Algorithmica (2010) 57: 484–498
DOI 10.1007/s00453-009-9282-7

On Metric Clustering to Minimize the Sum of Radii

**Matt Gibson · Gaurav Kanade · Erik Krohn ·
Imran A. Pirwani · Kasturi Varadarajan**

Received: 22 August 2008 / Accepted: 19 January 2009 / Published online: 12 February 2009

Abstract Given an n-point metric (P, d) and an integer $k > 0$, we consider the problem of covering P by k balls so as to minimize the sum of the radii of the balls. We present a randomized algorithm that runs in $n^{O(\log n \cdot \log \Delta)}$ time and returns with high probability the optimal solution. Here, Δ is the ratio between the maximum and minimum interpoint distances in the metric space. We also show that the problem is NP-hard, even in metrics induced by weighted planar graphs and in metrics of constant doubling dimension.

Keywords Clustering · Polynomial time · Approximation algorithm

Work of M. Gibson, G. Kanade, E. Krohn, and K. Varadarajan was partially supported by NSF CAREER award CCR 0237431.
Work of I.A. Pirwani was partially supported by Alberta Ingenuity. Most of this work was done while I.A. Pirwani was at the University of Iowa, Iowa City, IA 52242, USA.
Part of this work was done while K. Varadarajan was visiting the Institute for Mathematical Sciences, Chennai, India.

M. Gibson · G. Kanade · E. Krohn · K. Varadarajan (✉)
Department of Computer Science, University of Iowa, Iowa City, IA 52242-1419, USA
e-mail: kvaradar@cs.uiowa.edu

M. Gibson
e-mail: mrgibson@cs.uiowa.edu

G. Kanade
e-mail: gkanade@cs.uiowa.edu

E. Krohn
e-mail: eakrohn@cs.uiowa.edu

I.A. Pirwani
Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2E8, Canada
e-mail: pirwani@cs.ualberta.ca

1 Introduction

Clustering is an important problem in computer science. This is evidenced by the fact the researchers have studied numerous kinds of clustering problems (for a recent survey, see [16]), each with a different flavor and suitability to a particular application. The typical input is a collection of data points along with some relationship between pairs of points (usually, distance in some metric space). The typical task is to group the data points by some measure of similarity with the goal of minimizing some objective function. For example, in facility location problems, a reasonable objective is to try to minimize the sum of distances of clients to their nearest facilities, whereas in k-center problems, the goal is to try to minimize the diameter of the largest cluster. In some applications, for example, the k-server problem, there is a bound on the maximum number of clusters that one is allowed to place to satisfy requests that arise in an online setting from points that reside in some metric space. The objective then is to place the at most k servers so as to minimize the sum of distances of demand points to their nearest servers, over some period of time [15]. In applications where the demand points are known *a-priori* and the demand frequency is continuous, the goal is to satisfy the demand points by placing at most k base-stations. The task then is to assign broadcast ranges to the base-stations so as to cover all the demand points while minimizing the sum of these ranges. In such applications, the radius of coverage assigned to any base-station models the amount of power spent to meet the assigned demand points, and the total amount of power needed, the total cost, is modeled by the sum of radii [1, 3–5, 13]. In this paper, we study a clustering problem motivated by applications in base-station coverage.

Given a metric d defined on a set P of n points, we define the ball $\mathrm{B}(v, r)$ centered at $v \in P$ and having radius $r \geq 0$ to be the set $\{q \in P | d(v, q) \leq r\}$. For $\kappa > 0$, a κ-cover for subset $Q \subseteq P$ is a set of at most κ balls, each centered at a point in P, whose union covers (contains) Q. The cost of a set $\mathcal{D}$ of balls, denoted $\mathrm{cost}(\mathcal{D})$, is the sum of the radii of those balls. In this paper, we consider the (metric) minimum cost k-cover problem:

> Given a metric d on a set P of n points as above, and an integer $k > 0$, compute a minimum cost k-cover for P.

Doddi et al. [5] consider the metric min-cost k-cover problem and the closely related problem of partitioning P into a set of k clusters so as to minimize the sum of the cluster diameters. Following their terminology, we will call the latter problem *clustering to minimize the sum of diameters*. They present a bicriteria poly-time algorithm that returns $O(k)$ clusters whose cost is within a multiplicative factor $O(\log(n/k))$ of the optimal. For clustering to minimize the sum of diameters, they also show that the existence of a polynomial time algorithm that returns k clusters whose cost is strictly within 2 of the optimal would imply that $P = NP$. Notice that because the “the sum of diameters” is within a factor of 2 of “the sum of radii” (k-cover problem), the hardness result does not imply the NP-hardness of the k-cover problem. Charikar and Panigrahy [4] give a poly-time algorithm based on the primal-dual method that gives a constant factor approximation—around 3.504—for the k-cover problem, and thus also a constant factor approximation for clustering to minimize the sum of diameters.

The well known k-center problem is a variant of the k-cover problem where the cost of a set of balls is defined to be the maximum radius of any ball in the set. The problem is NP-hard and admits a polynomial time algorithm that yields a 2-approximation [10]. Several other formulations of clustering such as k-median and min-sum k-clustering are NP-hard as well [6, 11].

Recently, Gibson et al. [9] consider the geometric version of the k-cover problem where $P \subset \Re^l$ for some constant l. When the L_1 or L_∞ norm is used to define the metric, they obtain a polynomial time algorithm for the k-cover problem. With the L_2 norm, they give an algorithm that runs in time polynomial in n, the number of points, and in $\log(1/\epsilon)$ and returns a k-cover whose cost is within $(1+\epsilon)$ of the optimal, for any $0 < \epsilon < 1$.

Our Results

Our first result generalizes the algorithmic approach of Gibson et al. [9] to the metric case. For the k-cover problem in the general metric setting, we obtain an exact algorithm whose running time is $n^{O(\log n \cdot \log \Delta)}$, where Δ is the *aspect ratio* of the metric space, the ratio between the maximum interpoint distance and the minimum interpoint distance. The algorithm is randomized and succeeds with high probability. Thus when Δ is bounded by a polynomial in n, the running time of the algorithm is quasi-polynomial. This result for the k-cover problem should be contrasted with the NP-hardness results for problems such as k-center, k-median, and min-sum k-clustering, which hold when the aspect ratio is bounded by a polynomial in n.

The main idea that underlies this result is that if we probabilistically partition the metric into sets with at most half the original diameter [2, 7], then with high probability only $O(\log n)$ balls in the optimal k-cover of P are "cut" by the partition. A recursive approach is then used to compute the optimal k-cover.

This algorithmic result raises the question of whether an algorithm whose running time is quasi-polynomial in n is possible even when the aspect ratio is not polynomially bounded. Our second result shows that this is unlikely by establishing the NP-hardness of the k-cover problem. The aspect ratio in the NP-hardness construction is about 2^n. The metrics obtained are induced by weighted planar graphs, thus establishing the NP-hardness of the k-cover problem for this special case.

Our final result is that the k-cover problem is NP-hard in metrics of constant doubling dimension for a large enough constant. This result is somewhat surprising given the positive results of [9] for fixed dimensional geometric spaces—algorithmic results for such spaces often generalize to metrics of constant doubling dimension.

The rest of this article is organized as follows. In Sect. 2, we present our algorithm for the k-cover problem. In Sect. 3, we point out that our algorithmic result for the metric k-cover problem readily yields a randomized approximation algorithm that runs in time $n^{O(\log n \log \frac{n}{\epsilon})}$ and returns with high probability a k-cover whose cost is at most $(1+\epsilon)$ times the cost of the optimal k-cover. Notice that the running time does not depend on the aspect ratio of the input metric space. This approximation algorithm is obtained by applying a simple transformation (involving discretization) that reduces the approximate problem to several instances of the exact metric κ-cover problem with aspect ratio bounded by $\mathrm{poly}(n/\epsilon)$. In Sect. 4, we establish the NP-hardness of the k-cover problem for metrics induced by weighted planar graphs. In

Sect. 5, we establish NP-hardness for metrics of constant doubling dimension. We conclude in Sect. 6 with some directions for future work.

2 Algorithm for General Metrics

We consider the k-cover problem whose input is a metric (P,d), where P is a set of n points and d is a function giving the interpoint distances, and an integer $k>0$. We assume without loss of generality that the minimum interpoint distance is 1. Let Δ denote diam(P), the maximum interpoint distance. We present a randomized algorithm that runs in $n^{O(\log n \log \Delta)}$ time and with high probability returns the best k-cover for P. We will assume below that $k \le n$.

The main idea for handling the metric case is that probabilistic partitions [2, 7] can play a role analogous to the line separators were used in the geometric case [9]. To formalize this, let Q denote some subset of P such that $|Q| \ge 2$, and consider the following randomized algorithm (taken from [7]) that partitions Q into sets of diameter at most diam$(Q)/2$:

Algorithm 1 Partition(Q)

```
Require: A subset $Q \subseteq P$, with $|Q| \ge 2$.
Ensure: A partition of $Q$ into $\{Q_1, Q_2, \ldots, Q_\tau\}$ such that $\mathrm{diam}(Q_i) \le \mathrm{diam}(Q)/2$, $1 \le i \le \tau$.
Let $\pi$ denote a random permutation of the points in $Q$.
Let $\beta$ denote a random number in the range $[\mathrm{diam}(Q)/8, \mathrm{diam}(Q)/4]$.
Let $R \leftarrow Q$.
for all $i \leftarrow 1$ to $|Q|$ do
    Let $Q_i \leftarrow \{p \in R | d(p, \pi(i)) \le \beta\}$.
    Let $R \leftarrow R \setminus Q_i$.
```

Since each Q_i is contained in a ball of radius at most diam$(Q)/4$, we have that diam$(Q_i) \le$ diam$(Q)/2$. Clearly, the Q_i also partition Q. Let us say that a ball $B \subseteq P$ is *cut* by this partition of Q if there are two distinct indices i and j such that $(B \cap Q) \cap Q_i \ne \emptyset$ and $(B \cap Q) \cap Q_j \ne \emptyset$. The main property that the probabilistic partition enjoys is encapsulated by the following lemma, whose proof follows via the methods of Fakcharoenphol et al. [7].

Lemma 1 *Let $B \subseteq P$ be some ball of radius r. The probability that B is cut by the partition of Q output by Partition(Q) is at most* $\frac{16r}{\mathrm{diam}(Q)} \cdot (1 + \log |Q|)$.

Proof Let $q_1, \ldots q_{|Q|}$ denote the ordering of the points in Q according to increasing order of distance from $B' = B \cap Q$, with ties broken arbitrarily. We may assume that $B' \ne \emptyset$ for otherwise the lemma trivially holds. For each q_j let a_j (resp. b_j) denote the distance to the closest (resp. furthest) point in B'. By the triangle inequality it follows that $b_j - a_j \le 2r$. We say that $\pi(i)$ *settles* B if i is the first index for which some point in B' belongs to Q_i. Note that exactly one point in Q settles B. We say that $\pi(i)$ *cuts B* if $\pi(i)$ settles B and at least one point in B' is not assigned to Q_i. The probability that B is cut by the partition equals

$$\sum_i \Pr[\pi(i) \text{ cuts } B] = \sum_j \Pr[q_j \text{ cuts } B].$$

The event that q_j cuts B requires the occurrence of two events: E_1, the event that β lands in the interval $[a_j, b_j)$, and E_2, the event that q_j appears before $q_1, \dots, q_{j-1}$ in the ordering π. Using independence,

$$\begin{aligned}\Pr[q_j \text{ cuts } B] &\le \Pr[E_1]\cdot \Pr[E_2 | E_1] = \Pr[E_1]\cdot \Pr[E_2]\\ &\le \frac{2r}{\operatorname{diam}(Q)/8}\cdot\frac{1}{j} = \frac{16r}{\operatorname{diam}(Q)}\cdot\frac{1}{j}.\end{aligned}$$

So the probability that B is cut by the partition is bounded above by

$$\frac{16r}{\operatorname{diam}(Q)}\sum_j \frac{1}{j} \le \frac{16r}{\operatorname{diam}(Q)}\cdot(1+\log|Q|),$$

where the last inequality follows from the fact that $\sum_{j=1}^{|Q|}\frac{1}{j} \le 1 + \log|Q|$. □

Let S denote the optimal κ-cover for Q some $\kappa > 0$. The following lemma states the main structural property that S enjoys.

Lemma 2 *The expected number of balls in S that are cut by Partition(Q) is at most $c_0\cdot \log|Q|$, where $0 < c_0 \le 32$ is a constant. Consequently, the probability is at least $1/2$ that the number of balls in S that are cut by Partition(Q) is at most $c\log n$, where $c = 2\cdot c_0$.*

Proof The expected number of balls in S cut is equal to

$$\sum_{B\in S}\Pr[B \text{ is cut}] \le 16\cdot(1+\log|Q|)\sum_{B\in S}\frac{\operatorname{radius}(B)}{\operatorname{diam}(Q)} = 16\cdot(1+\log|Q|)\frac{\operatorname{cost}(S)}{\operatorname{diam}(Q)}.$$

Observe that $\operatorname{cost}(S) \le \operatorname{diam}(Q)$ since Q can be covered by a single ball of radius $\operatorname{diam}(Q)$. So,

$$\mathbb{E}\left[{\#\text{ of balls in } S \atop \text{that are cut}}\right] \le 16\cdot(1+\log|Q|) \le c_0\cdot\log|Q| \le c_0\cdot\log n.$$

In the penultimate inequality, we may assume $c_0 \le 32$ since $1+\log|Q| \le 2\log|Q|$, which in turn follows because $|Q| \ge 2$.

By Markov's inequality, $\Pr\left[{\text{more than } 2\cdot c_0\cdot\log n \atop \text{balls in } S \text{ are cut}}\right] \le \frac{1}{2}$. □

The Randomized Algorithm

We describe a recursive algorithm `BC-Compute` that takes as arguments a set $Q \subseteq P$ and an integer $0 \le \kappa \le n$ and returns with high probability an optimal κ-cover for Q. We begin by noting that we may restrict our attention to balls $B(x, r)$

Algorithm 2 `BC-Compute`(Q, κ)

```
Require: A subset $Q \subseteq P$, and an integer $0 \le \kappa \le k$.
Ensure: Return a $\kappa$-cover of $Q$ having optimal cost, with high probability.
If $|Q| = 0$, return the empty set.
Otherwise, if $\kappa = 0$, return $\{\mathcal{I}\}$ (not possible to cover).
Otherwise, if $|Q| = 1$, return the singleton set consisting of the ball centered at
  the unique point in $Q$ and having radius 0.
for all $2\log_2 n$ iterations do
    Call Partition($Q$) to obtain a partition of $Q$ into two or more non-empty
      sets. Let $Q_1, \dots, Q_\tau$ denote the nonempty sets in this collection.
    for all sets $C$ of at most $c \log n$ balls, where $c$ is the constant in Lemma 2 do
        Let $Q'_i$ be the points in $Q_i$ not covered by $C$. For each $1 \le i \le \tau$ and $0 \le$
          $\kappa_1 \le \kappa$, recursively call BC-Compute($Q'_i, \kappa_1$) and store the set returned in
          the local variable best($Q'_i, \kappa_1$).
        For $0 \le i \le \tau - 1$, let $R_i = \bigcup_{j=i+1}^{\tau} Q'_j$. Note that $R_{\tau-1} = Q'_\tau$ and $R_i =$
          $Q'_{i+1} \cup R_{i+1}$ for $0 \le i \le \tau - 2$.
        for all $i \leftarrow \tau - 2$ down to 0 and $0 \le \kappa_1 \le \kappa$, do
            set local variable best($R_i, \kappa_1$) to be the lowest cost solution among
              $\{\text{best}(Q'_{i+1}, \kappa') \cup \text{best}(R_{i+1}, \kappa_1 - \kappa') | 0 \le \kappa' \le \kappa_1\}$.
            Let $S$ denote the lowest cost solution best($R_0, \kappa - |C|$) $\cup$ $C$ over all
              choices of $C$ tried above with $|C| \le \kappa$.
Return the lowest cost solution $S$ obtained over the $\Theta(\log n)$ trials.
```

whose radius r equals $d(x, q)$ for some $q \in P$. Henceforth in this section we only refer to this set of balls. For easing the description of the algorithm, it is convenient to add to this set of balls an element $\mathcal{I}$ whose cost is ∞. Any subset of this enlarged set of balls that includes $\mathcal{I}$ will also have a cost of ∞.

Running Time To solve an instance (Q, κ) of the problem with $\text{diam}(Q) \ge 50$, the algorithm makes $n^{O(\log n)}$ recursive calls to instances with diameter at most $\text{diam}(Q)/2$. The additional book keeping takes $n^{O(\log n)}$ time. It follows that the running time of the algorithm invoked on the original instance (P, k) is $n^{O(\log n \cdot \log \Delta)}$.

Correctness We will show that `BC-Compute`(P, k) computes an optimal k-cover for P with high probability. We begin by noting that the base case instances (Q, κ) are solved correctly with a probability of 1. We will show by induction on $|Q|$ that any instance (Q, κ) with $|Q| \ge 2$ is optimally solved with a probability of at least $1 - \frac{|Q|-1}{n^2}$.

If the (Q, κ) instance happens to fit in one of the base cases, we are done. Otherwise, consider an optimal κ-cover OPT for Q. It is enough to show that `BC-Compute`(Q, κ) returns a κ-cover of optimal cost with a probability of at least $1 - \frac{|Q|-1}{n^2}$.

By Lemma 2, the probability is at least $1 - \frac{1}{n^2}$ that one of the $2\log_2 n$ calls to `Partition`(Q) returns a partition $(Q_1, \dots, Q_\tau)$ of Q into $\tau \ge 2$ sets such that no more than $c \log n$ balls in OPT are cut by the partition. Assuming this good event happens, fix such a partition $(Q_1, \ldots, Q_\tau)$ of Q and consider the choice of C that exactly equals the balls in OPT that are cut by the partition. The balls in OPT $\setminus C$ are not cut by the partition and can be partitioned into subsets $(\mathrm{OPT}_1, \ldots, \mathrm{OPT}_\tau)$ (some of these can be empty) such that for any ball $B \in \mathrm{OPT}_i$, we have $B \cap Q \subseteq Q_i$. It is easy to see that OPT_i must be an optimal $|\mathrm{OPT}_i|$-cover for Q'_i. By the induction hypothesis, BC-Compute(Q'_i, $|\mathrm{OPT}_i|$) returns an optimal $|\mathrm{OPT}_i|$-cover for Q'_i with a probability of at least $1 - \frac{|Q'_i|-1}{n^2}$ if $|Q'_i| \geq 2$ and with a probability of 1 otherwise. The probability that BC-Compute(Q'_i, $|\mathrm{OPT}_i|$) returns an optimal $|\mathrm{OPT}_i|$-cover for Q'_i for every i is at least

$$\prod_{i:|Q'_i|\geq 2} 1 - \frac{|Q'_i|-1}{n^2} \geq \prod_i 1 - \frac{|Q_i|-1}{n^2} \geq 1 - \frac{|Q|-2}{n^2}.$$

Assuming this second good event also happens, it follows from an easy backwards induction on i that $\mathrm{best}(R_i, \sum_{j>i} |\mathrm{OPT}_j|)$ is a $(\sum_{j>i} |\mathrm{OPT}|_j)$-cover for R_i with cost at most $\sum_{j>i} \mathrm{cost}(\mathrm{OPT}_j)$. Thus $\mathrm{best}(R_0, \kappa - |C|)$ is an $(\kappa - |C|)$-cover for $R_0 = \sum_{i=1}^{\tau} Q'_i$ with cost at most $\sum_{i=1}^{\tau} \mathrm{cost}(\mathrm{OPT}_i)$. Thus $\mathrm{best}(R_0, \kappa - |C|) \cup C$ is a κ-cover of Q with cost at most cost(OPT). The probability of this happening is at least the product of the probabilities of the two good events we assumed, which is at least $(1 - \frac{|Q|-1}{n^2})$. This completes the inductive step, because BC-Compute(Q, κ) returns the lowest cost κ-cover among the $2\log_2 n$ κ-covers that it sees.

Theorem 3 *There is a randomized algorithm that, given a set P of n points in a metric space and an integer k, runs in $n^{O(\log n \cdot \log \Delta)}$ time and returns, with probability at least $1/2$, an optimal k-cover for P. Here Δ is an upper bound on the ratio between the maximum and minimum interpoint distances within P.*

3 Approximation in Quasi-Polynomial Time

In this section, we describe how the result of the previous section can be used to obtain a quasi-polynomial time approximation scheme for the k-cover problem. That is, we describe a randomized algorithm that takes as input a set P of n points in a metric space, an integer k, and a parameter $0 < \epsilon < 1$, and returns a k-cover whose cost is, with probability at least 1/2, within a multiplicative factor of $(1 + \epsilon)$ of the optimal k-cover; the running time of the algorithm is $n^{O(\log n \log \frac{n}{\epsilon})}$. Since our algorithm is a rather standard way of reducing the problem to instances whose aspect ratio is bounded by a polynomial in $\frac{n}{\epsilon}$, we describe it here only for completeness and only sketch the proof of correctness. We assume without loss of generality that $1 \leq k \leq n$.

Let λ^* denote the cost of the optimal k-cover of P. We first obtain a crude approximation to λ^* by computing in polynomial time a 2-approximation λ to the optimal k-center cost for P [10]. Observe that $\frac{\lambda}{2} \leq \lambda^* \leq k\lambda$. We then compute a minimum spanning tree of P (under the input metric). Let $P_1, \ldots, P_\tau$ denote the connected components obtained by removing from the minimum spanning tree all edges of length strictly greater than $k\lambda$. Notice that the distance between any two points that are in different components is strictly larger than $k\lambda$. This implies that any ball in an optimal k-cover of P is contained within one of the P_i. The maximum inter-point distance within each P_i is at most $nk\lambda$.

For each P_i, and for each $1 \le k_1 \le k$, we compute an approximation best(P_i, k_1) to the optimal k_1-cover of P_i as described below. Compute a minimal subset $Q_i \subseteq P_i$ with the property that for each $p \in P_i$, there is a $q \in Q_i$ such that $d(p,q) < \frac{\epsilon\lambda}{8n^2}$. The aspect ratio of Q_i is bounded by a polynomial in $\frac{n}{\epsilon}$. Run the algorithm of the previous section $O(\log n)$ times with Q_i and k_1, and take the minimum cost solution. This is, with probability at least $1 - \frac{1}{2n^2}$, the optimal k_1-cover of Q_i. Expand the radii of each of these balls by $\frac{\epsilon\lambda}{8n^2}$ to obtain a k_1-cover best(P_i, k_1) of P_i. It is not hard to see that with probability at least $1 - \frac{1}{2n^2}$ the cost of best(P_i, k_1) exceeds the cost of the optimal k_1-cover of P_i by at most $\frac{\epsilon\lambda}{2n}$.

Let $R_1 = P_1$, and let $R_i = R_{i-1} \cup P_i$ for $2 \le i \le \tau$; note that $R_\tau = P$. Assign best(R_1, k_1) to best(P_1, k_1) for each $1 \le k_1 \le k$. For each $i = 2, \ldots, \tau$ and for each $i \le k_1 \le k$, set best(R_i, k_1) to be the min-cost solution among

$$\{\text{best}(R_{i-1}, j) \cup \text{best}(P_i, k_1 - j) \mid i - 1 \le j \le k_1 - 1\}.$$

We return best(R_τ, k) as our k-cover of P. It can be verified that with probability at least $1/2$ the cost of this k-cover is within an additive $\frac{\epsilon\lambda}{2} \le \epsilon\lambda^*$ of λ^*, the cost of an optimal k-cover.

Theorem 4 *There is a randomized algorithm that takes as input a set P of n points in a metric space, an integer k, and a parameter $0 < \epsilon < 1$, and returns, in $n^{O(\log n \log \frac{n}{\epsilon})}$ time, a k-cover for P whose cost is, with probability at least $1/2$, within a multiplicative factor of $(1+\epsilon)$ of the optimal k-cover.*

4 NP-Hardness of Min-Cost k-Cover

A natural question is whether there is a quasipolynomial time algorithm in n (that returns the exact optimum) for the case where the input metric has unbounded aspect ratio. This is unlikely to be the case because, as we show in this section, the general problem is NP-hard even in case of a planar metric. We give a reduction from a version of the planar 3-SAT problem—the *pn-planar* 3-SAT problem. This problem was shown to be NP-complete in [14]. Planar 3-SAT is defined as follows: Let $\Phi = (X, C)$ be an instance of 3SAT, with variable set $X = \{x_0, \ldots, x_{n-1}\}$ and clauses $C = \{c_1, \ldots, c_m\}$ such that each clause consists of exactly 3 literals. Define a *formula graph* $G_\Phi = (V, E)$ with vertex set $V = X \cup C$ and edges $E = E_1 \cup E_2$ where $E_1 = \{(x_i, x_{i+1}) | 0 \le i \le n-1\}$[1] and $E_2 = \{(x_i, c_j) | c_j$ contains x_i or $\overline{x_i}\}$. A 3SAT formula Φ is called *planar* if the corresponding formula graph G_Φ is planar. The edge set E_1 defines a cycle on the vertices X, and thus divides the plane into exactly 2 faces. Each node $c_j \in C$ lies in exactly one of those two faces. In the *pn-planar* 3SAT problem,

[1] Here we assume that the arithmetic wraps around i.e. $(n-1)+1 = 0$.

we have the additional restriction that there exists a planar drawing of G_Φ such that if c_j and $c_{j'}$ contain opposite occurrences of the same variable x_i, then they lie in opposite faces. In other words, all clauses with the literals x_i lie in one of the two faces and all clauses with $\overline{x_i}$ lie in the other face. We have to determine whether there exists an assignment of truth values to the variables in X that satisfies all the clauses in C.

We describe a simple transformation, easily seen to be effected by a polynomial time algorithm, from such a *pn-planar* 3SAT instance to an instance of the decision version of the k-cover problem, where in addition to the input metric and k, we are also given a target τ, and we wish to determine if there is a k-cover with cost at most τ. In the instance produced by our transformation, the metric is induced by a weighted planar graph $G = (V, E)$, and the target τ equals $2^k - 1$. The transformation has the property that there is a k-cover in the metric of cost at most $2^k - 1$ if and only if the original *pn-planar* 3SAT instance is satisfiable.

We set k to be n, the number of variables in the 3SAT instance. The vertex set V of the graph is a union of $k + 2$ sets: (a) a set $X = \{x_0, \overline{x_0}, \dots, x_{k-1}, \overline{x_{k-1}}\}$ that can be identified with the set of variables of the *pn-planar* 3SAT instance with each variable occurring twice—once as a positive literal and once as a negative literal, (b) a set $C = \{c_1, \dots, c_m\}$ that can be identified with the set of clauses of the *pn-planar* 3SAT instance, and (c) sets $W^0, \dots, W^{k-1}$, where each W^l consists of $k + 1$ vertices. To obtain the edge set E, we add an edge between each vertex x_l and $\overline{x_l}$ in X with weight 2^l for $0 \le l \le k - 1$. For each vertex $x_l \in X$ we add an edge between x_l and every vertex in W^l of weight 2^l for $0 \le l \le k - 1$. Analogously, we add an edge between each vertex $\overline{x_l}$ and every vertex in W^l again of weight 2^l. In addition we add edges between every vertex $c_i \in C$ and every variable vertex x_l or its negation $\overline{x_l}$ whichever appears in it of weight 2^l. Observe that this graph G is planar. It is crucial for this that in the planar drawing of the formula graph G_Φ, the clauses that contain the literal x_i lie in one face of the cycle induced by E_1, whereas the clauses that contain $\overline{x_i}$ lie in the opposite face, for each variable x_i. See Fig. 1 for an illustration.

Claim 5 *Any k-cover of V whose cost is at most $2^k - 1$ includes, for each $0 \le l \le k - 1$, a ball centered at either x_l or $\overline{x_l}$ with radius at least 2^l.*

Proof Consider any k-cover of V and let t be the largest index such that there is no ball in the k-cover centered at either x_t or $\overline{x_t}$ and having radius at least 2^t. So for each $t + 1 \le l \le k - 1$, there is a ball B_l in the k-cover centered at either x_l or $\overline{x_l}$ and having radius at least 2^l. Since W^t has $k + 1$ points in it, there is point $a \in W^t$ that is not the center of any ball in the k-cover. Let B be some ball in the k-cover that covers a. If $B = B_l$ for some $t + 1 \le l \le k - 1$, then B_l has radius at least $2^l + 2 \cdot 2^t$. In this case the k-cover has cost at least $2^{k-1} + 2^{k-2} \cdots 2^{t+1} + 2 \cdot 2^t = 2^k$. If $B \ne B_l$ for any $t + 1 \le l \le k - 1$, then the radius of B is at least $2 \cdot 2^t$, since the distance of a from any point other than x_t and $\overline{x_t}$ is at least $2 \cdot 2^t$. Thus in this case too the k-cover has cost at least $2^{k-1} + 2^{k-2} \cdots 2^{t+1} + 2 \cdot 2^t = 2^k$. □

Now suppose the original *pn-planar* 3SAT instance is a yes instance. So there is an assignment of truth values to $x_0, \dots, x_{k-1}$ such that all clauses in C are satisfied.

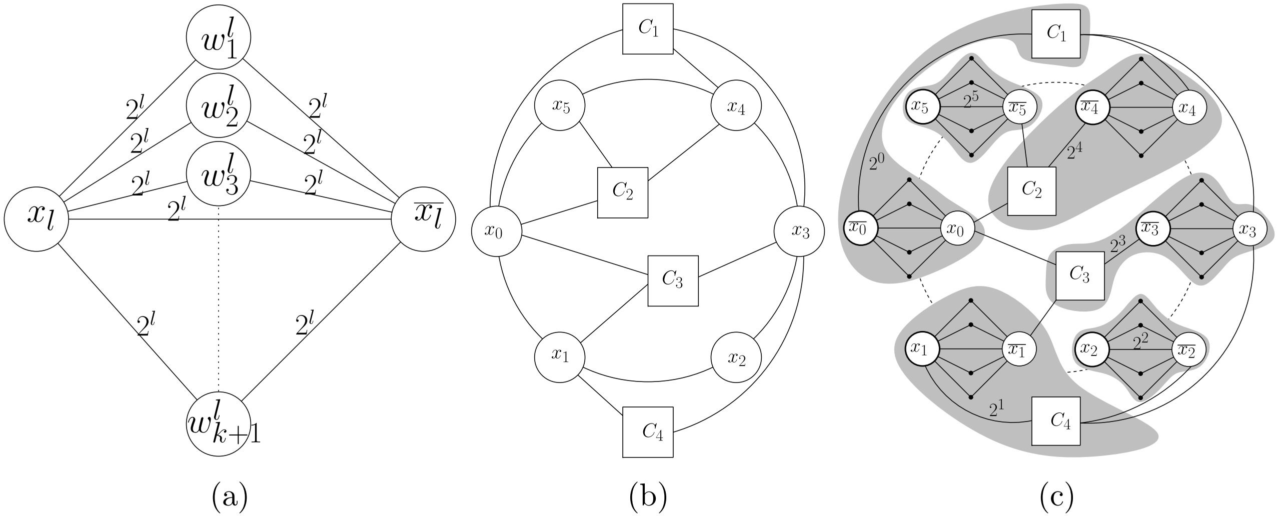

Fig. 1 **a** The gadget for variable x_l in Φ. **b** A planar embedding for $\Phi = (\neg x_0 \vee x_3 \vee x_4) \wedge (x_0 \vee \neg x_4 \vee \neg x_5) \wedge (x_0 \vee \neg x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)$. **c** Construction of the corresponding instance of k-clustering problem. All “clause-literal” edges have weight 2^l for the variable x_l. The optimal cover is highlighted with *grey “blobs”*. Satisfying assignment $X = (0, 1, 1, 0, 0, 1)$. Weight of the covering is exactly $2^6 - 1$

Consider the set of k balls $B_0, \dots, B_{k-1}$, where B_l is centered at x_l or $\overline{x_l}$ (whichever is satisfied by the assignment) and has radius 2^l. It is easily checked that these balls form a k-cover of V of cost $2^0 + 2^1 + \cdots + 2^{k-1} = 2^k - 1$.

Now suppose the original *pn-planar* 3SAT instance is a no instance. We claim that any k-cover of V has cost strictly greater than $2^k - 1$ in this case. Suppose this is not the case and consider a k-cover of cost at most $2^k - 1$. As a consequence of the claim, such a k-cover must consist of balls $B_0, \dots, B_{k-1}$ where B_l is centered at either x_l or $\overline{x_l}$ and has radius precisely 2^l. Since these balls must cover each vertex in C, it follows that the assignment of truth values to variables in X which comprises of x_l being true if the ball B_l is centered at x_l and false if it is centered at $\overline{x_l}$ satisfies all clauses in C. This contradicts the supposition that the original *pn-planar* 3SAT instance is a no instance.

Theorem 6 *The* (*decision version of the*) *problem of computing an optimal k-cover for an n-point planar metric* (P, d) *is NP-hard.*

5 The Doubling Metric Case

We now consider the k-cover problem when the input metric (P, d) has doubling dimension bounded by some constant $\rho \geq 0$. The doubling dimension of the metric (P, d) is said to be bounded by ρ if any ball $B(x, r)$ in (P, d) can be covered by 2^ρ balls of radius $r/2$ [12]. In this section, we show that for a large enough constant ρ, the k-cover problem for metrics of doubling dimension at most ρ is NP-hard.

The proof is by a reduction from a restricted version of 3SAT where each variable appears in at most 5 clauses [8]. Let Φ be such a 3-CNF formula with variables $x_0, \dots, x_{n-1}$ and clauses $c_1, \dots, c_m$. We describe a simple transformation, easily seen to be effected by a polynomial time algorithm, from such a 3SAT instance Φ to an instance of the decision version of the k-cover in a metric induced by a weighted graph $G = (V, E)$, and with the target cost being $2^k - 1$. The metric will have doubling dimension bounded by some constant. The transformation has the property that there is a k-cover in the metric of cost at most $2^k - 1$ if and only if the original 3SAT instance is satisfiable.

The transformation is similar to the one in the previous section with some modifications to ensure the doubling dimension property.

We set $k = n$, the number of variables in the 3SAT formula. The vertex set V of the graph is a union of $k + 2$ sets: (a) a set $X = \{x_0, \overline{x_0}, \dots, x_{k-1}, \overline{x_{k-1}}\}$ that can be identified with the set of literals in Φ, (b) a set $C = \{c_1, \dots, c_m\}$ that can be identified with the set of clauses of Φ, and (c) sets $W^0, \dots, W^{k-1}$, where each W^l consists of $n_l = 8(l + 1)^2 + 1$ vertices $w_1^l, \dots, w_{n_l}^l$. To obtain the edge set E, we add an edge between x_l and $\overline{x_l}$ with weight 2^l for $0 \leq l \leq k - 1$. We add an edge between x_l and every vertex in W^l of weight 2^l for $0 \leq l \leq k - 1$. Analogously, we add an edge between $\overline{x_l}$ and every vertex in W^l again of weight 2^l. In addition we add edges between every vertex $c_i \in C$ and every literal that appears in the clause c_i. If the literal is either x_l or $\overline{x_l}$, the weight of the corresponding edge is 2^l. Finally for each $0 \leq l \leq n - 1$ and each $1 \leq i \leq n_l - 1$, we add an edge of weight $2^l/(l + 1)^2$ between w_i^l and w_{i+1}^l. See Fig. 2 for an illustration of the transformation.

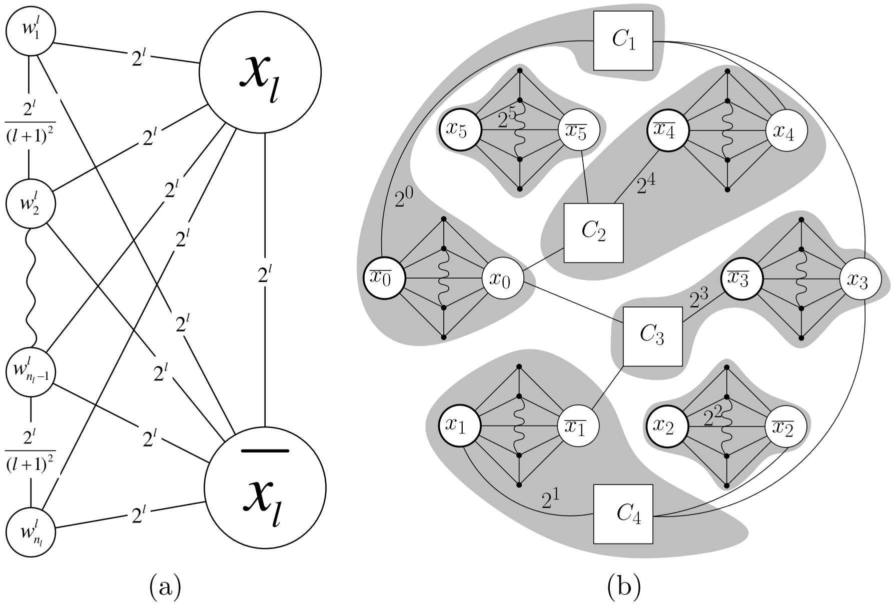

Fig. 2 **a** The gadget for the variable x_l in Φ. Each edge between w_i^l and w_{i+1}^l has weight $2^l/(l+1)^2$ and the number of w_i^l's is $8(l+1)^2+1$. **b** A representation of an instance of k-clustering on a doubling metric constructed from an instance of $\Phi = (\neg x_0 \vee x_3 \vee x_4) \wedge (x_0 \vee \neg x_4 \vee \neg x_5) \wedge (x_0 \vee \neg x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)$. Satisfying assignment $X = (0,1,1,0,0,1)$. All "clause-literal" edges have weight 2^l for variable x_l. The optimal cover is highlighted with *grey "blobs"*. Weight of the covering is 2^6-1

Lemma 7 *There is a constant $\rho \geq 0$ so that the doubling dimension of the metric induced by the graph $G=(V,E)$ is bounded by ρ.*

Proof Let $B(x,r)$ be some ball in the metric. If $r<1$, then either (a) the ball consists of a singleton vertex, or (b) $B(x,r) \subseteq W^l$ for some l and the subgraph of G induced by $B(x,r)$ is a path. In either case, it is easily verified that $O(1)$ balls centered within $B(x,r)$ and having radius $r/2$ cover $B(x,r)$.

We therefore consider the case $r \geq 1$. Let t be the largest integer that is at most $n-1$ such that $2^t \leq r$. For each $s \in \{t-3, t-2, t-1, t\}$, we place balls of radius $r/2$ centered at (i) $\{x_s, \overline{x_s}\} \cap B(x,r)$; (ii) clause vertices incident to x_s or $\overline{x_s}$ that are in $B(x,r)$; (iii) $O(1)$ points of $B(x,r) \cap W^s$ so that these balls cover $B(x,r) \cap W^s$ (this is possible because $B(x,r) \cap W^s$ induces a path of length at most 2^{s+3}). In addition, if $x \in W^l$ for some l, we place (iv) $O(1)$ balls of radius $r/2$ at points of $B(x,r) \cap W^l$ so that these balls cover $B(x,r) \cap W^l$. Finally, we place (v) a ball of radius $r/2$ at x. Observe that we have placed $O(1)$ balls and we will show that these cover $B(x,r)$. (Note that the assumption that each variable in the input formula appears in at most 5 clauses is used in concluding that the number of balls placed in (ii) is $O(1)$.) Let C denote the set of centers at which we have placed balls.

Let $y \in B(x,r)$ be a point that is not in C or in W^s for $s \in \{t-3, t-2, t-1, t\}$ or in W^l (if $x \in W^l$). Fix a shortest path from x to y and let x' be the last vertex on this path that is in C. We first prove that none of the internal vertices on the path from x to y is in W^q for any q.

Claim 8 *If both x and y do not belong to the same W^l, then no internal vertex w along the shortest path from x to y can belong to any W^q.*

Proof For the sake of contradiction, assume the contrary. Let w be the last internal vertex along the shortest path that is part of some W^q. We will prove that such a path cannot be of minimum length, deriving a contradiction. We examine two cases: (a) $y \in W^q$, and (b) $y \notin W^q$.

(a) Since $x \notin W^q$, there is an ancestor of w along the shortest path that is one of $\{x_q, \overline{x_q}\}$. Without loss of generality, let x_q be this ancestor. So, the shortest path pays a cost of at least $2^q + \frac{2^q}{(q+1)^2}$ to get to y from x_q. But, by construction, there is an edge $\{x_q, y\}$ having cost 2^q and we get a cheaper shortest path. A contradiction.

(b) Clearly there is a decendent from w along the shortest path that is one of $\{x_q, \overline{x_q}\}$. Without loss of generality, let it be x_q. Whether $x \in W^q$ or not, the shortest path pays of cost of at least $2^q + \frac{2^q}{(q+1)^2}$ to get to x_q from the immediate predecessor of w. However, in either case, there is an edge directly from this predecessor of w to x_q of cost 2^q, and we get a cheaper shortest path. A contradiction. □

Furthermore, if $x \in W^l$ for some l, then by assumption $y \notin W^l$. Thus all edges of the subpath from x' to y have weight 2^q for some $0 \le q \le n-1$. No such edge can have weight 2^{t+1} or greater because $2^{t+1} > r$ if $t \le n-2$. No such edge can have weight 2^s for $s \in \{t-3, t-2, t-1, t\}$ because otherwise the endpoint of the edge closer to y would be in C. Thus every edge on the subpath from x' to y has weight at most 2^{t-4}. It is easy to see that the subpath contains at most 3 edges of weight 2^q for any $q \le t-4$. Thus the weight of the subpath from x' to y is at most

$$3(2^{t-4} + 2^{t-5} + \cdots + 2^0) < 3 \cdot 2^{t-3} < 2^{t-1} < r/2.$$

So y is in the ball of radius $r/2$ centered at x'. □

Claim 9 *Any k-cover of V whose cost is at most $2^k - 1$ includes, for each $0 \le l \le k-1$, a ball centered at either x_l or $\overline{x_l}$ with radius at least 2^l.*

Proof Consider any k-cover of V and let t be the largest index such that there is no ball in the k-cover centered at either x_t or $\overline{x_t}$ and having radius at least 2^t. So for each $t+1 \le l \le k-1$, there is a ball B_l in the k-cover centered at either x_l or $\overline{x_l}$ and having radius at least 2^l.

If some point in W^t is covered by some B_l for $t+1 \le l \le k-1$, then B_l has radius at least $2^l + 2 \cdot 2^t$. In this case the k-cover has cost at least $2^{k-1} + 2^{k-2} \cdots 2^{t+1} + 2 \cdot 2^t = 2^k$. If some point in W^t is covered by a ball B different from the B_l's and not centered at any of the points in W^t, then the radius of B is at least $2 \cdot 2^t$. (Note that by assumption B can't be centered at x_t or $\overline{x_t}$.) Thus in this case too the k-cover has cost at least $2^{k-1} + 2^{k-2} \cdots 2^{t+1} + 2 \cdot 2^t = 2^k$.

The only remaining case is when each point in W^t is covered by some ball centered at a point in W^t. Since there can be at most $t+1$ balls in the k-cover centered within W^t, the sum of the radii of these balls is at least

$$\frac{1}{2}\left((n_t - 1)\frac{2^t}{(t+1)^2} - (t+1)\frac{2^t}{(t+1)^2}\right) > 2 \cdot 2^t.$$

The k-cover has cost at least $2^{k-1} + 2^{k-2} \cdots 2^{t+1} + 2 \cdot 2^t = 2^k$. □

We now argue that the transformation has the property that there is a k-cover in the metric of cost at most $2^k - 1$ if and only if the original 3SAT instance Φ is satisfiable.

Suppose that Φ is satisfiable. Then we can choose for each $0 \le l \le k-1$ exactly one of x_l or $\overline{x_l}$ such that within each clause of Φ there is a chosen literal. Consider the set of k balls $B_0, \ldots, B_{k-1}$ where B_l has radius 2^l and is centered at x_l or $\overline{x_l}$, whichever was chosen. These balls form a k-cover of V with cost $2^k - 1$.

For the reverse direction, consider a k-cover of the target metric space of cost at most $2^k - 1$. It follows from Claim 9 that the k-cover must consist of balls $B_0, \ldots, B_{k-1}$, where B_l is centered at either x_l or $\overline{x_l}$ and has radius precisely 2^l. Let us choose the literals corresponding to the centers of these balls. For each l, we clearly choose exactly one of x_l of $\overline{x_l}$. Consider any clause vertex c. It must be covered by at least one of the balls B_l. Given the radii of the balls, the only balls that can cover c are the ones centered at literals contained in the clause. It follows that our set of chosen literals contains, for each clause in Φ, at least one of the literals contained in the clause. Thus Φ is satisfiable.

Theorem 10 *For a large enough constant $\rho \ge 0$, the (decision version of the) k-cover problem for metrics of doubling dimension at most ρ is NP-hard.*

6 Future Work

We conclude with a short discussion on several tantalizing questions that remain open. A comparison of the positive and negative results (in Sects. 2 and 4, respectively) shows that the aspect ratio plays a crucial role in making the problem NP-hard. As a result, the existence of an exact, polynomial-time algorithm for the metric induced by an unweighted graph has not been ruled out. It would be interesting to develop such an algorithm. It would also be interesting to generalize this to an algorithm whose running time is polynomial in the aspect ratio and in the number of input points, for general metrics. On the question of approximation, our quasi-PTAS (in Sect. 3) raises the question of whether a PTAS for the k-cover problem is possible.

Acknowledgements We thank Chandra Chekuri for his suggestion to study the problem and the anonymous reviewers for their helpful comments.

References

1. Alt, H., Arkin, E.M., Brönnimann, H., Erickson, J., Fekete, S.P., Knauer, C., Lenchner, J., Mitchell, J.S.B., Whittlesey, K.: Minimum-cost coverage of point sets by disks. In: Amenta, N., Cheong, O. (eds.) Symposium on Computational Geometry, pp. 449–458. ACM, New York (2006)

2. Bartal, Y.: Probabilistic approximations of metric spaces and its algorithmic applications. In: FOCS, pp. 184–193 (1996)
3. Bilò, V., Caragiannis, I., Kaklamanis, C., Kanellopoulos, P.: Geometric clustering to minimize the sum of cluster sizes. In: Stølting Brodal, G., Leonardi, S. (eds.) ESA. Lecture Notes in Computer Science, vol. 3669, pp. 460–471. Springer, Berlin (2005)
4. Charikar, M., Panigrahy, R.: Clustering to minimize the sum of cluster diameters. J. Comput. Syst. Sci. **68**(2), 417–441 (2004)
5. Doddi, S., Marathe, M.V., Ravi, S.S., Taylor, D.S., Widmayer, P.: Approximation algorithms for clustering to minimize the sum of diameters. Nord. J. Comput. **7**(3), 185–203 (2000)
6. Fernandez de la Vega, W., Kenyon, C.: A randomized approximation scheme for metric max-cut. J. Comput. Syst. Sci. **63**(4), 531–541 (2001)
7. Fakcharoenphol, J., Rao, S., Talwar, K.: A tight bound on approximating arbitrary metrics by tree metrics. In: STOC, pp. 448–455. ACM, New York (2003)
8. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
9. Gibson, M., Kanade, G., Krohn, E., Pirwani, I.A., Varadarajan, K.: On clustering to minimize the sum of radii. In: SODA, pp. 819–825. SIAM, Philadelphia (2008)
10. Hochbaum, D.S., Shmoys, D.B.: A best possible heuristic for the k-center problem. Math. Oper. Res. **10**, 180–184 (1985)
11. Kariv, O., Hakimi, S.L.: An algorithmic approach to network location problems. Part II: The p-medians. SIAM J. Appl. Math. **37**, 539–560 (1982)
12. Krauthgamer, R., Lee, J.R.: Navigating nets: simple algorithms for proximity search. In: Munro, J.I. (ed.) SODA, pp. 798–807. SIAM, Philadelphia (2004)
13. Lev-Tov, N., Peleg, D.: Polynomial time approximation schemes for base station coverage with minimum total radii. Comput. Netw. **47**(4), 489–501 (2005)
14. Lichtenstein, D.: Planar formulae and their uses. SIAM J. Comput. **11**(2), 329–343 (1982)
15. Manasse, M.S., McGeoch, L.A., Sleator, D.D.: Competitive algorithms for server problems. J. Algorithms **11**(2), 208–230 (1990)
16. Schaeffer, S.E.: Graph clustering. Comput. Sci. Rev. **1**(1), 27–64 (2007)