New Algorithms and Hardness Results for Connected Clustering

Jan Eube
University of Bonn, Germany

Heiko Röglin
University of Bonn, Germany

Abstract

Connected clustering denotes a family of constrained clustering problems in which we are given a distance metric and an undirected connectivity graph G that can be completely unrelated to the metric. The aim is to partition the n vertices into a given number k of clusters such that every cluster forms a connected subgraph of G and a given clustering objective gets minimized. The constraint that the clusters are connected has applications in many different fields, like for example community detection and geodesy.

So far, k-center and k-median have been studied in this setting. It has been shown that connected k-median is $\Omega(n^{1-\epsilon})$-hard to approximate which also carries over to the connected k-means problem, while for connected k-center it remained an open question whether one can find a constant approximation in polynomial time. We answer this question by providing an $\Omega(\log^*(k))$-hardness result for the problem. Given these hardness results, we study the problems on graphs with bounded treewidth. We provide exact algorithms that run in polynomial time if the treewidth w is a constant. Furthermore, we obtain constant approximation algorithms that run in FPT time with respect to the parameter $\max(w, k)$.

Additionally, we consider the min-sum-radii (MSR) and min-sum-diameter (MSD) objectives. We prove that on general graphs, connected MSR can be approximated with an approximation factor of $(3+\epsilon)$ and connected MSD with an approximation factor of $(4+\epsilon)$. The latter also directly improves the best known approximation guarantee for unconstrained MSD from $(6+\epsilon)$ to $(4+\epsilon)$. We complement this with a reduction showing that connected MSR is NP-hard to approximate with an approximation factor smaller than $(4/3)$.

2012 ACM Subject Classification Theory of computation → Facility location and clustering

Keywords and phrases Clustering, Connectivity constraints, Approximation algorithms, Hardness of approximation

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.135

Related Version *Full Version*: `https://arxiv.org/abs/2511.19085` [20]

Funding This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 390685813; 459420781 and by the Lamarr Institute for Machine Learning and Artificial Intelligence lamarr-institute.org.

1 Introduction

Clustering describes a family of problems that, given a set of data points, aim at grouping together similar points into a limited number of clusters. These kinds of problems have been studied widely both in practice as well as in theory. There are numerous objective functions to evaluate the quality of a clustering, among them k-center, k-median, k-means, and min-sum-radii (MSR). In all of the listed objectives, the data points are usually contained in a metric space which defines a distance function and the number of clusters k is given beforehand. In addition to the clusters, one also needs to determine a center for each cluster and the quality of the clustering is determined by the distance of the data points to their respective centers.

In k-center one aims to minimize the maximum distance among any data point to its respective center. In k-median the sum of the distances of all points to the centers and in k-means the sum of the squared distances to the centers should be minimized. In min-sum-radii the objective value is the sum of the radii of all clusters, with the radius being the maximum distance of any point in this cluster to the center. There is also a variant of this problem called min-sum-diameter (MSD) where no centers are needed and instead of the radius one considers the diameter of each cluster, which is the maximum distance between two points in this cluster.

All of these problems are NP-hard but there exist numerous approximation results. For k-center multiple 2-approximation algorithms are known [26, 23], which is the best possible, unless $\mathsf{P} = \mathsf{NP}$ [28]. For k-median the currently best known algorithm achieves a $(2+\epsilon)$-approximation guarantee [11] and for k-means there exists a $(9+\epsilon)$-approximation algorithm for general metrics and a $(5.912+\epsilon)$-approximation algorithm for Euclidean metrics [2, 10]. Recently, Buchem et al. provided a $(3+\epsilon)$-approximation algorithm for MSR, which also directly implies a $(6+\epsilon)$-approximation guarantee for MSD [6].

In many applications the clusters should fulfill additional requirements besides having a good objective value, which is why researchers have studied these objectives under various side constraints. For example, one can require that the clusters are not too small or too large which results in private [1] and capacitated clustering [32]. Additionally, different variants of fair clustering have been studied over the last decade [7, 40, 38].

In this work, we deal with a connectivity side constraint. In this setting, we are not only given a distance metric but the data points are also vertices in a connectivity graph which is independent of the metric. The aim is to find clusters that minimize the respective objective while ensuring that every cluster induces a connected subgraph of the connectivity graph. This side constraint was introduced by Ge et al. [22], who motivated it in the context of community detection. In this setting we are given two kinds of data for a group of entities (for example humans, organizations, or events) called attribute and relationship data. The attribute data corresponds to the intrinsic properties of the entities and can be used to define the similarity of two entities. Ge et al. modeled this using the distance metric. The relationship data corresponds to extrinsic relationships between entities, for example, a common paper of two researchers if we want to identify research communities. When identifying a community it seems reasonable to require that there are no parts of this community that are totally isolated from the rest. As a result, Ge et al. modeled this using the connectivity graph by adding edges between entities if they are related. Given that in many real world scenarios also very dissimilar entities may interact (for example joint work of researchers from different fields) while similarity between entities does not necessarily imply that there exists a relationship between them, it seems reasonable to not assume any clear correlation between the distance metric and the connectivity graph. Other researchers have also observed that in practice incorporating both attribute and relationship data yields better results than purely attribute-based or relationship-based approaches [42, 27], and some explored connectivity constraints in this context [5, 4]. Gupta et al. [24] and Eube et al. [19] studied connected k-median motivated by this interpretation.

Another natural application of connected clustering is the clustering of geographic regions or objects. We are given information (e.g. regarding the population or climate) for a set of regions and want to cluster similar regions together while ensuring that the resulting clusters actually correspond to continuous areas. Connected k-median and related problems have been utilized to determine political districts [39, 41], aggregate areas in a scaled-down map [25, 36], and cluster sensor data [34]. Many researchers also studied the p-regions problem, which is a connected clustering variant where one aims at optimizing the sum of all pairwise distances between vertices in the same cluster [18, 17, 14, 21, 33, 43]. To the authors' knowledge all results for this problem are either based on integer linear programs or heuristic approaches. Most recently, Drexler et al. studied connected k-center in the context of clustering tide gauge stations [16]. Here one is given a sequence of measurements of the sea levels of different stations and wants to group stations with similar measurements together such that the groups are again reasonably compact.

Unfortunately it turned out that most clustering objectives get very hard to approximate once a connectivity constraint is introduced. Ge et al. already showed that even for $k = 2$ connected k-center cannot be solved optimally anymore [22]. While they claimed to have a constant approximation algorithm for connected k-center [22], Drexler et al. later pointed out that this is not the case [16]. The latter provided an $O(\log(k)^2)$-approximation algorithm, which is the best known guarantee to this day. They also provided constant approximations if the metric is an L_p metric of constant dimension or has a constant doubling dimension[1] but it remained an open question whether or not there exists a constant approximation in the general case. As we show in this work this is not the case, unless $\mathsf{P} = \mathsf{NP}$, as connected k-center is $\Omega(\log^*(k))$-hard to approximate[2].

The connected k-median problem turned out to be even harder, as Eube et al. [19] showed that even for $k = 2$ connected k-median is $\Omega(n^{1-\epsilon})$-hard to approximate. The same reduction also works for connected k-means. They relaxed the problem by allowing vertices to be part of multiple clusters. In this setting they were able to provide an $O(k^2 \log(n))$-approximation but also showed by slightly adapting a previous reduction by Gupta et al. [24] that the problem remains $\Omega(\log(n))$-hard to approximate. Already before this, it was observed that one can slightly adapt the classic k-center algorithm by Hochbaum and Shmoys to obtain a 2-approximation for connected k-center if the clusters may overlap [22]. However, while this might be a reasonable relaxation for some settings, the possibility to assign an entity to multiple groups would appear unnatural in many situations. Especially if we cluster geographic data, having overlapping areas may not be reasonable in many contexts. Previous research mostly focuses on the disjoint variant. In the remainder of this work we will focus exclusively on the disjoint setting.

While both connected k-center and k-median are very hard to approximate on general graphs, there exist dynamic programs for both problems that calculate the optimal solution in polynomial time if the graph is a tree [22, 19]. In this work, we extend these results for a broader class of "treelike" graphs. This is based on the concept of treewidth and tree decompositions defined by Robertson and Seymour [37], whose definition we present later. For many NP-hard problems it has been shown that they can be solved in polynomial time if the treewidth w of the input graph is constant (see for example Chapter 7 of [12]). By combining the dynamic programs for connected k-center and k-median with the ideas of an algorithm for Steiner trees on graphs with constant treewidth [8], we can also solve these problems exactly for constant treewidths. One may observe that especially in the setting that we want to cluster tide gauge stations as presented by Drexler et al., it is not unreasonable to assume that the graph is treelike. This is due to the fact that these stations are usually placed at the coast which means that for large parts there are sequences of stations just following the coastline. While islands close to the mainland and almost enclosed bodies of water like the Mediterranean Sea can lead to cycles, the number of parallel cycles will not be too large. Also in other settings we might end up with a reasonably small treewidth.

[1] The approximation factor depends on the dimension in both cases.

[2] $\log^*(k)$ denotes how often we have to iteratively apply the logarithmic function to k until we end up with a number that is at most 1.

However, the running times of the resulting algorithms lie in $\Omega(n^w)$. Using ideas similar to the nesting property for hierarchical clustering [35], we are able to obtain constant approximation algorithms, with a fixed parameter tractable (FPT) running time with respect to the parameter $\max(w, k)$ (i.e. the running time can be bounded by the product of a polynomial function in n and another function that only depends on $\max(w, k)$).

Lastly, we introduce the connectivity constraint to the MSR and MSD problem, which to the authors' knowledge has not been done before[3]. While it is NP-hard to obtain constant approximations for the other clustering objectives under the connectivity constraint, we show that the approximation algorithm for MSR by Buchem et al. [6] can be modified to also ensure connectivity. This results in a polynomial time $(3+\epsilon)$-approximation algorithm for connected MSR. Additionally, we noticed that Buchem et al. only mentioned that their algorithm calculates a $(6+\epsilon)$-approximation for MSD while one can also prove a $(4+\epsilon)$-approximation guarantee. We provide the corresponding proof for the connected setting. Since the algorithm also works in the non-connected setting (which is the same as using the complete connectivity graph) this also improves the best known approximation guarantee for unconstrained MSD from $(6+\epsilon)$ to $(4+\epsilon)$.

Also, while unconstrained MSD is NP-hard to approximate with an approximation factor smaller than 2 [15] no lower bounds for the approximability of unconstrained MSR are known. We show that connected MSR is NP-hard to α-approximate for any $\alpha < \frac{4}{3}$.

1.1 Definitions

Let us first formally define the clustering problems considered in this work:

▶ **Definition 1.** *In an instance of a connected clustering problem, we are given a set V with $n = |V|$ vertices, a distance metric $d : V \times V \to \mathbb{R}_{\geq 0}$ on V, a positive integer $k \geq 2$, and an undirected connectivity graph $G = (V, E)$. A feasible solution is a partition of V into k disjoint clusters $P_1, \dots, P_k$ with corresponding centers $c_1, c_2, \dots, c_k \in V$ which satisfies that for every $i \in [k] := \{1, \dots, k\}$ the subgraph of G induced by P_i is connected and $c_i \in P_i$. The aim is to minimize one of the following objective functions:*

k-center: $\max_{i\in[k]} \max_{v\in P_i} d(v, c_i)$

k-median: $\sum_{i\in[k]} \sum_{v\in P_i} d(v, c_i)$

k-means: $\sum_{i\in[k]} \sum_{v\in P_i} d(v, c_i)^2$

min-sum-radii: $\sum_{i\in[k]} \max_{v\in P_i} d(v, c_i)$

min-sum-diameter: $\sum_{i\in[k]} \max_{v,w\in P_i} d(v, w)$

One might note that for MSD the centers are not necessary.

To extend the dynamic programs for connected clustering on tree connectivity graphs we define the concept of *tree decompositions* introduced by Robertson and Seymour [37]:

▶ **Definition 2** (Tree Decomposition). *A* tree decomposition *of a graph $G = (V, E)$ consists of a tree $T = (V_T, E_T)$ and a corresponding bag $B_t \subseteq V$ for every $t \in V_T$ such that $\bigcup_{t\in V_T} B_t = V$ and:*

- *For every $\{u, v\} \in E$, there exists at least one node $t \in V_T$ with $u, v \in B_t$.*
- *For every $v \in V$, the set $S_v = \{t \mid v \in B_t\}$ of nodes whose bags contain v forms a connected subtree of T.*

[3] There exists a paper by An and Kao [3] that deals with the "Connected Minimum Sum of Radii Problem". However, their problem is not related to the connectivity constraint.

The treewidth *of a tree decomposition* $(T, \{B_t\}_{t \in V_T})$ *is the size of the largest bag minus one. The treewidth of a graph is the minimum treewidth among all possible tree decompositions of* G*.*

Kloks later introduced the concept of *nice tree decompositions*, which fulfill structural properties that are often desirable [29]. In this work we will use an extended version of this concept by Cygan et al. that also explicitly deals with the introduction of edges in the decomposition [13]:

▶ **Definition 3** (Nice Tree Decomposition). *A nice tree decomposition is a tree decomposition with a special node* z *called the root with* $B_z = \emptyset$ *and in which each node* $t \in V_T$ *is one of the following types:*

- ***Leaf node:*** t *is a leaf of* T *and* $B_t = \emptyset$
- ***Introduce vertex node:*** t *has precisely one child* s *such that* $B_t = B_s \cup \{v\}$ *for a vertex* $v \in V \setminus B_s$*. We say that* t *introduces* v*.*
- ***Introduce edge node:*** t *is labeled with an edge* $\{u, v\} \in E$ *and has a single child* s *such that* $B_t = B_s$ *and* $u, v \in B_t$*. We say that* t *introduces* $\{u, v\}$ *and every edge is introduced precisely once in our tree decomposition.*
- ***Forget node:*** t *has precisely one child* s *such that* $B_t = B_s \setminus \{v\}$ *for a vertex* $v \in B_s$*. We say that* t *forgets* v*. Every vertex* v *is forgotten exactly once.*
- ***Join node:*** t *has precisely two children* s_1 *and* s_2 *with* $B_{s_1} = B_{s_2} = B_t$*.*

It has been shown that if a graph admits a tree decomposition of width w then it also admits a nice tree decomposition of this width. Furthermore, one can always transform a tree decomposition into a nice one of bounded size:

▶ **Lemma 4** (Lemma 7.4 in [12]). *Given a tree decomposition* $(T, \{B_t\}_{t \in V_T})$ *of a graph* $G = (V, E)$ *with width* w*, one can compute a nice tree decomposition of* G *of width* w *and with* $O(w|V|)$ *nodes in time* $O(\max(|V_T|, |V|)w^2)$*.*

Together with the fact that Korhonen and Lokshtanov proved that one can calculate a tree decomposition of the same treewidth w as the graph G in $O(2^{O(w^2)}|V|^4)$ [31], this enables us to calculate a nice tree decomposition of a graph G with minimum width in FPT time. Alternatively, there also exist approximation algorithms for tree decompositions [30].

1.2 Outline and Our Results

Table 1 Overview over the connected clustering results given a tree decomposition of width w. The algorithms for connected k-median and connected k-means are based on an unconstrained k-median algorithm with running time $O(p_1(n))$ and approximation factor α_1 and an unconstrained k-means algorithm with running time $O(p_2(n))$ and approximation factor α_2, respectively.

Objective	Running Time	Approximation factor	Theorem
k-center	$n^{O(w)}$	1	11
k-center	$n^2 \log(n) + n \log(n)(wk)^{O(w)}$	6	13
k-median	$p_1(n) + n \cdot (wk)^{O(w)}$	$2\alpha_1 + 4$	14
k-means	$p_2(n) + n \cdot (wk)^{O(w)}$	$8\alpha_2 + 32$	15

In Section 2 we give an overview over the techniques we used to obtain our results. The complete presentation of the results can be found in the full version of this work.

In Section 2.1 we focus on our results for graphs with bounded treewidth which are summarized in Table 1. In Section 2.1.1 the exact algorithm for connected k-center with running time $n^{O(w)}$ is presented. In Section 2.1.2 we modify this algorithm to obtain a constant approximation while achieving an FPT running time. By similar methods as for k-center, one can also obtain exact algorithms for connected k-median and k-means with running time $n^{O(w)}$. In Section 2.1.3, we present how one can apply the same techniques as for k-center to compute approximations for connected k-median and k-means in FPT time.

In Section 2.2 we complement these results with a family of instances proving that in the general case one cannot hope for a constant approximation for connected k-center unless $\mathsf{P} = \mathsf{NP}$:

▶ **Theorem 5.** *It is* NP*-hard to approximate connected* k*-center with an approximation ratio in* $o(\log^*(k))$*. The same is also true if the centers are given in advance.*

In Section 2.3 we argue that unlike connected k-median and k-center, connected min-sum-radii actually admits a constant approximation:

▶ **Theorem 6.** *For any* $\epsilon > 0$ *one can calculate a* $(3 + \epsilon)$*-approximation for the connected min-sum-radii problem in polynomial time.*

Additionally, we show that the algorithm also calculates a $(4 + \epsilon)$-approximation for min-sum-diameter which also improves the previously known $(6+\epsilon)$-approximation guarantee in the unconstrained case and add a lower bound for the approximability of connected MSR:

▶ **Theorem 7.** *For any* $\epsilon > 0$ *one can calculate a* $(4+\epsilon)$*-approximation for both the connected as well as the regular min-sum-diameter problem in polynomial time.*

▶ **Theorem 8.** *It is* NP*-hard to approximate connected min-sum-radii with an approximation factor smaller than* $\frac{4}{3}$*. This is even true if there are only two clusters.*

2 Our Techniques

2.1 Results for Graphs with Bounded Treewidth

2.1.1 An Exact Algorithm for Connected k-Center

Let a graph $G = (V, E)$ with $n = |V|$ vertices, a natural number k and a nice tree decomposition $(T = (V_T, E_T), \{B_t\}_{t \in V_T})$ of G with treewidth w and size $O(wn)$ be given. For any node $t \in V_T$ we define T_t to be the nodes contained in the subtree rooted at t and the graph $G_t = (V_t, E_t)$, where $V_t = \bigcup_{u \in T_t} B_u$ and $E_t = \big\{\{v, u\} \in E \mid \{v, u\} \text{ has been introduced in } T_t\big\}$.

As pointed out by Hochbaum and Shmoys [26], every possible objective value of the k-center objective corresponds to the distance between a pair of vertices, of which there are only $O(n^2)$ many. This remains true for the connected k-center problem. Thus, we can guess the optimal radius r^* by sorting these distances and then applying binary search. For a guessed radius r, our goal is then to use dynamic programming to decide whether a connected k-center solution with radius r exists. This enables us to find the optimal radius r^* and later obtain the corresponding solution via backtracking. To apply dynamic programming, we need to be able to calculate, for a given node t, the number of clusters necessary to partition the vertices in the subgraph G_t independently of the remainder of the graph. We can use the property of the tree decomposition that the vertices in B_t form a vertex separator between $V \setminus V_t$ and $V_t \setminus B_t$. Thus, all clusters that contain vertices in V_t and $V \setminus V_t$ must be connected via at least one vertex in B_t.

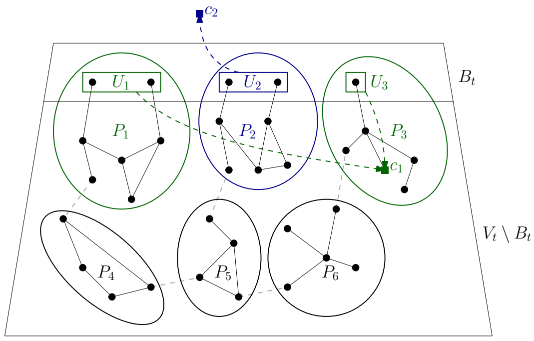

Figure 1 A solution for the graph G_t fulfilling the specification $(a, \{U_1, U_2, U_3\})$ where $a(U_1) = a(U_3) = c_1$ and $a(U_2) = c_2$ with $c_2 \notin V_t$. The solution requires 3 clusters, finished ones are circled in black, the unfinished ones in the color corresponding to their assignment.

Following the same idea as in the dynamic program for tree connectivity graphs [22], we can fix the centers to which the vertices in B_t get assigned to. Then we know exactly to which centers in $V \setminus V_t$ we could possibly assign the vertices in V_t and vice versa. However, this is not sufficient to make the partial solutions for the separated subgraphs entirely independent. If multiple vertices in B_t get assigned to the same center we know that they must be connected in the final solution. The connecting path could either pass through G_t or through the remainder of the graph G. Thus, we may or may not need to assign the vertices in V_t in such a way that the respective vertices in B_t are connected which could separate other clusters and increase the number of clusters necessary to cover G_t.

To circumvent this problem, we do not only fix the assignments of the vertices in B_t, but also provide a partition specifying which vertices must be connected in G_t. A similar approach was already used to solve the Steiner tree problem on graphs with bounded treewidth [8]. For obvious reasons, we only want to consider partitions of B_t that cluster vertices together that are assigned to the same center. More generally:

▶ **Definition 9.** *Let $S \subseteq V$ be a set of vertices and $a : S \to V$ be an assignment function on S. We say a partition $\mathcal{U}$ of S is* compatible *with a if for any $U \in \mathcal{U}$ and any vertices $v, w \in U$ it holds that $a(v) = a(w)$. With a slight abuse of notation we say for any set $U \in \mathcal{U}$ that $a(U) = a(v)$, where v is an arbitrary vertex in U. We call the pair $(a, \mathcal{U})$ of an assignment function and a compatible partition $\mathcal{U}$ a* specification *for S.*

Now we would like to calculate, for a given t, and any specification $(a, \mathcal{U})$ for B_t the minimum number of clusters necessary to cover G_t if we assign and connect the vertices in B_t accordingly. For technical reasons, we will only count the clusters that do not contain any vertices in B_t. We call these clusters *finished*. For every $U \in \mathcal{U}$ we have to make sure that all the vertices in U are contained in the same cluster already since we require them to be connected in G_t. All clusters need to be connected and every vertex should be within radius r of its center. While the finished clusters are required to contain their center the unfinished clusters might be connected to it in the upper layers of the tree. If there exists a vertex $c \in V_t$ and a vertex $v \in B_t$ with $a(v) = c$, c must be contained in an unfinished cluster that is assigned to c (possibly but not necessarily the same cluster that also contains v), since otherwise it would not be possible to connect v to c:

▶ **Definition 10.** *Let $t \in V_T$ and $(a, \mathcal{U})$ with $\mathcal{U} = \{U_1, \ldots, U_l\}$ be a specification for B_t. We say that a partition $P_1, \ldots, P_l, P_{l+1}, \ldots, P_{l+k'}$ of V_t is a connected k-center solution for G_t fulfilling $(a, \mathcal{U})$, if:*

1. *For all $i \in [l + k']$ the cluster P_i induces a connected subgraph of G_t.*
2. *For all $i \in [l]$ it holds that $U_i \subseteq P_i$ and for all $v \in P_i$ we have $d(v, a(U_i)) \leq r$.*
3. *For all $c \in V_t$ for which there exists a vertex $v \in B_t$ with $a(v) = c$, there must exist an index $i \in [l]$ such that $a(U_i) = c$ and $c \in P_i$.*
4. *For all $i \in \{l + 1, \dots, l + k'\}$ it holds that there exists a center $c \in P_i$ such that for all $v \in P_i$ $d(v, c) \leq r$.*

We say that such a solution requires k' clusters and define $p_t(a,\mathcal{U})$ to be the minimum value $k' \in \mathbb{N}_0$ such that there exists a solution for G_t fulfilling $(a,\mathcal{U})$ that only requires k' clusters. If such a k' does not exist, we set $p_t(a,\mathcal{U}) = \infty$.

Figure 1 depicts how a solution for a node fulfilling a specification can look like.

Starting with the leaves, our algorithm iterates over all nodes $t \in V_T$ ordered by their height in T, calculates the value $p_t(a,\mathcal{U})$ for any specification $(a,\mathcal{U})$ of B_t and stores it in a table. We show in the complete version of this work one can calculate each of these values with a running time bounded by a function depending only on w, if we already filled the table for the children of t. Note that for a fixed treewidth w the number of specifications for the bag B_t is polynomially bounded in n. Consequently, our algorithm runs in polynomial time if the treewidth is a constant. Once we reach the root z we have that $G_z = G$ and $B_z = \emptyset$. Thus, for the unique function $a_\emptyset : \emptyset \to V$ the value $p_z(a_\emptyset, \emptyset)$ is exactly equal to the minimum number of centers necessary to cover G with radius r and we can check if this number is at most k. We end up with the following theorem:

▶ **Theorem 11.** *Given a nice tree decomposition $(T, \{B_t\}_{t \in V_T})$ of a graph G with treewidth w and $O(nw)$ nodes and a natural number k, one can calculate an optimal connected k-center solution on G in time $O\left(\log(n) \cdot n^{w+2}(b_{w+1})^2 p(w)\right)$, where p is a polynomial function and b_{w+1} is the $(w + 1)$-th. Bell-number.*

2.1.2 Obtaining a Constant Approximation in FPT Time

One might note that the running time of the previous algorithm is not FPT. This is due to the fact that for a node t there exist up to $n^{|B_t|}$ different assignment functions from B_t to V, since for any vertex $v \in B_t$ there exist n possible assignment choices. Thus, there can be $\Omega(n^{w+1})$ specifications for a node. We can reduce this factor by limiting the number of possible center choices in advance. We do this by introducing the connected k-center with facilities problem:

▶ **Definition 12.** *In the* connected k-center with facilities *problem we are given a graph $G = (V, E)$, a set of facilities $F \subseteq V$, a constant k and a distance metric $d : V^2 \to \mathbb{R}_{\geq 0}$. The goal is to find a tuple $(c_1, \dots, c_k) \in F^k$ of k facilities (where one facility might appear multiple times) and clusters $P_1, \dots, P_k \subseteq V$ such that:*
- *$P_1, \dots, P_k$ is a partition of V.*
- *For all $i \in [k]$ the vertices in P_i form a connected subgraph of G.*
- *$\max_{i \in [k]} \max_{v \in P_i} d(v, c_i)$ gets minimized.*

One may note that this definition does not require the clusters to contain the respective centers, which will enable us to choose a facility set independently of the connectivity graph.

One can modify the dynamic program for connected k-center to also calculate the optimal connected k-center with facilities solution, while reducing the factor n^{w+1} to $|F|^{w+1}$. If the facility set F does not contain the optimal centers for connected k-center, this will obviously not correspond to the optimal connected k-center solution. However, if we choose a good unconnected k-center solution on the vertices V as F (which can be obtained using a

2-approximation [26]), then there exists a facility close to every optimal center, which means that the connected k-center with facilities solution is not too much more expensive than the regular connected k-center solution. Since the clusters of the connected k-center with facilities solution may not contain the respective facilities, we then have to find a feasible center for each cluster to transform the resulting solution into a regular connected k-center solution. To do this, one can just take the vertex closest to the center, which only increases the radius by at most 2. In total we end up with a 6-approximation while reducing the number of possible assignment functions to $|F|^{w+1} = k^{w+1}$ which results in an FPT running time w.r.t. the parameter $\max(k, w)$:

▶ **Theorem 13.** *Given a nice tree decomposition of a graph G with treewidth w and $O(nw)$ bag nodes and a natural number k, one can calculate a 6-approximation of connected k-center in time $O\left(n^2 \log(n) + n \log(n)(wk)^{O(w)}\right)$, which lies in FPT with respect to the parameter $\max(w, k)$.*

2.1.3 Results for Connected k-Median and k-Means

The dynamic program can also be adapted to the k-median and k-means objective. The main difference is that we cannot guess the optimal radius in advance. Inspired by an approach of Eube et al. [19], we extend the specification for the bag B_t of a node $t \in V_T$ by an additional parameter $k' \leq k$ that specifies how many centers may be placed in the subgraph. This allows us to compute, for every node t and every specification for B_t, the cost of the optimal clustering for G_t fulfilling this specification.

Also for these two objectives, we can use a nesting argument to show that if we choose the centers of a good unconnected k-clustering solution as facilities, the cost of the optimal solution only increases by a constant factor if we restrict ourselves to these facilities. By calculating the optimal connected clustering with facilities solution and choosing appropriate centers for each cluster afterwards, we can calculate constant approximations for both objectives in FPT time:

▶ **Theorem 14.** *Given a nice tree decomposition of a graph G with treewidth w and $O(nw)$ nodes and a natural number k, as well as an α-approximation algorithm for unconnected k-median with running time $p(n)$, one can calculate a $(2\alpha + 4)$-approximation for connected k-median in $O\left(p(n) + n \cdot (wk)^{O(w)}\right)$ time.*

▶ **Theorem 15.** *Given a nice tree decomposition of a graph G with treewidth w and $O(nw)$ bag nodes and a natural number k, as well as an α-approximation algorithm for unconnected k-means with running time $p(n)$, one can calculate an $(8\alpha + 32)$-approximation for connected k-means in $O\left(p(n) + n \cdot (wk)^{O(w)}\right)$ time.*

2.2 Hardness of Approximation for Connected k-Center

To prove that the connected k-center problem is $\Omega(\log^*(k))$-hard to approximate we consider the assignment variant of the problem. Here we are given the centers in advance and only have to decide to which center the vertices get assigned (i.e. determine the corresponding clusters). Intuitively, one would expect this to make the problem easier rather than harder. Indeed, we can prove that any α-approximation algorithm for the regular variant can be transformed into a 2α-approximation algorithm for the assignment variant, meaning that the hardness also carries over if the centers are not given. The hardness is based on a reduction from the 3-SAT problem, in which we have to decide if a given CNF formula ϕ can be satisfied. There already exist reductions that show that even with two centers the assignment problem is NP-hard [22, 16] to approximate with an approximation factor smaller than 3. We use the respective construction as a gadget in our reduction.

Given that our construction is quite technical, we only present a very broad idea of its structure in this section. The complete proof can be found in the full version of this work and some of the terms we use in this section (gadget, input, output) are not used in the actual description of the reduction. One might also note that for the asymmetric k-center problem there exists an $o(\log^*(n))$-hardness result with a similar bound [9]. However, there are no similarities between the respective hardness constructions. Notably, the asymmetric k-center problem also becomes polynomially solvable if the centers are given.

To get stronger inapproximability results we use a layer approach. In total we have L layers and our vertices are placed in an L-dimensional space, meaning that their positions have L coordinates where each coordinate basically corresponds to one of the layers. Using a metric similar to the Manhattan distance, we can enforce that the distance between two vertices is equal to the number of unequal coordinates of their positions[4]. For every position appearing in our instance, there exists a center that is also placed at this position. However, not every vertex placed at that position is directly connected to the center. We construct the instance in such a way that every vertex can be assigned to a center differing in at most one coordinate if ϕ can be satisfied while otherwise at least one vertex needs to be assigned to a center that differs in every coordinate, resulting in a gap of L between the radius of these two cases.

Each layer is only connected with the previous and the next layer and the centers are only connected with the first, i.e. the outermost layer. Thus, paths connecting the vertices in the L-th layer to their respective center have to pass through every layer. For a vertex v we say that an assignment does a *mistake* in the i-th layer if the assignments in this layer separate v from every center with the same i-th coordinate as v, meaning that v also needs to be assigned to a center with a different i-th coordinate.

The L-layer instance is defined iteratively and contains multiple copies of instances with $L-1$ layers. In these copies, we replace the centers by regular vertices which we call the *inputs* of the subinstance. They are the only vertices adjacent to vertices from the first layer of the L-layer instance. We also add an additional coordinate at the beginning of the positions of the vertices, which is the same for every vertex in this subinstance. As a result we can also say that we make a mistake for this entire subinstance if we assign all inputs to centers with a different first coordinate. A 0-layer instance simply consists of an isolated vertex.

To construct the first layer, we use gadgets based on the reduction by Ge et al. [22]. These gadgets simulate Boolean decisions by assigning a vertex to one of two centers, which enables connections to one center while potentially blocking connections to the other. Following this idea, we can create gadgets modeling the formula ϕ that use two centers whose positions only differ in the first coordinate and contains a special set of vertices that we call *outputs*. Each output is supposed to be assigned to one specific center (among the two contained in the gadget). If ϕ can be satisfied, we can assign all outputs accordingly, but if ϕ cannot be satisfied, at least one output needs to be assigned incorrectly. We create such gadgets for every possible choice of the last $L-1$ coordinates and connect their outputs with the inputs of the instances with $L-1$ layers (where the same output can be used multiple times). The different inputs of the same subinstance are each connected with an output from

[4] This is a slight simplification, actually their distance is not equal but proportional to this value.

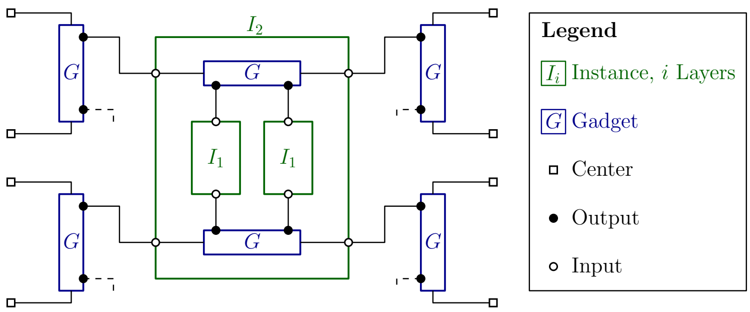

Figure 2 A sketch of a 2-layer subinstance included in a 3-layer instance. Note that the number of in- and outputs is not accurate and there are further gadgets, centers and subinstances in the 3-layer instance.

a different gadget and the connections between the in- and outputs have to fulfill certain properties which are not presented in this section. Figure 2 sketches the overall structure of this construction.

By adding sufficiently many $(L-1)$-layer instances, one can ensure that if ϕ cannot be satisfied (meaning each gadget has an incorrect output), then there also exists at least one of these subinstances which is only connected to incorrect outputs resulting in a mistake for this subinstance. By an inductive argument we can then show that in this $(L-1)$-layer subinstance we have to again make a mistake for one of the contained $(L-2)$-layer subinstances and so on. This guarantees a vertex for which we make a mistake in every layer resulting in a radius of L. However, if ϕ can be satisfied, it is possible to assign the vertices without any mistakes resulting in a radius of 1.

To make this construction work, we have to cover many different possibilities which outputs are assigned incorrectly by the gadgets. This requires many centers, gadgets, and copies of $(L-1)$-layer subinstances in an L-layer instance and the number of centers increases exponentially when we add an additional layer. As a result, the number of layers is bounded by $O(\log^*(k))$ which then also limits our hardness result accordingly.

2.3 Approximation Algorithms for Connected MSR and MSD

Our approximation for connected MSR is based on the $(3+\epsilon)$-approximation algorithm for regular MSR by Buchem et al. [6]. Their algorithm first guesses the $(1/\epsilon)$ largest clusters and then calculates the remaining ones using a primal dual approach. The variables of the primal LP (and thus the constraints of the dual LP) correspond to balls $B(c,r)$ that contain all vertices around a center c with radius r. These variables indicate whether or not the respective ball is used as a cluster in the LP solution. To ensure connectivity we have to modify these balls to instead contain all vertices which are reachable from the center via a path of vertices at distance r from it. It is easy to verify that the resulting balls are indeed connected.

Since the original algorithm already merges overlapping balls together, most of the analysis by Buchem et al. can be adapted to incorporate the connectivity constraint. However, one needs to be careful with the $(1/\epsilon)$ largest clusters that have been guessed in the beginning. In the regular setting, we can just assign all vertices within a certain radius around an optimal center (similarly to the balls) to the center and know that it does not affect the other clusters negatively, if they cannot use the respective vertices anymore. However, in the connected setting, this might split up some of the other optimal clusters. To circumvent this, we allow the balls in the LP to also contain vertices that are already contained in these guessed clusters. In the end, we merge them together with the guessed clusters if they overlap. By being careful in the analysis, one can show that this does not increase the approximation factor in the worst case. As a result, we end up with a $(3+\epsilon)$-approximation algorithm for connected MSR.

To improve the trivial $(6+\epsilon)$-guarantee that this algorithm gives for the MSD problem (both in the connected and unconnected case), one needs to observe that the cost of the optimal MSD solution is lower-bounded by the optimal MSR solution. As a result we can compare against the optimal MSR solution to bound the approximation factor for MSD. In their analysis, Buchem et al. were able to prove via an elaborate argument that the radius of a cluster, resulting from merging a specific set of overlapping balls, can be bounded by three times the contribution of these balls to the cost of the dual LP solution. Using an actually far less involved argument, one can show that the diameter of the resulting cluster is bounded by 4 times the contribution of these balls to the cost of the dual solution. By inserting this bound into the remainder of the proof, we obtain that the MSR algorithm also calculates a $(4+\epsilon)$-approximation for the optimal MSD solution.

To show that it is NP-hard to approximate connected MSR with an approximation factor smaller than $4/3$ (even if $k=2$), we use a reduction from 3-SAT similar to the gadgets in our hardness proof for connected k-center. We need to be slightly more careful when constructing the connected MSR instance, because in MSR one can replace two clusters by a single cluster with the combined radius without increasing the cost.

References

1 Gagan Aggarwal, Rina Panigrahy, Tomás Feder, Dilys Thomas, Krishnaram Kenthapadi, Samir Khuller, and An Zhu. Achieving anonymity via clustering. *ACM Transactions on Algorithms (TALG)*, 6(3):1–19, 2010. doi:10.1145/1798596.1798602.

2 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 61–72, 2017. doi:10.1109/FOCS.2017.15.

3 Hyung-Chan An and Mong-Jen Kao. On the connected minimum sum of radii problem. In *Proceedings of the 35th International Symposium on Algorithms and Computation (ISAAC)*, pages 7:1–7:13, 2024. doi:10.4230/LIPIcs.ISAAC.2024.7.

4 Stefano Benati, Diego Ponce, Justo Puerto, and Antonio M Rodríguez-Chía. A branch-and-price procedure for clustering data that are graph connected. *European Journal of Operational Research*, 297(3):817–830, 2022. doi:10.1016/j.ejor.2021.05.043.

5 Stefano Benati, Justo Puerto, and Antonio M Rodriguez-Chia. Clustering data that are graph connected. *European Journal of Operational Research*, 261(1):43–53, 2017. doi:10.1016/j.ejor.2017.02.009.

6 Moritz Buchem, Katja Ettmayr, Hugo K.K. Rosado, and Andreas Wiese. A $(3+\epsilon)$-approximation algorithm for the minimum sum of radii problem with outliers and extensions for generalized lower bounds. In *Proceedings of the 35th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1738–1765, 2024. doi:10.1137/1.9781611977912.69.

7 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 5036–5044, 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/978fce5bcc4eccc88ad48ce3914124a2-Paper.pdf.

8 Markus Chimani, Petra Mutzel, and Bernd Zey. Improved Steiner tree algorithms for bounded treewidth. *Journal of Discrete Algorithms*, 16:67–78, 2012. doi:10.1016/j.jda.2012.04.016.

9 Julia Chuzhoy, Sudipto Guha, Eran Halperin, Sanjeev Khanna, Guy Kortsarz, Robert Krauthgamer, and Joseph Naor. Asymmetric k-center is log* n-hard to approximate. *Journal of the ACM (JACM)*, 52(4):538–551, 2005. doi:10.1145/1082036.1082038.

10 Vincent Cohen-Addad, Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Improved approximations for euclidean k-means and k-median, via nested quasi-independent sets. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1621–1628, 2022. doi:10.1145/3519935.3520011.

11 Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, Chris Schwiegelshohn, and Ola Svensson. A $(2+\varepsilon)$-approximation algorithm for metric k-median. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 615–624, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3717823.3718299.

12 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

13 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Joham MM van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 150–159, 2011. doi:10.1145/3506707.

14 Wooseok Do, Hyeongjun Park, Koohong Chung, and Dongjoo Park. An effects analysis of logistics collaboration: The case of pharmaceutical supplies in seoul. *Sustainability*, 11(8), 2019. doi:10.3390/su11082442.

15 Srinivas R. Doddi, Madhav V. Marathe, Sekharipuram S. Ravi, David Scot Taylor, and Peter Widmayer. Approximation algorithms for clustering to minimize the sum of diameters. In *Proceedings of the 7th Scandinavian Workshop on Algorithm Theory(SWAT)*, pages 237–250, 2000. doi:10.1007/3-540-44985-X_22.

16 Lukas Drexler, Jan Eube, Kelin Luo, Dorian Reineccius, Heiko Röglin, Melanie Schmidt, and Julian Wargalla. Connected k-center and k-diameter clustering. *Algorithmica*, 86(11):3425–3464, 2024. doi:10.1007/s00453-024-01266-9.

17 Juan C Duque, Luc Anselin, and Sergio J Rey. The max-p-regions problem. *Journal of Regional Science*, 52(3):397–419, 2012. doi:10.1111/j.1467-9787.2011.00743.x.

18 Juan C Duque, Richard L Church, and Richard S Middleton. The p-regions problem. *Geographical Analysis*, 43(1):104–126, 2011. doi:10.1111/j.1538-4632.2010.00810.x.

19 Jan Eube, Kelin Luo, Dorian Reineccius, Heiko Röglin, and Melanie Schmidt. Connected k-median with disjoint and non-disjoint clusters. In *Proceedings of the 33rd Annual European Symposium on Algorithms (ESA)*, pages 63:1–63:14, 2025. doi:10.4230/LIPIcs.ESA.2025.63.

20 Jan Eube and Heiko Röglin. New algorithms and hardness results for connected clustering, 2025. doi:10.48550/arXiv.2511.19085.

21 Christian Etienne Fleischer. Using the max-p regions problem algorithm to define regions for energy system modelling. *MethodsX*, 8:101211, 2021. doi:10.1016/j.mex.2021.101211.

22 Rong Ge, Martin Ester, Byron J. Gao, Zengjian Hu, Binay K. Bhattacharya, and Boaz Ben-Moshe. Joint cluster analysis of attribute data and relationship data: The connected k-center problem, algorithms and applications. *ACM Trans. Knowl. Discov. Data*, 2(2):7:1–7:35, 2008. doi:10.1145/1376815.1376816.

23 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.

24 Neelima Gupta, Aditya Pancholi, and Yogish Sabharwal. Clustering with internal connectedness. In *Proceedings of the 5th Intl. Workshop on Algorithms and Computation (WALCOM)*, volume 6552 of *Lecture Notes in Computer Science*, pages 158–169, 2011. doi:10.1007/978-3-642-19094-0_17.

25 Jan-Henrik Haunert and Alexander Wolff. Area aggregation in map generalisation by mixed-integer programming. *International Journal of Geographical Information Science*, 24(12):1871–1897, 2010. doi:10.1080/13658810903401008.

26 Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985. doi:10.1287/moor.10.2.180.

27 Darko Hric, Richard K Darst, and Santo Fortunato. Community detection in networks: Structural communities versus ground truth. *Physical Review E*, 90(6):062805, 2014. doi:10.1103/PhysRevE.90.062805.

28 Wen-Lian Hsu and George L Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979. doi:10.1016/0166-218X(79)90044-1.

29 Ton Kloks. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994. doi:10.1007/BFB0045375.

30 Tuukka Korhonen. A single-exponential time 2-approximation algorithm for treewidth. In *Proceedings of the 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 184–192, 2022. doi:10.1137/22M147551X.

31 Tuukka Korhonen and Daniel Lokshtanov. An improved parameterized algorithm for treewidth. In *Proceedings of the 55th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 528–541, 2023. doi:10.1145/3564246.3585245.

32 Shi Li. Approximating capacitated k-median with $(1+\epsilon)$ k open facilities. In *Proceedings of the 2016 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 786–796, 2016. doi:10.1137/1.9781611974331.ch56.

33 Wenwen Li, Richard L. Church, and Michael F. Goodchild. An extendable heuristic framework to solve the p-compact-regions problem for urban economic modeling. *Comput. Environ. Urban Syst.*, 43:1–13, 2014. doi:10.1016/j.compenvurbsys.2013.10.002.

34 Zhung-Xun Liao and Wen-Chih Peng. Clustering spatial data with a geographic constraint: exploring local search. *Knowl. Inf. Syst.*, 31(1):153–170, 2012. doi:10.1007/s10115-011-0402-8.

35 Guolong Lin, Chandrashekhar Nagarajan, Rajmohan Rajaraman, and David P. Williamson. A general approach for incremental approximation and hierarchical clustering. *SIAM Journal on Computing*, 39(8):3633–3669, 2010. doi:10.1137/070698257.

36 Johannes Oehrlein and Jan-Henrik Haunert. A cutting-plane method for contiguity-constrained spatial aggregation. *Journal of Spatial Information Science*, 15:89–120, 2017. doi:10.5311/JOSIS.2017.15.379.

37 Neil Robertson and Paul D Seymour. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, 1984. doi:10.1016/0095-8956(84)90013-3.

38 Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Fair coresets and streaming algorithms for fair k-means. In *Proceedings of the 17th International Workshop on Approximation and Online Algorithms (WAOA)*, pages 232–251, 2020. doi:10.1007/978-3-030-39479-0_16.

39 Takeshi Shirabe. Districting modeling with exact contiguity constraints. *Environment and Planning B: Planning and Design*, 36(6):1053–1066, 2009. doi:10.1068/b34104.

40 Ali Vakilian and Mustafa Yalciner. Improved approximation algorithms for individually fair clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 8758–8779. PMLR, 2022. URL: https://proceedings.mlr.press/v151/vakilian22a.html.

41 Hamidreza Validi, Austin Buchanan, and Eugene Lykhovyd. Imposing contiguity constraints in political districting models. *Oper. Res.*, 70(2):867–892, 2022. doi:10.1287/opre.2021.2141.

42 Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *Proceedings of the IEEE 13th international conference on data mining*, pages 1151–1156, 2013. doi:10.1109/ICDM.2013.167.

43 Xinyue Ye, Bing She, and Samuel Benya. Exploring regionalization in the network urban space. *Journal of Geovisualization and Spatial Analysis*, 2(1):4, 2018. doi:10.1007/s41651-018-0013-y.