assumption that there is no vertex cover $V' \subseteq V$ of size $\deg(z) + k$ such that $N(z) \subseteq V'$, we ensure that there is such a node t'.

- Let t be a neighbor of the nodes in P, and z is not adjacent to a neighbor of the nodes in P. Furthermore, assume that there are at least two nodes $u, v \in P$ such that $\{z, u\} \in E \cup S$ and $\{z, v\} \in S$. Then the distance between z and v decreases from 3 to 1 after adding the edge $\{z, v\}$. Other distances between nodes in G and z are unaffected, as z is already adjacent to $u \in P$. Hence, c_z increases by $\frac{2}{3}$ after adding $\{z, v\}$ to the graph. However, if we replace $\{z, v\}$ by $\{z, t\}$, then the distance between z and t decreases from 2 to 1, and the distance between z and v decreases from 3 to 2. Hence, c_z increases by $\frac{1}{2} + \frac{1}{3} > \frac{2}{3}$.

Hence, in each case, we can replace $\{z, v\}$ by an edge $\{z, t\}$.

□

Based on the knowledge about the structure of an optimal solution from Lemma 10, we now show that Closeness Improvement is fixed-parameter tractable with respect to the parameter vertex cover size.

Theorem 5. Closeness Improvement *can be solved in $\mathcal{O}(2^\ell \cdot (2^{2^\ell} + k) \cdot (n + m))$ time, where ℓ is the vertex cover size.*

Proof. Let $I = (G = (V, E), z \in V, k, r)$ be a Closeness Improvement instance, and let $V_{VC} \subseteq V$ be a vertex cover of G of size ℓ. Let d_0 be the number of isolated nodes in G.

In this proof, we assume that G does not admit a vertex cover $V'_{VC} \subseteq V$ of size at most $\deg(z) + k - d_0$, such that $N(z) \subseteq V'_{VC}$. In particular, this means that $k < \ell + d_0$. Otherwise, we can solve I in $\mathcal{O}(1.2738^\ell + \ell n)$ time by computing a vertex cover of size $k - d_0$ for $G - N(z)$ (Lemma 4). Furthermore, we assume that the set V_{VC} is given. Let S be an optimal solution for I.

We partition V into the vertex cover V_{VC} of size ℓ, and the sets V_N and V_I. The set V_N contains all nodes in $V \setminus V_{VC}$ with degree at least 1; they are the set of non-isolated nodes of the induced independent set; the set V_I contains all isolated nodes of the induced independent set, that is $V_I := V \setminus (V_{VC} \cup V_N)$. Moreover, we denote the V_I union V_N by V_{IS} Each $u \in V_N$ has at least one neighbor in V_{VC}, otherwise V_{VC} is not a vertex cover. Furthermore, the subgraph induced by V_{IS} is edgeless, which means that for each $u \in V_{IS} : N(u) \subseteq V_{VC}$. Second, we define a partition $\mathcal{P}$ of V_N such that all nodes in a subset $P \in \mathcal{P}$ have the same neighbors in V_{VC}. As the vertex cover has size ℓ, the size of $\mathcal{P}$ is at most 2^ℓ. From Lemma 10 we know that we obtain an optimal solution if, for each $P \in \mathcal{P}$, there is at most one node $u \in P$ such that $\{u, z\} \in S$. Furthermore, the number of degree-0 nodes in V_I is arbitrary; however, if S contains a certain number $\ell' \leq \ell$ of edges between z and nodes in V_I, then it does not make a difference which nodes are chosen to be adjacent to z. Moreover, adding an edge $\{z, w\}, w \in V_I$ increases c_z by exactly 1.

We now describe an algorithm $\mathcal{A}$ to solve Closeness Improvement in $\mathcal{O}(2^\ell \cdot (2^{2^\ell} + k) \cdot (n + m))$ time, where ℓ is the vertex cover size. Let S be an optimal solution that contains edges to nodes in $V_{VC} \cup V_I$ and at most one edge to each $P \in \mathcal{P}$. As shown in Lemma 10, such a solution exists. The algorithm guesses S by finding a set $V'_{VC} \subseteq V_{VC}$ and a set $\mathcal{P}' \subseteq \mathcal{P}$, such that S only contains edges $\{z, u\}$ for each $u \in V'_V C$, an edge to a node in each $\mathcal{P}'$, and at most k arbitrary edges between z and nodes in V_I if $|V'_{VC} \cup \mathcal{P}'| < k$. We now show how $\mathcal{A}$ works.

First, $\mathcal{A}$ iterates over all subsets $V'_{VC} \subseteq V_{VC}$ of size at most k. For each V'_{VC}, the algorithm iterates over all subsets $\mathcal{P}' \subseteq \mathcal{P}$ of size less than $k - |V'_{VC}|$. We then add edges between z and each node in V'_{VC}, and between z and one node in each $P \in \mathcal{P}'$. If $k' := |\mathcal{P}'| + |V'_{VC}| < k$, that less than k edges have been added and we add the remaining k' edges between z and arbitrary nodes in V_I. We do not need to differ between the nodes in V_I, as adding an edge to any of these nodes does not manipulate the distances to any other nodes, and for each edge $\{z, w\}$, the centrality of z increases by exactly 1. If there are less than k' nodes in V_I, then the guess for V'_{VC} or $\mathcal{P}'$ does not constitute S. For each such guess, we determine c_z, possibly saving the current guess if it increases c_z the most, and remove the added edges before starting the next iteration. Finally, after computing c_z for each V'_{VC} and $\mathcal{P}'$, we return 1 if the largest value found for c_z is larger than r, and 0 otherwise.

We now analyze the running time of the algorithm. As $V_{VC} = \ell$, there are $\mathcal{O}(2^\ell)$ subsets $V'_{VC} \subseteq V_{VC}$. For each such subset, there are $\mathcal{O}(2^{2^l})$ subsets $\mathcal{P}' \subseteq \mathcal{P}$, as the size of $\mathcal{P}$ is $\mathcal{O}(2^l)$. The remaining edges to nodes in V_I are computed in $\mathcal{O}(k)$ time. For each combination of $V'VC$ and $\mathcal{P}'$, the closeness centrality of z is computed in $\mathcal{O}(n+m)$ time where appropriate. Hence, the total running time of $\mathcal{A}$ is $\mathcal{O}(2^\ell \cdot (2^{2^\ell} + k) \cdot (n+m))$. This result directly implies that Closeness Improvement is fixed-parameter tractable with respect to the size ℓ of a vertex cover.

□

Theorem 6. Closeness Improvement *is solvable in* $\mathcal{O}(2^\ell \cdot 2^{2^{2^\ell}} \cdot 2^{2^l} \cdot (m+n))$ *time, where* ℓ *is the vertex deletion distance of* G *to a cluster graph.*

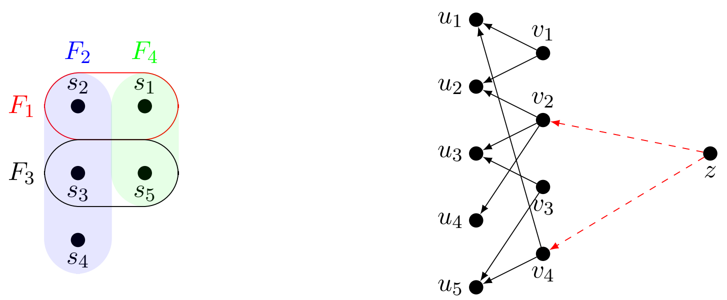
Proof. Let (G, z, k, r) be a Closeness Improvement instance, where $V_{VDS} \subset V$ is a cluster vertex deletion set of size ℓ such that $G_C = (V_C, E_C) := G - V_{VDS}$ is a cluster graph with clusters $\{c_1, \dots, c_s\} =: C$. Assume that V_{VDS} is given. The idea is to compute the optimal solution by guessing an optimal solution in two steps.

Before we describe the algorithm, we define a partition $\mathcal{P}$ of the set of clusters C such that the following statement holds for all subsets $P \in \mathcal{P}$ and all clusters $c_i, c_j \in C$: If $c_i, c_j \in P$, then for each $u_i \in c_i$ there is a node $u_j \in c_j$ such that $N_{VDS}(u_i) = N_{VDS}(u_j)$. We then say that c_i and c_j have the same *cluster signature*. We now analyze the size of $\mathcal{P}$: Let $u_i, u_j \in V_c$ such that $N_{VDS}(u_i) = N_{VDS}(u_j)$; we say that u_i and u_j have the same *node signature*. As V_{VDS} has size ℓ, there are at most 2^l nodes in V_C with different node signatures. Hence, there are at most 2^{2^ℓ} clusters with different cluster signatures.

We next describe an algorithm which solves Closeness Improvement in $\mathcal{O}(2^\ell \cdot 2^{2^{2^\ell}} \cdot 2^{2^l} \cdot (m+n))$ time.

First, we guess a subset of V_{VDS} of size $b \leq k$ by trying all possibilities and add edges between z and these nodes. This step involves $\mathcal{O}(2^\ell)$ branches. Second, we add the remaining $k - b$ edges between z and nodes in V_C in the following two steps: In Step 1, we guess the signatures of the clusters containing these nodes. As there are $\mathcal{O}(2^{2^\ell})$ cluster signatures, this involves $\mathcal{O}(2^{2^{2^\ell}})$ branches. In Step 2, for each cluster signature, we guess the signatures of nodes we want to add edges to. As there are $\mathcal{O}(2^\ell)$ node signatures, this step involves $\mathcal{O}(2^{2^\ell})$ branches. Finally, for each guess we need $\mathcal{O}(n+m)$ time to compute the closeness centrality of c_z.

We now show that the algorithm is correct. For each cluster signature we guess in Step 1, we can find the best cluster with that signature, i.e. adding edges to nodes in that cluster maximizes the closeness centrality of z in polynomial time: Let $P \in \mathcal{P}$ be a set of clusters with the same signature. Then adding an edge to a node in the largest cluster in P which does not contain nodes that z is connected to by an edge is part of an optimal solution



(a) A Set Cover instance $I = (U, \mathcal{F}, k = 2)$ with solution $\{F_2, F_4\}$.

(b) The constructed Directed Closeness Improvement instance $I' = (G, z, k = 2, r = 4\frac{5}{6})$. The red dashed edges imply a solution for I'.

Figure 5: Parameterized reduction from Set Cover to Directed Closeness Improvement .

constructed by this algorithm: If we choose a node u in a cluster c which contains a node adjacent to z, then the distance between z and u is reduced from 2 to 1, and the distances to all other nodes in c remain unchanged. However, if we choose a node u' with the same node signature of a cluster c' with the same cluster signature which does not contain a node that is adjacent to z, then the distance between z and u' decreases from at least 2 to 1, and the distance between z and each other node in c' is 2.

Moreover, we show the correctness of Step 2. For each cluster signature, we construct an optimal solution by adding at most one edge to a node with a specific node signature s. Assume there is an edge between z and a node with a specific node signature which is part of a cluster $c \in P$. Then by adding another edge to a node with the same signature in a cluster $c' \in P$, we only decrease the distances between z and the nodes in c', but not to the rest of the graph. Hence, by adding an edge to any other node in c' which is not adjacent to z, we increase c_z by at least the same value. □

2.3 Closeness improvement on directed graphs

In this section, we introduce the problem of improving the closeness centrality on directed, unweighted graphs. We show that the problem remains W[2]-hard with respect to the number k of added arcs, even on directed acyclic graphs and even if the diameter of the graph is 3. Before we present the hardness proofs, we introduce formal definition of the decision problem:

DIRECTED CLOSENESS IMPROVEMENT
Input: A directed, unweighted graph $G = (V, A)$, a node $z \in V$, an integer k and a rational number r.
Question: Is there an arc set S of size at most k such that $c_z \geq r$ in $G' = (V, E \cup S)$?

Analogously to the undirected variant, we show that we can maximize the closeness centrality of a node z in a directed graph by adding arcs adjacent to z:

Lemma 11. *Let $I = (G = (V, E), z, k, r)$ be a non-trivial* DIRECTED CLOSENESS IMPROVEMENT *instance, i.e. $deg(z)+k \leq n-1$. If I is a* YES*-instance, then there is a solution S for I where for each arc $a \in S$, the source node is z.*

Proof. The proof is analogous to the one of Lemma 2: If an optimal solution S contains an arc $a := (u, v), u, v \neq z$, then any shortest path from z to some node w containing the arc a becomes even shorter if (u, v) is replaced by (z, v). If (z, v) already exists, then no shortest path from z contains a; hence, it can be replaces by an arbitrary arc with source z. Furthermore, an arc a' where z is the endpoint does not improve the closeness centrality of z at all, as any path from z containing a' contains a loop and thus is no shortest path. □

Lemma 11 directly implies that DIRECTED CLOSENESS IMPROVEMENT is in XP with respect to the number of arc additions:

Lemma 12. DIRECTED CLOSENESS IMPROVEMENT *can be solved in $\mathcal{O}(n^k \cdot (n+m))$ time, where k is the number of arc additions and thus is in XP with respect to the parameter number of arc additions.*

Proof. As shown in Lemma 11, an optimal solution for a DIRECTED CLOSENESS IMPROVEMENT instance $I = (G, z, k, r)$ contains only arcs where the source is z. Hence, a brute-force approach can be used which iterates over all subsets of $V' \subseteq V$ of size at most k such that $(z, v) \notin E$ for all $v \in V'$. For each V', the closeness centrality of z after adding arcs between z and each node in V' is computed and the subset which increases c_z at most forms the solution. The total running time of this algorithm is $\mathcal{O}(n^k \cdot (n + m))$. □

Theorem 7. DIRECTED CLOSENESS IMPROVEMENT *is W[2]-hard with respect to the number of edge additions k.*

Proof. The proof uses a parameterized reduction from SET COVER with the parameter *number of subsets k*, which is known to be W[2]-hard (Downey and Fellows [DF12]). Let $I = (\mathcal{F} = \{F_1, \ldots, F_m\}, U = \{s_1, \ldots, s_n\}, k)$ be a SET COVER instance. We reduce I to a CLOSENESS IMPROVEMENT instance $I' = (G = (V, A), z, k, k + \frac{n}{2})$, where G is a directed, unweighted graph constructed as follows: For each $s_i \in U$ and each $F_j \in \mathcal{F}$, we add a node v_i or u_j to the graph, respectively. Furthermore, if $s_i \in F_j$ for any $s_i \in U, F_j \in \mathcal{F}$, then we add an arc (v_j, u_i). Finally, we add a node z to the constructed graph. We provide an example in Figure 5.

Before showing the correctness of the reduction, we state and prove the following observation: The closeness centrality of z can be maximally increased by adding arcs from z to v_j.

First of all, the closeness centrality of z can be maximally increased if the source of each added arc is z. Otherwise, if z is the target of an arc, then we either introduced a loop, or the source of the arc remains unreachable from z. If z is neither the source nor the target of the arc, then all introduced shortest paths containing this arc become even shorter if we replace the source of the arc by z. Last, an arc (z, u_i) can be replaced by (z, v_j), where $(v_j, u_i) \in E$. By adding the arc (z, u_i), the distance from z to u_i is decreased to 1. An arc (z, v_j) decreases the distance from z to v_j to 1 and the distance of at least one more node u_i to 2. Hence, by adding arcs from z to v_j, we obtain a larger closeness centrality of z compared to adding edges from z to u_i.

We show that the reduction is correct, that is, I is a Yes-instance if and only if I' is a Yes-instance.

$\Rightarrow$: If I is a Yes-instance, then there is an $\mathcal{F}' \subseteq \mathcal{F}$ of size k such that $\bigcup_{F_j \in \mathcal{F}'} = U$. By adding k arcs (z, v_j), $F_j \in \mathcal{F}'$, there are k nodes with distance 1 from z, and each node in $\{u_1, \ldots, u_n\}$ has distance 2 from z. Hence, c_z can be increased to $k + \frac{n}{2}$ and I' is a Yes-instance.

$\Leftarrow$: If I is not a Yes-instance, then there is no such set $\mathcal{F}' \subseteq \mathcal{F}$ of size k such that $\cup_{F_j \in \mathcal{F}'} = U$. After adding k arcs from z to nodes in $\{v_1, \ldots, v_m\}$, there is at least one node u_i such that there is no path from z to u_i. Summing up, c_z can be increased to at most $k + \frac{n'}{2}$ for $n' < n$ and I' is a No-instance. □

Corollary 5. Directed Closeness Improvement *is W[2]-hard on directed acyclic graphs.*

Proof. Theorem 7 directly implies W[2]-hardness on directed acyclic graphs, as the constructed graphs in the reduction are acyclic. □

Hence, there is no hope for fixed-parameter algorithm for Directed Closeness Improvement with respect to the number of arc additions. Although we do not know the relationship of the hardness between Closeness Improvement and Directed Closeness Improvement, we propose that the directed variant is even harder than the undirected one: An optimal solution may contain arcs where the source is z or the target is z, while for the undirected variant there is no such distinction. In the next theorem, we slightly modify the reduction in the proof of Theorem 7 in order to show that DIRECTED CLOSENESS IMPROVEMENT remains W[2]-hard on directed graphs with diameter 4.

Theorem 8. DIRECTED CLOSENESS IMPROVEMENT *remains W[2]-hard on directed graphs, even with diameter 3.*

Proof. Let $I = (\mathcal{F} = \{F_1, \ldots, F_m\}, U = \{s_1, \ldots, s_n\}, k)$ be a SET COVER instance. We construct a DIRECTED CLOSENESS IMPROVEMENT instance $I' = (G = (V, E), z, k, r)$ as follows. First we construct a directed graph as described in the reduction in the proof of Theorem 7. Then we add m nodes w_i and $2m$ arcs $(z, w_i), (w_i, v_i)$ for each $1 \leq i \leq m$. Additionally, for each u_i, v_i and $w_i \in V$, we add the arcs $(u_i, z), (v_i, z)$ and (w_i, z) to G. Last, we set $r = k + 2n - \frac{k}{2}$.

The constructed graph is a directed graph with diameter 3: From z, the length of shortest paths to the other nodes is at most 3. The distance from each node w_i, v_i and u_i to any node u_j is at most 4, and the distance from these nodes to any node v_j is at most 3. Hence, G is a strongly connected directed graph with diameter 4.

Analogously to the reduction in Theorem 7, there is an optimal solution for I' which only contains arcs where z is the source and some of the nodes v_i are the target - the proof for this statement is the same as the one in the referred theorem.

It remains to show that the reduction is correct, that is I' is a YES-instance if and only if I is a YES-instance:

$\Rightarrow$: If I is a YES-instance, then there is an $\mathcal{F}' \subseteq \mathcal{F}$ of size k such that $\bigcup_{F_j \in \mathcal{F}'} = U$. By adding k arcs (z, v_j), $F_j \in \mathcal{F}'$, there are k nodes v_i with distance 1 from z, and each node in $\{u_1, \ldots, u_n\}$ has distance 2 from z. Moreover, the other $n - k$ nodes v_i have distance 2 from z, and each node w_i has distance 1 from z. Hence, c_z can be increased to $r = k + n + \frac{n+(n-k)}{2}$ and I' is a YES-instance.

$\Leftarrow$: If I is not a YES-instance, then there is no such set $\mathcal{F}' \subseteq \mathcal{F}$ of size k such that $\cup_{F_j \in \mathcal{F}'} = U$. After adding k arcs from z to nodes in $\{v_1, \ldots, v_m\}$, there is at least one node u_i such that there is no path from z to u_i. Hence, there are n nodes w_i and k nodes v_i with distance 1 from z and $n - k$ nodes u_i with distance 2 from z. Furthermore, there are $n' < n$ nodes u_i with distance 2 from z, and there is at least one node u_i with distance 3 from z. Summing up, c_z can be increased to at most $k + n + \frac{n'+(n-k)}{2} + \frac{1}{3}$ for $n' < n$ and I' is a NO-instance.

□

It is unknown whether DIRECTED CLOSENESS CENTRALITY is W[2]-hard on directed graphs with diameter 3. However, analogously to the problem variant with undirected input graphs, in Lemma 13 we show that the problem is polynomial-time solvable on graphs with diameter at most 2.

Lemma 13. DIRECTED CLOSENESS IMPROVEMENT *is solvable in* $\mathcal{O}(1)$ *time on directed graphs with diameter 2, and solvable in* $\mathcal{O}(n)$ *time on directed graphs with diameter 2.*

Proof. If the diameter of the graph is 1, then it is a clique and the distance from z to any node is 1 and thus maximal. If the diameter of the graph is 2, then the distance from z to any node is at most 2. It does not make a difference in terms of centrality improvement to which of the nodes with distance 2 from z we add an arc. Assume an arc (z, u) is added, for an arbitrary node u with distance 2 from z. Each path from z to a node $u', u' \neq u$ which contains u has length at least 2; hence, the arc (z, u) introduces no shortest paths except the one from z to u. □

2.4 Solution space reduction rules

This section introduces some *solution space reduction rules* for CLOSENESS IMPROVEMENT. Solution space reduction rules are rules that allow us to exclude certain parts of the solution space from an optimal solution. For a CLOSENESS IMPROVEMENT instance $(G = (V, E), z, k, r)$, the solution space is the set of edges E' such that for each $u \in V$, $\{z, u\} \in E'$ if and only if $\{z, u\} \notin E$.

In the following, let $I = (G = (V, E), z, k, r)$ be a CLOSENESS IMPROVEMENT instance. Furthermore, assume that there is no dominating set in G of size at most k.

Rule 1. Let $u \in V, u \neq z$ be a node such that $\deg(u) = 1$ and $N(u) = \{v\}$. Then there is an optimal solution for I which does not contain (z, u).

Proof. Assume that there is an optimal solution S for I such that $u \in S$. Then after adding $\{z, u\}$ to G, the distance between z and u is 1 and the distance between z and v is 2. Moreover, the length of shortest paths between z and the neighbors of v which contain u is 3. If we replace $\{z, u\}$ by $\{z, v\}$, then the distance between z and v is 1 and the distance between z and u is 2; the shortest paths between z and the neighbors of v which contain u is 2. Hence, if an optimal solution contains $\{z, u\}$, then we can replace this edge by $\{z, v\}$ without decreasing the closeness centrality of z. □

Rule 2. Let $V' = \{u_1, u_2, \ldots, u_\ell\} \subseteq V$ such that $N[u_1] = N[u_2] = \ldots N[u_\ell]$. Then an optimal solution S contains at most one edge between z and a node in V' and all but one node of V' can be excluded from the search space.

Proof. If $\{z, u_i\} \in S$ for an $u_i \in V'$, then the distance between z and u_i is 1 and the distance between z and each neighbor of u_i, including all other nodes in V', is at most 2. By adding another edge $\{z, u_j\}$ for an $u_j \in V'$, only the distance between z and u_j is decreased from 2 to 1, as the neighborhoods of u_i and u_j are the same. Hence, replacing $\{z, u_j\}$ by any other edge where an endpoint is z does not decrease the closeness centrality of z. □

Last, we introduce a search space reduction rule that involves the open neighborhoods of the nodes in the graph.

Rule 3. Let $V' = \{u_1, u_2, \ldots, u_\ell\} \subseteq V$ such that $N(u_1) = N(u_2) = \ldots N(u_\ell)$ and z is not connected to any node in V' by an edge. Then all except k nodes of V' can be excluded from the search space.

Proof. As the open neighborhood for each node in V' is the same, none of the nodes in V' are pairwise adjacent. Moreover, as none of the nodes in V' are adjacent to z, the distance between z and any node in V' is the same. If S contains an edge $\{z, u_i\}$ for $u_i \in V'$, then for any node $u_j \in V', u_j \neq u_i$, only the distance between z and u_j is decreased from 3 to 1 if S also contains $\{z, u_j\}$. There is no difference in terms of closeness centrality improvement, which of the nodes in V' are contained in S. As we may add at most k edges to G, we can exclude any additional nodes in V' from the search space. □

Unfortunately, we cannot transform these rules into *data* reduction rules, which is desirable in order to decrease the memory consumption by shrinking the input instance. If we are able exclude a part of the input from an optimal solution, such as degree-1 nodes (Rule 1), we cannot reduce the size of the input instance by removing degree-1 nodes from the input graph, as the sum of the multiplicative inverses of the distances of these nodes may be required to increase c_z to the specified target value r.

Chapter 3

Betweenness Centrality Improvement

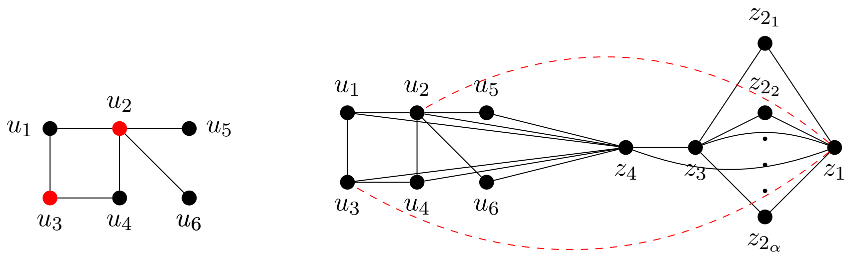
This chapter covers the problem of increasing the betweenness centrality of a specific node in a graph by inserting a certain number of edges into the graph. We show that, similar to CLOSENESS IMPROVEMENT, BETWEENNESS IMPROVEMENT is W[2]-hard with respect to the parameter number of edge additions and in FPT with respect to the parameter distance to clique. Furthermore, we introduce the problem of improving the betweenness centrality on directed, unweighted graphs and also show W[2]-hardness with respect to the parameter number of arc additions. Before we introduce the hardness proofs, we provide the Lemma 14 which is used to show the correctness of the parameterized reductions in this chapter:

Lemma 14. *Let $I = (G, z, k, r)$ be a* BETWEENNESS IMPROVEMENT *instance. If I is a* YES*-instance, then there is an optimal solution that only contains edges where one endpoint is z.*

Proof. Let S be a solution for I, and let $e := \{u_i, u_j\} \in S$. Furthermore, assume that e introduces at least one shortest path containing z (if it does not, then e can be replaced by any edge containing z). Without loss of generality, assume u_i precedes u_j on each of these paths. Then by replacing e by $e' := \{z, u_j\}$ in S, the distance between z and u_j decreases to 1 and the shortest paths previously containing e now contain e'. Hence, b_z does not decrease. □

Hence, if we compute a solution for some BETWEENNESS IMPROVEMENT instance, we need to find a subset of the graph's nodes of size k such that adding an edge between z and these nodes maximally increases the betweenness centrality of z. This directly implies the following corollary:

Corollary 6. BETWEENNESS IMPROVEMENT *is solvable in $\mathcal{O}(n^k)$ time where k is the number of edge additions and thus is in XP with respect to the parameter number of edge additions.*



(a) A Dominating Set instance ($I = (G, k = 2)$). The red colored nodes imply a solution for I.

(b) The constructed Betweenness Improvement instance $I' = (G, z_1, k, r)$. The red dashed edges imply a solution for I'.

Figure 6: Parameterized reduction from Dominating Set to Betweenness Improvement.

Proof. As shown in Lemma 14, an optimal solution for a Betweenness Improvement instance $I = (G, z, k, r)$ contains only edges where one endpoint is z. Hence, a brute-force approach can be used which iterates over all subsets of $V' \subseteq V \setminus \{z\}$ of size at most k such that $(z, v) \notin E$ for all $v \in V'$. For each V', the betweenness centrality of z after adding edges between z and each node in V' is computed and the subset which increases b_z at most forms the solution. The total running time of this algorithm is $\mathcal{O}(n^k \cdot (n + m))$. □

Moreover, in order to determine a proper running time for the algorithms in Section 3.2, we note that the betweenness centrality of a node in an undirected graph can be computed in $\mathcal{O}(nm)$ time (Brandes [Bra01]).

3.1 Hardness results

We show that Betweenness Improvement is W[2]-hard with respect to the parameter number of edge additions by a parameterized reduction from Dominating Set.

Theorem 9. Betweenness Improvement *is W[2]-hard with respect to the parameter number of edge additions k.*

Proof. We give a parameterized reduction from Dominating Set. Let $I = (G = (U, E), k)$ be a Dominating Set instance, where $U = \{u_1, \ldots, u_n\}$. We construct a Betweenness Improvement instance

$$I' = (G' = (V, E'), z_1, k, r = \alpha k + \frac{2}{3}\alpha(n - k) + \frac{1}{2}(k + \alpha + \binom{\alpha}{2})),$$

where $\alpha > \frac{3k(k-1)}{2}$. The graph G' is constructed as follows:

For each $u_i \in U$, we add a node u'_i to G'. Also, for each edge $\{u_i, u_j\} \in E$, we add an edge $\{u'_i, u'_j\}$ to E'. We set $U' := \{u'_1, \ldots, u'_n\}$. Next, we add the nodes $\{z_1, z_3, z_4\}$ and $Z_2 = \{z_{2_1}, \ldots, z_{2_\alpha}\}$ to G'. For each $z_{2_i} \in Z_2$, we add two edges $\{z_1, z_{2_i}\}$ and $\{z_{2_i}, z_3\}$ to G'. Furthermore, we add the edges $\{z_1, z_3\}, \{z_1, z_4\}$ and $\{z_3, z_4\}$. Finally, for each node $u'_i \in U'$, we add an edge $\{z_4, u'_i\}$. Figure 6 illustrates the construction.

As z_1 is adjacent to all nodes except the ones in U', a solution S for I' contains only edges where one endpoint is z_1 and each other one is in U' (Lemma 14). We now show that I' is a YES-instance if and only if I is a YES-instance: First, if I is a NO-instance, we show that there is an upper bound $r_u < r$ such that b_{z_1} can be increased to at most r_u by adding at most k edges to G'. Second, if I is a YES-instance, we provide a lower bound $r_\ell \geq r$ such that b_{z_1} can be increased to at least r_ℓ by adding at most k edges to G'. Both r_ℓ and r_u depend on α, which determines the size of G'. Finally, we determine a minimum value for α such that r_ℓ and r_u are strict bounds.

$\Rightarrow$ The input graph contains a dominating set $U_{DS} \subseteq U$ of size k. We say that U'_{DS} is the set of nodes in the constructed graph which correspond to the nodes in U_{DS}. Then, by adding k edges between z_1 and the nodes in U'_{DS}, for the following pairs of nodes there are shortest paths containing z_1:

- For each pair $(u' \in U'_{DS}, z \in Z_2)$, there is one shortest path of length 2, containing z_1. The number of such pairs is αk.
- For each pair $(u' \in U' \setminus U'_{DS}, z \in Z_2)$, two out of three shortest paths of length 3 between u' and the nodes in z contain z_1: One contains z_1 and a member of the dominating set, one contains z_1 and z_4, and one contains z_3 and z_4. The number of such pairs is $\alpha(n-k)$.
- For each pair $(u' \in U'_{DS}, z_3)$, there are two shortest paths of length 2 between u' and z_3: One contains z_1, the other one contains z_4. The number of such pairs is k.
- For each pair $(z_{2_i}, z_{2_j} \in Z_2 \mid i \neq j)$, there are two shortest paths of length 2 between z_{2_i} and z_{2_j}: One contains z_1, the other one contains z_3. The number of such pairs is $\binom{\alpha}{2}$.
- For each pair $(z_{2_i} \in Z_2, z_4)$, there are two shortest paths of length 2: One contains z_3 and the other one contains z_1. The number of such pairs is α

In total,

$$b_{z_1} \geq \alpha k + \frac{2\alpha(n-k)}{3} + \frac{k}{2} + \frac{\binom{\alpha}{2}}{2} + \frac{\alpha}{2},$$

which can be simplified to

$$b_{z_1} \geq \alpha k + \frac{2\alpha(n-k)}{3} + \frac{k + \alpha + \binom{\alpha}{2}}{2} =: r_\ell.$$

$\Leftarrow$ We prove the reverse direction by contraposition. That is, we show that I' is a NO-instance if I is a NO-instance. If the input instance does not admit a dominating set of size at most k, then there is at least one node which cannot be dominated. We analyze the number of shortest paths in the constructed BETWEENNESS IMPROVEMENT instance after adding k edges between z_1 and nodes in U'. We set $U'' \subseteq U' := \{u'_i \in U' \mid \{z_1, u'_i\} \in S\}$. Furthermore, let ℓ be the number of nodes that are undominated in G' after adding the edges in S, i.e. which are not adjacent to z_1 and which do not have a neighbor adjacent to z_1. As G does not admit a dominating set of size k, it holds that $\ell \geq 1$.

- For each pair $(u' \in U'', z \in Z_2)$, there is one shortest path of length 2, containing z_1. The number of such pairs is αk.
- For each pair $(u' \in U' \setminus U'', z \in Z_2)$ where u' is a neighbor of one of the nodes in U'', two out of three shortest paths of length 3 between u' and the nodes in z contain z_1: One contains z_1 and a member of the dominating set, one contains z_1 and z_4, and one contains z_3 and z_4. The number of such pairs is $\alpha(n - k - \ell)$.
- For each pair $(u'_i, u'_j \in U')$, there is a path of length 2 containing z_4. If the nodes in U' are not adjacent, then this is the shortest path. Additionally, there may be another shortest path containing z_1 of length 2, introduced by the edges in S. Hence, for each of up to $\binom{k}{2}$ pairs of nodes, one out of two shortest path contain z_1.
- For each pair $(u' \in U' \setminus U'', z \in Z_2)$ where u' is *not* a neighbor of one of the nodes in U'', there are two shortest paths betweens u' and z of length 3: One contains z_1 and z_4, the other one contains z_3 and z_4. The number of such pairs is $\alpha\ell$.
- For each pair $(u' \in U'', z_3)$, there are two shortest paths of length 2 between u' and z_3: One contains z_1, the other one contains z_4. The number of such pairs is k.
- For each pair $(z_{2_i}, z_{2_j} \in Z_2 \mid i \neq j)$, there are two shortest paths of length 2 between z_{2_i} and z_{2_j}: One contains z_1, the other one contains z_3. The number of such pairs is $\binom{\alpha}{2}$.

- For each pair $(z_{2_i} \in Z_2, z_4)$, there are two shortest paths of length 2: One contains z_3 and the other one contains z_1. The number of such pairs is α.

In total,

$$b_{z_1} \leq \alpha k + \frac{2\alpha(n-k-\ell)}{3} + \frac{\binom{k}{2}}{2} + \frac{\alpha\ell}{2} + \frac{k}{2} + \frac{\binom{\alpha}{2}}{2} + \frac{\alpha}{2},$$

which can be simplified to

$$b_{z_1} \leq \alpha k + \frac{2\alpha(n-k-\ell)}{3} + \frac{\binom{k}{2} + \alpha(\ell+1) + k + \binom{\alpha}{2}}{2} =: r_u.$$

In the last step, we need to determine a proper value for α such that $r_u < r_\ell$. Hence, the inequality that needs to be satisfied is

$$\alpha k + \frac{2\alpha(n-k-\ell)}{3} + \frac{\binom{k}{2} + \alpha(\ell+1) + k + \binom{\alpha}{2}}{2} < \alpha k + \frac{2\alpha(n-k)}{3} + \frac{k + \alpha + \binom{\alpha}{2}}{2}$$

for each $n, k, \ell \in \mathbb{N}, k \leq n, 1 \leq \ell \leq n$. This equation can be transformed to

$$\frac{\alpha\ell}{3} > \binom{k}{2}.$$

By setting $\ell = 1$ and transforming the binomial coefficient, we get

$$\alpha > \frac{3k(k-1)}{2}.$$

Hence, by setting α to a value strictly larger than $\frac{3k(k-1)}{2}$, the reduction is correct. Furthermore, the reduction is computable in fpt time: As the size of I' is polynomial to the size of I, G' can be constructed even in polynomial time. □

As social networks are usually power-law distributed, they tend to have a rather low h-index. However, as we see in Corollary 7, Betweenness Improvement remains NP-hard even on graphs with h-index 4.

Corollary 7. Betweenness Improvement *is NP-hard even on graphs with h-index 4.*

Proof. Let $I = (G, k)$ be a Dominating Set instance, where G is a graph with maximum degree three. Let G' be the graph constructed by the reduction used in the proof of Theorem 9. Then each node except z_1, z_3, and z_4 has degree at most four. Hence, the h-index of G' is at most four. As Dominating Set is NP-hard even on planar graphs with degree three [GJ90], Betweenness Improvement remains NP-hard on graphs with h-index four. □

3.2 Algorithmic results

Analogously to Section2.2, we derive some positive results for Betweenness Improvement. We show that the problem is fixed-parameter tractable with respect to the combined parameter vertex cover size and number of edge additions, and with respect to the combined parameter distance to cluster and number of edge additions.

Theorem 10. Betweenness Improvement *can be solved in* $\mathcal{O}(2^\ell \cdot 2^{\ell^k} \cdot (nm))$ *time, where* ℓ *is the vertex cover size and* k *is the number of edge additions.*

Proof. Let $(G = (V, E), z \in V, k, r)$ be a Betweenness Improvement instance. Furthermore, let $V_{VC} \subseteq V$ be a minimal vertex cover of size at most ℓ and $V_{IS} := V \setminus V_{VC}$ be its complement of size $n - \ell$. The proof is very similar to the proof of Theorem 5.

First, we observe that each $u \in V_{IS}$ has at least one neighbor in V_{VC}, otherwise V_{VC} is not a vertex cover. Furthermore, the subgraph induced by V_{IS} is edgeless, which means that for each $u \in V_{IS} : N(u) \subseteq V_{VC}$. Hence, there is a partitioning $\mathcal{P}$ of V_{IS} such that all nodes in a subset $p \in \mathcal{P}$ have the same neighbors in V_{VC}. As the vertex cover has size ℓ, the size of $\mathcal{P}$ is at most 2^ℓ.

Second, we observe that all nodes within a subset $p \in \mathcal{P}$ are equivalent in terms of their neighborhood. Hence, if an optimal solution contains edges from z to nodes in p, then it does not play a role which nodes in p are chosen.

We now describe an algorithm to solve Betweenness Improvement in $\mathcal{O}(2^\ell \cdot 2^{\ell^k} \cdot (n+m))$ time, where ℓ is the vertex cover size. The algorithm first iterates over all subsets $V'_{VC} \subseteq V_{VC}$ with size at most to k. As $V_{VC} = \ell$, there are $\mathcal{O}(2^\ell)$ such subsets. For a subset V'_{VC}, we add an edge between z and each node in V'_{VC}. If the size of V'_{VC} is strictly less than k, then we finally add the remaining $k - |V'_{VC}|$ edges between z and nodes in V_{IS}. As the size of $\mathcal{P}$ is 2^ℓ there are $\mathcal{O}(2^{\ell^k})$ ways to chose k partitions, where we cannot preclude that a partition is chosen multiple times. The total number of combinations is $\mathcal{O}(2^\ell \cdot 2^{\ell^k})$, and each combination takes another $\mathcal{O}(nm)$ time to compute b_z. Hence, Betweenness Improvement can be solved in $\mathcal{O}(2^\ell \cdot 2^{\ell^k} \cdot (nm))$ time. It follows that Betweenness Improvement is fixed-parameter tractable with respect to the combined parameter vertex cover size and number of edge additions. □

3.3 Directed Betweenness Improvement

This subsection covers results for the problem of improving the betweenness centrality of directed, unweighted graphs. First, we define betweenness centrality for directed, unweighted graphs, as the definition due to Freeman [Fre77] only measures the centrality over all unordered subsets of nodes of size two. A very natural definition, which is equivalent to the one used in further literature (e.g. by White and Borgatti [WB94]) is to measure the ratio of shortest paths containing a certain node z for both orders of any pair of nodes:

$$b_z = \sum_{s \in V} \sum_{\substack{t \in V \\ t \neq s; s, t \neq z \\ \sigma_{st} \neq 0}} \frac{\sigma_{stz}}{\sigma_{st}}$$

Next, we introduce the problem of improving the betweenness centrality of a directed, unweighted graph:

Directed Betweenness Improvement
Input: A directed, unweighted graph $G = (V, A)$, a node $z \in V$, an integer k and a rational number r.
Question: Is there an arc set S of size at most k such that $b_z \geq r$ in $G' = (V, A \cup S)$?

Analogously to the undirected problem variant, we show that we can maximally improve the betweenness centrality of a node z by adding arc where one of the endpoints is z.

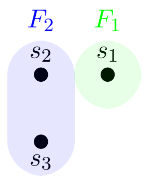
Lemma 15. *If a* Directed Betweenness Improvement *instance* $I = (G = (V, A), z, k, r)$ *is a* Yes*-instance, then there is a solution* S *that only contains arcs where either the source or the target is* z*.*

Proof. Assume S contains an arc (u_1, u_2) such that $u_1 \neq z$ and $u_2 \neq z$. Let $v_1, v_2 \in V$ such that (u_1, u_2) introduced a shortest path from v_1 to v_2 containing z and the arc (u_1, u_2). It is clear that u_2 must have been connected to z by a path before adding the arc (u_1, u_2). Furthermore, it is clear $\sigma_{v_1 v_2 z} \leq 1$, as there max be other paths from v_1 to v_2 not containing z.

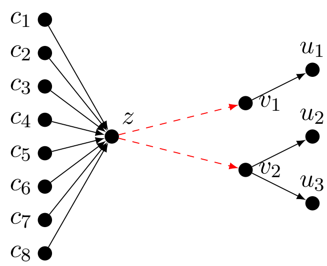
However, the shortest paths introduced by (u_1, u_2) necessarily contain u_1 and u_2; hence, these paths can be contracted by replacing (u_1, u_2) by (u_1, z). By this, we do not decrease b_z: Let ℓ be the length of a shortest path from v_1 to v_2 which contains (u_1, u_2) and z. Then, after replacing (u_1, u_2) by (u_1, z), there is exactly one shortest path of length $\ell' < \ell$ from v_1 to v_2. Hence, $\sigma_{stz} = 1$.

□

However, note that a solution S for a Yes-instance $I = (G, z, k, r)$ may also contain arcs where z is the source. For instance, $(G = \{z, v_1, v_2\}, A =$



(a) A SET COVER instance $I = (U, \mathcal{F}, k = 2)$ with solution $\{F_1, F_2\}$.



(b) The constructed DIRECTED BETWEENNESS IMPROVEMENT instance $I' = (G, z, k = 2, r = 4\frac{5}{6})$. The red dashed edges imply a solution for I'.

Figure 7: Parameterized reduction from SET COVER to DIRECTED BETWEENNESS IMPROVEMENT.

$\{(v_1, z)\}, z, 1, 1)$ is a YES-instance with solution $S = \{(z, v_2)\}$.

Corollary 8. DIRECTED BETWEENNESS IMPROVEMENT *is solvable in* $\mathcal{O}((2n)^k)$ *time where* k *is the number of edge additions, and thus is in XP with respect to the parameter number of edge additions.*

Proof. As shown in Lemma 15, an optimal solution for a DIRECTED BETWEENNESS IMPROVEMENT instance $I = (G, z, k, r)$ contains k arcs where one endpoint is z. For each node $u \neg z$, the arc (z, u) and (u, z) can be part of the optimal solution. Hence, we need to chose k arcs from the set of at most $2n$ possible arcs. For each such subset of size at most k, we add the corresponding arcs to the graph and measure the resulting betweenness centrality of z; the subset with the maximum increase is the solution. The total running time of this algorithm is $\mathcal{O}(n^k \cdot nm)$. □

We now show that DIRECTED BETWEENNESS IMPROVEMENT is also W[2]-hard with respect to the parameter number k of arc additions.

Theorem 11. DIRECTED BETWEENNESS IMPROVEMENT *is W[2]-hard with respect to the parameter number of arc additions* k*.*

Proof. We prove the hardness using a parameterized reduction from SET COVER. Let $I = (\mathcal{F} = \{F_1, \ldots, F_m\}, U = \{s_1, \ldots, s_n\})$ be a SET COVER instance. We construct a DIRECTED BETWEENNESS IMPROVEMENT instance $I' = (G = (V, A), z, k, k(1+n)+n))$, where G is a directed, unweighted graph. The construction is as follows:

- For each $s_i \in U$, add a node u_i. Set $V_U := \{u_1, \ldots, u_n\}$.
- For each $F_j \in \mathcal{F}$, add a node v_j. Set $V_V := \{v_1, \ldots, v_m\}$.
- Add the node z.

- Add the nodes $c_1, \dots, c_{m(m+n-1)}$; the set of these nodes is denoted as V_C.
- For each $c \in V_C$, add the arcs (c, z).
- For each $u_i \in V_U$ and each $v_j \in V_V$, add an arc (v_j, u_i) if $s_i \in F_J$.

In Figure 7, the reduction is illustrated.

Let I be a YES-instance and S be an arc set of size at most k, such that $b_z \geq r$ in $G' = (V, A \cup S)$. We now show that for each arc $a \in S$, the source is z and the target is one of the nodes in V_V. First, from Lemma 15 we know that there is a solution S' of the same size where z is an endpoint of each arc $a \in S'$. Hence, in the following we assume that for each $a \in S$, one of its endpoints is z.

Moreover, if a solution S contains an arc $(z, u_i), u_i \in V_U$, we can replace it by an arc $(z, v_i), v_i \in V_V$ such that $s_i \in F_j$ without decreasing b_z: The arc (z, u_i) introduces paths from the nodes z and all its predecessors to u_i. By replacing (z, u_i) by (z, v_j), the paths remain, but additionally paths from z and its predecessors to v_j are added. Hence, b_z does not decrease.

Furthermore, by adding the node set V_C of size $m(m+n-1)$, we ensure that by adding arcs where the source is z, we obtain more (shortest) paths containing z than by adding arcs where the endpoint is z: Each arc from z to a node in V_V introduces at least $m(m+n-1)$ shortest paths containing z. However, adding an arc from a node in V_U to z introduces at most $m((m-1)+(n-1))$ paths containing z: Each node in V_U has at most m predecessors in V_V. Furthermore, z may have at most m successors in V_V and at most $(n-1)$ successors in V_U. Hence, by adding an arc from a node in V_U to z, c_z is increased by at most $m((m-1)+(n-1))$.

We now show that the reduction is correct, i.e. that I is a YES-instance if and only if I' is a YES-instance.

$\Rightarrow$: If I is a YES-instance, then there is a $\mathcal{F}' \subseteq \mathcal{F}$ of size k such that $\cup_{F_j \in \mathcal{F}'} = U$. By adding arcs (z, v_j) for each $F_j \in \mathcal{F}'$, the following shortest paths contain z:

- For each v_j such that $F_j \in \mathcal{F}'$ and each $c \in V_C$, there is a shortest path from c to v_j containing z. As $|\mathcal{F}'| = k$, the number of such shortest paths is $k(m(m+n-1))$.
- For each u_i and each $c \in V_C$, there is a shortest path from c to u_i containing z. In total, the number of such paths is $n(m(m+n-1))$.

Hence, b_z can be increased to $(k+n)(m(m+n-1))$ and I' is a YES-instance.

$\Leftarrow$: If I is not a YES-instance, then there is no such set $\mathcal{F}' \subseteq \mathcal{F}$ of size k such that $\sum_{\mathcal{F}'} = U$. Let S be a set of size k which contains arcs from nodes v_j to z. The graph $G' = (V, A \cup S)$ contains the following shortest paths, each containing z:

- For each $v \in V_V$ which is the endpoint in an arc in S, and each $c \in V_C$, there is a shortest path from c to v containing z. As the target of all arcs in S is a node in V_V and $|S| = k$, the number of such shortest paths is $k(m(m+n-1))$.
- As I is a NO-instance, there is at least one node u in G' such that there is no path from the nodes in V_C to u. Hence, the number of paths from nodes in V_C to nodes in V_U is at most $n-1(m(m+n-1))$.

Hence, b_z can be increased to at most $(k+n-1)(m(m+n-1))$ and I' is a YES-instance. □

Hence, the problem of improving the betweenness centrality is W[2]-hard on both directed and undirected graphs. A last result is that DIRECTED BETWEENNESS IMPROVEMENT even remains W[2]-hard on directed, acyclic graphs.

Corollary 9. DIRECTED BETWEENNESS IMPROVEMENT *is W[2]-hard even on directed acyclic graphs.*

Proof. The graphs constructed in the reduction in the proof of Theorem 11 are directed acyclic graphs. □

3.4 Relationship to Independent Set

In Chapter 2, we show that there is a relationship between DOMINATING SET and CLOSENESS IMPROVEMENT: If our task is to maximally increase the closeness centrality of a node z by adding at most k edges to the graph, then by adding edges between z and all nodes of a dominating set we obtain an optimal solution, if there is a dominating set of size at most k. In this section, we want to show similarities between BETWEENNESS IMPROVEMENT and INDEPENDENT SET in order to have a better understanding for the structure of BETWEENNESS IMPROVEMENT. Furthermore, we managed to exploit the similarities between DOMINATING SET and CLOSENESS IMPROVEMENT: If the number k of edge additions is larger than the size of a dominating set of the input graph, then adding edges to the nodes of the dominating set forms an optimal solution. Hence, we want to show whether there is a similar relationship between INDEPENDENT SET and BETWEENNESS IMPROVEMENT.

However, we show that for a Betweenness Improvement instance $I = (G = (V, E), z, k, r)$, adding edges from z to nodes u and v where $\{u, v\} \in E$ can be necessary to maximize the betweenness centrality of z.

Let $I = (G = (V, E), z, k, r)$ be a Betweenness Improvement instance, where $V_{IS} \subseteq V$ is an independent set of G of size at most k. At first sight, it seems like adding edges between z and nodes in V_{IS} yields a high betweenness centrality of z: As none of the nodes in V_{IS} are adjacent, the distance between each pair of these nodes after adding the edges is 2, and for each pair of nodes in V_{IS} there is a shortest path containing z. If V_{IS} is a Distance-3 Independent Set, then by adding edges between z and nodes in V_{IS} we achieve a betweenness centrality for z of at least $\frac{k(k-1)}{2}$, as for all shortest paths between nodes in V_{IS} there is exactly one shortest path, each containing z. Moreover, we can even improve this lower bound if we find a Distance-7 Independent Set of size k: Then after adding edges between z and nodes in V_{IS}, all shortest path between the nodes in V_{IS} have length 2 and contain z. Additionally, all shortest paths between neighbors of nodes in V_{IS} have length 4 and contain z, as their distance before adding the edges was at least five (otherwise, V_{IS} was no Distance-7 Independent Set).

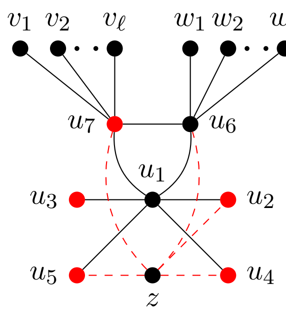


Figure 8: Betweenness Improvement instance $I = (G, z, k = 5, r)$. The red colored nodes form an independent set of size five. The red dashed edges illustrate a solution where the endpoints u_2, u_4, u_5, u_6, u_7 do not form an independent set, but which yields a larger improvement of b_z than the solution obtained by adding edges between z and the red colored nodes.

Unfortunately, for some instances it may be better in terms of betweenness centrality improvement to add edges between z and nodes $V' \subseteq V$ which are not an independent set. We provide an example in Figure 8: By adding edges between z and the red colored nodes forming an independent set of size five, we have $b_z = 5+2\ell$: For each pair of the red colored nodes, there are two shortest paths, one of which contains z. Moreover, for each of the ℓ nodes v_i, there are two shortest paths between v_i and u_2, u_3, u_4 and u_5, one of which contains z. However, we obtain $b_z = 9/2 + 3\ell$ by adding the five red dashed edges to the graph: Between each pair of the nodes in $\{u_2, u_4, u_5, u_6, u_7\}$ except $\{u_6, u_7\}$, there are two shortest paths, one of which contains z; these paths increase b_z by $\frac{9}{2}$. Moreover, for each node $v_1, \ldots, v_\ell, w_1, \ldots, w\ell$, there are two shortest paths to the nodes u_2, u_4 and u_5, half of which contain z; these paths increase b_z by 3ℓ. Note that in the latter solution, the endpoints of the edges do not form an independent set, but the quality of this solution is better than the quality of the first one, where the endpoints form an independent set. The quality of the solution containing edges to the red colored nodes may even become arbitrarily bad with a large value for ℓ.

It is not clear whether there is a strategy to compute an optimal solution for a Betweenness Improvement instance by exploiting the similarities to Independent Set. However, adding edges to nodes that form an independent set, e.g. in combination with a polynomial-time local search algorithm, may be helpful for heuristically solving Betweenness Improvement.

Chapter 4

Destructive Betweenness Improvement

In this chapter, we introduce DESTRUCTIVE BETWEENNESS IMPROVEMENT, which is strongly related to BETWEENNESS IMPROVEMENT: Given an undirected graph G, we ask if we can increase the betweenness centrality of a node z to at least r by *removing* edges from the input graph. The term *destructive* phrases that ask whether a property, such as a certain value for the betweenness centrality of a node, can be achieved by "destroying" parts of the graphs, such as edges. A real-world application for this scenario is, e.g., that in a computer network, an adversary wants to tear down connections between nodes in order to increase the amount of information passing it. We refer to this problem as the DESTRUCTIVE BETWEENNESS IMPROVEMENT (DBI) problem, using the following formal definition:

DESTRUCTIVE BETWEENNESS IMPROVEMENT

Input: An undirected, unweighted graph $G = (V, E)$, a node $u \in V$, an integer k and a rational number r.

Question: Is there a set of edges $S \subseteq E$, $|S| \leq k$, such that $b_u \geq r$ in $G' = (V, E \setminus S)$?

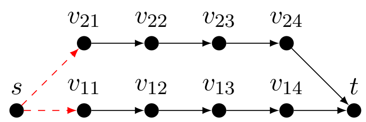
Furthermore, we introduce the directed variant referred to as DDBI, where the input graph is a directed, unweighted graph:

DIRECTED DESTRUCTIVE BETWEENNESS IMPROVEMENT

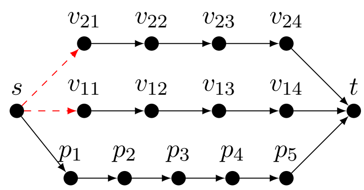
Input: A directed, unweighted graph $G = (V, A)$, a node $u \in V$, an integer k and a rational number r.

Question: Is there a set of arcs $S \subseteq A$, $|S| \leq k$, such that $b_u \geq r$ in $G' = (V, A \setminus S)$?

We first show that DIRECTED DESTRUCTIVE BETWEENNESS IMPROVEMENT is W[1]-hard with respect to the number of arc additions - unfortu-



(a) A BOUNDED EDGE DIRECTED (s,t)-CUT instance $I = (G, k = 2, \ell = 5, s, t)$. The red dashed lines form a solution for I.



(b) The constructed DIRECTED DESTRUCTIVE BETWEENNESS IMPROVEMENT instance $I' = (G', p_1, k = 2, r = \ell - 1 + \frac{1}{(n-2)!})$. The red dashed edges imply a solution for I'.

Figure 9: Parameterized reduction from BOUNDED EDGE DIRECTED (s,t)-CUT to DIRECTED DESTRUCTIVE BETWEENNESS IMPROVEMENT with respect to the number k of arc deletions.

nately, we have no result for the undirected variant. Then, we introduce a natural and simple greedy strategy and show that it has an arbitrarily small approximation ratio for the undirected variant.

4.1 Hardness result

We show that DIRECTED DESTRUCTIVE BETWEENNESS IMPROVEMENT (DDBI) is W[1]-hard with respect to the number of arc deletions by a parameterized reduction from the W[1]-hard BOUNDED EDGE DIRECTED (s,t)-CUT (BEDC) with parameter number k of arc deletions (Golovach and Thilikos [GT11]). Unfortunately, we do not provide a parameterized reduction to the undirected variant of this problem.

BOUNDED EDGE DIRECTED (s,t)-CUT

Input: A directed, unweighted graph $G = (V, A)$, two distinct nodes $s, t \in V$, tho positive integers k, ℓ.

Question: Is there a set of arcs $S \subseteq A$ of size at most k such that $G' = (V, A \setminus S)$ contains no (s,t)-path of length at most ℓ?

Theorem 12. DIRECTED DESTRUCTIVE BETWEENNESS IMPROVEMENT *is W[1]-hard with respect to the parameter number of arc deletions.*

Proof. Using a parameterized reduction, we reduce a BEDC instance $I = (G, k, \ell, s, t)$ to a DDBI instance $I' = (G', p_1, k, \ell - 1 + \frac{1}{(n-2)!})$ in polynomial time. The construction of G' is as follows: Starting with a copy of G, we additionally introduce $\ell - 1$ new nodes $p_1, \ldots, p_\ell$ and the arc set $\{(s, p_1), (p_1, p_2), \ldots, (p_{\ell-1}, p_\ell), (p_\ell, t)\}$. Furthermore, we remove all arcs from G' where s is the target node and all nodes where t is the source node: As we regard the shortest paths where s is the source and t is the target, all

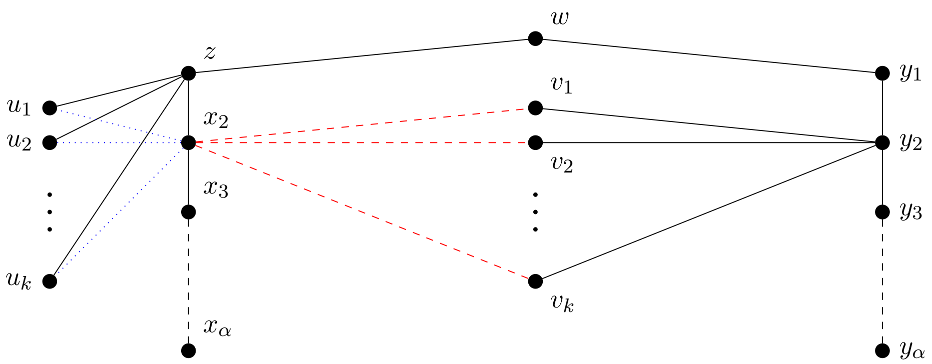


Figure 10: Destructive Betweenness Improvement instance I. The greedy strategy introduced in Section 4.2 has an arbitrarily low approximation ratio on this instance. The red dashed edges are part of the optimal solution, the blue dotted edges are the result of the greedy strategy: Deleting up to $k-1$ of the red dashed edges from the graph does not increase b_z, whereas deleting all red dashed edges maximally increases c_z.

paths containing s or t more than once contain a loop and are not minimal. An example is provided in Figure 9. We now show that the reduction is correct:

$\Rightarrow$ If the input instance I is a Yes-instance, then there is an arc set $S \subseteq A$ of size k such that after removing S from A, there is no path of length at most ℓ from s to t. Removing the arcs corresponding to S from G', it holds that the path from s to t containing the nodes $p_1, \ldots, p_\ell$ is a shortest path from s to t with length $\ell+1$. Hence, the betweenness centrality of p_1 is larger than $\ell - 1 + \frac{1}{(n-2)!}$: The paths from s to $p_2, \ldots, p_\ell$ and to t all contain p_1. However, there may be arbitrary other paths from s to t of length $\ell + 1$.

$\Leftarrow$ If the input instance is a No-instance, then for all arc sets $S \subseteq A$ of size at most k it holds that after removing S from A, the shortest path between s and t has length less than ℓ. Thus, by removing k arcs from G', none of the shortest paths from s to t contains p_1, as the only path from s to t containing p_1 has length $\ell+1$. It follows that $b_{p_1} = \ell-1$ and I' is a No-instance.

From the parameterized reduction follows that DDBI is W[1]-hard with respect to the number k of arc deletions, as BEDC is W[1]-hard on directed graphs with respect to the size of the arc deletion set S. □

4.2 Greedy strategy

In this subsection, we introduce a very simple greedy strategy for Destructive Betweenness Improvement. As a very simple and natural greedy strategy for Directed Closeness Improvement and Directed Closeness Improvement provides a $1 - \frac{1}{e}$-approximation (Crescenzi et al. [Cre+16]), but have as arbitrarily small approximation ratio for Betweenness Improvement (D'Angelo et al. [DSV16]). Our goal is to analyze whether a similar strategy for Directed Betweenness Improvement also provides an approximation. Unfortunately, this strategy does not serve as an approximation algorithm, as the ratio between the size of an optimal solution and the size of the solution generated by the greedy strategy may become arbitrarily low. We now introduce the greedy strategy and such an instance.

The greedy strategy is as follows: Given an instance $I = G, z, k, r$, we introduce k steps. In each step, we determine which edge deletion increases b_z at most and remove that edge from G.

For a graph $G = (V, E)$, a node $z \in V$ and a positive integer k, we denote $b_{OPT}(z)$ the maximum betweenness centrality of z in G after removing k edges, and $b_{GRE}(z)$ the betweenness centrality of z in G after deleting k edges as computed by the greedy strategy.

We introduce a Directed Betweenness Improvement instance $I = (G, z, k, r)$ as visualized in Figure 10 such that the ratio between $b_{OPT}(z)$ and $b_{GRE}(z)$ is arbitrarily small. We now proof this statement by computing $b_{OPT}(z)$ and $b_{GRE}(z)$.

An optimal solution is $S = \{\{x_2, v_i\} \mid 1 \leq i \leq k\}$. By removing S from G, the following shortest paths contain z:

- For each $u_i, 1 \leq i \leq k$, and each $y_j, 1 \leq j \leq \alpha$, there is one shortest path from u_i to y_j containing z.
- For each $u_i, 1 \leq i \leq k$, and each $v_j, 1 \leq j \leq k$, there is one shortest path from u_i to v_j containing z.
- For each $u_i, 1 \leq i \leq k$, and each $x_j, 2 \leq j \leq \alpha$, there is one shortest path from w to u_i and one from w to x_j containing z.
- For each $x_i, 2 \leq i \leq \alpha$, and each $v_j, 1 \leq j \leq k$, there is one shortest path from x_i to v_j containing z.
- For each $x_i, 2 \leq i \leq \alpha$, and each $y_j, 1 \leq j \leq \alpha$, there is one shortest path from x_i to y_j containing z.

In total, by removing S from G, we increase b_z to $(k + (\alpha - 1))(\alpha + k + 1)$.

Now, we analyze the result of the greedy strategy. In each step, we must remove an edge from u_i to $x_2, 1 \leq i \leq k$, as the removal of any other edge does not increase b_z. After k steps, all edges from u_i to x_2 are removed. The following shortest paths then contain z:

- For each $u_i, 1 \leq i \leq k$, and each $x_j, 2 \leq j \leq \alpha$, there is one shortest path from u_i to y_j containing z.
- For each $u_i, 1 \leq i \leq k$, and each $x_j, 2 \leq j \leq \alpha$, there is one shortest path from w to u_i and one from w to x_j. Each path contains z.
- For each $v_i, 1 \leq i \leq k$, there are two shortest paths from v_i to w. One contains x_2 and z and the other one contains y_2 and y_1.
- For each $x_i, 2 \leq i \leq \alpha$, there is one shortest path from x_i to y_1 containing z.
- For each $u_i, 1 \leq i \leq k$ and each $y_j, 1 \leq j \leq \alpha$, each shortest path from u_i to y_j contains z.

In total, the greedy strategy increases b_z to $(k+1)(\alpha-1)+k(\frac{3}{2}+\alpha)+\alpha$. Hence, using the greedy strategy, the betweenness centrality of z increases linear with α. However, by deleting the edges from an optimal solution, b_z increases quadratically with α. Hence, the solution quality becomes arbitrary bad with a large value for α.

However, this result does not necessarily mean that the greedy strategy is useless for real-world data. Practical evaluation and comparison between the solution quality of the greedy strategy and an optimal solver may be a future task in order to determine the practical usability of the greedy strategy.

Chapter 5

Conclusion and Outlook

We studied the parameterized complexity of CLOSENESS IMPROVEMENT and BETWEENNESS IMPROVEMENT for the perhaps most natural parameter, that is the number of edge additions. We showed that both problems are in XP, but unfortunately they are W[2]-hard with this parameter. The same holds for the problem variants on directed graphs. Moreover, we showed that CLOSENESS IMPROVEMENT is NP-hard even on planar graphs with maximum degree 3 and W[2]-hard on graphs with diameter 3 with parameter number of edge additions. BETWEENNESS IMPROVEMENT remains NP-hard on graphs with h-index 6, and the directed variant of both centrality improvement problems remain W[2]-hard even on directed acyclic graphs, with respect to the number of edge additions.

However, we were also able to show that CLOSENESS IMPROVEMENT is in FPT for some non-standard parameters, such as the node deletion distance to a clique or a cluster graph, or the size of a vertex cover; CLOSENESS IMPROVEMENT and BETWEENNESS IMPROVEMENT are even polynomial-time solvable on graphs with diameter 2.

For DIRECTED BETWEENNESS IMPROVEMENT, we showed that the problem is W[1]-hard with respect to the number of edge deletions, and that a natural greedy strategy has an arbitrary low approximation ratio.

Most of the results in this work are negative hardness or inapproximability results, and the few polynomial-time or fixed-parameter tractability results might not be of much practical use, due to the high running time or very restricted graph classes. However, they provide first insights into the structure of centrality improvement problems. We propose some related subjects and open questions which might be worth further research.

5.1 Betweenness vs. Closeness Improvement

One aspect which has not been discussed in this paper is whether Betweenness Improvement is harder to solve than Closeness Improvement, in terms of parameterized complexity. Unfortunately, we were not able to find (parameterized) reductions from Betweenness Improvement to Closeness Improvement or vice versa, as the structures of the problems are very unsimilar. One major problem which occurred when designing a reduction was the impact of an edge addition on the betweenness centrality gain or closeness centrality gain, respectively. Let z be the node whose closeness or betweenness centrality shall be maximized. Then decreasing the distance between two nodes $u, v \neq z$ as a side effect of an edge addition does not impact the closeness centrality of z. The same is not true for the betweenness centrality. Moreover, there is no relationship between the absolute distance between z and other nodes and the betweenness centrality of z. Hence, it is unclear how a reduction from one problem to the other might work.

5.2 More centrality improvement variants

In this section, we introduce some more problems related to Betweenness Improvement and Closeness Improvement: We present Betweenness Balancing, where we ask to minimize the maximum betweenness centrality of all nodes in the graph. A practical scenario is to evenly distribute traffic in the network by avoiding bottlenecks in form of few nodes with a high betweenness centrality. Furthermore, we introduce Relative Closeness Improvement, where the task is to relatively improve a node's closeness centrality by removing edges from the graph. For the latter problems, this work does not cover any results, but analysis of the complexity and work on algorithms of these problems might be interesting.

As an outlook, this section introduce some decision problems that are related to Closeness Improvement or Betweenness Improvement.

5.2.1 Betweenness Editing

A very canonical variant of Betweenness Improvement is to allow a certain number k of edge deletions *and* additions, in order to improve a node's betweenness centrality. As both Betweenness Improvement and Directed Betweenness Improvement are W[2]-hard, it seems likely that Betweenness Editing remains W[2]-hard with respect to the number of edge additions and deletions. Another variant may be to improve a node's closeness or betweenness centrality by *replacing* at most k adjacent to a node z by k other nodes. This problem better models the cost of infrastructure: For instance, it might be too expensive for a logistics company to add new routes, but it may replace routes by new ones in order to decrease the distances, and hence the cost, to its customers.

5.2.2 Betweenness Balancing

Another modification of BETWEENNESS IMPROVEMENT is motivated from the following scenario: A network provider wants to add edges to her infrastructure such that for each node, the workload is as low as possible. In other words, the provider wants to avoid hotspots in his network where much traffic passes through. Defined as an optimization problem, the goal is to add or delete a certain number k of edges such that the maximum betweenness centrality of all nodes is minimized.

5.2.3 Relative Improvement

In contrast to the problem of improving the betweenness centrality, it is not possible to improve the closeness centrality of a node by removing edges from the graph. However, it is possible to *decrease* the closeness or betweenness centrality of other nodes by deleting edges. Hence, deleting edges may be beneficial when our goal is to have a *relatively* high closeness or betweenness centrality compared to other nodes in the graph. Therefore, we might ask whether it is possible to remove a certain number k of edges such that the centrality of a certain node z is larger than the centrality of any other node (or that at most ℓ other nodes have a higher centrality.)

A real-world application for relative closeness improvement might be that a competitor is not able to improve her own centrality by adding edges, but destructively removing edges is possible and being better than as many other competitors as possible is beneficial (this might be true for almost any economic scenario). Another real-world application might be found in (social) networks: Assume that a social network is modeled as a graph, and the closeness centrality of a member, for instance a voting candidate, somehow reflects its influence on other members (for instance, the voters) in the network. Then decreasing the popularity, and hence the influence of another candidate might be easier than increasing the own popularity.

5.3 Open questions

While our work provides first classifications for CLOSENESS IMPROVEMENT and BETWEENNESS IMPROVEMENT in terms of parameterized complexity and for few restricted graph classes, there are still many open questions. For instance, for many graph parameters it is not clear whether the centrality improvement problems are fixed-parameter tractable or not. Moreover, there may be fpt results for multivariate parameters such as maximum degree and number of edge additions. Finally, the fpt-time algorithms provided in this work are useful for the classification of these problems, but are not of practical use due to the high running time. However, further research may uncover more insights into the structures of these problems, and ideas how to practically exploit them.

Bibliography

[Bav48] Alex Bavelas. “A mathematical model for group structures.” In: *Human Organization* 7.3 (1948), pp. 16–30 (cit. on p. 6).

[Bra01] Ulrik Brandes. “A faster algorithm for betweenness centrality.” In: *Journal of Mathematical Sociology* 25.2 (2001), pp. 163–177 (cit. on pp. 6, 36).

[Bra08] Ulrik Brandes. “On variants of shortest-path betweenness centrality and their generic computation.” In: *Social Networks* 30.2 (2008), pp. 136–145 (cit. on pp. 3, 6).

[CKX10] Jianer Chen, Iyad A. Kanj, and Ge Xia. “Improved upper bounds for vertex cover.” In: *Theoretical Computer Science* 411.40 (2010), pp. 3736–3756 (cit. on pp. 12, 16).

[Cre+16] Pierluigi Crescenzi, Gianlorenzo D’angelo, Lorenzo Severini, and Yllka Velaj. “Greedily improving our own closeness centrality in a network.” In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11.1 (2016), p. 9 (cit. on pp. 6, 50).

[Cyg+15] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Vol. 3. Springer, 2015 (cit. on p. 10).

[DF12] Rodney G Downey and Michael Ralph Fellows. *Parameterized complexity.* Springer Science & Business Media, 2012 (cit. on p. 29).

[DF13] Rodney G Downey and Michael R Fellows. *Fundamentals of parameterized complexity.* Vol. 4. Springer, 2013 (cit. on p. 10).

[DSV16] Gianlorenzo D’Angelo, Lorenzo Severini, and Yllka Velaj. “On the maximum betweenness improvement problem.” In: *Electronic Notes in Theoretical Computer Science* 322 (2016), pp. 153–168 (cit. on pp. 6, 50).

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory, volume XIV of Texts in Theoretical Computer Science. An EATCS Series.* 2006 (cit. on p. 10).

[Fre77] Linton C Freeman. “A set of measures of centrality based on betweenness.” In: *Sociometry* 40 (1977), pp. 35–41 (cit. on pp. 6, 41).

[Fre78] Linton C. Freeman. “Centrality in social networks conceptual clarification.” In: *Social Networks* 1.3 (1978), pp. 215–239 (cit. on p. 3).

[GJ90] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990 (cit. on pp. 17, 39).

[GT11] Petr A. Golovach and Dimitrios M. Thilikos. “Paths of bounded length and their cuts: Parameterized complexity and algorithms.” In: *Discrete Optimization* 8.1 (2011). Parameterized Complexity of Discrete Optimization, pp. 72–86 (cit. on p. 48).

[Hüf+10] Falk Hüffner, Christian Komusiewicz, Hannes Moser, and Rolf Niedermeier. “Fixed-parameter algorithms for cluster vertex deletion.” In: *Theory of Computing Systems* 47.1 (2010), pp. 196–217 (cit. on p. 22).

[Kan+11] U Kang, Spiros Papadimitriou, Jimeng Sun, and Hanghang Tong. “Centralities in large networks: Algorithms and observations.” In: *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM. 2011, pp. 119–130 (cit. on p. 4).

[LKF05] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. “Graphs over time: densification laws, shrinking diameters and possible explanations.” In: *Proceedings of the eleventh ACM International Conference on Knowledge Discovery in Data Mining (SIGKDD)*. ACM. 2005, pp. 177–187 (cit. on p. 17).

[Lok+13] Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, M. S. Ramanujan, and Saket Saurabh. “Hardness of r-dominating set on Graphs of Diameter (r + 1).” In: *Parameterized and Exact Computation - 8th International Symposium, IPEC 2013, Sophia Antipolis, France, September 4-6, 2013, Revised Selected Papers*. 2013, pp. 255–267 (cit. on p. 19).

[New05] Mark EJ Newman. “A measure of betweenness centrality based on random walks.” In: *Social networks* 27.1 (2005), pp. 39–54 (cit. on p. 3).

[NG04] Mark EJ Newman and Michelle Girvan. “Finding and evaluating community structure in networks.” In: *Physical review E* 69.2 (2004), p. 026113 (cit. on p. 3).

[Nie06] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006 (cit. on p. 10).

[OAS10] Tore Opsahl, Filip Agneessens, and John Skvoretz. "Node centrality in weighted networks: Generalizing degree and shortest paths." In: *Social Networks* 32.3 (2010), pp. 245–251 (cit. on p. 3).

[OCL08] Kazuya Okamoto, Wei Chen, and Xiang-Yang Li. "Ranking of closeness centrality for large-scale social networks." In: *International Workshop on Frontiers in Algorithmics*. Springer. 2008, pp. 186–195 (cit. on p. 3).

[RK16] Matteo Riondato and Evgenios M Kornaropoulos. "Fast approximation of betweenness centrality through sampling." In: *Data Mining and Knowledge Discovery* 30.2 (2016), pp. 438–475 (cit. on p. 6).

[Rom+14] M Puck Rombach, Mason A Porter, James H Fowler, and Peter J Mucha. "Core-periphery structure in networks." In: *SIAM Journal on Applied Mathematics* 74.1 (2014), pp. 167–190 (cit. on p. 8).

[RS10] Mikail Rubinov and Olaf Sporns. "Complex network measures of brain connectivity: uses and interpretations." In: *Neuroimage* 52.3 (2010), pp. 1059–1069 (cit. on p. 3).

[SW17] Manuel Sorge and Martin Weller. *The graph parameter hierarchy (Manuscript)*. 2017. URL: `http://fpt.akt.tu-berlin.de/msorge/parameter-hierarchy.pdf` (cit. on p. 7).

[Tho99] Mikkel Thorup. "Undirected single-source shortest paths with positive integer weights in linear time." In: *Journal of the ACM* 46.3 (1999), pp. 362–394 (cit. on p. 14).

[WB94] Douglas R. White and Stephen P. Borgatti. "Betweenness centrality measures for directed graphs." In: *Social Networks* 16.4 (1994), pp. 335–346 (cit. on p. 41).