Coresets for Clustering in Graphs of Bounded Treewidth

Daniel Baker [1] **Vladimir Braverman** [1] **Lingxiao Huang** [2] **Shaofeng H.-C. Jiang** [3] **Robert Krauthgamer** [3] **Xuan Wu** [1]

Abstract

We initiate the study of coresets for clustering in graph metrics, i.e., the shortest-path metric of edge-weighted graphs. Such clustering problems are essential to data analysis and used for example in road networks and data visualization. A coreset is a compact summary of the data that approximately preserves the clustering objective for every possible center set, and it offers significant efficiency improvements in terms of running time, storage, and communication, including in streaming and distributed settings. Our main result is a near-linear time construction of a coreset for k-MEDIAN in a general graph G, with size $O_{\epsilon,k}(\mathrm{tw}(G))$ where $\mathrm{tw}(G)$ is the treewidth of G, and we complement the construction with a nearly-tight size lower bound. The construction is based on the framework of Feldman and Langberg [STOC 2011], and our main technical contribution, as required by this framework, is a uniform bound of $O(\mathrm{tw}(G))$ on the shattering dimension under any point weights. We validate our coreset on real-world road networks, and our scalable algorithm constructs tiny coresets with high accuracy, which translates to a massive speedup of existing approximation algorithms such as local search for graph k-MEDIAN.

1. Introduction

We initiate the study of coresets for clustering in *graph metrics*, i.e., the shortest-path metrics of graphs. As usual in these contexts, the focus is on edge-weighted graphs $G = (V, E)$ with a restricted topology, and in our case bounded treewidth. Previously, coresets were studied extensively but mostly under geometric restrictions, e.g., for Euclidean metrics.

Coresets for k-Clustering We consider the *metric* k-MEDIAN problem, whose input is a metric space $M = (V, d)$ and an n-point data set $X \subseteq V$, and the goal is to find a set $C \subseteq V$ of k points, called *center set*, that minimizes the objective function

$$\mathrm{cost}(X, C) := \sum_{x \in X} d(x, C),$$

where $d(x, C) := \min\{d(x, c) : c \in C\}$. The metric k-MEDIAN generalizes the well-known Euclidean case, in which $V = \mathbb{R}^{\mathsf{d}}$ and $d(x, y) = \|x - y\|_2$. k-MEDIAN problem and related k-clustering problems (like k-MEANS, whose objective is $\sum_{x \in X} (d(x, C))^2$), are essential tools in data analysis and are used in many application domains, such as genetics, information retrieval, and pattern recognition. However, finding an optimal clustering is a nontrivial task, and even in settings where polynomial-time algorithms are known, it is often challenging in practice because data sets are huge, and potentially distributed or arriving over time. To this end, a powerful data-reduction technique, called *coresets*, is of key importance.

Roughly speaking, a coreset is a compact summary of the data points by weighted points, that approximates the clustering objective for every possible choice of the center set. Formally, an ϵ-coreset for k-MEDIAN is a subset $D \subseteq V$ with weight $w : D \to \mathbb{R}_+$, such that for every k-subset $C \subseteq V$,

$$\sum_{x \in D} w(x) \cdot d(x, C) \in (1 \pm \epsilon) \cdot \mathrm{cost}(X, C).$$

This notion, sometimes called a strong coreset, was proposed in (Har-Peled & Mazumdar, 2004), following a weaker notion of (Agarwal et al., 2004). Small-size coresets (where size is defined as $|D|$) often translate to faster algorithms, more efficient storage/communication of data, and streaming/distributed algorithms via the merge-and-reduce framework, see e.g. (Har-Peled & Mazumdar, 2004; Fichtenberger et al., 2013; Balcan et al., 2013; Huang et al., 2018;

Authors are listed in alphabetical order. Full version: (Baker et al., 2019) [1]Johns Hopkins University, USA [2]Yale University, USA [3]Weizmann Institute of Science, Israel. Correspondence to: Daniel Baker <dnb@cs.jhu.edu>, Vladimir Braverman <vova@cs.jhu.edu>, Lingxiao Huang <huanglingxiao1990@126.com>, Shaofeng H.-C. Jiang <shaofeng.jiang@weizmann.ac.il>, Robert Krauthgamer <robert.krauthgamer@weizmann.ac.il>, Xuan Wu <wu3412790@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

Friggstad et al., 2019) and recent surveys (Phillips, 2017; Munteanu & Schwiegelshohn, 2018; Feldman, 2020).

Coresets for k-MEDIAN were studied extensively in Euclidean spaces, i.e., when $V = \mathbb{R}^{\mathsf{d}}$ and $d(x, y) = \|x - y\|_2$. The size of the first ϵ-coreset for k-MEDIAN, when they were first proposed (Har-Peled & Mazumdar, 2004), was $O(k(\frac{1}{\epsilon})^{\mathsf{d}} \cdot \log n)$, and it was improved to $O(k(\frac{1}{\epsilon})^{\mathsf{d}})$, which is independent of n, in (Har-Peled & Kushal, 2007). Feldman and Langberg (Feldman & Langberg, 2011) drastically improved the dependence on the dimension d, from exponential to linear, achieving an ϵ-coreset of size $O(\frac{k}{\epsilon^2} \cdot \mathsf{d})$, and this bound was recently generalized to doubling metrics (Huang et al., 2018). Recently, coresets of size *independent* of d and polynomial in $\frac{k}{\epsilon}$ were devised by (Sohler & Woodruff, 2018).

Clustering in Graph Metrics While clustering in Euclidean spaces is very common and well studied, clustering in graph metrics is also of great importance and has many applications. For instance, clustering is widely used for community detection in social networks (Fortunato, 2010), and is an important technique for the visualization of graph data (Herman et al., 2000). Moreover, k-clustering on graph metrics is one of the central tasks in data mining of spatial (e.g., road) networks (Shekhar & Liu, 1997; Yiu & Mamoulis, 2004), and it has been applied in various data analysis methods (Rattigan et al., 2007; Cui et al., 2008), and many other applications can be found in a survey (Tansel et al., 1983).

Despite the importance of graph k-MEDIAN, coresets for this problem were not studied before, and to the best of our knowledge, the only known constructions applicable to graph metrics are coresets for general n-point metrics $M = (V, d)$ with $X = V$ (Chen, 2009; Feldman & Langberg, 2011), that have size $\mathrm{poly}\log n$. In contrast, as mentioned above, coresets for Euclidean spaces usually have size independent of $n = |V|$ and sometimes even independent of the dimension d. Moreover, this generic construction assumes efficient access to the distance function, which is expensive in graphs and requires to compute all-pairs shortest paths.

To fill this gap, we study coresets for k-MEDIAN on the shortest-path metric of an edge-weighted graph G. As a baseline, we confirm that the $O(\log n)$ factor in coreset size is really necessary for general graphs, which motivates us to explore whether structured graphs admit smaller coresets. We achieve this by designing coresets whose size are independent of n when G has a bounded *treewidth* (see Definition 2.1), which is a special yet common graph family. Moreover, our algorithm for constructing the coresets runs in *near-linear time* (for every graph regardless of treewidth).

Indeed, treewidth is a well-studied parameter that measures how close a graph is to a tree (Robertson & Seymour, 1986; Kloks, 1994), and intuitively it guarantees a (small) vertex separator in every subgraph. Several important graph families have bounded treewidth: trees have treewidth at most 1, series-parallel graphs have treewidth at most 2, and k-outerplanar graphs, which are an important special case of planar graphs, have treewidth $O(k)$. In practice, treewidth is a good complexity measure for many types of graph data. A recent experimental study showed that real data sets in various domains including road networks of the US power grid networks and social networks such as an ego-network of Facebook, have small to moderate treewidth (Maniu et al., 2019).

1.1. Our Results

Our main result is a near-linear time construction of a coreset for k-MEDIAN whose size depends linearly on the treewidth of G and is completely independent of $|X|$ (the size of the data set). This significantly improves the generic $O(\frac{k}{\epsilon^2} \cdot \log n)$ size bound from (Feldman & Langberg, 2011) whenever the graph has small treewidth.

Theorem 1.1 (Fast Coresets for Graph k-MEDIAN; see Theorem 3.1)**.** *For every edge-weighted graph $G = (V, E)$, $0 < \epsilon < 1$, and integer $k \geq 1$, k-MEDIAN of every data set $X \subseteq V$ (with respect to the shortest-path metric of G) admits an ϵ-coreset of size $\tilde{O}(\frac{k^3}{\epsilon^2}) \cdot \mathrm{tw}(G)$.[1] Furthermore, the coreset can be computed in time $\tilde{O}(|E|)$ with high probability.*[2]

We complement our coreset construction with a size lower bound, which is information-theoretic, i.e., regardless of computational power.

Theorem 1.2 (Coreset Size Lower Bound; proved in full version)**.** *For every $0 < \epsilon < 1$ and integers $k, t \geq 1$, there exists a graph $G = (V, E)$ with $\mathrm{tw}(G) \leq t$, such that every ϵ-coreset for k-MEDIAN on $X = V$ in G has size $\Omega(\frac{k}{\epsilon} \cdot t)$.*

This matches the linear dependence on $\mathrm{tw}(G)$ in our coreset construction, and we show in a corollary (in full version) that the same hard instance actually implies for the first time that the $O(\log n)$ factor is optimal for general metrics, which justifies considering restricted graph families.

Experiments We evaluate our coreset on real-world road networks. Thanks to our new near-linear time algorithm, the coreset construction scales well even on data sets with millions of points. Our coreset consistently achieves $< 5\%$ error using only 1000 points on various distributions of data points X, and the small size of the coreset results in a 100x-

[1] Throughout, we use $\tilde{O}(f)$ to denote $O(f \cdot \mathrm{polylog}(f))$.

[2] We note that our size bound can be improved to $\tilde{O}(\frac{k^2}{\epsilon^2}) \cdot \mathrm{tw}(G)$, by replacing Lemma 3.2 with Theorem 31 of a recent work (Feldman et al., 2020).

1000x speedup of local search approximation algorithm for graph k-MEDIAN. When experimenting with our coreset on different data sets X, we observe that coresets of similar size yield similar error, which confirms our theoretical bounds (for structured graphs) where the coreset size is independent of the data set.

In fact, our experiments demonstrate that the algorithm performs well even without knowing the treewidth of the graph G. More precisely, the algorithm can be executed on an arbitrary graph G, and the treewidth parameter is needed only to tune the coreset size. We do not know the treewidth of the graphs used in the experiments (we made no attempt to compute it, even approximately). Our experiments validate the algorithm's effectiveness in practice, with coreset size much smaller than our worst-case theoretical guarantees. In fact, it is also plausible that while the graphs have moderate treewidth, they are actually "close" to having an even smaller treewidth. Another possible explanation is that the algorithm actually works well on a wider family of graphs than bounded treewidth, hence it is an interesting open question to analyze our construction for graphs that are planar or excluding a fixed minor.

1.2. Technical Contributions

Our coreset construction employs the importance sampling framework proposed by Feldman and Langberg (Feldman & Langberg, 2011), although implemented differently as explained in Remark 3.1. A key observation of the framework is that it suffices to give a *uniform* upper bound on the *shattering dimension* (see Definition 2.2), denoted $\mathrm{sdim}_v(M)$, of the metric $M = (V, d)$ weighted by any point weight $v : V \to \mathbb{R}_+$. Our main technical contribution is a (uniform) shattering-dimension bound that is *linear* in the treewidth, and this implies the size bound of our coreset.

Theorem 1.3 (Shattering Dimension Bound; see Theorem 3.5)**.** *For every edge-weighted graph $G = (V, E)$ and every point weight function $v : V \to \mathbb{R}_+$, the shortest-path metric M of G satisfies* $\mathrm{sdim}_v(M) \le O(\mathrm{tw}(G))$.

The shattering dimension of many important spaces was studied, including for Euclidean spaces (Feldman & Langberg, 2011) and for doubling spaces (Huang et al., 2018). For graphs, the shattering dimension of an K_r-minor free graph (which includes bounded-treewidth graphs) is known to be $O(r)$ (Bousquet & Thomassé, 2015) for *unit weight* $v \equiv 1$, see Section 2 for details. However, a general point weight $v : V \to \mathbb{R}_+$ introduces a significant technical challenge which is illustrated below.

In our context, the shattering dimension is defined with respect to the set system of all *v-weighted* metric balls, where every such ball has a center $x \in V$ and a radius $r \ge 0$, and is defined by

$$B_v(x, r) := \{y \in V : v(y) \cdot d(x, y) \le r\}. \tag{1}$$

Roughly speaking, a bounded shattering dimension means that for every subset $H \subseteq V$, the number of ways this H is intersected by v-weighted metric balls is at most $\mathrm{poly}(|H|)$. The main technical difficulty is that an arbitrary weight v can completely break the "continuity" of the space, which can be illustrated even in one-dimensional line $V = \mathbb{R}$ (and analogously in a simple path graph on $V = \{0, 1, \ldots, n\}$), where under unit weight $v \equiv 1$, every ball is a *contiguous* interval, but under a general weight v an arbitrary subset of points could form a ball; indeed, for a center $x = 0$ and radius $r = 1$, every point $y \ge 1$ can be made inside or outside of the ball $B_v(x, r)$ by setting $v(y) = \frac{1}{2y}$ or $v(y) = \frac{2}{y}$.

Our main technical contribution is to analyze the shattering dimension with general weight functions, which we outline now briefly (see Section 3 for a more formal overview). We start by showing a slightly modified balanced-separator theorem for bounded-treewidth graphs (Lemma 3.6), through which the problem of bounding the shattering dimension is reduced to bounding the "complexity" of shortest paths that cross one of a few vertex separators, each of size $O(\mathrm{tw}(G))$. An important observation is that, if $S \subset V$ is a vertex separator and $x, y \in V$ belong to different components after removing S, then every path connecting x to y must *cross* S, and hence

$$d(x, y) = \min\{d(x, s_i) + d(s_i, y) : s_i \in S\}.$$

If we fix $x \in V$ and consider all $y \in V$, then we can think of each $d(s_i, y)$ as a real variable $z_i \in \mathbb{R}$, so instead of varying over all $y \in V$, which depends on the graph structure, we can vary over $|S|$ real variables, and each $d(x, \cdot)$ is the minimum of $|S|$ linear (actually affine) functions, or in short a min-linear function. Finally, we consider different $x \in V$ with the same separator S, and hence the same $|S|$ real variables, and we bound the "complexity" of these min-linear functions by relating it to the *arrangement number* of hyperplanes, which is a well-studied concept in computational geometry. We believe our techniques may be useful for more general graph families, such as minor-free graphs.

1.3. Related Work

Approximation algorithms have been extensively studied for k-MEDIAN in graph metrics, and here we only mention a small selection of results. In general graphs (which is equivalent to general metrics), it is NP-hard to approximate k-MEDIAN within $1 + \frac{2}{e}$ factor (Jain et al., 2002), and the state-of-art is a 2.675-approximation (Byrka et al., 2017). For planar graphs and more generally graphs excluding a fixed minor, a PTAS for k-MEDIAN was obtained

in (Cohen-Addad et al., 2019a) based on local search, and it has been improved to be FPT (i.e. the running time is of the form $f(k,\epsilon)\cdot n^{O(1)}$) recently (Cohen-Addad et al., 2019b). For general graphs, (Thorup, 2005) proposed an $O(1)$-approximation that runs in near-linear time.

Coresets have been studied for many problems in addition to k-MEDIAN, such as PCA (Feldman et al., 2020) and regression (Maalouf et al., 2019), but in our context we focus on discussing results for other clustering problems only. For k-CENTER clustering in Euclidean space $\mathbb{R}^{\mathsf{d}}$, an ϵ-coreset of size $O(\frac{k}{\epsilon^{\mathsf{d}}})$ can be constructed in near-linear time (Agarwal & Procopiuc, 2002; Har-Peled, 2004). Recently, coreset for generalized clustering objective receives attention from the research community, for example, (Braverman et al., 2019) obtained simultaneous coreset for ORDERED k-MEDIAN, (Schmidt et al., 2018; Huang et al., 2019) gave a coresets for k-clustering with fairness constraints, and (Marom & Feldman, 2019) presented a coreset for k-MEANS clustering on lines in Euclidean spaces where inputs are lines in $\mathbb{R}^{\mathsf{d}}$ while the centers are points.

2. Preliminaries

Definition 2.1 (Tree Decomposition and Treewidth)**.** A tree decomposition of a graph $G=(V,E)$ is a tree $\mathcal{T}$ with node set $\mathcal{V}$, such that each node in $\mathcal{V}$, called a *bag*, is a subset of V, and the following conditions hold:

1. $\bigcup_{S\in\mathcal{V}} S = V$.
2. $\forall u \in V$, the nodes of $\mathcal{T}$ that contain u form a connected component in $\mathcal{T}$.
3. $\forall (u,w)\in E$, $\exists S\in\mathcal{V}$, such that $\{u,w\}\subseteq S$.

The treewidth of a graph G, denoted $\mathrm{tw}(G)$, is the smallest integer t, such that there exists a tree decomposition with maximum bag size $t+1$.

A *nice* tree decomposition is a tree decomposition such that each bag has a degree at most 3.[3] It is well known that there exists a nice tree decomposition of G with maximum bag size $O(\mathrm{tw}(G))$ (Kloks, 1994).

Shattering Dimension As mentioned in Section 1, our coreset construction employs the Feldman-Langberg framework (Feldman & Langberg, 2011). A key notion in the Feldman-Langberg framework is the *shattering dimension* of a metric space with respect to a point weight function.

Definition 2.2 (Shattering Dimension)**.** Given a point weight function $v: V\to\mathbb{R}_+$, the shattering dimension of $M=(V,d)$ with respect to v, denoted as $\mathrm{sdim}_v(M)$, is the smallest integer t, such that for every $H\subseteq V$ with $|H|\geq 2$, it holds that

$$|\{H\cap B_v(x,r) : x\in V, r\geq 0\}| \leq |H|^t.$$

Observe that the left-hand side counts the number of ways that H is intersected by all weighted balls, which were defined in (1). We remark that our notion shattering dimension is tightly related to the well-known *VC-dimension* (see for example (Kearns & Vazirani, 1994)). In particular, let $\mathcal{B}_v := \{B_v(x,r) : x\in V, r\geq 0\}$ be the collection of all v-weighted balls, then the VC-dimension of the set system $(V,\mathcal{B}_v)$ is within a logarithmic factor to the $\mathrm{sdim}_v(M)$. It was shown in (Bousquet & Thomassé, 2015) that the VC-dimension of a K_r-minor free graph with unit weights $v\equiv 1$ is at most $O(r)$, which immediately implies an $O(r)$ bound also for the shattering dimension (under unit weight $v\equiv 1$).

3. Coresets for k-MEDIAN in Graph Metrics

In this section, we present a near-linear time construction for ϵ-coreset for k-MEDIAN in graph metrics, whose size is linear in the treewidth. This is formally stated in the following theorem.

Theorem 3.1 (Coreset for Graph k-MEDIAN)**.** *For every edge-weighted graph $G=(V,E)$, $0<\epsilon,\delta<1$, and integer $k\geq 1$, k-MEDIAN of every data set $X\subseteq V$ (with respect to the shortest path metric of G) admits an ϵ-coreset of size $\tilde{O}\left(\frac{k^2}{\epsilon^2}\cdot(k\cdot\mathrm{tw}(G)+\log(1/\delta))\right)$. Furthermore, it can be computed in time $\tilde{O}(|E|)$ with success probability $1-\delta$.*

Our construction is based on the Feldman-Langberg framework (Feldman & Langberg, 2011), in which the coreset is constructed using *importance sampling*. While this framework is quite general, their implementation is tailored to Euclidean spaces and is less suitable for graphs metrics. In addition, their algorithm runs in $\tilde{O}(kn)$ time assuming access to pairwise distances, which is efficient in Euclidean spaces but rather expensive in graphs.

We give an efficient implementation of the Feldman-Langberg framework in graphs, and also provide an alternative analysis that is not Euclidean-specific. A similar strategy was previously employed for constructing coresets in doubling spaces (Huang et al., 2018), but that implementation is not applicable here because of the same efficiency issue (i.e., it requires oracle access to distances). We present our implementation and analysis of the framework below, and then put it all together to prove Theorem 3.1.

Importance Sampling At a high level, the importance sampling method consists of two steps.

[3] Usually, nice tree decompositions are defined to have additional guarantees, but we only need the bounded degree.

1. For each data point $x \in X$, compute an importance $\sigma_x \in \mathbb{R}_+$.

2. Form a coreset by drawing N (to be determined later) independent samples from X, where each sample picks every $x \in X$ with probability proportional to σ_x, i.e., $p_x := \frac{\sigma_x}{\sum_{x' \in X} \sigma_{x'}}$, and assigns it weight $\frac{1}{p_x}$.

To implement the algorithm, we need to define σ_x and N. Following the Feldman-Langberg framework, each importance σ_x is an *upper bound* on the *sensitivity*

$$\sigma_x^\star := \max_{C \subseteq V, |C|=k} \frac{d(x,C)}{\mathrm{cost}(X,C)},$$

which was introduced in (Langberg & Schulman, 2010) and represents the maximum possible contribution of x to the objective over all center sets C.

Let the *total importance* be $\sigma_X := \sum_{x \in X} \sigma_x$. Our key tool is the following bound on coreset size N in terms of σ_X and a uniform upper bound on $\mathrm{sdim}_v(M)$. It follows by combining their Theorem 4.1 in (Feldman & Langberg, 2011) with a PAC sampling bound from (Vapnik & Chervonenkis, 2015).

Lemma 3.2 (Theorem 4.1 of (Feldman & Langberg, 2011) together with (Vapnik & Chervonenkis, 2015))**.** *Let* $\mathrm{sdim}_{\max} := \max_{v: V \to \mathbb{R}_+} \mathrm{sdim}_v(M)$. *Then for*

$$N = \tilde{O}\Big(\big(\frac{\sigma_X}{\epsilon}\big)^2 \big(k \cdot \mathrm{sdim}_{\max} + \log \frac{1}{\delta}\big)\Big),$$

the importance sampling procedure returns an ϵ-coreset with probability at least $1 - \delta$.

Computing σ_x An efficient algorithm to compute σ_x was presented in (Varadarajan & Xiao, 2012), assuming that an $O(1)$-approximation to k-MEDIAN is given. Furthermore, an $O(k)$ bound on the total importance σ_X was shown.

Lemma 3.3 ((Varadarajan & Xiao, 2012))**.** *Suppose $C^\star$ is a ρ-approximate solution to the k-MEDIAN instance. Let $\sigma_x := \rho \cdot \big(\frac{d(x,C^\star)}{\mathrm{cost}(X,C^\star)} + \frac{1}{|C^\star(x)|}\big)$, where $C^\star(x) \subseteq X$ is the cluster of $C^\star$ that contains x. Then $\sigma_X = O(\rho k)$ and*

$$\forall x \in X, \quad \sigma_x \geq \Omega(\sigma_x^\star).$$

Thus, to construct the coreset in near-linear time, we need to compute an $O(1)$-approximation $C^\star$ fast, for which we use the following result of (Thorup, 2005).

Lemma 3.4. *There is an algorithm that, given as input a weighted undirected graph $G = (V, E)$ and data set $X \subseteq V$, computes an $O(1)$-approximate solution for graph k-MEDIAN in time $\tilde{O}(|E|)$ with probability $1 - o(1)$.*

Finally, we need a uniform shattering-dimension bound (with respect to treewidth). Such a bound, stated next, is our main technical contribution and its proof is presented in Section 3.1.

Theorem 3.5 (Shattering Dimension)**.** *For every edge-weighted graph $G = (V, E)$ and every point weight function $v : V \to \mathbb{R}_+$, the shortest-path metric M of G satisfies* $\mathrm{sdim}_v(M) \leq O(\mathrm{tw}(G))$.

Remark 3.1. Our implementation of the framework of (Feldman & Langberg, 2011) differs in several respects. First, their shattering dimension is defined with respect to *hyperbolic* balls instead of usual metric balls (as the underlying set system). Second, the choice of σ_x and the sampling bound are different. While they achieve an improved coreset size (linear in k), their analysis relies on Euclidean-specific properties and does not apply in graph metrics.

Putting It Together We are now in position to conclude our main result.

Proof of Theorem 3.1. Construct a coreset by the importance sampling procedure, where the importance σ_x is computed using Lemma 3.4. Then we can apply Lemma 3.3 with $\rho = O(1)$ to bound the total importance $\sigma_X = O(k)$. Combining this and the shattering dimension from Theorem 3.5, we can apply Lemma 3.2 with coreset size

$$N = \tilde{O}\Big(\frac{k^2}{\epsilon^2}\big(k \cdot \mathrm{tw}(G) + \log \frac{1}{\delta}\big)\Big).$$

The running time is dominated by computing the importance σ_x for all $x \in X$, which we claim can be computed in time $\tilde{O}(|E|)$ by using Lemmas 3.3 and 3.4. Indeed, first compute $C^\star$ in time $\tilde{O}(|E|)$ using Lemma 3.4, then compute the clustering of X with respect to $C^\star$ and the associated distances $\{d(x, C^\star) : x \in X\}$ using a *single* Dijkstra execution in time $\tilde{O}(|E|)$ time (see Observation 1 of (Thorup, 2005)). Finally, use this information to compute σ_x for all $x \in X$, and sample according to it, in total time $\tilde{O}(|X|)$. □

3.1. Bounding the Shattering Dimension

We give a technical overview before presenting the detailed proof of Theorem 3.5. Recall that the unit-weight case of shattering dimension was already proved in (Bousquet & Thomassé, 2015), and our focus is when $v : V \to \mathbb{R}_+$ is a general weight function.

The proof starts with a slightly modified balanced-separator theorem for bounded treewidth graphs (Lemma 3.6), through which the problem of bounding the shattering dimension is reduced to bounding the complexity of *bag-crossing* shortest paths for every bag.

Lemma 3.6 (Structural Lemma)**.** *Given graph $G(V, E)$, and $H \subseteq V$, there exists a collection $\mathcal{S} \subseteq 2^V$ of subsets of V, such that the following holds.*

1. $\bigcup_{A\in\mathcal{S}} A = V$.
2. $|\mathcal{S}| \le \text{poly}(|H|)$.
3. *For each $A \in \mathcal{S}$, either $|A| \le O(\text{tw}(G))$, or i) $|A \cap H| \le O(\text{tw}(G))$ and ii) there exists $P \subseteq V$ with $|P| \le O(\text{tw}(G))$ such that there is no edge in E between A and $V \setminus (A \cup P)$.*

A well-known fact is that every bag $\{s_i, \dots, s_m\} \subseteq V$ in the tree decomposition is a vertex cut of size $m = O(\text{tw}(G))$, and this leads to an important observation: if x and y belong to different components after removing this bag, then every path connecting x with y *crosses* the bag, and hence

$$d(x, y) = \min\{d(x, s_i) + d(s_i, y) : i \in [m]\}.$$

Now suppose we fix $x \in V$ and let y vary over V; then we can write $d(x, \cdot)$ as a min-linear function (which means the minimum of m linear functions) $f_x : \mathbb{R}^m \to \mathbb{R}_+$, whose variables are $z_i = d(s_i, y)$ for $i \in [m]$; notice that the terms $d(x, s_i)$ are constant with respect to y.

This alternative view of distances enables us to bound the complexity of shortest-paths, because the functions $\{f_x\}_x$ all have common variables $\{z_i = d(s_i, y)\}_{i\in[m]}$ in real domain (instead of variables in V), and more importantly, the domain of these functions has low dimension $m = O(\text{tw}(G))$. Furthermore, the min-linear description also handles weights because $v(x) \cdot f_x$ is min-linear too. Finally, in a technical lemma (Lemma 3.7), we relate the complexity of a collection of min-linear functions of low dimension to the *arrangement number* of hyperplanes, which is a well-studied quantity in computational geometry.

Lemma 3.7 (Complexity of Min-Linear Functions)**.** *Suppose $f_1, \dots, f_s$ are s functions such that for every $i \in [s]$,*

- *$f_i : \mathbb{R}^l \to \mathbb{R}$, and*
- *$f_i(x) = \min_{j\in[l]}\{g_{ij}(x)\}$ where each $g_{ij} : \mathbb{R}^l \to \mathbb{R}$ is a linear function.*

For $x \in \mathbb{R}^l$, let σ_x be the permutation of $[s]$ such that $i \in [s]$ is ordered by $f_i(x)$ (in non-increasing order), and ties are broken consistently. Then $|\{\sigma_x : x \in \mathbb{R}^l\}| \le O(sl)^{O(l)}$.

Due to space limit, we omit the detailed proofs and they can be found in the full version.

4. Experiments

We implement our algorithm and evaluate its performance on real-world road networks. Our implementation generally follows the importance sampling algorithm as in Section 3.

Algorithm 1 ITERATEDTHORUPSAMPLING

input edge-weighted graph $G = (V, E)$, data set $X \subseteq V$, number of centers k, parameters n and m that control the number of iterations
output bicriteria solution $F \subseteq X$

let $X_0 \leftarrow X$, and $\forall u \in X_0$ let $w_{X_0}(u) \leftarrow 1$
for $i = 1$ to n **do**
  expand X_{i-1} into a multi-set X', such that each $u \in X'$ has multiplicity $w_{X_{i-1}}(u)$
  let $F_i \leftarrow$ THOSAMPLEBEST(G, X', k, m)
  let $X_i \leftarrow F_i$ and $\forall u \in X_i$, let

$$w_{X_i}(u) \leftarrow |\{v \in X : \text{NN}_{F_i}(v) = u\}|$$

  // $\text{NN}_{F_i}(v)$ is the nearest point in F_i from v; this step implements the projection of X to F_i
end for
return F_n

Algorithm 2 THOSAMPLEBEST

input edge-weighted graph $G = (V, E)$, data set $X \subseteq V$, number of centers k, number of iterations m
output bicriteria solution $F \subseteq X$

for $i = 1$ to m **do**
  let $F_i \leftarrow$ THOSAMPLE(G, X, k)
  // THOSAMPLE is Algorithm D of (Thorup, 2005)
end for
return F_i such that $i = \arg\min_{1\le j\le m} \text{cost}(F_i, X)$

We observe that the running time is dominated by computing an $O(1)$-approximation for k-MEDIAN (used to assign importance σ_x), for which we use Thorup's $\tilde{O}(|E|)$-time algorithm (Lemma 3.4). However, the straightforward implementation of Thorup's algorithm is very complicated and scales with $k \log^3 n$ which is already near the size of our data set, and thus we employ an optimized implementation based on it.

Optimized Implementation Thorup's algorithm starts with an $O(|E| \log |E|)$-time procedure to find a bicriteria solution F (Algorithm D in (Thorup, 2005)), namely, $|F| = O(k \log^2 n)$ such that $\text{cost}(F, X) = O(1) \cdot \text{OPT}$. Then a modified Jain-Vazirani algorithm (Jain & Vazirani, 2001) is applied on F to produce the final $O(1)$-approximation in $\tilde{O}(|E|)$ time. However, the modified Jain-Vazirani algorithm is complicated to implement, and the hidden polylogarithmic factor in its running time is quite large. Thus, we replace the Jain-Vazirani algorithm with a simple local search algorithm (Arya et al., 2001) to find an $O(1)$-approximation on F. The performance of the local search relies heavily on $|F|$, but $|F| = O(k \log^2 n)$ is not much smaller than n for our data set. Therefore, we run

the bicriteria approximation *iteratively* to further reduce $|F|$. Specifically, after we obtain F_i, we project X to F_i (i.e., map each $x \in X$ to its nearest point in F_i) to form X_i, and run the bicriteria algorithm again on X_i to form F_{i+1}. We use a parameter to control the number of iterations, and we observe that F reduces significantly in our data set with only a few iterations.

The procedure for finding F iteratively is described in Algorithm 1, which uses Algorithm 2 as a subroutine. Algorithm 2 essentially corresponds to the above-mentioned Thorup's bicriteria approximation algorithm THOSAMPLE (Algorithm D in (Thorup, 2005)), except that we execute it multiple times (m times in Algorithm 1) to boost the success probability. As can be seen in our experiments, the improved implementation scales very well on road networks and achieves high accuracy.

Table 1. Comparison of empirical error of our coreset with the baseline of uniform sampling when $k = 25$ and varying coreset sizes, for both data sets X_{uni} and X_{man}.

SIZE	X_{uni}		X_{man}	
	OURS	UNI.	OURS	UNI.
25	32.1%	35.8%	32.1%	151.6%
50	26.6%	23.0%	22.1%	90.3%
75	17.8%	23.2%	23.2%	62.3%
100	17.2%	17.2%	15.2%	49.9%
500	7.72%	8.53%	8.34%	31.7%
1250	4.57%	5.32%	4.87%	21.2%
2500	4.14%	4.03%	3.29%	9.53%
3750	2.49%	3.21%	2.89%	14.39%
6561	2.00%	2.11%	2.38%	5.83%
13122	1.50%	1.70%	1.53%	6.53%
19683	1.27%	1.36%	1.39%	3.73%

Experimental Setup Throughout the experiments the graph G is a road network of New York State extracted from OpenStreetMap (OpenStreetMap contributors, 2020) and clipped by bounding box to enclose New York City (NYC). This graph consists of 1 million vertices and 1.2 million edges whose weight are the distances calculated using the Haversine formula between the endpoints. It is illustrated in Figure 1. Our software is open source and freely available, and implemented in C++17. All experiments were performed on a Lenovo x3850 X6 system with 4 2.0 GHz Intel E7-4830 CPUs, each with 14 processor cores with hyperthreading enabled. The system had 1 TB of RAM.

4.1. Performance of Coresets

Our first experiments evaluate how the accuracy of our coresets depends on their size. Here, the data X may be interpreted as a set of customers to be clustered, and their distribution could have interesting geographical patterns. We experiment with X chosen uniformly at random from V (all of NYC), mostly for completeness as it is less likely in practice, and denote this scenario as X_{uni}. We also experiment with a "concentrated" scenario where X is highly concentrated in Manhattan but also has much fewer points picked uniformly from other parts of NYC, denoted as X_{man}. We demonstrate the two types of data sets X in Figure 2.

We define the *empirical error* of a coreset D and a center set $C \subseteq V$ as $\text{err}(D, C) := \left|\frac{\text{cost}(D,C)}{\text{cost}(X,C)} - 1\right|$ (corresponding to ϵ in the definition of a coreset). Since by definition a coreset preserves the objective for all center sets, we evaluate the empirical error by randomly picking 2000 center sets $C \subseteq V$ from V, and reporting the *maximum* empirical error $\text{err}(D, C)$ over all these C. For the sake of evaluation, we compare the maximum empirical error of our coreset with a baseline of a uniform sample, where points are drawn uniformly at random from X and assigned equal weight (that sums to $|X|$). To reduce the variance introduced by the randomness in the coreset construction, we repeat each construction 10 times and report the average of their maximum empirical error.

Results We report the empirical error of our coresets and that of the uniform sampling baseline in Table 1. Our coreset performs consistently well and quite similarly on the two data sets X, achieving for example 5% error using only about 1000 points. Compared to the uniform sampling baseline, our coreset is $3 - 5$ times more accurate on the Manhattan-concentrated data X_{man}, and (as expected) is comparable to the baseline on the uniform data X_{uni}.

In addition, we show the accuracy of our coresets with respect to varying sizes of data sets X in Figure 3 (left). We find that coresets of the same size have similar accuracy regardless of $|X|$, which confirms our theory that the size of the coreset is independent of $|X|$ in structured graphs. We also verify in Figure 3 (right) that a coreset constructed for a target value $k = 25$ performs well also as a coreset for fewer centers (various $k' < k$). While this should not be surprising and follows from the coreset definition, it is very useful in practice when k is not known in advance, and a coreset (constructed for large enough k) can be used to experiment and investigate different $k' < k$.

4.2. Speedup of Local Search

An important application of coresets is to speed up existing approximation algorithms. To this end, we demonstrate the speedup of the local search algorithm of (Arya et al., 2001) achieving 5-approximation for graph k-MEDIAN by using our coreset. In particular, we run the local search on top of our coreset D (denoted as $D \times V$), and then compare the accuracy and the overall running time with those of running

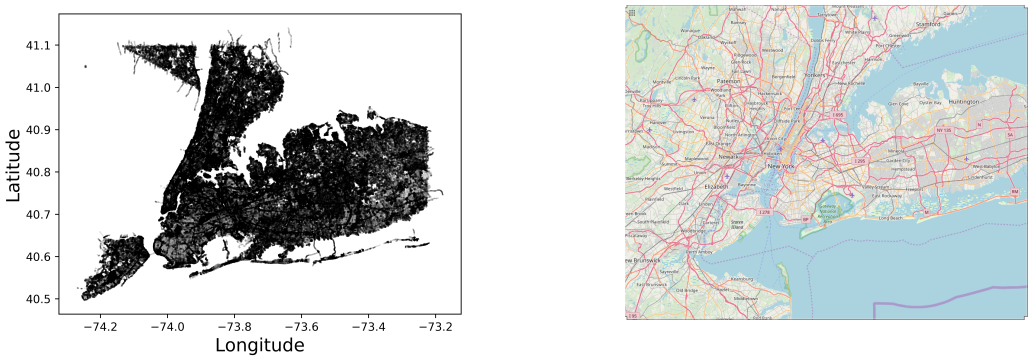

Figure 1. Illustration of our graph G, plotting (on left) the vertices according to their geographic coordinates, and showing (on right) a map, taken from OpenStreetMap, of the bounding box used to form G.

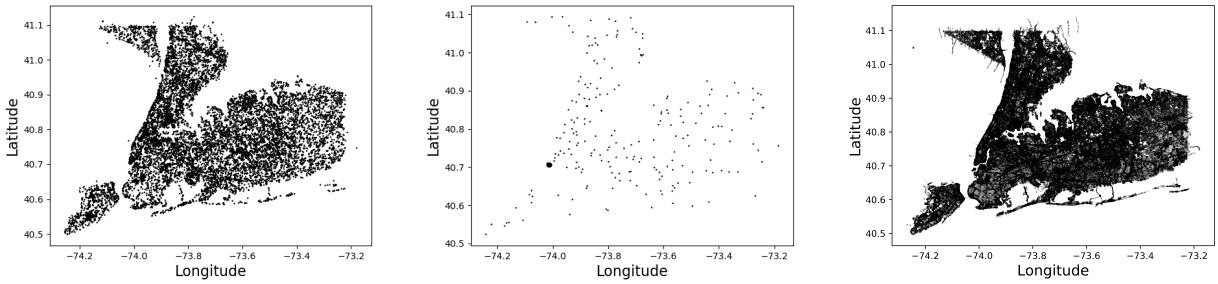

Figure 2. Illustration of data set X used in the accuracy-vs-size experiment. The left plot is a uniform data X_{uni}, the middle is X_{man} that is highly concentrated in Manhattan, where in both cases $|X| \approx 14000$, and the right plot is all of V which is the full NYC.

the local search on the original data X (denoted as $X \times V$). Notice that by definition of k-MEDIAN, the centers always come from V, which defines the search space, and a smaller data set can only affect the time required to evaluate the objective. This limits the potential speedup of local search, and therefore we additionally evaluate the running time and accuracy of local search on D when also the centers come from D (denoted as $D \times D$).

We report separately the running time of the coreset construction, denoted T_{cs}, and that of the local search on the coreset. Indeed, as mentioned in Section 4.1, a coreset D constructed for large k can be used also when clustering for $k' < k$, and since one can experiment with any clustering algorithm on D (e.g. Jain-Vazirani, local search, etc.), the coreset construction is one-time effort that may be averaged out when successive clustering tasks are performed on D.

The results are illustrated in Figure 4, where we find that the coreset construction is very efficient, about 100 times faster than local search on X, not to mention that the coreset may be used for successive clustering tasks. We see that the speedup of local search $D \times V$ is only moderate (which matches the explanation above), but the alternative local search on $D \times D$ performs extremely well — for example using $|D| \approx 1000$, it is about 1000 times faster than the naive local search on X, and it achieves similar objective value (i.e. $5\% - 10\%$ error). This indicates that local search on $D \times D$ may be a good candidate for practical use.

Acknowledgements

This research was supported in part by NSF CAREER grant 1652257, ONR Award N00014-18-1-2364, the Lifelong Learning Machines program from DARPA/MTO, Israel Science Foundation grant #1086/18, and a Minerva Foundation grant.

References

Agarwal, P. K. and Procopiuc, C. M. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2): 201–226, 2002.

Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. Approximating extent measures of points. *J. ACM*, 51(4):606–635, July 2004. ISSN 0004-5411. doi:10.1145/1008731.1008736.

Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala,

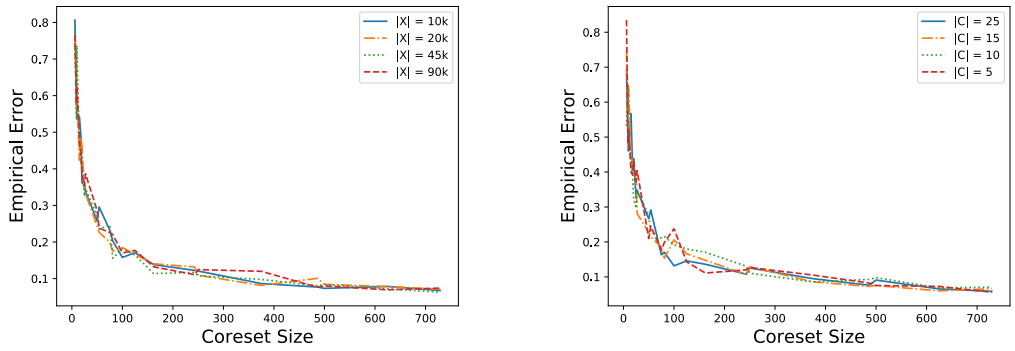

Figure 3. The left plot shows the accuracy of coresets ($k = 25$) on uniform X's with varying sizes. Each line is labeled with the size of each respective $X \subseteq V$. The right plot shows the accuracy of coresets constructed with $k = 25$ but evaluated with smaller center sets C on the same uniform X with $|X| = 10^4$.

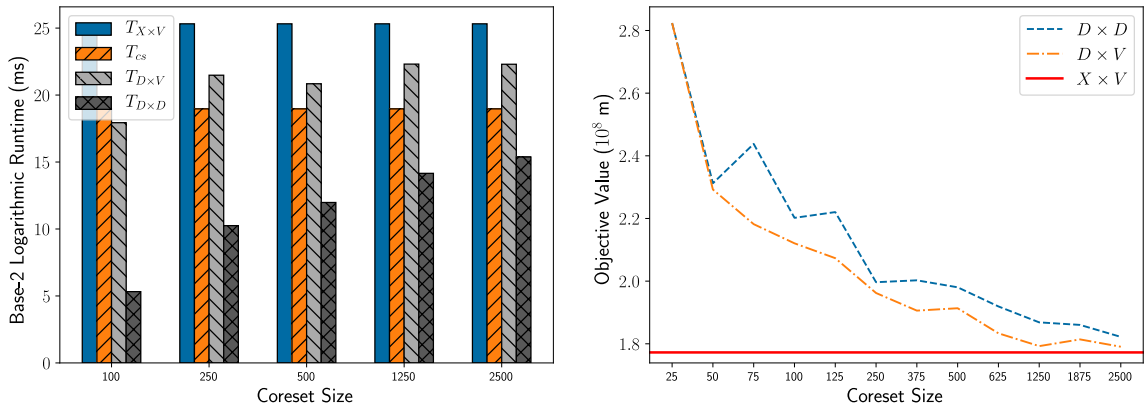

Figure 4. Performance of local search on $X \times V$, $D \times V$, and $D \times D$. The running time is shown on the left, where the coreset construction time T_{cs} is separated out (so $T_{D\times V}$ and $T_{D\times D}$ do not include T_{cs}). The objective values reached are shown on the right. Here $k = 25$ and $X = V$ is the whole NYC of size $|X| \approx 10^6$.

K., and Pandit, V. Local search heuristic for k-median and facility location problems. In *STOC*, pp. 21–29. ACM, 2001.

Baker, D., Braverman, V., Huang, L., Jiang, S. H., Krauthgamer, R., and Wu, X. Coresets for clustering in graphs of bounded treewidth. *CoRR*, abs/1907.04733, 2019. https://arxiv.org/abs/1907.04733.

Balcan, M.-F. F., Ehrlich, S., and Liang, Y. Distributed k-means and k-median clustering on general topologies. In *NIPS*, pp. 1995–2003, 2013.

Bousquet, N. and Thomassé, S. VC-dimension and Erdős–Pósa property. *Discrete Mathematics*, 338(12):2302–2317, 2015.

Braverman, V., Jiang, S. H., Krauthgamer, R., and Wu, X. Coresets for ordered weighted clustering. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 744–753. PMLR, 2019.

Byrka, J., Pensyl, T., Rybicki, B., Srinivasan, A., and Trinh, K. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Trans. Algorithms*, 13(2):23:1–23:31, 2017.

Chen, K. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

Cohen-Addad, V., Klein, P. N., and Mathieu, C. Local search yields approximation schemes for k-means and k-median in Euclidean and minor-free metrics. *SIAM J. Comput.*, 48(2):644–667, 2019a. doi:10.1137/17M112717X.

Cohen-Addad, V., Pilipczuk, M., and Pilipczuk, M. Efficient approximation schemes for uniform-cost clustering problems in planar graphs. In *ESA*, volume 144 of *LIPIcs*,

pp. 33:1–33:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019b.

Cui, W., Zhou, H., Qu, H., Wong, P. C., and Li, X. Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1277–1284, 2008.

Feldman, D. Core-sets: An updated survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 10(1), 2020.

Feldman, D. and Langberg, M. A unified framework for approximating and clustering data. In *43rd Annual ACM Symposium on Theory of computing*, pp. 569–578. ACM, 2011. Full version at https://arxiv.org/abs/1106.1379.

Feldman, D., Schmidt, M., and Sohler, C. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering. *SIAM J. Comput.*, 49(3): 601–657, 2020.

Fichtenberger, H., Gillé, M., Schmidt, M., Schwiegelshohn, C., and Sohler, C. BICO: BIRCH meets coresets for k-means clustering. In *ESA*, volume 8125 of *Lecture Notes in Computer Science*, pp. 481–492. Springer, 2013. doi:10.1007/978-3-642-40450-4_41.

Fortunato, S. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

Friggstad, Z., Rezapour, M., and Salavatipour, M. R. Local search yields a PTAS for k-means in doubling metrics. *SIAM J. Comput.*, 48(2):452–480, 2019.

Har-Peled, S. Clustering motion. *Discrete & Computational Geometry*, 31(4):545–565, 2004.

Har-Peled, S. and Kushal, A. Smaller coresets for k-median and k-means clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007. doi:10.1007/s00454-006-1271-x.

Har-Peled, S. and Mazumdar, S. On coresets for k-means and k-median clustering. In *36th Annual ACM Symposium on Theory of Computing*, pp. 291–300, 2004. doi:10.1145/1007352.1007400.

Herman, I., Melançon, G., and Marshall, M. S. Graph visualization and navigation in information visualization: A survey. *IEEE Trans. Vis. Comput. Graph.*, 6(1):24–43, 2000.

Huang, L., Jiang, S., Li, J., and Wu, X. Epsilon-coresets for clustering (with outliers) in doubling metrics. In *59th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 814–825. IEEE, 2018.

Huang, L., Jiang, S. H., and Vishnoi, N. K. Coresets for clustering with fairness constraints. In *NeurIPS*, pp. 7587–7598, 2019.

Jain, K. and Vazirani, V. V. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.

Jain, K., Mahdian, M., and Saberi, A. A new greedy approach for facility location problems. In *STOC*, pp. 731–740. ACM, 2002.

Kearns, M. J. and Vazirani, U. V. *An introduction to computational learning theory*. MIT press, 1994.

Kloks, T. *Treewidth: computations and approximations*, volume 842. Springer Science & Business Media, 1994.

Langberg, M. and Schulman, L. J. Universal epsilon-approximators for integrals. In *SODA*, pp. 598–607. SIAM, 2010.

Maalouf, A., Jubran, I., and Feldman, D. Fast and accurate least-mean-squares solvers. In *NeurIPS*, pp. 8305–8316, 2019.

Maniu, S., Senellart, P., and Jog, S. An experimental study of the treewidth of real-world graph data. In *ICDT*, volume 127 of *LIPIcs*, pp. 12:1–12:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.

Marom, Y. and Feldman, D. k-means clustering of lines for big data. In *NeurIPS*, pp. 12797–12806, 2019.

Munteanu, A. and Schwiegelshohn, C. Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms. *KI*, 32(1):37–53, 2018.

OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2020.

Phillips, J. M. Coresets and sketches. In Toth, C. D., O'Rourke, J., and Goodman, J. E. (eds.), *Handbook of discrete and computational geometry*, chapter 48. Chapman and Hall/CRC, 3rd edition, 2017. doi:10.1201/9781315119601-48.

Rattigan, M. J., Maier, M., and Jensen, D. Graph clustering with network structure indices. In *Proceedings of the 24th international conference on Machine learning*, pp. 783–790. ACM, 2007.

Robertson, N. and Seymour, P. D. Graph minors. II. algorithmic aspects of tree-width. *Journal of algorithms*, 7 (3):309–322, 1986.

Schmidt, M., Schwiegelshohn, C., and Sohler, C. Fair coresets and streaming algorithms for fair k-means clustering. *CoRR*, abs/1812.10854, 2018.

Shekhar, S. and Liu, D. CCAM: A connectivity-clustered access method for networks and network computations. *IEEE Trans. Knowl. Data Eng.*, 9(1):102–119, 1997.

Sohler, C. and Woodruff, D. P. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *FOCS*, pp. 802–813. IEEE Computer Society, 2018.

Tansel, B. C., Francis, R. L., and Lowe, T. J. State of the art—location on networks: a survey, part i and ii. *Management Science*, 29(4):482–497, 1983.

Thorup, M. Quick k-median, k-center, and facility location for sparse graphs. *SIAM J. Comput.*, 34(2):405–432, 2005. ISSN 0097-5397. doi:10.1137/S0097539701388884.

Vapnik, V. N. and Chervonenkis, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pp. 11–30. Springer, 2015.

Varadarajan, K. and Xiao, X. On the sensitivity of shape fitting problems. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

Yiu, M. L. and Mamoulis, N. Clustering objects on a spatial network. In *SIGMOD Conference*, pp. 443–454. ACM, 2004.