




# On the Budgeted Priority $p$ -Median Problem in High-Dimensional Euclidean Spaces

Zhen Zhang<sup>1,2</sup> · Zi-Yun Huang<sup>3</sup> · Zhi-Ping Tian<sup>1,2</sup> · Li-Mei Liu<sup>1,2</sup>  · Xue-Song Xu<sup>1,2</sup> · Qi-Long Feng<sup>2,4</sup>

Received: 4 September 2023 / Revised: 20 November 2023 / Accepted: 24 November 2023  
© Operations Research Society of China, Periodicals Agency of Shanghai University, Science Press, and Springer-Verlag GmbH Germany, part of Springer Nature 2024

## Abstract

Given a set of clients and a set of facilities with different priority levels in a metric space, the BUDGETED PRIORITY  $p$ -MEDIAN problem aims to open a subset of facilities and connect each client to an opened facility with the same or a higher priority level, such that the number of opened facilities associated with each priority level is no more than a given upper limit, and the sum of the client-connection costs is minimized.

---

This work was supported by the National Natural Science Foundation of China (Nos. 62202161, 62172446, and 62376092), Natural Science Foundation of Hunan Province (No. 2023JJ40240), and Open Project of Xiangjiang Laboratory (Nos. 22XJ02002 and 22XJ03005).

---

✉ Li-Mei Liu  
seagullm@163.com

Zhen Zhang  
csuzz@foxmail.com

Zi-Yun Huang  
zxh201@psu.edu

Zhi-Ping Tian  
tian\_hutb@163.com

Xue-Song Xu  
dannyxu@hnuc.edu.cn

Qi-Long Feng  
csufeng@mail.csu.edu.cn

- <sup>1</sup> School of Advanced Interdisciplinary Studies, Hunan University of Technology and Business, Changsha 410205, Hunan, China
- <sup>2</sup> Xiangjiang Laboratory, Changsha 410205, Hunan, China
- <sup>3</sup> Department of Computer Science and Software Engineering, Penn State Erie, The Behrend College, Erie, PA 16563, USA
- <sup>4</sup> School of Computer Science and Engineering, Central South University, Changsha 410012, Hunan, China

In this paper, we present a data reduction-based approach for limiting the solution search space of the BUDGETED PRIORITY  $p$ -MEDIAN problem, which yields a  $(1 + \varepsilon)$ -approximation algorithm running in  $O(nd \log n) + (p\varepsilon^{-1})^{p\varepsilon^{-O(1)}} n^{O(1)}$  time in  $d$ -dimensional Euclidean space, where  $n$  is the size of the input instance and  $p$  is the maximal number of opened facilities. The previous best approximation ratio for this problem obtained in the same time is  $(3 + \varepsilon)$ .

**Keywords** Approximation algorithms · Facility location ·  $p$ -Median

**Mathematics Subject Classification** 90C27 · 68Q27 · 68Q25

## 1 Introduction

Locating facilities to serve a given set of clients is a fundamental problem in the field of operational research. This problem becomes difficult when considering the following trade-off: Each client should be served by a neighboring facility, while we cannot open too many facilities.

The  $p$ -MEDIAN problem is one of the most commonly studied facility location-type problems, which balances the client-connection and facility-opening costs by limiting the number of opened facilities and minimizing the sum of distances of the clients from the corresponding facilities. It was known that the  $p$ -MEDIAN problem is NP-hard [1, 2], and considerable attention has been paid on developing approximation algorithms for it. The current best known approximation guarantees for the problem are the ratios of 2.67059 [3] and 2.406 [4] in general and Euclidean metrics, respectively.

The  $p$ -MEDIAN problem assumes that each client can be connected to an arbitrary opened facility, and thus, its algorithms fail to work in applications where clients differ in levels of required services. One such example is in the construction of supply chain networks, where the algorithms for the  $p$ -MEDIAN problem are not guaranteed to yield feasible solutions since they tend to minimize the supply chain cost without concerning the target service levels of the chain members. Motivated by such applications, Kumar and Sabharwal [5] introduced the BUDGETED PRIORITY  $p$ -MEDIAN problem, which can be formally defined as follows.

**Definition 1** (BUDGETED PRIORITY  $p$ -MEDIAN [5]) An instance  $(\mathcal{C}, \mathcal{F}, \mathcal{L}, \mathcal{P}, \ell)$  of the BUDGETED PRIORITY  $p$ -MEDIAN problem is specified by a set  $\mathcal{C}$  of clients, a set  $\mathcal{F}$  of facilities, a set  $\mathcal{L} = \{1, \dots, |\mathcal{L}|\}$  of service levels (or called priority levels), and a set  $\mathcal{P} = \{p_1, \dots, p_{|\mathcal{L}|}\}$  of real numbers no less than 1, where each  $x \in \mathcal{C} \cup \mathcal{F}$  is located in a metric space and associated with a service level  $\ell(x) \in \mathcal{L}$ . A feasible solution  $(\mathcal{D}, \varphi)$  to this instance opens a subset  $\mathcal{D} \subseteq \mathcal{F}$  of facilities satisfying  $|\{f \in \mathcal{D} : \ell(f) = i\}| \leq p_i$  for each  $i \in \mathcal{L}$  and connects each client  $c \in \mathcal{C}$  to an opened facility  $\varphi(c) \in \mathcal{D}$  satisfying  $\ell(c) \leq \ell(\varphi(c))$ . The cost of such a solution is  $\sum_{c \in \mathcal{C}} \delta(c, \varphi(c))$ , where  $\delta(c, \varphi(c))$  denotes the distance of  $c$  from  $\varphi(c)$ , and the BUDGETED PRIORITY  $p$ -MEDIAN problem aims to identify a feasible solution with minimal cost.

Despite its important role in applications involving different service levels, the BUDGETED PRIORITY  $p$ -MEDIAN problem is far less understood from the theoretical

point of view compared with the standard  $p$ -MEDIAN problem (the latter is a special case of the BUDGETED PRIORITY  $p$ -MEDIAN problem where all the clients and facilities are associated with the same service level). Kumar and Sabharwal [5] showed that the natural linear programming relaxation of the BUDGETED PRIORITY  $p$ -MEDIAN problem has an unbounded integrality gap if the number of service levels is more than four, and gave a constant-factor approximation algorithm for instances with no more than two service levels. Without such a limitation on the number of service levels, whether the BUDGETED PRIORITY  $p$ -MEDIAN problem admits an  $O(1)$ -approximation algorithm still remains elusive.

In most real-world applications of facility location-type problems, the facilities providing services are much less than the clients with service requirements. Thus, taking the number of opened facilities as a fixed parameter is a feasible way for relaxing these problems in practice, as done in [6–8]. However, the BUDGETED PRIORITY  $p$ -MEDIAN problem has turned out to be still difficult when the maximal number of opened facilities (denoted by  $p$ ) is small: Reductions to the SET COVERING problem given by Guha and Khuller [1] indicate that the BUDGETED PRIORITY  $p$ -MEDIAN problem, even for the case where the service levels associated with the clients and facilities are uniform, is W[2]-hard if parameterized by  $p$ ; Cohen-Addad et al. [2] showed that the approximation ratio of any FPT( $p$ )-time (i.e.,  $g(p)n^{O(1)}$  time for an instance of size  $n$  and a positive function  $g$ ) algorithm for the problem cannot be better than  $(1 + 2e^{-1})$  under the Gap-Exponential Time Hypothesis. On the positive side, Feng et al. [9] gave a  $(3 + \varepsilon)$ -approximation algorithm for the BUDGETED PRIORITY  $p$ -MEDIAN problem running in  $(p\varepsilon^{-1})^{O(p)}n^{O(1)}$  time based on a greedy sampling approach, which significantly improves the performance guarantees of polynomial-time algorithms.

## 1.1 Our Results

The negative result given by Cohen-Addad et al. [2] implies that we are unlikely to approximate the BUDGETED PRIORITY  $p$ -MEDIAN problem with a ratio better than  $1 + 2e^{-1}$  in FPT time if taking  $p$  as the fixed parameter, but we can still hope to near-exactly solve the problem in Euclidean spaces using FPT-time algorithms since such a lower bound only holds in general metrics. Indeed, for continuous Euclidean instances where the services have only a single level and each facility can be opened at an arbitrary location of the space, a series of sampling-based  $(1 + \varepsilon)$ -approximation algorithms running in FPT time are known for the BUDGETED PRIORITY  $p$ -MEDIAN problem [10–12]. Unfortunately, without such limitations on the Euclidean instances, the current best FPT-time approximation guarantee for the problem is still the ratio of  $(3 + \varepsilon)$  given in [9], and whether the problem can be near-exactly solved in FPT time remains an open question. The main result in this paper affirmatively answers this question.

**Theorem 1** *Given a constant  $\varepsilon \in (0, 1)$  and an instance  $(\mathcal{C}, \mathcal{F}, \mathcal{L}, \mathcal{P}, \ell)$  of the BUDGETED PRIORITY  $p$ -MEDIAN problem with  $\mathcal{C} \cup \mathcal{F} \subset \mathbb{R}^d$ , there is a  $(1 + \varepsilon)$ -*

approximation algorithm running in  $O(nd \log n) + (p\varepsilon^{-1})^{p\varepsilon^{-O(1)}} n^{O(1)}$  time, where  $n = |\mathcal{C}| + |\mathcal{F}|$  and  $p = \sum_{i=1}^{|\mathcal{L}|} \lfloor p_i \rfloor$ .

## 1.2 Our Techniques

As mentioned above, an Euclidean instance  $(\mathcal{C}, \mathcal{F}, \mathcal{L}, \mathcal{P}, \ell)$  of the BUDGETED PRIORITY  $p$ -MEDIAN problem involving only a single service level (i.e.,  $|\mathcal{L}| = |\mathcal{P}| = 1$ ) has several known FPT-time  $(1 + \varepsilon)$ -approximation algorithms [10–12]. The main challenge encountered when removing such a limitation on the instance lies in the significantly increased difficulty of constructing feasible solutions: We need to simultaneously satisfy the different service requirements of the clients and the budgeted constraints defined by  $\mathcal{P}$ . This invalidates existing algorithms in Euclidean spaces.

In this paper, we deal with the BUDGETED PRIORITY  $p$ -MEDIAN problem based on a quite different data reduction-based method. Given an Euclidean instance of the BUDGETED PRIORITY  $p$ -MEDIAN problem, we show that a combination of an adapted coresets-construction approach and a Johnson–Lindenstrauss transform-type method can significantly reduce its size and map it into a low-dimensional Euclidean space. Based on such a reduced instance, we estimate the locations of the facilities opened in optimal solutions and in turn construct a small solution search space by exploring the properties of the Euclidean metrics. It is shown that enumerating over this search space yields the desired  $(1 + \varepsilon)$ -approximation solution in FPT time.

## 1.3 Preliminaries

Let  $\varepsilon$  denote a constant from  $(0, 1)$  and  $\mathcal{I} = (\mathcal{C}, \mathcal{F}, \mathcal{L}, \mathcal{P}, \ell)$  be an instance of the BUDGETED PRIORITY  $p$ -MEDIAN problem satisfying  $\sum_{i=1}^{|\mathcal{L}|} \lfloor p_i \rfloor = p$ ,  $|\mathcal{C}| + |\mathcal{F}| = n$ , and  $\mathcal{C} \cup \mathcal{F} \subset \mathbb{R}^d$ . Define  $[t] = \{1, \dots, t\}$  for each integer  $t \geq 1$ . For each  $i \in \mathcal{L}$ , let  $\mathcal{C}_i = \{c \in \mathcal{C} : \ell(c) = i\}$  and  $\mathcal{F}_i = \{f \in \mathcal{F} : \ell(f) = i\}$ , and define  $\mathcal{F}_i^+ = \bigcup_{j=i}^{|\mathcal{L}|} \mathcal{F}_j$ . Let  $(\mathcal{D}^*, \varphi^*)$  denote an optimal solution to  $\mathcal{I}$ . Without loss of generality, we can assume that  $|\mathcal{D}^*| = p$  and let  $\mathcal{D}^* = \{f_1^*, \dots, f_p^*\}$ . Let  $\ell_i^* = \ell(f_i^*)$  for each  $i \in [p]$ . Given two data points  $x, y$  and a set  $\mathcal{X}$  located in an Euclidean space, define  $\delta(x, y)$  as the distance of  $x$  from  $y$ , and let  $\delta(\mathcal{X}, y) = \sum_{z \in \mathcal{X}} \delta(z, y)$  and  $\delta(y, \mathcal{X}) = \min_{z \in \mathcal{X}} \delta(y, z)$ . Let  $\text{opt} = \sum_{c \in \mathcal{C}} \delta(c, \varphi^*(c))$  denote the cost of an optimal solution to  $\mathcal{I}$ .

The running time of our algorithm is analyzed based on the following algebraic fact.

**Lemma 1** *We have  $\log^j i \leq \max\{i, j^{O(j)}\}$  for each  $i, j > 1$ .*

**Proof** If  $j \geq \frac{\log i}{\log \log i}$ , then we have  $\log i \leq j^{O(1)}$  and hence  $\log^j i \leq j^{O(j)}$ . If

$j < \frac{\log i}{\log \log i}$ , then we have  $\log^j i < \log^{\frac{\log i}{\log \log i}} i = i$ . Thus, Lemma 1 is true.

Our algorithm constructs nets for the facilities to identify a small solution search space. Such nets can be formally defined as follows.

**Definition 2** ( $\lambda$ -net [13]) Given a set  $\mathcal{X} \subset \mathbb{R}^d$  and a real number  $\lambda > 0$ , a subset  $\mathcal{X}' \subseteq \mathcal{X}$  is called a  $\lambda$ -net of  $\mathcal{X}$  if  $\delta(x, \mathcal{X}' \setminus \{x\}) > \lambda$  for each  $x \in \mathcal{X}'$  and  $\delta(x, \mathcal{X}') < \lambda$  for each  $x \in \mathcal{X}$ .

Har-Peled and Mendel [14] showed that a small-size net of a given set located in a low-dimensional Euclidean space can be constructed in near-linear time.

**Lemma 2** ([14]) *Given a set  $\mathcal{X} \subset \mathbb{R}^d$  and a real number  $\lambda > 0$ , one can construct a  $\lambda$ -net of  $\mathcal{X}$  of size no more than  $\min\{(\lambda^{-1} \max_{x,y \in \mathcal{X}} \delta(x, y))^d, |\mathcal{X}|\}$  in  $|\mathcal{X}| \log|\mathcal{X}| 2^{O(d)}$  time.*

## 2 Data Reduction

In this section we reduce the size of instance  $\mathcal{I}$  based on the stronger version of Johnson–Lindenstrauss transform [15] given by Narayanan and Nelson [16] and the coresset-construction approach given by Chen [11]. Lemma 3 and Lemma 4 are the guarantees of these two approaches.

**Lemma 3** ([16]) *Given a set  $\mathcal{X} \subset \mathbb{R}^d$  and a real number  $\varepsilon \in (0, 1)$ , a mapping  $h : \mathbb{R}^d \rightarrow \mathbb{R}^{\tilde{d}}$  satisfying  $\tilde{d} = O(\varepsilon^{-2} \log|\mathcal{X}|)$  and  $\delta(h(x), h(y)) \in [1, 1 + \varepsilon]\delta(x, y)$  for each  $x \in \mathcal{X}$  and  $y \in \mathbb{R}^d$  can be constructed in  $O(|\mathcal{X}|d \log|\mathcal{X}|)$  time.*

We can assume that the mapping  $h : \mathcal{X}' \rightarrow \mathbb{R}^{\tilde{d}}$  constructed by Lemma 3 is injective for each finite set  $\mathcal{X}' \subset \mathbb{R}^d$ . This is without loss of generality since we can make copies of the points from  $\mathbb{R}^{\tilde{d}}$  that have multiple inverse images under  $h$ . Such an assumption means that we need to differentiate  $h(x)$  and  $h(y)$  for each two distinct points  $x, y \in \mathbb{R}^{\tilde{d}}$ , even for the case where  $h(x)$  and  $h(y)$  share the same value in each dimension (differentiating  $h(x)$  and  $h(y)$  is necessary since two points with the same dimension values can be associated with different service levels).

**Lemma 4** ([11]) *Given a set  $\mathcal{X} \subset \mathbb{R}^d$ , a real number  $\varepsilon \in (0, 1)$ , and an integer  $k > 0$ , one can construct a weighted subset  $\tilde{\mathcal{X}} \subseteq \mathcal{X}$  with a weight function  $w : \tilde{\mathcal{X}} \rightarrow [1, +\infty)$  satisfying  $\sum_{c \in \tilde{\mathcal{X}}} w(c) = |\mathcal{X}|$  in  $O(|\mathcal{X}|dk)$  time, such that  $|\tilde{\mathcal{X}}| \leq d(k\varepsilon^{-1} \log|\mathcal{X}|)^{O(1)}$  and  $\sum_{c \in \tilde{\mathcal{X}}} w(c)\delta(c, \mathcal{D}) \in [1 - \varepsilon, 1 + \varepsilon] \sum_{c \in \mathcal{X}} \delta(c, \mathcal{D})$  for each  $\mathcal{D} \subset \mathbb{R}^d$  with  $|\mathcal{D}| \leq k$ .*

The following corollary of Lemma 4 says that the coresset-construction approach given in [11] can be adapted to the BUDGETED PRIORITY  $p$ -MEDIAN problem.

**Corollary 1** *Given a real number  $\varepsilon \in (0, 1)$  and an instance  $\mathcal{I} = (\mathcal{C}, \mathcal{F}, \mathcal{L}, \mathcal{P}, \ell)$  of the BUDGETED PRIORITY  $p$ -MEDIAN problem with  $\sum_{i=1}^{|\mathcal{L}|} \lfloor p_i \rfloor = p$  and  $\mathcal{C} \cup \mathcal{F} \subset \mathbb{R}^d$ , one can construct a weighted subset  $\tilde{\mathcal{C}} \subseteq \mathcal{C}$  with a weight function  $w : \tilde{\mathcal{C}} \rightarrow [1, +\infty)$  satisfying  $\sum_{c \in \tilde{\mathcal{C}}} w(c) = |\mathcal{C}|$  in  $O(|\mathcal{C}|dp)$  time, such that  $|\tilde{\mathcal{C}}| \leq d(p\varepsilon^{-1} \log|\mathcal{C}|)^{O(1)}$  and  $\sum_{c \in \tilde{\mathcal{C}}} w(c)\delta(c, \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+) \in [1 - \varepsilon, 1 + \varepsilon] \sum_{c \in \mathcal{C}} \delta(c, \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+)$  for each  $\mathcal{D} \subseteq \mathcal{F}$  with  $|\mathcal{D}| \leq p$ .*

**Proof** For each  $i \in \mathcal{L}$ , denote by  $\tilde{\mathcal{C}}_i$  the weighted subset of  $\mathcal{C}_i$  and  $w : \tilde{\mathcal{C}}_i \rightarrow [1, +\infty)$  the corresponding weight function constructed by Lemma 4 when taking  $\mathcal{C}_i$ ,  $\varepsilon$ , and  $p$  as inputs. We have  $\ell(c) = i$  for each  $c \in \tilde{\mathcal{C}}_i$  due to the definition of  $\mathcal{C}_i$  and the fact that

$\tilde{\mathcal{C}}_i \subseteq \mathcal{C}_i$ . Let  $\tilde{\mathcal{C}} = \bigcup_{i=1}^{|\mathcal{L}|} \tilde{\mathcal{C}}_i$ . Lemma 4 implies that

$$\sum_{c \in \tilde{\mathcal{C}}} w(c) = \sum_{i=1}^{|\mathcal{L}|} \sum_{c \in \tilde{\mathcal{C}}_i} w(c) = \sum_{i=1}^{|\mathcal{L}|} |\mathcal{C}_i| = |\mathcal{C}| \quad (1)$$

and

$$|\tilde{\mathcal{C}}| = \sum_{i=1}^{|\mathcal{L}|} |\tilde{\mathcal{C}}_i| \leq \sum_{i=1}^{|\mathcal{L}|} d(p\varepsilon^{-1} \log |\mathcal{C}_i|)^{O(1)} \leq d(p\varepsilon^{-1} \log |\mathcal{C}|)^{O(1)}. \quad (2)$$

Also by Lemma 4, we know that constructing  $\tilde{\mathcal{C}}$  and the corresponding weight function takes

$$O\left(\sum_{i=1}^{|\mathcal{L}|} |\mathcal{C}_i| d p\right) = O(|\mathcal{C}| d p) \quad (3)$$

time in total. Moreover, it is the case that

$$\begin{aligned} \sum_{c \in \tilde{\mathcal{C}}_i} w(c) \delta(c, \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+) &= \sum_{c \in \tilde{\mathcal{C}}_i} w(c) \delta(c, \mathcal{D} \cap \mathcal{F}_i^+) \\ &\in [1 - \varepsilon, 1 + \varepsilon] \sum_{c \in \mathcal{C}_i} \delta(c, \mathcal{D} \cap \mathcal{F}_i^+) \\ &= [1 - \varepsilon, 1 + \varepsilon] \sum_{c \in \mathcal{C}_i} \delta(c, \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+) \end{aligned} \quad (4)$$

for each  $i \in \mathcal{L}$  and  $\mathcal{D} \subseteq \mathcal{F}$  with  $|\mathcal{D}| \leq p$ , where the first and third steps are due to the fact that  $\ell(c) = i$  for each  $c \in \tilde{\mathcal{C}}_i$ , and the second step follows from Lemma 4. Summing both sides of equality (4) over  $i \in \mathcal{L}$ , we know that each  $\mathcal{D} \subseteq \mathcal{F}$  with  $|\mathcal{D}| \leq p$  satisfies

$$\begin{aligned} \sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+) &= \sum_{i=1}^{|\mathcal{L}|} \sum_{c \in \tilde{\mathcal{C}}_i} w(c) \delta(c, \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+) \\ &\in [1 - \varepsilon, 1 + \varepsilon] \sum_{i=1}^{|\mathcal{L}|} \sum_{c \in \mathcal{C}_i} \delta(c, \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+) \\ &= [1 - \varepsilon, 1 + \varepsilon] \sum_{c \in \mathcal{C}} \delta(c, \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+), \end{aligned}$$

combining which with equality (1), inequality (2), and equality (3), we complete the proof of Corollary 1.

Based on Lemma 3 and Corollary 1, we obtain the following result about data reduction for the BUDGETED PRIORITY  $p$ -MEDIAN problem.

**Lemma 5** *Given a real number  $\varepsilon \in (0, 1)$  and an instance  $\mathcal{I} = (\mathcal{C}, \mathcal{F}, \mathcal{L}, \mathcal{P}, \ell)$  of the BUDGETED PRIORITY  $p$ -MEDIAN problem satisfying  $\sum_{i=1}^{|\mathcal{L}|} \lfloor p_i \rfloor = p$  and  $\mathcal{C} \cup \mathcal{F} \subset \mathbb{R}^d$ , one can construct a mapping  $h : \mathbb{R}^d \rightarrow \mathbb{R}^{\tilde{d}}$  and a weighted set  $\tilde{\mathcal{C}} \subset \mathbb{R}^{\tilde{d}}$  with a weight function  $w : \tilde{\mathcal{C}} \rightarrow [1, +\infty)$  satisfying  $\sum_{c \in \tilde{\mathcal{C}}} w(c) = |\mathcal{C}|$  and a level function  $\ell : \tilde{\mathcal{C}} \rightarrow \mathcal{L}$  in  $O(|\mathcal{C}|d \log |\mathcal{C}| + (p\varepsilon^{-1}|\mathcal{C}|)^{O(1)})$  time, such that  $\tilde{d} = \varepsilon^{-O(1)}(\log p + \log \log |\mathcal{C}|)$ ,  $|\tilde{\mathcal{C}}| \leq (p\varepsilon^{-1} \log |\mathcal{C}|)^{O(1)}$ , and  $\sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \{h(f) : f \in \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+\}) \in [1 - \varepsilon, 1 + O(\varepsilon)] \sum_{c \in \mathcal{C}} \delta(c, \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+)$  for each  $\mathcal{D} \subseteq \mathcal{F}$  with  $|\mathcal{D}| \leq p$ .*

**Proof** Denote by  $h_1 : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$  the mapping constructed by Lemma 3 when taking  $\varepsilon$  and  $\mathcal{C}$  as inputs, and let  $\ell(h_1(x)) = \ell(x)$  for each  $x \in \mathcal{C} \cup \mathcal{F}$ . Denote by  $\mathcal{C}'$  the weighted subset of  $\{h_1(c) : c \in \mathcal{C}\}$  constructed by Corollary 1 when taking  $\varepsilon$  and  $(\{h_1(c) : c \in \mathcal{C}\}, \{h_1(f) : f \in \mathcal{F}\}, \mathcal{L}, \mathcal{P}, \ell)$  as inputs, and let  $w : \mathcal{C}' \rightarrow [1, +\infty)$  be the corresponding weight function. Let  $h_2 : \mathbb{R}^{d'} \rightarrow \mathbb{R}^{\tilde{d}}$  be the mapping constructed by Lemma 3 when taking  $\mathcal{C}'$  and  $\varepsilon$  as inputs. Define  $\tilde{\mathcal{C}} = \{h_2(c) : c \in \mathcal{C}'\}$ . Let  $h(x) = h_2(h_1(x))$  for each  $x \in \mathbb{R}^d$ . Let  $w(c) = w(h_2^{-1}(c))$  and  $\ell(c) = \ell(h_2^{-1}(c))$  for each  $c \in \tilde{\mathcal{C}}$ . The above process for data reduction is illustrated in Fig. 1.

Observe that

$$\sum_{c \in \tilde{\mathcal{C}}} w(c) = \sum_{c \in \mathcal{C}'} w(h_2^{-1}(c)) = \sum_{c \in \mathcal{C}'} w(c) = |\{h_1(c) : c \in \mathcal{C}\}| = |\mathcal{C}|, \tag{5}$$

where the second and fourth steps are due to the assumption that the mappings constructed by Lemma 3 are injective, and the third step is due to Corollary 1. Moreover, Lemma 3 and Corollary 1 imply that

$$d' = O(\varepsilon^{-2} \log |\mathcal{C}|)$$

and

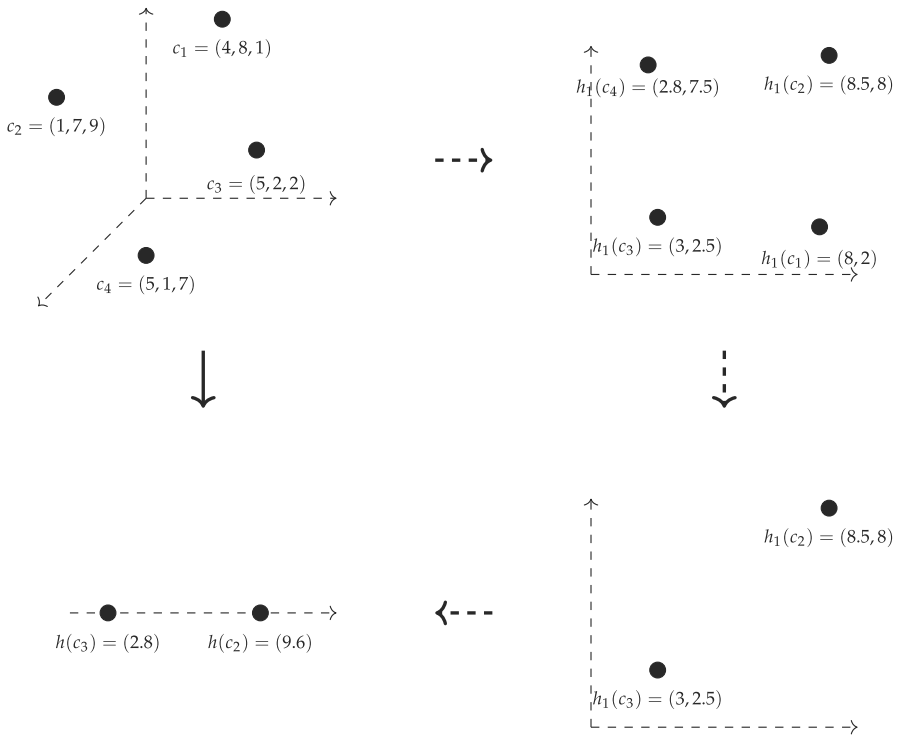
$$|\tilde{\mathcal{C}}| = |\mathcal{C}'| \leq d'(p\varepsilon^{-1} \log |\mathcal{C}|)^{O(1)} = (p\varepsilon^{-1} \log |\mathcal{C}|)^{O(1)}, \tag{6}$$

combining which with Lemma 3 and Corollary 1, we know that constructing  $\tilde{\mathcal{C}}$  and  $h : \mathbb{R}^d \rightarrow \mathbb{R}^{\tilde{d}}$  takes

$$O(|\mathcal{C}|d \log |\mathcal{C}| + |\mathcal{C}|d'p + |\mathcal{C}'|d' \log |\mathcal{C}'|) \leq O(|\mathcal{C}|d \log |\mathcal{C}| + (p\varepsilon^{-1}|\mathcal{C}|)^{O(1)}) \tag{7}$$

time in total, and

$$\tilde{d} = O(\varepsilon^{-2} \log |\mathcal{C}'|) \leq O(\varepsilon^{-2} \log (p\varepsilon^{-1} \log |\mathcal{C}|)) = \varepsilon^{-O(1)}(\log p + \log \log |\mathcal{C}|). \tag{8}$$



**Fig. 1** Illustration of our data reduction method, which reduces the dimensionality twice and constructs a coreset. Here, we have  $\mathcal{C} = \{c_1, \dots, c_4\}$ ,  $\mathcal{C}' = \{h_1(c_2), h_1(c_3)\}$ , and  $\tilde{\mathcal{C}} = \{h(c_2), h(c_3)\}$

Considering a set  $\mathcal{D} \subseteq \mathcal{F}$  with  $|\mathcal{D}| \leq p$ , we have

$$\begin{aligned}
 & \sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \{h(f) : f \in \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+\}) \\
 & \in [1 - \varepsilon, 1 + \varepsilon] \sum_{c \in \mathcal{C}'} w(c) \delta(c, \{h_1(f) : f \in \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+\}) \\
 & \in [1 - \varepsilon, 1 + O(\varepsilon)] \sum_{c \in \mathcal{C}} \delta(h_1(c), \{h_1(f) : f \in \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+\}) \\
 & \in [1 - \varepsilon, 1 + O(\varepsilon)] \sum_{c \in \mathcal{C}} \delta(c, \mathcal{D} \cap \mathcal{F}_{\ell(c)}^+), \tag{9}
 \end{aligned}$$

where the first and third steps follow from Lemma 3, and the second step is due to Corollary 1. Combining formula (9) with equality (5), inequality (6), inequality (7), and inequality (8), we complete the proof of Lemma 5.



### 3 Our Algorithm

In this section we show how the desired  $(1 + \varepsilon)$ -approximation solution can be constructed based on the reduced instance given by Lemma 5. We first introduce some notations. Let  $h : \mathbb{R}^d \rightarrow \mathbb{R}^{\tilde{d}}$  and  $\tilde{\mathcal{C}} \subset \mathbb{R}^{\tilde{d}}$  be the mapping and weighted set given by Lemma 5 when taking  $\varepsilon$  and  $\mathcal{I}$  as inputs, and let  $w : \tilde{\mathcal{C}} \rightarrow [1, +\infty)$  and  $\ell : \tilde{\mathcal{C}} \rightarrow \mathcal{L}$  denote the corresponding weight function and level function, respectively. As done in Sect. 2, we can make  $h : \mathcal{F} \rightarrow \mathbb{R}^{\tilde{d}}$  injective by copying the points from  $\mathbb{R}^{\tilde{d}}$  that have multiple inverse images. For each  $f \in \mathcal{F}$ , let  $\ell(h(f)) = \ell(f)$ . Define  $\tilde{\mathcal{F}} = \{h(f) : f \in \mathcal{F}\}$  and  $\tilde{\mathcal{D}}^* = \{h(f) : f \in \mathcal{D}^*\}$ . Let  $\tilde{\mathcal{F}}_i = \{h(f) : f \in \mathcal{F}_i\}$  and  $\tilde{\mathcal{F}}_i^+ = \{h(f) : f \in \mathcal{F}_i^+\}$  for each  $i \in \mathcal{L}$ . Define  $\delta_{\max} = \max_{c \in \tilde{\mathcal{C}}} \delta(c, \tilde{\mathcal{D}}^* \cap \tilde{\mathcal{F}}_{\ell(c)}^+)$ . The following lemma provides a way of estimating the cost of an optimal solution to  $\mathcal{I}$ .

**Lemma 6**  $\delta_{\max} \leq \sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \tilde{\mathcal{D}}^* \cap \tilde{\mathcal{F}}_{\ell(c)}^+) \leq (1 + O(\varepsilon)) \text{opt}$ .

**Proof** Observe that

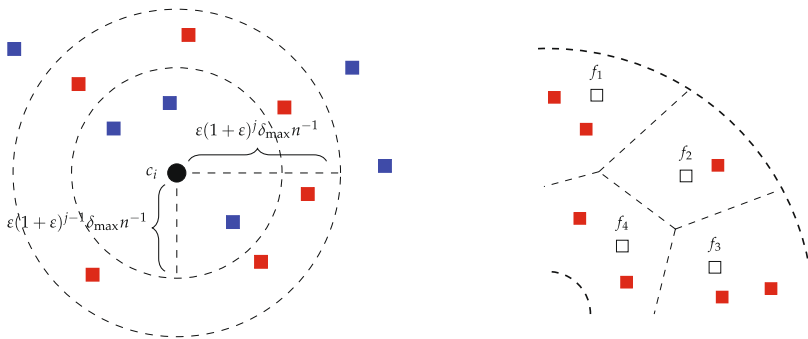
$$\begin{aligned} \delta_{\max} &\leq \max_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \tilde{\mathcal{D}}^* \cap \tilde{\mathcal{F}}_{\ell(c)}^+) \\ &\leq \sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \tilde{\mathcal{D}}^* \cap \tilde{\mathcal{F}}_{\ell(c)}^+) \\ &= \sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \{h(f) : f \in \mathcal{D}^* \cap \mathcal{F}_{\ell(c)}^+\}) \\ &\leq (1 + O(\varepsilon)) \sum_{c \in \mathcal{C}} \delta(c, \mathcal{D}^* \cap \mathcal{F}_{\ell(c)}^+) \\ &= (1 + O(\varepsilon)) \text{opt}, \end{aligned}$$

where the first step follows from the fact that  $w(c) \geq 1$  for each  $c \in \tilde{\mathcal{C}}$  and the definition of  $\delta_{\max}$ , the fourth step is due to Lemma 5, and the fifth step follows from the fact that  $(\mathcal{D}^*, \varphi^*)$  is an optimal solution to  $\mathcal{I}$ . This completes the proof of Lemma 6.

#### 3.1 The Partition Step

In this section, we estimate the locations of the facilities from  $\tilde{\mathcal{D}}^*$  based on their neighboring clients. For each  $i \in [p]$ , let  $c_i$  denote the client from  $\{c \in \tilde{\mathcal{C}} : \ell(c) \leq \ell_i^*\}$  nearest to  $h(f_i^*)$ , and define  $\mathcal{B}_i(\alpha) = \{f \in \tilde{\mathcal{F}} : \delta(c_i, f) \leq \alpha\}$  for each  $\alpha > 0$ . We partition the facilities from  $\tilde{\mathcal{F}}$  into a set of ring cells centered at the clients from  $\{c_1, \dots, c_p\}$ : Let  $\mathcal{Q}(i, 0) = \mathcal{B}_i(\varepsilon \delta_{\max} n^{-1})$  and  $\mathcal{Q}(i, j) = \mathcal{B}_i(\varepsilon(1+\varepsilon)^j \delta_{\max} n^{-1}) \setminus \mathcal{B}_i(\varepsilon(1+\varepsilon)^{j-1} \delta_{\max} n^{-1})$  for each  $i \in [p]$  and  $j \in [\lceil \varepsilon^{-2} \log n \rceil]$ , as shown in Fig. 2a. We have

$$\varepsilon(1 + \varepsilon)^{\lceil \varepsilon^{-2} \log n \rceil} \delta_{\max} n^{-1} > \delta_{\max} = \max_{c \in \tilde{\mathcal{C}}} \delta(c, \tilde{\mathcal{D}}^* \cap \tilde{\mathcal{F}}_{\ell(c)}^+) \geq \delta(c_i, h(f_i^*))$$



(a) Illustration of the ring cells centered at  $c_i$ , where the red points denote the members of  $\mathcal{Q}(i, j)$ .

(b) Illustration of the net constructed for a ring cell, where  $f_1, f_2, f_3,$  and  $f_4$  are the members of the net.

**Fig. 2** The images of candidates for the opened facilities under  $h$

for each  $i \in [p]$ , where the second step follows from the definition of  $\delta_{\max}$ , and the third step follows from the fact that  $h(f_i^*) \in \mathcal{D}^* \cap \tilde{\mathcal{F}}_{\ell(c_i)}^+$  (due to the definition of  $c_i$ ). Combining this inequality with the definition of  $\mathcal{Q}(i, j)$ , we know that there is an integer  $j \in [0, \lceil \varepsilon^{-2} \log n \rceil]$  satisfying  $h(f_i^*) \in \mathcal{Q}(i, j)$  for each  $i \in [p]$ . Denote by  $\mathcal{Q}_i$  such a set  $\mathcal{Q}(i, j)$  containing  $h(f_i^*)$ . Our idea for constructing solutions to  $\mathcal{I}$  in FPT( $p$ ) time is to guess the sets  $\mathcal{Q}_1, \dots, \mathcal{Q}_p$ , such that we can limit the search range of the opened facilities to the inverse images of the members of  $\bigcup_{i=1}^p \mathcal{Q}_i$  under  $h$ . The following result implies that we can guess the sets  $\mathcal{Q}_1, \dots, \mathcal{Q}_p$  with a  $(p\varepsilon^{-1})^{O(p)}n^{O(1)}$  multiplicative overhead in the running time of our algorithm.

**Lemma 7** *Given the reduced instance  $(\tilde{\mathcal{C}}, \tilde{\mathcal{F}}, \mathcal{L}, \mathcal{P}, \ell)$  of the BUDGETED PRIORITY  $p$ -MEDIAN problem, the set  $\{\mathcal{Q}_1, \dots, \mathcal{Q}_p\}$  can be guessed by enumerating over no more than  $(p\varepsilon^{-1})^{O(p)}n^{O(1)}$  items.*

**Proof** We have  $|\tilde{\mathcal{F}}| = |\mathcal{F}| \leq n$  and  $|\tilde{\mathcal{C}}| \leq (p\varepsilon^{-1} \log n)^{O(1)}$  due to Lemma 5. Using the definitions of  $\delta_{\max}$  and  $c_i$ , it can be seen that there are  $|\tilde{\mathcal{F}}| \cdot |\tilde{\mathcal{C}}| \leq n(p\varepsilon^{-1} \log n)^{O(1)}$  choices of the value of  $\delta_{\max}$  and  $|\tilde{\mathcal{C}}|^p \leq (p\varepsilon^{-1} \log n)^{O(p)}$  choices of the set  $\{c_1, \dots, c_p\}$ . Consequently, the definition of  $\mathcal{Q}(i, j)$  indicates that we have no more than  $n(p\varepsilon^{-1} \log n)^{O(p)}$  choices of the set  $\{\mathcal{Q}(1, 0), \dots, \mathcal{Q}(p, \lceil \varepsilon^{-2} \log n \rceil)\}$ . Combining this with the fact that each  $i \in [p]$  satisfies  $\mathcal{Q}_i \in \{\mathcal{Q}(i, 0), \dots, \mathcal{Q}(i, \lceil \varepsilon^{-2} \log n \rceil)\}$ , the number of choices of the set  $\{\mathcal{Q}_1, \dots, \mathcal{Q}_p\}$  can be upper-bounded by  $n(p\varepsilon^{-1} \log n)^{O(p)} \cdot (\varepsilon^{-2} \log n)^p$ , which is no more than  $(p\varepsilon^{-1})^{O(p)}n^{O(1)}$  due to Lemma 1. Thus, Lemma 7 is true.

### 3.2 The Construction Step

Given the set  $\{\mathcal{Q}_1, \dots, \mathcal{Q}_p\}$ , our algorithm constructs a net of  $\mathcal{Q}_i \cap \tilde{\mathcal{F}}_j$  for each  $i \in [p]$  and  $j \in \mathcal{L}$  and considers the inverse images of the members of the net under  $h$  as candidates for the opened facilities (see Fig. 2b), as described in Algorithm 1. Define  $\mathcal{J}, \mathbb{D}$ , and  $\mathcal{N}(i, j)$  in the same way as Algorithm 1 for each  $i \in [p]$  and  $j \in \mathcal{L}$ . Let

**Algorithm 1** Construct the approximation solution

**Require:** A constant  $\varepsilon \in (0, 1)$ , an instance  $\mathcal{I} = (\mathcal{C}, \mathcal{F}, \mathcal{L}, \mathcal{P}, \ell)$  of the BUDGETED PRIORITY  $p$ -MEDIAN problem satisfying  $\mathcal{C} \cup \mathcal{F} \subset \mathbb{R}^d$  and  $\sum_{i=1}^{|\mathcal{L}|} \lfloor p_i \rfloor = p$ , an injective mapping  $h : \mathcal{F} \rightarrow \mathbb{R}^{\tilde{d}}$ , a set  $\tilde{\mathcal{C}} \subset \mathbb{R}^{\tilde{d}}$  of no more than  $(p\varepsilon^{-1} \log n)^{O(1)}$  weighted clients with a weight function  $w : \tilde{\mathcal{C}} \rightarrow [1, +\infty)$  and a level function  $\ell : \tilde{\mathcal{C}} \rightarrow \mathcal{L}$ ,  $p$  subsets  $\mathcal{Q}_1, \dots, \mathcal{Q}_p$  of  $\{h(f) : f \in \mathcal{F}\}$ , a set  $\mathcal{J}$  storing the images of facilities selected from  $\mathcal{F}$  under  $h$ , and a set  $\mathbb{D}$  storing selected subsets of  $\mathcal{J}$ .

**Ensure:** A solution  $(\mathcal{D}^\dagger, \varphi^\dagger)$  to  $\mathcal{I}$

```

1  $\mathcal{J} \leftarrow \emptyset, \mathbb{D} \leftarrow \emptyset$ 
2 for  $i \leftarrow 1$  to  $|\mathcal{L}|$  do
3    $\mathcal{F}_i \leftarrow \{f \in \mathcal{F} : \ell(f) = i\}$ 
4    $\tilde{\mathcal{F}}_i \leftarrow \{h(f) : f \in \mathcal{F}_i\}$ 
5 end for
6 for  $i \leftarrow 1$  to  $|\mathcal{L}|$  do
7    $\mathcal{F}_i^+ \leftarrow \bigcup_{j=i}^{|\mathcal{L}|} \mathcal{F}_j$ 
8    $\tilde{\mathcal{F}}_i^+ \leftarrow \{h(f) : f \in \mathcal{F}_i^+\}$ 
9 end for
10 for  $i \leftarrow 1$  to  $p$  do
11   for  $j \leftarrow 1$  to  $|\mathcal{L}|$  do
12     if  $|\mathcal{Q}_i \cap \tilde{\mathcal{F}}_j| \leq 1$  then
13        $\mathcal{J} \leftarrow \mathcal{J} \cup (\mathcal{Q}_i \cap \tilde{\mathcal{F}}_j)$ 
14     else
15       Construct a  $\max_{x,y \in \mathcal{Q}_i} \varepsilon \delta(x, y)$ -net  $\mathcal{N}(i, j)$  of  $\mathcal{Q}_i \cap \tilde{\mathcal{F}}_j$ 
16        $\mathcal{J} \leftarrow \mathcal{J} \cup \mathcal{N}(i, j)$ 
17     end if
18   end for
19 end for
20 for each  $\tilde{\mathcal{D}} \subseteq \mathcal{J}$  do
21   if  $|\tilde{\mathcal{D}} \cap \tilde{\mathcal{F}}_j| \leq p_j \forall j \in \mathcal{L}$  then
22      $\mathbb{D} \leftarrow \mathbb{D} \cup \{\tilde{\mathcal{D}}\}$ 
23   end if
24 end for
25  $\tilde{\mathcal{D}}^\dagger \leftarrow \arg \min_{\tilde{\mathcal{D}} \in \mathbb{D}} \sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \tilde{\mathcal{D}} \cap \tilde{\mathcal{F}}_{\ell(c)}^+)$ 
26  $\mathcal{D}^\dagger \leftarrow \{h^{-1}(f) : f \in \tilde{\mathcal{D}}^\dagger\}$ 
27 for each  $c \in \mathcal{C}$  do
28    $\varphi^\dagger(c) \leftarrow \arg \min_{f \in \mathcal{D}^\dagger \cap \mathcal{F}_{\ell(c)}^+} \delta(c, f)$ 
29 end for
30 return  $(\mathcal{D}^\dagger, \varphi^\dagger)$ 

```

$(\mathcal{D}^\dagger, \varphi^\dagger)$  denote the solution returned by the algorithm, and let  $\tilde{\mathcal{D}}^\dagger = \{h(f) : f \in \mathcal{D}^\dagger\}$ . The following lemma implies that  $(\mathcal{D}^\dagger, \varphi^\dagger)$  can be constructed in  $\text{FPT}(p)$  time.

**Lemma 8** *We can obtain the solution  $(\mathcal{D}^\dagger, \varphi^\dagger)$  to  $\mathcal{I}$  in no more than  $O(nd \log n) + (p\varepsilon^{-1})^{p\varepsilon^{-O(1)}} n^{O(1)}$  time.*

**Proof** Define  $\mathcal{G} = \{(i \in [p], j \in \mathcal{L}) : |\mathcal{Q}_i \cap \tilde{\mathcal{F}}_j| > 1\}$ . The definition of  $\mathcal{N}(i, j)$  and Lemma 2 imply that

$$|\mathcal{N}(i, j)| \leq \left( \frac{\max_{x,y \in \mathcal{Q}_i \cap \tilde{\mathcal{F}}_j} \delta(x, y)}{\varepsilon \max_{x,y \in \mathcal{Q}_i \cap \tilde{\mathcal{F}}_j} \delta(x, y)} \right)^{\tilde{d}} = \varepsilon^{-\tilde{d}} \leq (p \log n)^{\varepsilon^{-O(1)}} \quad (10)$$

for each  $(i, j) \in \mathcal{G}$ , where the third step follows from the fact that  $\tilde{d} \leq \varepsilon^{-O(1)}(\log p + \log \log n)$  (due to Lemma 5). Consequently, we know that

$$\begin{aligned}
 |\mathcal{J}| &= \sum_{(i,j) \in \mathcal{G}} |\mathcal{N}(i, j)| + p|\mathcal{L}| - |\mathcal{G}| \\
 &\leq |\mathcal{L}|(p \log n)^{\varepsilon^{-O(1)}} \\
 &\leq (p \log n)^{\varepsilon^{-O(1)}} \sum_{i=1}^{|\mathcal{L}|} \lfloor p_i \rfloor \\
 &= (p \log n)^{\varepsilon^{-O(1)}}, \tag{11}
 \end{aligned}$$

where the first step follows from the definition of  $\mathcal{J}$ , the second step is due to inequality (10), the third step follows from the fact that  $p_i \geq 1$  for each  $i \in \mathcal{L}$ , and the fourth step is derived from the fact that  $p = \sum_{i=1}^{|\mathcal{L}|} \lfloor p_i \rfloor$ .

Observe that Algorithm 1 constructs a net of  $\mathcal{Q}_i \cap \tilde{\mathcal{F}}_j$  for each  $(i, j) \in \mathcal{G}$  according to the maximal distance between the facilities from  $\mathcal{Q}_i \cap \tilde{\mathcal{F}}_j$ . Lemma 2 and the fact that  $\tilde{d} \leq \varepsilon^{-O(1)}(\log p + \log \log n)$  imply that this takes no more than

$$\sum_{(i,j) \in \mathcal{G}} 2^{O(\tilde{d})} |\mathcal{Q}_i \cap \tilde{\mathcal{F}}_j|^{O(1)} \leq 2^{O(\tilde{d})} n^{O(1)} \leq (p \log n)^{\varepsilon^{-O(1)}} n^{O(1)} \tag{12}$$

time in total. After this, Algorithm 1 enumerates all subsets  $\tilde{\mathcal{D}} \subseteq \mathcal{J}$  with  $|\tilde{\mathcal{D}} \cap \tilde{\mathcal{F}}_j| \leq p_i, \forall j \in \mathcal{L}$  and selects the one corresponding to the solution with minimal cost to the reduced instance  $(\tilde{\mathcal{C}}, \tilde{\mathcal{F}}, \mathcal{L}, \mathcal{P}, \ell)$ . Equality  $\sum_{i=1}^{|\mathcal{L}|} \lfloor p_i \rfloor = p$  implies that each subset considered in this enumeration is of size no more than  $p$ , and we can complete this enumeration in

$$O(|\mathcal{J}|^p |\tilde{\mathcal{C}}| \tilde{d} p) \leq (p \log n)^{p\varepsilon^{-O(1)}} |\tilde{\mathcal{C}}| \tilde{d} p = (p \log n)^{p\varepsilon^{-O(1)}} \tag{13}$$

time, where the first step follows from inequality (11), and the second step follows from the fact that  $|\tilde{\mathcal{C}}| \leq (p\varepsilon^{-1} \log n)^{O(1)}$  and  $\tilde{d} \leq \varepsilon^{-O(1)}(\log p + \log \log n)$  (due to Lemma 5). Using inequality (12) and inequality (13), we know that Algorithm 1 runs in no more than  $(p \log n)^{p\varepsilon^{-O(1)}} n^{O(1)}$  time.

Lemma 5 implies that constructing the mapping  $h : \mathbb{R}^d \rightarrow \mathbb{R}^{\tilde{d}}$  and the reduced instance  $(\tilde{\mathcal{C}}, \tilde{\mathcal{F}}, \mathcal{L}, \mathcal{P}, \ell)$  takes  $O(nd \log n + (p\varepsilon^{-1}n)^{O(1)})$  time, and Lemma 7 says that there are at most  $(p\varepsilon^{-1})^{O(p)} n^{O(1)}$  choices of the set  $\{\mathcal{Q}_1, \dots, \mathcal{Q}_p\}$ . Therefore, all inputs of Algorithm 1 can be known with a  $(p\varepsilon^{-1})^{O(p)} n^{O(1)}$  multiplicative and an  $O(nd \log n + (p\varepsilon^{-1}n)^{O(1)})$  additive overheads in the running time of the algorithm. Combining this with the fact that Algorithm 1 runs in at most  $(p \log n)^{p\varepsilon^{-O(1)}} n^{O(1)}$  time, we know that constructing  $(\mathcal{D}^\dagger, \varphi^\dagger)$  takes no more than  $O(nd \log n) + (p \log n)^{p\varepsilon^{-O(1)}} n^{O(1)}$  time, which can be upper-bounded by  $O(nd \log n) + (p\varepsilon^{-1})^{p\varepsilon^{-O(1)}} n^{O(1)}$  due to Lemma 1, as desired.

We now consider the cost of  $(\mathcal{D}^\dagger, \varphi^\dagger)$ . The following lemma says that it is a  $(1 + O(\varepsilon))$ -approximation solution to  $\mathcal{I}$ .

**Lemma 9**  $\sum_{c \in \mathcal{C}} \delta(c, \varphi^\dagger(c)) \leq (1 + O(\varepsilon))\text{opt}$ .

**Proof** The fact that  $h(f_i^*) \in \mathcal{Q}_i$  and  $\ell(h(f_i^*)) = \ell(f_i^*) = \ell_i^*$  implies that  $h(f_i^*) \in \mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}$  and thus  $|\mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}| \geq 1$  for each  $i \in [p]$ . Given a real number  $i \in [p]$ , we, respectively, consider the following three cases: (1)  $|\mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}| = 1$ , (2)  $|\mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}| > 1$  and  $\mathcal{Q}_i = \mathcal{Q}(i, 0)$ , and (3)  $|\mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}| > 1$  and  $\mathcal{Q}_i \neq \mathcal{Q}(i, 0)$ .

For case (1), the facility from  $\mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}$  is added to  $\mathcal{J}$  by Algorithm 1, and the fact that  $h(f_i^*) \in \mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}$  indicates that

$$\{h(f_i^*)\} = \mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*} \subseteq \mathcal{J}. \tag{14}$$

For the case where  $|\mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}| > 1$ , it can be seen that Algorithm 1 adds the facilities from a net  $\mathcal{N}(i, \ell_i^*)$  of  $\mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}$  to  $\mathcal{J}$ , and the fact that  $\mathcal{N}(i, \ell_i^*) \subseteq \mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}$  implies that  $(\mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}) \cap \mathcal{J} \neq \emptyset$ . For case (2), let  $f'_i \in \mathcal{F}$  be a facility satisfying  $h(f'_i) \in (\mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}) \cap \mathcal{J}$ . We have

$$\begin{aligned} \delta(h(f'_i), h(f_i^*)) &\leq \delta(c_i, h(f'_i)) + \delta(c_i, h(f_i^*)) \\ &\leq 2 \max_{f \in \mathcal{Q}_i} \delta(c_i, f) \\ &\leq \frac{2\varepsilon}{n} \delta_{\max} \\ &\leq \frac{2\varepsilon}{n} (1 + O(\varepsilon))\text{opt}, \end{aligned} \tag{15}$$

where the first step is due to triangle inequality, the second step follows from the fact that  $\{h(f'_i), h(f_i^*)\} \subseteq \mathcal{Q}_i$ , the third step is due to the assumption that  $\mathcal{Q}_i = \mathcal{Q}(i, 0)$  and the definition of  $\mathcal{Q}(i, 0)$ , and the fourth step follows from Lemma 6.

For case (3), let  $h(f'_i)$  be the facility from  $\mathcal{N}(i, \ell_i^*)$  nearest to  $h(f_i^*)$ . We have  $h(f'_i) \in \mathcal{N}(i, \ell_i^*) \subseteq \mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}$ , and

$$\begin{aligned} \delta(h(f'_i), h(f_i^*)) &\leq \varepsilon \max_{x, y \in \mathcal{Q}_i} \delta(x, y) \\ &\leq 2\varepsilon \max_{f \in \mathcal{Q}_i} \delta(c_i, f) \\ &\leq 2\varepsilon(1 + \varepsilon)\delta(c_i, h(f_i^*)) \\ &= O(\varepsilon)\delta(c_i, h(f_i^*)), \end{aligned} \tag{16}$$

where the first step follows from the fact that  $h(f'_i)$  is the nearest facility to  $h(f_i^*)$  in a  $\max_{x, y \in \mathcal{Q}_i} \varepsilon\delta(x, y)$ -net of  $\mathcal{Q}_i \cap \tilde{\mathcal{F}}_{\ell_i^*}$ , the second step follows from triangle inequality, and the third step is due to the assumption that  $\mathcal{Q}_i \neq \mathcal{Q}(i, 0)$  (which says that there

exists a real number  $j \in [\lceil \varepsilon^{-2} \log n \rceil]$  satisfying  $\mathcal{Q}_i = \mathcal{Q}(i, j)$ ) and the definition of  $\mathcal{Q}(i, j)$  for  $j > 0$ .

The argument above implies the existence of a set  $\mathcal{D}' = \{f'_1, \dots, f'_p\} \subseteq \mathcal{F}$  with  $h(f'_i) \in \mathcal{J} \cap \tilde{\mathcal{F}}_{\ell_i^*}$  for each  $i \in [p]$ , which satisfies

$$\delta(h(f'_i), h(f_i^*)) \leq \frac{2\varepsilon}{n}(1 + O(\varepsilon))\text{opt} + O(\varepsilon)\delta(c_i, h(f_i^*)) \tag{17}$$

for each  $i \in [p]$  due to equality (14), inequality (15), and inequality (16). Define  $\tilde{\mathcal{D}}' = \{h(f) : f \in \mathcal{D}'\}$ . The fact that  $h(f'_i) \in \tilde{\mathcal{F}}_{\ell_i^*}$  for each  $i \in [p]$  and  $(\mathcal{D}^*, \varphi^*)$  is a feasible solution to  $\mathcal{I}$  implies that inequality  $|\tilde{\mathcal{D}}' \cap \tilde{\mathcal{F}}_j| = |\mathcal{D}^* \cap \mathcal{F}_j| \leq p_j$  holds for each  $j \in \mathcal{L}$ . Combining this with the definition of  $\mathbb{D}$ , we know that  $\tilde{\mathcal{D}}' \in \mathbb{D}$ .

For each  $i \in [p]$ , define  $\tilde{\mathcal{C}}_i^* = \{c \in \tilde{\mathcal{C}} : \arg \min_{f \in \tilde{\mathcal{D}}' \cap \tilde{\mathcal{F}}_{\ell(c)}^+} \delta(c, f) = h(f'_i)\}$ . It is the case that

$$\begin{aligned} \sum_{i=1}^p \sum_{c \in \tilde{\mathcal{C}}_i^*} w(c)\delta(h(f'_i), h(f_i^*)) &\leq \sum_{i=1}^p \sum_{c \in \tilde{\mathcal{C}}_i^*} w(c) \left( \frac{2\varepsilon}{n}(1 + O(\varepsilon))\text{opt} + O(\varepsilon)\delta(c_i, h(f_i^*)) \right) \\ &\leq O(\varepsilon)\text{opt} + O(\varepsilon) \sum_{i=1}^p \sum_{c \in \tilde{\mathcal{C}}_i^*} w(c)\delta(c_i, h(f_i^*)) \\ &\leq O(\varepsilon)\text{opt} + O(\varepsilon) \sum_{i=1}^p \sum_{c \in \tilde{\mathcal{C}}_i^*} w(c)\delta(c, h(f_i^*)) \\ &= O(\varepsilon)\text{opt} + O(\varepsilon) \sum_{c \in \tilde{\mathcal{C}}} w(c)\delta(c, \tilde{\mathcal{D}}^* \cap \tilde{\mathcal{F}}_{\ell(c)}^+) \\ &\leq O(\varepsilon)\text{opt}, \end{aligned} \tag{18}$$

where the first step follows from inequality (17), the second step is derived from the fact that  $\sum_{c \in \tilde{\mathcal{C}}} w(c) = |\mathcal{C}| < n$  (due to Lemma 5), the third step follows from the fact that  $c_i$  is the client from  $\{c \in \tilde{\mathcal{C}} : \ell(c) \leq \ell_i^*\}$  nearest to  $h(f_i^*)$  for each  $i \in [p]$ , the fourth step is due to the definition of  $\tilde{\mathcal{C}}_i^*$ , and the fifth step is derived from Lemma 6. Consequently, we have

$$\begin{aligned} \sum_{c \in \tilde{\mathcal{C}}} w(c)\delta(c, \tilde{\mathcal{D}}' \cap \tilde{\mathcal{F}}_{\ell(c)}^+) &= \sum_{i=1}^p \sum_{c \in \tilde{\mathcal{C}}_i^*} w(c)\delta(c, \tilde{\mathcal{D}}' \cap \tilde{\mathcal{F}}_{\ell(c)}^+) \\ &\leq \sum_{i=1}^p \sum_{c \in \tilde{\mathcal{C}}_i^*} w(c)(\delta(c, h(f_i^*)) + \delta(h(f_i^*), \tilde{\mathcal{D}}' \cap \tilde{\mathcal{F}}_{\ell(c)}^+)) \\ &\leq \sum_{i=1}^p \sum_{c \in \tilde{\mathcal{C}}_i^*} w(c)(\delta(c, h(f_i^*)) + \delta(h(f_i^*), h(f'_i))) \end{aligned}$$

$$\begin{aligned}
 &= \sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \tilde{\mathcal{D}}^* \cap \tilde{\mathcal{F}}_{\ell(c)}^+) \\
 &\quad + \sum_{i=1}^p \sum_{c \in \tilde{\mathcal{C}}_i^*} w(c) \delta(h(f_i^*), h(f_i')) \\
 &= (1 + O(\varepsilon)) \text{opt}, \tag{19}
 \end{aligned}$$

where the second step is due to triangle inequality, the third step follows from the fact that  $\ell(c) \leq \ell_i^*$  and  $h(f_i') \in \tilde{\mathcal{F}}_{\ell(c)}^* \subseteq \tilde{\mathcal{F}}_{\ell(c)}^+$  for each  $c \in \tilde{\mathcal{C}}_i^*$  and  $i \in [p]$ , the fourth step follows from the definition of  $\tilde{\mathcal{C}}_i^*$ , and the fifth step is due to Lemma 6 and inequality (18).

Combining the fact that  $\tilde{\mathcal{D}}' \in \mathbb{D}$  and  $\tilde{\mathcal{D}}^\dagger = \arg \min_{\tilde{\mathcal{D}} \in \mathbb{D}} \sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \tilde{\mathcal{D}} \cap \tilde{\mathcal{F}}_{\ell(c)}^+)$  with inequality (19) yields

$$\sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \tilde{\mathcal{D}}^\dagger \cap \tilde{\mathcal{F}}_{\ell(c)}^+) \leq \sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \tilde{\mathcal{D}}' \cap \tilde{\mathcal{F}}_{\ell(c)}^+) \leq (1 + O(\varepsilon)) \text{opt}. \tag{20}$$

Therefore, we get

$$\begin{aligned}
 \sum_{c \in \mathcal{C}} \delta(c, \varphi^\dagger(c)) &= \sum_{c \in \mathcal{C}} \delta(c, \mathcal{D}^\dagger \cap \mathcal{F}_{\ell(c)}^+) \\
 &\leq \frac{1}{1 - \varepsilon} \sum_{c \in \tilde{\mathcal{C}}} w(c) \delta(c, \tilde{\mathcal{D}}^\dagger \cap \tilde{\mathcal{F}}_{\ell(c)}^+) \\
 &\leq \frac{1 + O(\varepsilon)}{1 - \varepsilon} \text{opt} \\
 &= (1 + O(\varepsilon)) \text{opt},
 \end{aligned}$$

where the first step is due to the definition of  $\varphi^\dagger$ , the second step follows from Lemma 5, and the third step is due to inequality (20). This completes the proof of Lemma 9.

Lemma 8 and Lemma 9 imply that a  $(1 + O(\varepsilon))$ -approximation solution to  $\mathcal{I}$  can be constructed in  $O(nd \log n) + (p\varepsilon^{-1})^{p\varepsilon^{-O(1)}} n^{O(1)}$  time, which in turn implies that Theorem 1 is true.

### 4 Conclusions

In this paper we consider the BUDGETED PRIORITY  $p$ -MEDIAN problem in high-dimensional Euclidean spaces. We give a  $(1 + O(\varepsilon))$ -approximation algorithm running in  $\text{FPT}(p)$  time. Our main technical contribution is a data reduction-based approach for

constructing small solution search spaces, which is quite different from the commonly used sampling-based methods and we believe is of independent interest.

**Author Contributions** L.-M. Liu, X.-S. Xu, and Q.-L. Feng designed the study and contributed the central idea. Z. Zhang and Z.-Y. Huang did the analyses and wrote the initial draft of the paper. All authors discussed the results and revised the manuscript.

**Conflict of interest** The authors declare no conflict of interest.

## References

- [1] Guha, S., Khuller, S.: Greedy strikes back: improved facility location algorithms. *J. Algorithm.* **31**(1), 228–248 (1999)
- [2] Cohen-Addad V., Gupta A., Kumar A., Lee E., Li J.: Tight FPT approximations for  $k$ -median and  $k$ -means. In: *The Proceedings of the 46th International Colloquium on Automata, Languages, and Programming*, pp. 42 (2019)
- [3] Cohen-Addad V., Grandoni F., Lee E., Schwiegelshohn C.: Breaching the 2 LMP approximation barrier for facility location with applications to  $k$ -median. In: *The Proceedings of the 34th ACM-SIAM Symposium on Discrete Algorithms*, pp. 940–986 (2023)
- [4] Cohen-Addad V., Esfandiari H., Mirrokni V. S., Narayanan S.: Improved approximations for Euclidean  $k$ -means and  $k$ -median, via nested quasi-independent sets. In: *The Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1621–1628 (2022)
- [5] Kumar A., Sabharwal Y.: The priority  $k$ -median problem. In: *The Proceedings of the 27th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, pp. 71–83 (2007)
- [6] Goyal, D., Jaiswal, R.: Tight FPT approximation for socially fair clustering. *Inf. Process. Lett.* **182**, 106383 (2023)
- [7] Goyal, D., Jaiswal, R.: Tight FPT approximation for constrained  $k$ -center and  $k$ -supplier. *Theor. Comput. Sci.* **940**, 190–208 (2023)
- [8] Bandyapadhyay S., Lochet W., Saurabh S.: FPT constant-approximations for capacitated clustering to minimize the sum of cluster radii. In: *The Proceedings of the 39th International Symposium on Computational Geometry*, pp. 12 (2023)
- [9] Feng Q., Zhang Z., Huang Z., Xu J., Wang J.: A unified framework of FPT approximation algorithms for clustering problems. In: *The Proceedings of the 31st International Symposium on Algorithms and Computation*, pp. 5 (2020)
- [10] Kumar, A., Sabharwal, Y., Sen, S.: Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM* **57**(2), pp. 5 (2010)
- [11] Chen K.: On  $k$ -median clustering in high dimensions. In: *The Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1177–1185 (2006)
- [12] Jaiswal, R., Kumar, A., Sen, S.: A simple  $D^2$ -sampling based PTAS for  $k$ -means and other clustering problems. *Algorithmica* **70**(1), 22–46 (2014)
- [13] Gupta, A., Krauthgamer, R., Lee, J. R.: Bounded geometries, fractals, and low-distortion embeddings. In: *The Proceedings of the 44th IEEE Annual Symposium on Foundations of Computer Science*, pp. 534–543 (2003)
- [14] Har-Peled, S., Mendel, M.: Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.* **35**(5), 1148–1184 (2006)
- [15] Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. *Contemp. Math.* **26**, 189–206 (1984)
- [16] Narayanan S., Nelson J.: Optimal terminal dimensionality reduction in Euclidean space. In: *The Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1064–1069 (2019)

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.