



# Capacitated Covering Problems in Geometric Spaces

Sayan Bandyapadhyay<sup>1</sup> · Santanu Bhowmick<sup>1</sup> · Tanmay Inamdar<sup>1</sup> ·  
Kasturi Varadarajan<sup>1</sup>

Received: 25 June 2018 / Revised: 29 July 2019 / Accepted: 13 August 2019 / Published online: 29 August 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

We consider the following capacitated covering problem. We are given a set  $P$  of  $n$  points and a set  $\mathcal{B}$  of balls from some metric space, and a positive integer  $U$  that represents the *capacity* of each of the balls in  $\mathcal{B}$ . We would like to compute a subset  $\mathcal{B}' \subseteq \mathcal{B}$  of balls and assign each point in  $P$  to some ball in  $\mathcal{B}'$  that contains it, so that the number of points assigned to any ball is at most  $U$ . The objective function that we would like to minimize is the cardinality of  $\mathcal{B}'$ . We consider this problem in arbitrary metric spaces as well as Euclidean spaces of constant dimension. In the metric setting, even the uncapacitated version of the problem is hard to approximate to within a logarithmic factor. In the Euclidean setting, the best known approximation guarantee in dimensions 3 and higher is logarithmic in the number of points. Thus we focus on obtaining “bi-criteria” approximations. In particular, we are allowed to expand the balls in our solution by some factor, but optimal solutions do not have that flexibility. Our main result is that allowing constant factor expansion of the input balls suffices to obtain constant approximations for this problem. In fact, in the Euclidean setting, only  $1 + \epsilon$  factor expansion is sufficient for any  $\epsilon > 0$ , with the approximation factor being a polynomial in  $1/\epsilon$ . We obtain these results using a unified scheme for rounding the natural LP relaxation; this scheme may be useful for other capacitated covering problems. We also complement these bi-criteria approximations by obtaining hardness of approximation results that shed light on our understanding of these problems.

**Keywords** Capacitated covering · Geometric set cover · LP rounding · Bi-criteria approximation

**Mathematics Subject Classification** 65D18 · 90C05 · 68W25 · 05B40

---

Editor in Charge: Kenneth Clarkson

---

This material is based upon work supported by the National Science Foundation under Grants CCF-1318996 and CCF-1615845.

---

Extended author information available on the last page of the article

## 1 Introduction

In this paper, we consider the following capacitated covering problem. We are given a set  $P$  of  $n$  points and a set  $\mathcal{B}$  of balls from some metric space, and a positive integer  $U$  that represents the *capacity* of each of the balls in  $\mathcal{B}$ . We would like to compute a subset  $\mathcal{B}' \subseteq \mathcal{B}$  of balls and assign each point in  $P$  to some ball in  $\mathcal{B}'$  that contains it, so that the number of points assigned to any ball is at most  $U$ . The objective function that we would like to minimize is the cardinality of  $\mathcal{B}'$ . We call this the *Metric Capacitated Covering* (MCC) problem.

An important special case of this problem arises when  $U = \infty$ , and we refer to this as *Metric Uncapacitated Covering* (MUC). The MUC requires us to cover the points in  $P$  using a minimum number of balls from  $\mathcal{B}$ , and we can therefore solve it using the efficient greedy algorithm for set cover and obtain an approximation guarantee of  $O(\log n)$ . The approximation factor of  $O(\log n)$  for Set Cover cannot be improved unless  $P = NP$  [14]. The same is true for the MUC, as demonstrated by the following reduction from Set Cover. We construct a graph with a vertex corresponding to each set and element, and an edge of length 1 between a set and an element if the element is contained in the set. In the metric induced by this graph, we create an MUC instance: we include a ball of radius 1 at each set vertex, and let the points that need to be covered be the element vertices. It is easy to see that any solution for this instance of MUC directly gives a solution for the input instance of the general Set Cover, implying that for MUC, it is not possible to get any approximation guarantee better than the  $O(\log n)$  bound for Set Cover.

The MUC in fixed dimensional Euclidean spaces has been extensively studied. One interesting variant is when the allowed set  $\mathcal{B}$  of balls consists of all *unit* balls. Hochbaum and Maass [19] gave a polynomial time approximation scheme (PTAS) for this variant using a grid shifting strategy. When  $\mathcal{B}$  is an arbitrary finite set of balls, the problem seems to be much harder. An  $O(1)$  approximation algorithm in the 2-dimensional Euclidean plane was given by Brönnimann and Goodrich [9]. More recently, a PTAS was obtained by Mustafa and Ray [27]. In dimensions 3 and higher, the best known approximation guarantee is still  $O(\log n)$ . Motivated by this, Har-Peled and Lee [18] gave a PTAS for a bi-criteria version where the algorithm is allowed to expand the input balls by a  $(1 + \epsilon)$  factor. Covering with geometric objects other than balls has also been extensively studied; see [4, 10, 12, 17, 23, 30] for a sample.

The MCC is a special case of the *Capacitated Set Cover* (CSC) problem. In the latter problem, we are given a set system  $(X, \mathcal{F})$  with  $n = |X|$  elements and  $m = |\mathcal{F}|$  subsets of  $X$ . For each set  $\mathcal{F}_i \in \mathcal{F}$ , we are also given an integer  $U_i$ , which is referred to as its *capacity*. We are required to find a minimum size subset  $\mathcal{F}' \subseteq \mathcal{F}$  and assign each element in  $X$  to a set in  $\mathcal{F}'$  containing it, so that for each set  $\mathcal{F}_i$ , the number of points assigned to  $\mathcal{F}_i$  is at most  $U_i$ . The MCC is obtained as a special case of CSC by setting  $X = P$ ,  $\mathcal{F} = \mathcal{B}$ , and  $U_i = U$  for all  $i$ . Set Cover is a special case of CSC where the capacity of each set is  $\infty$ .

Applications of Set Cover include placement of wireless sensors or antennas to serve clients, VLSI design, and image processing [7, 19]. It is natural to consider capacity constraints that appear in many applications, for instance, an upper bound on the number of clients that can be served by an antenna. Such constraints lead to

the natural formulation of CSC. For the CSC problem, Wolsey [31] used a greedy algorithm to give an  $O(\log n)$  approximation. For the special case of vertex cover (where each element in  $X$  belongs to exactly two sets in  $\mathcal{F}$ ), Chuzhoy and Naor [11] presented an algorithm with approximation ratio 3, which was subsequently improved to 2 by Gandhi et al. [15]. The generalization where each element belongs to at most a bounded number  $f$  of sets has been studied in a sequence of works, culminating in [20,32]. Berman et al. [7] have considered the “soft” capacitated version of the CSC problem that allows making multiple copies of input sets. Another closely related problem to the CSC problem is the so-called *Replica Placement* problem. For the graphs with treewidth bounded by  $t$ , an  $O(t)$  approximation algorithm for this problem is presented in [1]. Finally, PTASs for the Capacitated Dominating Set, and Capacitated Vertex Cover problems on planar graphs are presented in [6], under the assumption that the demands and capacities of the vertices are upper bounded by a constant.

Compared to MUC, relatively fewer geometric versions of the MCC problem have been studied in the literature. We refer to the version of MCC where the underlying metric is Euclidean as the *Euclidean Capacitated Covering* (ECC) problem. The dimension of the Euclidean space is assumed to be a constant. One such version arises when  $\mathcal{B}$  comprises of all possible unit balls. This problem appeared in the Sloan Digital Sky Survey project [25]. Building on the shifting strategy of Hochbaum and Maass [19], Ghasemi and Razzazi [16] obtain a PTAS for this problem. When the set  $\mathcal{B}$  of balls is arbitrary, the best known approximation guarantee is  $O(\log n)$ , even in the plane.

Given this state of affairs for MCC and ECC, we focus our efforts on finding a bi-criteria approximation. Specifically, we allow the balls in our solution to expand by at most a constant factor  $\lambda$ , without changing their capacity constraints (but optimal solution does not expand). We formalize this as follows. An  $(\alpha, \beta)$ -approximation for a version of MCC, is a solution in which the balls may be expanded by a factor of  $\beta$  (i.e., for any ball  $B_i$ , and any point  $p_j \in P$  that is assigned to  $B_i$ ,  $d(c_i, p_j) \leq \beta \cdot r_i$ —where  $r_i$  is the radius of the ball  $B_i$ ), and its cost is at most  $\alpha$  times that of an optimal solution (which does not expand the balls). From the reduction of Set Cover to MUC described above, we can see that it is NP-hard to get an  $(f(n), \lambda)$ -approximation for any  $\lambda < 3$  and  $f(n) = o(\log n)$ . This follows from the observation that in the constructed instance of MUC, the distance between a set vertex and a vertex corresponding to an element not in that set is at least 3. We note that it is a common practice in the wireless network setting to expand the radii of antennas at the planning stage to improve the quality of service. For example, Bose et al. [8] propose a scheme for replacing omni-directional antennas by directional antennas that expands the antennas by a constant factor.

*Related Work.* Capacitated versions of facility location and clustering type problems have been well-studied over the years. One such clustering problem is the capacitated  $k$ -center. In the version of this problem with uniform capacities, we are given a set  $P$  of points in a metric space, along with an integer capacity  $U$ . A feasible solution to this problem is a choice of  $k$  centers to open, together with an assignment of each point in  $P$  to an open center, such that no center is assigned more than  $U$  points. The objective is to minimize the maximum distance of a point to its assigned center.  $O(1)$  approximations are known for this problem [5,21]; the version with non-uniform capacities is addressed

in [2,13]. Notice that the decision version of the uniform capacitated  $k$ -center with the radius parameter  $r$  is the same as the decision version of a special case of MCC, where the set  $\mathcal{B}$  consists of balls of radius  $r$  centered at each point of the capacitated  $k$ -center instance. The capacity of each ball is the same as the uniform capacity  $U$  of the points. We want to find whether there is a subset of  $\mathcal{B}$  consisting of  $k$  balls that can serve all the points without violating the capacity constraint. There have been recent developments on the capacitated versions of related optimization problems. For the metric facility location problem with non-uniform capacities, constant approximations were known via the local search technique [26,28]. For the special case of capacitated facility location problem, where the opening costs are uniform, Levi et al. [22] gave an LP rounding based constant approximation. However, for the general case with non-uniform opening costs, the first LP-based constant approximation was given by An et al. [3], who need to strengthen the natural LP relaxation. There has also been some recent work on the capacitated  $k$ -median problem [24].

## 1.1 Our Results and Contributions

In this article, we make significant progress on both the MCC and ECC problems.

- We present a  $(21, 6.47)$ -approximation for the MCC problem. Thus, if we are allowed to expand the input balls by a constant factor, we can obtain a solution that uses at most 21 times the number of balls used by the optimal solution. We note that we have not tried to optimize the approximation guarantee of 21. As noted above, if we are not allowed to expand by a factor of at least 3, we are faced with a hardness of approximation of  $\Omega(\log n)$ .
- We present an  $(O(\epsilon^{-4d} \log(1/\epsilon)), 1 + \epsilon)$ -approximation for the ECC problem in  $\mathbb{R}^d$ . Thus, assuming we are allowed to expand the input balls by an arbitrarily small constant factor, we can obtain a solution with at most a corresponding constant times the number of balls used by the optimal solution. Without expansion, the best known approximation guarantee for  $d \geq 3$  is  $O(\log n)$ , even without capacity constraints.

Both results are obtained via a unified scheme for rounding the natural LP relaxation for the problem. This scheme, which is instantiated in different ways to obtain the two results, may be of independent interest for obtaining similar results for related capacitated covering problems. Though the LP rounding technique is a standard tool that has been used in the literature of the capacitated problems, our actual rounding scheme is different from the existing ones. In fact, the standard rounding scheme for facility location, for example the one in [22], is not useful for our problems, as there a point can be assigned to any facility. But in our case, each point must be assigned to a ball that contains it (modulo constant factor expansion). This hard constraint makes the covering problems more complicated to deal with.

When the input balls have the same radius, it is easier to obtain the above guarantees for the MCC and the ECC using known results or techniques. For the MCC, this (in fact, even a  $(1, O(1))$ -approximation) follows from the results for capacitated  $k$ -center [2,5,13,21]. This is because of the connection between capacitated  $k$ -center and MCC as pointed out before. The novelty in our work lies in handling the challenging

scenario where the input balls have widely different radii. For geometric optimization problems, inputs with objects at multiple scales are often more difficult to handle than inputs with objects at the same scale.

As a byproduct of the rounding schemes we develop, the bicriteria approximations can be extended to a more general capacity model. In this model, the capacities of the balls are not necessarily the same. In particular, suppose ball  $B_i$  has capacity  $U_i$  and radius  $r_i$ . Then for any two balls  $B_i, B_j \in \mathcal{B}$ , our model assumes that the following holds:  $r_i > r_j \Rightarrow U_i \geq U_j$ . We refer to this capacity model as the *monotonic capacity model*. We refer to the generalizations of the MCC and the ECC problems with the monotonic capacity model as the *Metric Monotonic Capacitated Covering* (MMCC) problem and the *Euclidean Monotonic Capacitated Covering* (EMCC) problem, respectively. We note that the monotonicity assumption on the capacities is reasonable in many applications such as wireless networks—it might be economical to invest in capacity of an antenna to serve more clients, if it covers more area.

**Hardness.** We complement our algorithmic results with some hardness of approximation results that give a better understanding of the problems we consider. Firstly, we show that for any constant  $c > 1$ , there exists a constant  $\epsilon_c > 0$  such that it is NP-hard to obtain a  $(1 + \epsilon_c, c)$ -approximation for the MCC problem, even when the capacity of all balls is 3. This shows that it is not possible to obtain a  $(1, c)$  approximation even for an arbitrarily large constant  $c$ . In the hardness construction, not all the balls in the hard instance have the same radius. This should be contrasted with the case where the radii of all balls are equal—in this case one can use the results from capacitated  $k$ -center (such as [2,13]), to obtain a  $(1, O(1))$ -approximation.

It is natural to wonder if our algorithmic results can be extended to weighted versions of the problems. We derive hardness results that indicate that this is not possible. In particular, we show that for any constant  $c \geq 1$ , there exists a constant  $c' > 0$ , such that it is NP-hard to obtain a  $(c' \log n, c)$ -approximation for the weighted version of MMCC with a very simple weight function (a constant power of original radius).

We describe the natural LP relaxation for the MMCC problem in Sect. 2. We describe a unified rounding scheme in Sect. 3, and apply it in two different ways to obtain the algorithmic guarantees for MMCC and EMCC. We present the hardness results in Sect. 4.

## 2 LP Relaxation for MMCC

Recall that the input for the MMCC consists of a set  $P$  of points and a set  $\mathcal{B}$  of balls in some metric space, along with an integer capacity  $U_i > 0$  for ball  $B_i \in \mathcal{B}$ . We assume that for any two input balls  $B_i, B_j \in \mathcal{B}$ , it holds that  $r_i > r_j \Rightarrow U_i \geq U_j$ . The goal is to compute a minimum cardinality subset  $\mathcal{B}' \subseteq \mathcal{B}$  for which each point in  $P$  can be assigned to a ball  $B' \in \mathcal{B}'$  containing it in such a way that no more than  $U_i$  points are assigned to ball  $B_i$ . Let  $d(p, q)$  denote the distance between two points  $p$  and  $q$  in the metric space. Let  $B(c, r)$  denote the ball of radius  $r$  centered at point  $c$ . We let  $c_i$  and  $r_i$  denote the center and radius of ball  $B_i \in \mathcal{B}$ ; thus,  $B_i = B(c_i, r_i)$ .

First we consider an integer programming formulation of MMCC. For each set  $B_i \in \mathcal{B}$ , let  $y_i = 1$  if the ball  $B_i$  is selected in the solution, and 0 otherwise. Similarly, for each point  $p_j \in P$  and each ball  $B_i \in \mathcal{B}$ , let the variable  $x_{ij} = 1$  if  $p_j$  is assigned to  $B_i$ , and  $x_{ij} = 0$  otherwise. We relax these integrality constraints, and state the corresponding linear program as follows:

$$\text{minimize} \quad \sum_{B_i \in \mathcal{B}} y_i \quad (\text{MMCC-LP})$$

$$\text{s.t.} \quad x_{ij} \leq y_i \quad \forall p_j \in P, \forall B_i \in \mathcal{B}, \quad (1)$$

$$\sum_{p_j \in P} x_{ij} \leq y_i \cdot U_i \quad \forall B_i \in \mathcal{B}, \quad (2)$$

$$\sum_{B_i \in \mathcal{B}} x_{ij} = 1 \quad \forall p_j \in P, \quad (3)$$

$$x_{ij} = 0 \quad \forall p_j \in P, \forall B_i \in \mathcal{B} \text{ such that } p_j \notin \mathcal{B}_i, \quad (4)$$

$$x_{ij} \geq 0 \quad \forall p_j \in P, \forall B_i \in \mathcal{B}, \quad (5)$$

$$0 \leq y_i \leq 1 \quad \forall B_i \in \mathcal{B}. \quad (6)$$

Subsequently, we will refer to an assignment  $(x, y)$  that is feasible or infeasible with respect to Constraints (1)–(6) as just a *solution*. The *cost* of the LP solution  $\sigma = (x, y)$  (feasible or otherwise), denoted by  $\text{cost}(\sigma)$ , is defined as  $\sum_{B_i \in \mathcal{B}} y_i$ .

### 3 The Algorithmic Framework

In this section, we describe our framework for extracting an integral solution from a fractional solution to the above LP. The framework consists of two major steps—Preprocessing and the Main Rounding. The Main Rounding step is in turn divided into two smaller steps—Cluster Formation and Selection of Objects. For simplicity of exposition, we first describe the framework with respect to the MMCC problem as an algorithm and analyze the approximation factor achieved by this algorithm for MMCC. Later, we show how one or more steps of this algorithm can be modified to obtain the desired results for the EMCC.

#### 3.1 The Algorithm for the MMCC Problem

Before we describe the algorithm we introduce some definitions and notation which will heavily be used throughout this section. For point  $p_j \in P$  and ball  $B_i \in \mathcal{B}$ , we refer to  $x_{ij}$  as the *flow* from  $B_i$  to  $p_j$ ; if  $x_{ij} > 0$ , then we say that the ball  $B_i$  *serves* the point  $p_j$ . Each ball  $B_i \in \mathcal{B}$  can be imagined as a source of at most  $y_i \cdot U_i$  units of flow, which it distributes to some points in  $P$ .

We now define an important operation, called *rerouting of flow*. “Rerouting of flow for a set  $P' \subseteq P$  of points from a set of balls  $\mathcal{B}'$  to a ball  $B_k \notin \mathcal{B}'$ ” means obtaining a

new solution  $(\hat{x}, \hat{y})$  from the current solution  $(x, y)$  in the following way: (a) For all points  $p_j \in P'$ ,  $\hat{x}_{kj} = x_{kj} + \sum_{B_i \in \mathcal{B}'} x_{ij}$ ; (b) for all points  $p_j \in P'$  and balls  $B_i \in \mathcal{B}'$ ,  $\hat{x}_{ij} = 0$ ; (c) the other  $\hat{x}_{ij}$  variables are the same as the corresponding  $x_{ij}$  variables. The relevant  $\hat{y}_i$  variables may also be modified depending on the context where this operation is used.

Let  $0 < \alpha \leq 1/2$  be a parameter to be fixed later. A ball  $B_i \in \mathcal{B}$  is *heavy* if the corresponding  $y_i = 1$ , and *light*, if  $0 < y_i \leq \alpha$ . Corresponding to a feasible LP solution  $(x, y)$ , let  $\mathcal{H} = \{B_i \in \mathcal{B} \mid y_i = 1\}$  denote the set of heavy balls, and  $\mathcal{L} = \{B_i \in \mathcal{B} \mid 0 < y_i \leq \alpha\}$  denote the set of light balls. We emphasize that the set  $\mathcal{L}$  of light and  $\mathcal{H}$  of heavy balls are defined w.r.t. an LP solution; however, the reference to the LP solution may be omitted when it is clear from the context.

Now we move on towards the description of the algorithm. The algorithm, given a feasible fractional solution  $\sigma = (x, y)$ , rounds  $\sigma$  to a solution  $\hat{\sigma} = (\hat{x}, \hat{y})$  such that  $\hat{y}$  is integral, and the cost of  $\hat{\sigma}$  is within a constant factor of the cost of  $\sigma$ . The  $\hat{x}$  variables are non-negative but may be fractional. Furthermore, each point receives unit flow from the balls that are chosen ( $y$  values are 1), and the amount of flow each chosen ball sends is bounded by its capacity. Notably, no point gets any non-zero amount of flow from a ball that is not chosen ( $y$  value is 0). Moreover, for any ball  $B_i$  and any  $p_j \in P$ , if  $B_i$  serves  $p_j$ , then  $d(c_i, p_j)$  is at most a constant times  $r_i$ . We expand each ball by a constant factor so that it contains all the points it serves.

We note that in  $\hat{\sigma}$  points might receive fractional amount of flow from the chosen balls. However, since the capacity of each ball is integral, we can find, using a textbook argument for integrality of flow, another solution with the same set of chosen balls, such that the new solution satisfies all the properties of  $\hat{\sigma}$  and the additional property, that for each point  $p$ , there is a single chosen ball that sends one unit of flow to  $p$  [11]. Thus, choosing an optimal LP solution as the input  $\sigma = (x, y)$  of the rounding algorithm yields a constant approximation for MMCC by expanding each ball by at most a constant factor.

Our LP rounding algorithm consists of two steps. The first step is a preprocessing step where we construct a fractional LP solution  $\bar{\sigma} = (\bar{x}, \bar{y})$  from  $\sigma$ , such that each ball in  $\bar{\sigma}$  is either heavy or light, and for each point  $p_j \in P$ , the amount of flow that  $p_j$  can potentially receive from the light balls is at most  $\alpha$ . The latter property will be heavily exploited in the next step. The second step is the core step of the algorithm where we round  $\bar{\sigma}$  to the desired integral solution.

We note that throughout the algorithm, for any intermediate LP solution that we consider, we maintain the following two invariants: (i) Each ball  $B_i$  sends at most  $U_i$  units of flow to the points, and (ii) Each point receives exactly one unit of flow from the balls. With respect to a solution  $\sigma = (x, y)$ , we define the *available capacity* of a ball  $B_i \in \mathcal{B}$ , denoted  $\text{AvCap}(B_i)$ , to be  $U_i - \sum_{p_j \in P} x_{ij}$ . We now describe the preprocessing step.

### 3.1.1 The Preprocessing Step

**Lemma 3.1** *Given a feasible LP solution  $\sigma = (x, y)$ , and a parameter  $0 < \alpha \leq 1/2$ , there exists a polynomial time algorithm to obtain another LP solution  $\bar{\sigma} = (\bar{x}, \bar{y})$  that*



satisfies all the constraints of **MMCC-LP** (Constraints (1)–(6)), except Constraint (4). Additionally,  $\bar{\sigma}$  satisfies the following properties.

1. Any ball  $B_i \in \mathcal{B}$  with non-zero  $\bar{y}_i$  is either heavy ( $\bar{y}_i = 1$ ) or light ( $0 < \bar{y}_i \leq \alpha$ ).
2. For each point  $p_j \in P$ , we have that

$$\sum_{B_i \in \mathcal{L}: \bar{x}_{ij} > 0} \bar{y}_i \leq \alpha, \quad (7)$$

where  $\mathcal{L}$  is the set of light balls with respect to  $\bar{\sigma}$ .

3. For any heavy ball  $B_i$ , and any point  $p_j \in P$  served by  $B_i$ ,  $d(c_i, p_j) \leq 3r_i$ .
4. For any light ball  $B_i$ , and any point  $p_j \in P$  served by  $B_i$ ,  $d(c_i, p_j) \leq r_i$ .
5.  $\text{cost}(\bar{\sigma}) \leq \text{cost}(\sigma)/\alpha$ .

**Proof** The algorithm starts off by initializing  $\bar{\sigma}$  to  $\sigma$ . While there is a violation of inequality (7), we perform the following steps.

1. We pick an arbitrary point  $p_j \in P$ , for which inequality (7) is not met. Let  $\mathcal{L}_j$  be a subset of light balls serving  $p_j$  such that  $\alpha < \sum_{B_i \in \mathcal{L}_j} \bar{y}_i \leq 2\alpha$ . Note that such a set  $\mathcal{L}_j$  always exists because the  $\bar{y}_i$  variables corresponding to light balls are at most  $\alpha \leq 1/2$ . Let  $B_k$  be a ball with the largest radius from the set  $\mathcal{L}_j$ . (If there is more than one ball with the largest radius, we consider one having the largest capacity among those. Throughout the paper we follow this convention.) Since  $r_k \geq r_m$  for all other balls  $B_m \in \mathcal{L}_j$ , we have, by the *monotonicity* assumption, that  $U_k \geq U_m$ .
2. We set  $\bar{y}_k \leftarrow \sum_{B_i \in \mathcal{L}_j} \bar{y}_i$ , and  $\bar{y}_m \leftarrow 0$  for all  $B_m \in \mathcal{L}_j \setminus \{B_k\}$ . Note that  $\bar{y}_k \leq 2\alpha \leq 1$ . Let  $A = \{p_t \in P \mid \bar{x}_{it} > 0 \text{ for some } B_i \in \mathcal{L}_j \setminus \{B_k\}\}$  be the set of “affected” points. We reroute the flow for all the affected points in  $A$  from  $\mathcal{L}_j \setminus \{B_k\}$  to the ball  $B_k$ . Since  $U_k \geq U_m$  for all other balls  $B_m \in \mathcal{L}_j$ ,  $B_k$  has enough available capacity to “satisfy” all “affected” points. In  $\bar{\sigma}$ , all other  $\bar{x}_{ij}$  and  $\bar{y}_i$  variables remain same as before. (Note: Since  $B_k$  had the largest radius from the set  $\mathcal{L}_j$ , all the points in  $A$  are within distance  $3r_k$  from its center  $c_k$ , as seen using the triangle inequality. Also, since  $\bar{y}_k > \alpha$ ,  $B_k$  is no longer a light ball.)

Finally, for all balls  $B_i$  such that  $\bar{y}_i > \alpha$ , we set  $\bar{y}_i = 1$ , making them heavy. Thus  $\text{cost}(\bar{\sigma})$  is at most  $1/\alpha$  times  $\text{cost}(\sigma)$ , and  $\bar{\sigma}$  satisfies all the conditions stated in the lemma.  $\square$

**Remark** As a byproduct of Lemma 3.1, we get a simple (4, 3)-approximation algorithm for the *soft* capacitated version of our problem that allows making multiple copies of the input balls. We remind the reader that in this variant, we are allowed to open multiple identical copies of the given ball at the same location, and each such ball has the same capacity as the original ball. However, we need to pay a cost of 1 for each copy. The LP corresponding to the soft capacitated version, is the same as **MMCC-LP**, except that Constraint (6) is relaxed to simply  $y_i \geq 0$ . We solve this LP, and obtain an optimal solution  $(x^*, y^*)$ . Then, using the procedure from Lemma 3.1, we can ensure that the flow that each point receives from the set of *non-light* balls ( $\mathcal{B} \setminus \mathcal{L}$ ) is at least  $1 - \alpha$ . Then, opening  $\lceil y_i \rceil / (1 - \alpha)$  identical copies of each non-light ball  $B_i$  ensures



that for each point, at least one unit of demand is satisfied exclusively by these balls. We now expand each of the opened balls by a factor of 3. As  $y_i \geq \alpha$  for each non-light ball  $B_i$ , choosing  $\alpha = 1/2$  yields a simple 4-approximation for this version, where each ball is expanded by a factor of at most 3.

### 3.1.2 The Main Rounding Step

The main rounding step can be logically divided into two stages. The first stage, *Cluster Formation*, is the crucial stage of the algorithm. Note that there can be many light balls in the preprocessed solution. Including all these balls in the final solution may incur a huge cost. Thus we use a careful strategy based on flow rerouting to select a small number of balls. The idea is to use the capacity of a selected light ball to reroute as much flow as possible from other intersecting balls. This in turn frees up some capacity at those balls. The available capacity of each heavy ball is used, when possible, to reroute *all* the flow from some light ball intersecting it; this light ball is then added to a cluster centered around the heavy ball. Notably, for each cluster, the heavy ball is the only ball in it that actually serves some points, as we have rerouted flow from the other balls in the cluster to the heavy ball. In the second stage, referred to as *Selection of Objects*, we select exactly one ball (in particular, a largest ball) from each cluster as part of the final solution, and reroute the flow from the heavy ball to this ball, and expand it by the required amount. Together, these two stages ensure that we do not end up choosing many light balls.

We now describe the two stages in detail. Recall that any ball in the preprocessed solution is either heavy or light. Also  $\mathcal{L}$  denotes the set of light balls and  $\mathcal{H}$  the set of heavy balls. Note that any heavy ball  $B_i$  may serve a point  $p_j$  which is at a distance  $3r_i$  from  $c_i$ . We expand each heavy ball by a factor of 3 so that  $B_i$  can contain all points it serves.

**1. Cluster Formation.** In this stage, each light ball will be added to either a set  $\mathcal{O}$  (that will eventually be part of the final solution), or a cluster corresponding to some heavy ball. Till the very end of this stage, the sets of heavy and light balls remain unchanged. The set  $\mathcal{O}$  is initialized to  $\emptyset$ . For each heavy ball  $B_i$ , we initialize the cluster of  $B_i$ , denoted by  $\text{cluster}(B_i)$  to  $\{B_i\}$ . We say a ball is clustered if it is added to a cluster. At any point, let  $\Lambda$  denote the set consisting of each light ball that is (a) not in  $\mathcal{O}$ , and (b) not yet clustered. While the set  $\Lambda$  is non-empty, we perform the following steps.

(a) While there is a heavy ball  $B_i$  and a light ball  $B_t \in \Lambda$  such that (1)  $B_t$  intersects  $B_i$ ; and (2)  $\text{AvCap}(B_i)$  is at least the flow  $\sum_{p_j \in P} \bar{x}_{tj}$  out of  $B_t$ :

1. For all the points served by  $B_t$ , we reroute the flow from  $B_t$  to  $B_i$ .
2. We add  $B_t$  to  $\text{cluster}(B_i)$ .

After the execution of this *while* loop, if the set  $\Lambda$  becomes empty, we stop and proceed to the *Selection of Objects* stage. Otherwise, we proceed to the following.

(b) For any ball  $B_j \in \Lambda$ , let  $\mathcal{A}_j$  denote the set of points currently being served by  $B_j$ . Also, for  $B_j \in \Lambda$ , let  $k_j = \min\{U_j, |\mathcal{A}_j|\}$ , i.e.,  $k_j$  denotes the minimum

- of its capacity, and the number of points that it currently serves. We select the ball  $B_t \in \Lambda$  with the maximum value of  $k_j$ , and add it to the set  $\mathcal{O}$ .
- (c) Since we have added  $B_t$  to the set  $\mathcal{O}$  that will be among the selected balls, we use the available capacity at  $B_t$  to reroute flow to it. This is done based on the following three cases depending on the value of  $k_t$ .

1.  $k_t = |\mathcal{A}_t| \leq U_t$ . In this case, for each point  $p_l$  in  $B_t$  that gets served by  $B_t$ , we reroute the flow of  $p_l$  from  $\mathcal{B} \setminus \mathcal{O}$  to  $B_t$ . Note that after the rerouting,  $p_l$  is no longer being served by a ball in  $\Lambda$ . The rerouting increases the available capacity of other balls intersecting  $B_t$ . In particular, for each  $B_i \in \mathcal{H}$ ,  $\text{AvCap}(B_i)$  increases by  $\sum_{p_l: B_t \text{ serves } p_l} \bar{x}_{il}$ .

2.  $k_t = U_t < |\mathcal{A}_t|$ , but  $k_t = U_t > 1$ . Observe that the flow out of ball  $B_t$  is  $\sum_{p_j \in \mathcal{A}_t} x_{tj} \leq \alpha U_t$ ; thus  $\text{AvCap}(B_t) \geq (1 - \alpha)U_t = (1 - \alpha)k_t$ .

In this case, we select a point  $p_j \in \mathcal{A}_t$  arbitrarily, and reroute the flow of  $p_j$  from  $\mathcal{B} \setminus \mathcal{O}$  to  $B_t$ . This will increase the available capacity of other balls in  $\mathcal{B} \setminus \mathcal{O}$  that were serving  $p_j$ . Also note that  $p_j$  is no longer being served by a ball in  $\Lambda$ .

We repeat the above flow rerouting process for other points of  $\mathcal{A}_t$  until we encounter a point  $p_l$  such that rerouting the flow of  $p_l$  from  $\mathcal{B} \setminus \mathcal{O}$  to  $B_t$  violates the capacity of  $B_t$ . Thus the flow assignment of  $p_l$  remains unchanged. Note that we can reroute the flow of at least  $\lfloor (1 - \alpha)k_t \rfloor = \lfloor (1 - \alpha)U_t \rfloor \geq 1$  points of  $\mathcal{A}_t$  in this manner, since  $U_t > 1$  and  $\alpha \leq 1/2$ .

3.  $k_t = U_t = 1 < |\mathcal{A}_t|$ . Note that  $B_t$  has used  $\sum_{p_j \in \mathcal{A}_t} x_{tj} \leq \alpha U_t = \alpha$  capacity. In this case, we pick a point  $p_j \in \mathcal{A}_t$  arbitrarily, and then perform the following two steps:

- (i) Reroute the flow of  $p_j$  from  $\Lambda$  to  $B_t$ ; after this,  $p_j$  is no longer being served by a ball in  $\Lambda$ . Note that in this step, we reroute at most  $\alpha$  amount of flow—this follows from the LP Constraint (1) and Property (7) as maintained in Lemma 3.1. Therefore, at this point we have  $\text{AvCap}(B_t) \geq 1 - 2\alpha$ . Let  $f$  be the amount of flow  $p_j$  receives from the balls in  $\mathcal{O}$ .
- (ii) Then we reroute  $\min\{\text{AvCap}(B_t), 1 - f\}$  amount of flow of  $p_j$  from the set  $\mathcal{H}$  to  $B_t$ .

When the loop terminates, we have that each light ball is either in  $\mathcal{O}$  or clustered. We set  $\bar{y}_i \leftarrow 1$  for each ball  $B_i \in \mathcal{O}$ , thus making it heavy. For convenience, we also set  $\text{cluster}(B_i) = \{B_i\}$  for each  $B_i \in \mathcal{O}$ . Note that, throughout the algorithm we ensure that, if a point  $p_j \in P$  is currently served by a ball  $B_i \in \Lambda$ , then the amount of flow  $p_j$  receives from any ball  $B_{i'} \in \mathcal{O} \cup \Lambda$  is the same as that in the preprocessed solution, i.e., the flow assignment of  $p_j$  w.r.t. the balls in  $\mathcal{O} \cup \Lambda$  remains unchanged.

2. **Selection of Objects.** At the start of this stage, we have a collection of clusters, each centered around a heavy ball, such that the light balls in each cluster intersect the heavy ball. We are going to pick exactly one ball from each cluster and add it to a set  $\mathcal{C}$ . Let  $\mathcal{C} = \emptyset$  initially. For each heavy ball  $B_i$ , we consider  $\text{cluster}(B_i)$  and perform the following steps.

- (a) If  $\text{cluster}(B_i)$  consists of only the heavy ball, we add  $B_i$  to  $\mathcal{C}$ .
- (b) Otherwise, let  $B_j$  be a largest ball in  $\text{cluster}(B_i)$ . If  $B_j = B_i$ , then we expand it by a factor of 3. Otherwise,  $B_j$  is a light ball intersecting with  $B_i$ , in which case we expand it by a factor of 5. In this case, we also reroute the flow from the heavy ball to the selected ball  $B_j$ . Note that since we always choose a largest ball in the cluster, its capacity is at least that of the heavy ball, because of the *monotonicity* assumption. We add  $B_j$  to  $\mathcal{C}$ , and we set  $\bar{y}_s \leftarrow 0$  for any other ball  $B_s$  in the cluster.

Note that we always select a largest ball in from  $\text{cluster}(B_i)$ . In the first case, the selected ball  $B_i$  is also the center of the cluster. From the triangle inequality, it is easy to see that an expansion by a factor of 3 is sufficient to cover all balls in  $\text{cluster}(B_i)$ . Otherwise, in the second case, the selected ball intersects the center of the cluster,  $B_i$ . Again, from triangle inequality, it can be seen that an expansion by a factor of 5 suffices.

After processing the clusters, we set  $\bar{y}_t \leftarrow 1$  for each ball  $B_t \in \mathcal{C}$ . Finally, we return the current set of heavy balls (i.e.,  $\mathcal{C}$ ) as the set of selected balls. Note that the flow out of each such ball is at most its capacity, and each point receives one unit of flow from the (possibly expanded) balls that contain it. As mentioned earlier, this can be converted into an integral flow.

### 3.1.3 The Analysis of the Rounding Algorithm

Let  $OPT$  be the cost of an optimal solution. We establish a bound on the number of balls our algorithm outputs by bounding the size of the set  $\mathcal{C}$ . Then we conclude by showing that any input ball that is part of our solution expands by at most a constant factor to cover the points it serves.

For notational convenience, we refer to the solution  $\bar{\sigma} = (\bar{x}, \bar{y})$  at hand after preprocessing, as  $\sigma = (x, y)$ . Now we bound the size of the set  $\mathcal{O}$  computed during Cluster Formation. The basic idea is that each light ball added to  $\mathcal{O}$  creates significant available capacity in the heavy balls. Furthermore, whenever there is enough available capacity, a heavy ball clusters intersecting light balls, thus preventing them from being added to  $\mathcal{O}$ . The actual argument is more intricate because we need to work with a notion of  $y$ -accumulation, a proxy for available capacity. The way the light balls are picked for addition to  $\mathcal{O}$  plays a crucial role in the argument.

Let  $\mathcal{H}_1$  (resp.  $\mathcal{L}_1$ ) be the set of heavy (resp. light) balls after preprocessing, and  $I$  be the total number of iterations in the Cluster Formation stage. Also let  $L_j$  be the light ball selected (i.e., added to  $\mathcal{O}$ ) in iteration  $j$  for  $1 \leq j \leq I$ . Now,  $L_t$  maximizes  $k_j$  amongst all balls from  $\Lambda$  in iteration  $t$  (Recall that  $k_j$  was defined as the minimum of the number of points being served by  $L_j$ , and its capacity). Note that  $k_1 \geq k_2 \geq \dots \geq k_I$ . For any  $B_i \in \mathcal{H}_1$ , denote by  $F(L_t, B_i)$ , the total amount of flow rerouted in iteration  $t$  from  $B_i$  to  $L_t$  corresponding to the points  $B_i$  serves. This is the same as the increase in  $\text{AvCap}(B_i)$  when  $L_t$  is added to  $\mathcal{O}$ . Correspondingly, we define  $Y(L_t, B_i)$ , the “ $y$ -credit contributed by  $L_t$  to  $B_i$ ”, to be  $F(L_t, B_i)/k_t$ . Now, the increase in available capacity over all balls in  $\mathcal{H}_1$  is  $F_t = \sum_{B_i \in \mathcal{H}_1} F(L_t, B_i)$ . The approximation guarantee of the algorithm depends crucially on the following simple

lemma, which states that in each iteration we make “sufficiently large” amount of flow available for the set of heavy balls.

**Lemma 3.2** *Consider a ball  $B_t \in \mathcal{O}$  processed in the step (c) of the Cluster Formation stage. For  $0 < \alpha \leq 3/8$ ,  $F_t \geq k_t/5$ .<sup>1</sup>*

**Proof** The algorithm ensures that the flow assignment of each point in  $\mathcal{A}_t$  w.r.t. the balls in  $\mathcal{O} \cup \Lambda$  is the same as that in the preprocessed solution. Thus by Property 2 of Lemma 3.1, each such point gets at most  $\alpha$  amount of flow from the balls in  $\mathcal{O} \cup \Lambda$ . Now there are three cases corresponding to the three substeps of step (c).

1.  $k_t = |\mathcal{A}_t| \leq U_t$ . For each point in  $\mathcal{A}_t$ , at most  $\alpha$  amount of flow comes from the balls in  $\mathcal{O} \cup \Lambda$ . So the remainder is rerouted from the balls in  $\mathcal{H}_1$  resulting in a contribution of at least  $1 - \alpha$  towards  $F_t$ . Therefore, we get that  $F_t \geq (1 - \alpha)k_t \geq k_t/5$ , since  $0 < \alpha \leq 3/8$ .
2.  $1 < k_t = U_t < |\mathcal{A}_t|$ . It is possible to reroute the flow of at least  $\lfloor (1 - \alpha)U_t \rfloor = \lfloor (1 - \alpha)k_t \rfloor$  points of  $\mathcal{A}_t$  from  $\mathcal{B} \setminus \mathcal{O}$  to  $B_t$ . Therefore, we get that  $F_t \geq (1 - \alpha)\lfloor (1 - \alpha)k_t \rfloor$ . When  $k_t > 1$ , the previous quantity is at least  $k_t/5$ , again by using the fact that  $0 < \alpha \leq 3/8$ .
3. When  $1 = k_t = U_t < |\mathcal{A}_t|$ ,  $F_t \geq (1 - 2\alpha) \geq k_t/5$ , as  $0 < \alpha \leq 3/8$ .  $\square$

Consider any heavy ball  $B_i \in \mathcal{H}_1$ . It gains  $y$ -credit when a light ball is added to  $\mathcal{O}$ , and loses  $y$ -credit when it adds a light ball to  $\text{cluster}(B_i)$ . We define a quantity called the  $y$ -accumulation of  $B_i$ —a proxy for the available capacity of  $B_i$ —which indicates the “remaining”  $y$ -credit of  $B_i$ . Formally, at any moment in the Cluster Formation stage, and any heavy ball  $B_i \in \mathcal{H}_1$ , define its  $y$ -accumulation as:

$$\tilde{y}(B_i) = \left( \sum_{L_t \in \mathcal{O}} Y(L_t, B_i) \right) - \left( \sum_{B_j \in \mathcal{L}_1 \cap \text{cluster}(B_i)} y_j \right).$$

The next lemma gives a relation between the  $y$ -accumulation of  $B_i$  and its available capacity.

**Lemma 3.3** *Fix a heavy ball  $B_i \in \mathcal{H}_1$ , and an integer  $1 \leq t \leq I$ . Suppose that  $L_1, L_2, \dots, L_t$  have been added to  $\mathcal{O}$ . Then  $\text{AvCap}(B_i) \geq \tilde{y}(B_i) \cdot k_t$ .*

**Proof** The proof is by induction on  $t$ . For this proof, we abbreviate  $\text{AvCap}(B_i)$  by  $A_i$ . In the first iteration, just after adding  $L_1$ ,  $A_i \geq F(L_1, B_i) = Y(L_1, B_i) \cdot k_1 \geq \tilde{y}(B_i) \cdot k_1$ .

Assume inductively that we have added balls  $L_1, \dots, L_{t-1}$  to the set  $\mathcal{O}$ , and that just after adding  $L_{t-1}$ , the claim is true. That is, if  $\tilde{y}(B_i)$  and  $A_i$  are, respectively, the  $y$ -accumulation and the available capacity of  $B_i$  just after adding  $L_{t-1}$ , then  $A_i \geq \tilde{y}(B_i) \cdot k_{t-1}$ .

Consider the iteration  $t$ . At step (a) of Cluster Formation,  $B_i$  uses up some of its available capacity to add 0 or more balls to  $\text{cluster}(B_i)$ , after which at step (b) we add  $L_t$  to  $\mathcal{O}$ . Suppose that at step (a), one or more balls are added to  $\text{cluster}(B_i)$ . Let  $B_j$  be

<sup>1</sup> We note that the factor of  $1/5$  in Lemma 3.2 is not tight, and can be improved slightly.

the first such ball, and let  $k$  and  $U_j$  be the number of points  $B_j$  serves and the capacity of  $B_j$ , respectively. Then the amount of capacity used by  $B_j$  is at most

$$\min\{U_j \cdot y_j, k \cdot y_j\} = \min\{U_j, k\} \cdot y_j \leq k_{t-1} \cdot y_j,$$

where the last inequality follows because of the order in which we add balls to  $\mathcal{O}$ . Now, after adding  $B_j$  to  $\text{cluster}(B_i)$ , the new  $y$ -accumulation becomes  $\tilde{y}(B_i)' = \tilde{y}(B_i) - y_j$ . As for the available capacity,

$$\begin{aligned} A'_i &\geq A_i - k_{t-1} \cdot y_j \geq (\tilde{y}(B_i) \cdot k_{t-1}) - k_{t-1} \cdot y_j \\ &= (\tilde{y}(B_i) - y_j) \cdot k_{t-1} = \tilde{y}(B_i)' \cdot k_{t-1} \end{aligned}$$

Therefore, the claim is true after addition of the first ball  $B_j$ . Note that  $B_i$  may add multiple balls to  $\text{cluster}(B_i)$ , and the preceding argument would work after each such addition.

Now consider the moment when  $L_t$  is added to  $\mathcal{O}$ . Let  $\tilde{y}(B_i)$  denote the  $y$ -accumulation just before this. Now, the new  $y$ -accumulation of  $B_i$  becomes  $\tilde{y}(B_i)' = \tilde{y}(B_i) + Y(L_t, B_i)$ . If  $\tilde{y}(B_i) \leq 0$ , then the new available capacity is

$$A'_i \geq F(L_t, B_i) = Y(L_t, B_i) \cdot k_t \geq \tilde{y}(B_i)' \cdot k_t.$$

If  $\tilde{y}(B_i) > 0$ , the new available capacity, using the inductive hypothesis, is

$$A'_i \geq \tilde{y}(B_i) \cdot k_{t-1} + Y(L_t, B_i) \cdot k_t \geq (\tilde{y}(B_i) + Y(L_t, B_i)) \cdot k_t = \tilde{y}(B_i)' \cdot k_t$$

where, in the second inequality we use  $k_t \leq k_{t-1}$ .  $\square$

Now, in the next lemma, we show that any ball  $B_i \in \mathcal{H}_1$  cannot have “too-much”  $y$ -accumulation at any moment during Cluster Formation.

**Lemma 3.4** *At any moment in the Cluster Formation stage, for any ball  $B_i \in \mathcal{H}_1$ , we have that  $\tilde{y}(B_i) \leq 1 + \alpha$ .*

**Proof** The proof is by contradiction. Let  $B_i \in \mathcal{H}_1$  be the first ball that violates the condition. As  $\tilde{y}(B_i)$  increases only due to addition of a light ball to set  $\mathcal{O}$ , suppose  $L_t$  was the ball whose addition to  $\mathcal{O}$  resulted in the violation.

Let  $\tilde{y}(B_i)$  and  $\tilde{y}(B_i)' = \tilde{y}(B_i) + Y(L_t, B_i)$  be the  $y$ -accumulations of  $B_i$  just before and just after the addition of  $L_t$ . Because of the assumption,  $\tilde{y}(B_i) \leq 1 + \alpha$ . So the increase in the  $y$ -accumulation of  $B_i$  must be because  $Y(L_t, B_i) > 0$ . Thus,  $L_t$  intersects  $B_i$ . However,  $Y(L_t, B_i) \leq 1$  by definition. Therefore, we have  $\tilde{y}(B_i) > \alpha$ .

Now, by Lemma 3.3, just before addition of  $L_t$ ,  $\text{AvCap}(B_i) \geq \tilde{y}(B_i) \cdot k_{t-1} > \alpha \cdot k_{t-1} \geq \alpha \cdot k_t$ , as  $k_t \leq k_{t-1}$ . However,  $L_t$  is a light ball, and so the total flow out of  $L_t$  is at most  $\alpha k_t$ . Therefore, the available capacity of  $B_i$  is large enough that we can add  $L_t$  to  $\text{cluster}(B_i)$ , instead of to the set  $\mathcal{O}$ , which is a contradiction.  $\square$

We can now show that the number of light balls added to  $\mathcal{O}$  during Cluster Formation is small. The idea is that each such ball, by Lemma 3.2, contributes a significant amount

of  $y$ -credit to the balls in  $\mathcal{H}_1$ . On the other hand, Lemma 3.4 upper bounds the total  $y$ -accumulation for the heavy balls.

**Lemma 3.5** *At the end of Cluster Formation stage, we have  $|\mathcal{O}| \leq 5 \cdot ((1 + \alpha) \cdot |\mathcal{H}_1| + \sum_{B_j \in \mathcal{L}_1} y_j)$ , where  $0 < \alpha \leq 3/8$ .*

**Proof** At the end of Cluster Formation stage,

$$\begin{aligned}
 \sum_{B_i \in \mathcal{H}_1} \tilde{y}(B_i) &\geq \sum_{\substack{B_i \in \mathcal{H}_1 \\ 1 \leq t \leq I}} Y(L_t, B_i) - \sum_{B_i \in \mathcal{H}_1} \sum_{B_j \in \text{cluster}(B_i)} y_j \\
 &\geq \sum_{1 \leq t \leq I} (F_t/k_t) - \sum_{B_j \in \mathcal{L}_1} y_j \\
 &\quad \left( \because F_t = \sum_{B_i \in \mathcal{H}_1} F(L_t, B_i) = k_t \cdot \sum_{B_i \in \mathcal{H}_1} Y(L_t, B_i) \right) \\
 &\geq \frac{1}{5} \cdot |\mathcal{O}| - \sum_{B_j \in \mathcal{L}_1} y_j, \tag{8}
 \end{aligned}$$

where we used Lemma 3.2 to get the last inequality. Now, adding the inequality of Lemma 3.4 over all  $B_i \in \mathcal{H}_1$ , we have that  $\sum_{B_i \in \mathcal{H}_1} \tilde{y}(B_i) \leq (1 + \alpha) \cdot |\mathcal{H}_1|$ . Combining this with (8) yields the desired inequality.  $\square$

The next two lemmas bound the cost of our solution and the expansion factor.

**Lemma 3.6** *The cost of the solution returned by the algorithm is at most 21 times the cost of an optimal solution.*

**Proof** Let  $\sigma = (x, y)$  be the preprocessed LP solution. Now, the total number of balls in the solution is  $|\mathcal{O}| + |\mathcal{H}_1|$ . Using Lemma 3.5,

$$\begin{aligned}
 |\mathcal{O}| + |\mathcal{H}_1| &\leq 5 \cdot \left( (1 + \alpha) \cdot |\mathcal{H}_1| + \sum_{B_j \in \mathcal{L}_1} y_j \right) + |\mathcal{H}_1| \\
 &\leq (6 + 5\alpha) \left( \sum_{B_j \in \mathcal{H}_1} y_j + \sum_{B_j \in \mathcal{L}_1} y_j \right) \\
 &\leq (6 + 5\alpha) \cdot \text{cost}(\sigma) \\
 &\leq \left( \frac{6 + 5\alpha}{\alpha} \right) \cdot OPT = 21 \cdot OPT \quad (\text{by setting } \alpha = 3/8). \quad \square
 \end{aligned}$$

**Lemma 3.7** *In the algorithm each input ball is expanded by at most a factor of 9.*

**Proof** Recall that when a light ball becomes heavy in the preprocessing step, it is expanded by a factor of 3. Therefore after the preprocessing step, any heavy ball in a solution may be an expanded or unexpanded ball.

Now, consider the selection of the balls in the second stage. If a cluster consists of only a heavy ball, then it does not expand any further. Since it might be an expanded ball, the total expansion factor is at most 3.

Otherwise, for a fixed cluster, let  $r_l$  and  $r_h$  be the radius of the largest light ball and the heavy ball, respectively. If  $r_l \geq r_h$ , then the overall expansion factor is 5. Otherwise, if  $r_l < r_h$ , then the heavy ball is chosen, and it is expanded by a factor of at most 3. Now as the heavy ball might already be expanded by a factor of 3 during the preprocessing step, here the overall expansion factor is 9.  $\square$

If the capacities of all balls are equal, then one can improve the expansion factor to 6.47 by using an alternative procedure to the *Selection of Objects* stage.

**Lemma 3.8** *If the capacities of all balls are equal, then there exists an alternative procedure to the Selection of Objects stage of the algorithm for MMCC, that guarantees that any ball is expanded by at most a factor of 6.47.*

**Proof** If the capacities of all balls are equal (say  $U$ ), then we proceed in the same way until the *Selection of Objects* stage. Then, we use the following scheme that guarantees a smaller expansion factor for this special case. We first describe the scheme and then analyze it.

Fix a cluster obtained after the *Cluster Formation* stage. If the cluster contains only a heavy ball, then we add it to a set  $\mathcal{C}$  (initialized to  $\emptyset$ ), without expansion.

Otherwise, let  $r_l$  denote the radius of a largest light ball in the cluster, and let  $r_h$  be the radius of the heavy ball. Let  $B_l$  and  $B_h$  be the corresponding balls. We consider the following three cases:

$r_l \geq r_h$ : In this case, let  $B = B_l$ . We set its new radius to be  $3r_l + 2r_h$ .

$r_h/\sqrt{3} \leq r_l < r_h$ : Let  $B = B_l$ . We set its new radius to be  $3r_l + 2r_h$ .

$r_l < r_h/\sqrt{3}$ : Let  $B = B_h$ . We set its new radius to be  $r_h + 2r_l$ .

Finally, if  $B \neq B_h$ , then we reroute the flow from  $B_h$  to  $B$ , set  $y_h \leftarrow 0$ , and add  $B$  to the set  $\mathcal{C}$  respectively. Finally, we set  $y_i \leftarrow 1$  for all balls  $B_i \in \mathcal{C}$ , and return  $\mathcal{C}$  as the solution.

To analyze the scheme, note that a heavy ball at the end of *Cluster Formation* stage may have been a light ball that was expanded by a factor of 3 in the preprocessing step. Therefore, if a cluster contains only a heavy ball, then the total expansion factor is at most 3. Otherwise, we analyze each of the 3 cases discussed above separately.

In the first case,  $3r_l + 2r_h \leq 5r_l$ .

In the second case,  $3r_l + 2r_h \leq (3 + 2 \cdot \sqrt{3})r_l < 6.47r_l$ .

In the third case,  $r_h + 2r_l \leq (1 + 2/\sqrt{3})r_h$ . But  $B_h$  might be originally a light ball that was expanded by a factor of 3 in the preprocessing step. Therefore, the total expansion factor is at most  $3 + 2 \cdot \sqrt{3} < 6.47$ .  $\square$

Lastly, from Lemmas 3.6 and 3.7, we get the following theorem.

**Theorem 3.9** *There is a polynomial time (21, 9)-approximation algorithm for the MMCC problem.*



### 3.2 The Algorithm for the EMCC Problem

For the EMCC problem, we can exploit the structure of  $\mathbb{R}^d$  to restrict the expansion factor of the balls to at most  $1 + \epsilon$ , while paying in terms of the cost of the solution. In the following, we give an overview of how to adapt the stages of the framework for obtaining this result. Note that in each iteration of the preprocessing stage for MMCC, we consider a point  $p_j$  and a cluster  $\mathcal{L}_j$  of light balls. We select a largest ball from this set and reroute the flow of other balls in  $\mathcal{L}_j$  to this ball. However, to ensure that the selected ball contains all the points it serves we need to expand this ball by a factor of 3. For the EMCC problem, for the cluster  $\mathcal{L}_j$ , we consider the bounding hypercube whose side is at most a constant times the maximum radius of the balls from  $\mathcal{L}_j$ , and subdivide it into multiple cells. The granularity of the cells is carefully chosen to ensure that (1) Selecting a maximum radius ball among the balls whose centers are lying in that cell, and expanding it by  $1 + \epsilon$  factor is enough to cover all such balls, and (2) The total number of cells is  $\text{poly}(1/\epsilon)$ . From each cell we select a maximum radius ball, expand it by  $1 + \epsilon$  factor, and reroute the flow from the other balls in that cell to it. It follows that the cost of the preprocessed solution increases by at most a  $\text{poly}(1/\epsilon)$  factor. The Cluster Formation stage for the EMCC problem is exactly the same as that for the MMCC problem. Note that in the Selection of Objects stage for MMCC, we select only one ball per cluster and expand it by an  $O(1)$  factor to cover all the balls in the cluster. But for EMCC, we want to restrict the expansion factor of the balls to at most  $1 + \epsilon$ . Hence we select multiple balls per cluster in a way so that the selected balls when expanded by a factor of  $1 + \epsilon$  can cover all the balls in the cluster. The ideas are similar to the ones in the Preprocessing stage, however one needs to be more careful to handle some technicalities that arise. We summarize the result for the EMCC problem in the following theorem.

**Theorem 3.10** *There is a polynomial time  $(O(\epsilon^{-3d} \log(1/\epsilon)), 1 + \epsilon)$ -approximation algorithm for the EMCC problem in  $\mathbb{R}^d$ , for any  $\epsilon > 0$ .*

In the rest of this section we prove this theorem. We describe in detail how the Preprocessing stage and the Selection of Objects stage are modified. For simplicity, we first assume that  $d = 2$ , and thus we are working in the plane. Our algorithm takes an additional input—a constant  $\epsilon > 0$ , and gives an  $O(\epsilon^{-6} \log(1/\epsilon))$  approximation, where each ball in the solution may be expanded by a factor of at most  $1 + \epsilon$ . For the EMCC problem, the Preprocessing stage is as follows.

**Lemma 3.11** *Given a feasible LP solution  $\sigma = (x, y)$  corresponding to an EMCC instance in  $\mathbb{R}^2$ , and parameters  $0 < \alpha \leq 1/2$  and  $\epsilon > 0$ , there exists a polynomial time algorithm to obtain another LP solution  $\bar{\sigma} = (\bar{x}, \bar{y})$  that satisfies all the constraints of **MMCC-LP** (Constraints (1)–(6)) except Constraint (4). Additionally,  $\bar{\sigma}$  satisfies the following properties.*

1. Any ball  $B_i \in \mathcal{B}$  with non-zero  $\bar{y}_i$  is either heavy ( $\bar{y}_i = 1$ ), or light ( $0 < \bar{y}_i \leq \alpha$ ).
2. For each point  $p_j \in P$ , we have that

$$\sum_{B_i \in \mathcal{L}: \bar{x}_{ij} > 0} \bar{y}_i \leq \alpha, \quad (9)$$

where  $\mathcal{L}$  is the set of light balls with respect to  $\bar{\sigma}$ .

3. For any heavy ball  $B_i$ , and any point  $p_j \in P$  served by  $B_i$ ,  $d(c_i, p_j) \leq (1 + \epsilon) \cdot r_i$ .
4. For any light ball  $B_i$ , and any point  $p_j \in P$  served by  $B_i$ ,  $d(c_i, p_j) \leq r_i$ .
5.  $\text{cost}(\bar{\sigma}) = O(\epsilon^{-2} \log(1/\epsilon)) \cdot \text{cost}(\sigma)$ .

**Proof** As in Lemma 3.1, in each iteration, we pick an arbitrary point  $p_j \in P$  for which the inequality (9) is not met, and consider a set  $\mathcal{L}_j$  of light balls serving  $p_j$  such that  $\alpha < \sum_{B_i \in \mathcal{L}_j} y_i \leq 2\alpha$ . Let  $r$  be the radius of a maximum radius ball from the set  $\mathcal{L}_j$ . Let  $\mathcal{T} \subset \mathcal{L}_j$ , the set of *tiny* balls, consist of those balls whose radius is at most  $\epsilon r/2$ . We partition the set  $\mathcal{L}_j \setminus \mathcal{T}$  of balls into groups such that the centers of balls in a group are nearby.

To compute these groups, we divide the balls in  $\mathcal{L}_j \setminus \mathcal{T}$  into  $O(\log(1/\epsilon))$  classes such that the  $i$ th class contains balls whose radii are in the interval  $(2^{i-1}r\epsilon, 2^i r\epsilon]$ , for  $0 \leq i \leq O(\log(1/\epsilon))$ . We consider each class separately. For the  $i$ th class, we overlay a grid on the plane where each grid cell has side length  $2^{i-2}r\epsilon^2$ . For each grid cell that has at least one ball center from the  $i$ th class, we create a group consisting of all balls (from the  $i$ 'th class) whose center lies within the cell. As all balls in  $\mathcal{L}_j$  contain the common point  $p_j$ , it follows that there exists an axis-parallel square of side  $2^{i+2}r\epsilon$  such that the centers of the balls in  $i$ th class are contained in it. Thus the number of groups from the  $i$ th class is  $O(\epsilon^{-2})$ , and the overall number of groups is  $O(\epsilon^{-2} \log 1/\epsilon)$ .

Consider a group  $\mathcal{B}'$  with balls in the  $i$ th class. Note that the radius of each ball in this group is at least  $2^{i-1}r\epsilon$ . Let  $B_m \in \mathcal{B}'$  be the ball with the maximum radius  $r_m$ ; recall that we break ties in favor of higher capacity. We refer to  $B_m$  as the *leader* of its group. Since the center of  $B_m$  is within distance  $2^{i-1}r\epsilon^2$  from the center of any ball  $B_l \in \mathcal{B}'$ , all the points contained in ball  $B_l \in \mathcal{B}'$  are within distance  $2^{i-1}r\epsilon^2 + r_l \leq \epsilon r_m + r_m = (1 + \epsilon) \cdot r_m$  from the center  $c_m$  of the ball  $B_m$ . Thus, a  $(1 + \epsilon)$ -expansion of the leader contains any ball within its group.

For each group  $\mathcal{B}'$ , we select the group leader  $B_m$ . We reroute the flow from each of the other balls in  $\mathcal{B}'$  to  $B_m$ . Note that the capacity  $U_m$  of the ball  $B_m$  is at least that of the capacity of any ball from the set  $\mathcal{B}'$ , because of the *monotonicity* property. Furthermore, the ball  $B_m$  has enough capacity to receive all the redirected flow, since

$$\sum_{p_j \in P, B_l \in \mathcal{B}'} x_{lj} \leq \sum_{B_l \in \mathcal{B}'} U_l \cdot y_l \leq U_m \cdot \sum_{B_l \in \mathcal{B}'} y_l \leq U_m \cdot 2 \cdot \alpha \leq U_m.$$

We set  $y_m \leftarrow 1$ , and  $y_t \leftarrow 0$  for the remaining balls in  $\mathcal{B}'$ .

Now we address the tiny balls. Note that the ball in  $\mathcal{L}_j$  with maximum radius is the leader of the group it belongs to. It is easy to see that a  $(1 + \epsilon)$ -expansion of this ball contains every ball in  $\mathcal{T}$ . We set  $y_t \leftarrow 0$  for every ball in  $\mathcal{T}$ , and redirect the flow from it to the maximum radius ball. As above, it is easy to see that the capacity constraint for the maximum radius ball is maintained.

This completes the description of one iteration of the preprocessing algorithm. After all the iterations are completed, we set  $y_t \leftarrow 1$  for any ball  $B_t$  such that  $\alpha < y_t < 1$ . Consider an iteration of the algorithm. At the beginning of the iteration, all balls in  $\mathcal{L}_j$  were light, and  $y(\mathcal{L}_j) > \alpha$ . After the iteration, none of the balls in  $\mathcal{L}_j$  are light. Furthermore, we select  $O(\epsilon^{-2} \log(1/\epsilon))$  balls from  $\mathcal{L}_j$  in the iteration. It follows that the entire

preprocessing algorithm increases the cost by a factor of at most  $O(\epsilon^{-2} \log(1/\epsilon))$ . It is easy to verify that the other properties in the statement of the lemma are also satisfied.  $\square$

As mentioned before, the Cluster Formation stage for EMCC is exactly the same as the one for MMCC. Note that the Cluster Formation stage increases the cost of the solution only by a constant factor. We describe and analyze the Selection of Objects stage in the following lemma. The main idea is similar to that of Lemma 3.11, but there is one complicating factor.

**Lemma 3.12** *There exists a scheme for the Selection of Objects stage for the EMCC problem, such that for any  $\epsilon > 0$ ,*

1. *The number of balls we select is  $O(\epsilon^{-4})$  times the cost of the fractional solution after the Cluster Formation stage.*
2. *For any chosen ball  $B(c_i, r_i)$  that serves a point  $p_j \in P$ , we have that  $d(c_i, p_j) \leq (1 + \epsilon) \cdot r_i$ .*
3. *For any chosen ball  $B_i$  with capacity  $U_i$ , we have that  $\sum_{p_j \in P} x_{ij} \leq U_i$ .*

**Proof** Let  $\sigma = (x, y)$  denote the fractional solution after the Cluster Formation stage. Let  $\mathcal{H}$  and  $\mathcal{L}$  denote, respectively, the set of heavy and light balls after the Cluster Formation stage. Recall that  $y_i = 1$  for each  $B_i \in \mathcal{H}$ . We use  $y(\mathcal{B}')$  to denote  $\sum_{B_i \in \mathcal{B}'} y_i$ .

Recall that the clusters are pairwise-disjoint. We consider each cluster in turn. Recall that for each heavy ball  $B_h$ ,  $\text{cluster}(B_h)$  contains  $B_h$  and possibly other intersecting light balls. If the cluster has no light balls, we just select the heavy ball, and do not change the flow from it.

So let us assume  $\text{cluster}(B_h)$  contains one or more light balls. During the Cluster Formation stage, when a light ball  $B_i$  was added to  $\text{cluster}(B_h)$ , we redirected flow from  $B_i$  to  $B_h$ . Let  $f_i$  denote the flow out of  $B_i$  prior to this redirection. That is, if  $(\bar{x}, \bar{y})$  denotes the solution just before we redirected flow from  $B_i$  to  $B_h$ , then  $f_i(p_j) = \bar{x}_{ij}$ . Notice that the amount of this flow is  $\sum_j f_i(p_j) \leq \bar{y}_i U_i = y_i U_i$ .

We begin by undoing these redirections. That is, for each light ball  $B_i$  in  $\text{cluster}(B_h)$  and each point  $p_j$  such that  $f_i(p_j) > 0$ , we set  $x_{ij} \leftarrow f_i(p_j)$  and  $x_{hj} \leftarrow x_{hj} - f_i(p_j)$ . Let  $\mathcal{B}_1$  denote the set of light balls in  $\text{cluster}(B_h)$  with radius larger than that of  $B_h$ , and  $\mathcal{B}_2$  denote the remaining light balls. We first consider  $\mathcal{B}_1$ , assuming it is non-empty. Let  $r$  denote the radius of the maximum radius ball in  $\mathcal{B}_1$ . Let  $\mathcal{T}_1 \subset \mathcal{B}_1$ , the set of tiny balls from  $\mathcal{B}_1$ , denote the balls with radius at most  $\epsilon r/4$ . Observe that for any  $r' > 0$ , the centers of all balls in  $\mathcal{B}_1$  with radius at most  $r'$  are contained in an axes-parallel square of side length  $O(r')$ . Thus, we may proceed as in Lemma 3.11 to partition  $\mathcal{B}_1 \setminus \mathcal{T}_1$  into  $O(\epsilon^{-2} \log 1/\epsilon)$  groups, such that the  $(1 + \epsilon)$ -expansion of the leader within each group contains any other ball in the group. In each group, we redirect the flow from the non-leaders to the leader. Now assume that  $\mathcal{T}_1$  is non-empty, and consider any  $B_i \in \mathcal{T}_1$ . Consider expanding the maximum radius ball of  $\mathcal{B}_1$  by an additive factor of  $2r_h + 2r_i \leq \epsilon r$ ;  $B_i$  is contained in the expanded ball. Thus, a  $(1 + \epsilon)$ -expansion of the maximum radius ball contains each ball  $B_i \in \mathcal{T}_1$ . Hence, we redirect the flow from  $\mathcal{T}_1$  to the maximum radius ball, which we may assume is the leader of the group that

contains it. As balls in  $\mathcal{B}_1$  have capacity at least that of the heavy ball  $B_h$ , each such ball has enough capacity to absorb the flow redirected to it.

We select the leaders in  $\mathcal{B}_1$ . The selection of balls in  $\mathcal{B}_2$  is more involved, because the capacity of some of these balls may be much smaller than that of  $B_h$ . Let  $\mathcal{T}_2 \subset \mathcal{B}_2$  denote the balls with radius at most  $\epsilon r_h/2$ . We partition the balls in  $\mathcal{B}_2 \setminus \mathcal{T}_2$  into groups using the following slightly different scheme. We overlay a grid on the plane where each cell has side length  $\epsilon^2 r_h/4$ . For each grid cell that has at least one center of some ball in  $\mathcal{B}_2 \setminus \mathcal{T}_2$ , we create a group consisting of the balls in  $\mathcal{B}_2 \setminus \mathcal{T}_2$  with centers within the grid cell. As all centers of balls in  $\mathcal{B}_2$  are contained in a square of side length  $O(r_h)$ , the number of groups is  $O(1/\epsilon^4)$ .

Let  $\mathcal{B}'$  denote any such group. We order the balls in  $\mathcal{B}'$  by non-increasing radii, breaking ties using capacity. By monotonicity, this ordering is also an ordering where capacity is non-increasing. Furthermore, for any two balls  $B_i$  and  $B_j$  such that  $B_i$  occurs before  $B_j$  in the ordering, the  $(1 + \epsilon)$ -expansion of  $B_i$  contains  $B_j$ . Let  $\mathcal{F}'$  be the first  $\lceil y(\mathcal{B}') \rceil$  balls in this ordering. Observe that

$$\sum_{B_i \in \mathcal{F}'} U_i \geq \sum_{B_i \in \mathcal{B}'} y_i U_i \geq \sum_{\substack{B_i \in \mathcal{B}' \\ p_j \in P}} x_{ij}.$$

We select the balls in  $\mathcal{F}'$ , and redirect the flow from the balls in  $\mathcal{B}' \setminus \mathcal{F}'$  to the balls in  $\mathcal{F}'$  in an arbitrary way that respects the capacities. As observed above, the balls in  $\mathcal{F}'$  have enough capacity for doing this.

Each ball in  $\mathcal{T}_2$  is contained in the  $(1 + \epsilon)$ -expansion of the heavy ball  $B_h$ . We select  $B_h$ , and redirect the flow from each ball in  $\mathcal{T}_2$  to  $B_h$ .

As the number of groups for  $\mathcal{B}_2$  is  $O(\epsilon^{-4})$ , the number of balls we select for  $\mathcal{B}_2$  is  $O(\epsilon^{-4} + y(\mathcal{B}_2))$ . As the number of balls selected for  $\mathcal{B}_1$  is  $O(\epsilon^{-2} \log 1/\epsilon)$ , the number of balls selected from  $\text{cluster}(B_h)$  is also  $O(\epsilon^{-4} + y(\mathcal{B}_2))$ . We have reassigned the flow that was originally from  $B_h$  to the selected balls in a way that respects capacity constraints. Furthermore, if a selected ball serves a point, then that point is contained in the  $(1 + \epsilon)$ -expansion of the ball.

Summing over all clusters, the number of balls selected is  $O(\epsilon^{-4} |\mathcal{H}| + y(\mathcal{L}))$ , implying the upper bound claimed in the statement of the lemma.  $\square$

We note that Lemmas 3.11 and 3.12 can be modified to work in  $\mathbb{R}^d$ . In this case, the dependence on  $\epsilon$  becomes  $O(\epsilon^{-d} \log(1/\epsilon))$ , and  $O(\epsilon^{-2d})$ , respectively (where the constants inside the Big-Oh may depend exponentially on the dimension  $d$ ). Therefore, with suitable modifications to Lemma 3.11, the analysis of the Cluster Formation stage from the MMCC algorithm, and Lemma 3.12, Theorem 3.10 follows. Finally, we also note that, after appropriate modifications, the algorithm of this section can also be made to work in metric spaces with constant doubling dimension, with similar bicriteria approximation guarantees.

## 4 Hardness of Approximation

In this section, we prove the APX-hardness results for the Metric Monotonic Capacitated Covering (MMCC) problem, and a generalization of MMCC with weights.

### 4.1 Hardness of Metric Monotonic Capacitated Covering

We show that for any constant  $c \geq 1$ , there exists a constant  $\epsilon_c > 0$  such that it is NP-hard to obtain a  $(1 + \epsilon_c, c)$ -approximation for the MMCC problem. We observed in Sect. 1 that it is NP-hard to obtain a  $(o(\log n), c)$ -approximation for the MMCC for  $1 \leq c < 3$ . The result we establish in this section shows that even if we allow expansion by a constant at least 3, it is not possible to obtain a PTAS for this problem. To show this, we use a gap-preserving reduction from (a version of) the 3-Dimensional Matching problem.

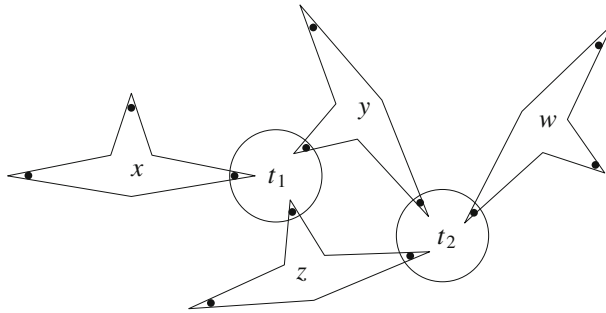
Consider the Maximum Bounded 3-Dimensional Matching (3DM-3) problem (defined in [29]). In this problem, we are given three disjoint sets of elements  $X, Y, Z$ , with  $|X| = |Y| = |Z| = N$ , and a set of “triples”  $T \subseteq X \times Y \times Z$ , such that each element  $w \in W := X \cup Y \cup Z$  appears in 1, 2 or 3 triples of  $T$ . A triple  $t = (x, y, z) \in T$  is said to cover  $x \in X, y \in Y, z \in Z$ . A subset  $M \subseteq T$  of triples such that no element in  $W$  appears in more than one triple of  $M$  is called a *matching*, and the set of elements  $U \subseteq W$  that are covered by the triples in  $M$  are said to be the *matched* elements. If  $W = U$ , then the corresponding  $M$  is said to be a *perfect matching*. The goal of the 3DM-3 problem is to find a maximum cardinality matching. We have the following result for the 3DM-3 problem due to Petrank [29].

**Lemma 4.1** (Restatement of Theorem 4.4 from [29]) *There exists a constant  $0 < \beta < 1$ , such that it is NP-hard to distinguish between the instances of the 3DM-3 problem in which a perfect matching exists, from the instances in which at most  $3\beta N$  elements are matched.*

Notice that if the given 3DM-3 instance has a perfect matching, then there is a *cover* for  $W$  with exactly  $N$  triples. On the other hand, if at most  $3\beta N$  elements can be matched in the given instance, then the minimum size cover has size significantly larger than  $N$ .

**Reduction from 3DM-3 to MMCC.** Given an instance  $I$  of 3DM-3 problem, we show how to reduce it to an instance  $I'$  of the MMCC problem. Recall that in this version of the MMCC problem, a solution is allowed to expand the balls in the input by the given constant  $c \geq 1$ .

First, we discuss the high level idea of the reduction. For each element  $w \in W$ , we create an *element gadget*, which is a system of balls and points that need to be covered. Three of the points in the gadget are designated as *ideal points*. In an element gadget, (a) the number of points in the gadget is one more than the total capacity of the balls, and (b) if we exclude an ideal point, the remaining points in the gadget can be served by the balls. For each triple  $t = (x, y, z)$ , we create a *triple gadget*, a single ball that contains exactly one ideal point from the element gadgets of  $x, y$ , and  $z$  (see Fig. 1).



**Fig. 1** There is an element gadget (represented here by a polygon) corresponding to each of the elements  $w, x, y, z$ , containing three ideal points each. Triple  $t_1$  contains elements  $x, y, z$ , therefore its triple gadget (a ball) contains the ideal points of those elements. Triple  $t_2$  contains elements  $w, y, z$

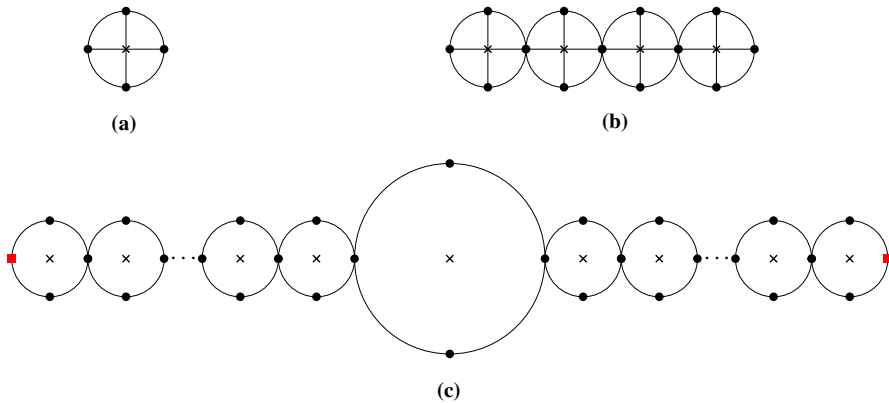
We ensure that for each element  $w$ , the balls for the (at most three) triples containing  $w$  are incident to different ideal points in the gadget for  $w$ .

Furthermore, the constructed instance of MMCC has the property that every ball in every element gadget must be chosen. From the property (a) of the element gadget, it follows that the triple gadgets chosen in a feasible solution for MMCC correspond to a cover of all elements in  $W$  in the original 3DM-3 instance. In particular, if the given 3DM-3 instance admits a perfect matching, then an optimal solution to the MMCC instance chooses precisely  $N$  triple gadgets; if the maximum matching in the given 3DM-3 instance is far from perfect, then an optimal solution to the MMCC instance chooses significantly more than  $N$  triple gadgets. The main complication in implementing this high level idea is that in the target instance, the solution can expand the input balls by a factor of up to  $c$ .

Now, we construct a metric space  $(P \cup C, d)$  for the MMCC instance  $I'$ , that is induced by the shortest path metric on a certain graph  $G = (P \cup C, E)$ . The target instance  $I'$  will have exactly one ball centered at each point of  $C$ , and  $P$  is the set of points that need to be covered by the balls. We have  $P \cap C = \emptyset$ . We construct the instance by combining fragments, as described next.

Consider a vertex  $c' \in C$ , that is connected to four other vertices  $p_1, \dots, p_4 \in P$  (for convenience we refer to them as *left*, *right*, *top*, *bottom* vertices respectively) by an edge of weight 1. We also consider the associated ball of radius 1 at the center  $c'$  that covers  $p_1, \dots, p_4$ . For convenience, we refer to this object (the graph induced by these five vertices and the ball) as a *small cluster*, and the ball as a *small ball*. A *large cluster* and a *large ball* are defined in the same way except the radius of the ball and the edge weights are  $c + 1$ .

**Element Gadget.** We combine small and large clusters in a careful way to form the element gadget. Consider  $p = \lceil c(c + 1)/2 \rceil + 1$  copies of *small clusters*, numbered  $\kappa_1, \dots, \kappa_p$ . For each  $1 < i \leq p$ , we “glue” small clusters  $\kappa_{i-1}$  and  $\kappa_i$ , by setting the *right* vertex of  $\kappa_{i-1}$  equal to the *left* vertex of  $\kappa_i$ , that is, *identifying* these two vertices. This forms an object in which two consecutive clusters share exactly one vertex. We refer to this object as a *small chain*. For a particular small chain, we refer to the left vertex of  $\kappa_1$  and right vertex of  $\kappa_p$  as its *leftmost* and *rightmost* vertices, respectively.



**Fig. 2** (a) A small cluster, (b) a small chain obtained by “gluing”  $p = 4$  small clusters, (c) a long chain with its endpoints depicted as red squares

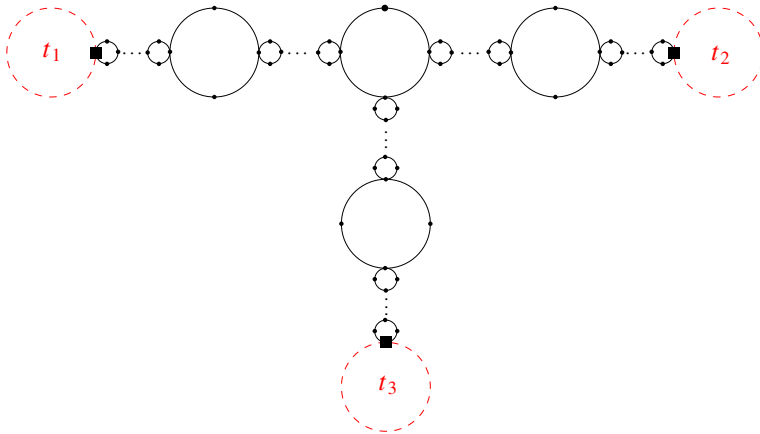
Note that there are  $p$  balls in a small chain and  $3p + 1$  points from  $P$ . A *long chain* is obtained by combining two small chains  $\psi_1$  and  $\psi_2$  and a single large cluster, by identifying (a) the rightmost vertex of  $\psi_1$  and the left vertex of the cluster, and (b) the leftmost vertex of  $\psi_2$  and the right vertex of the cluster. The leftmost vertex of  $\psi_1$  and the rightmost vertex of  $\psi_2$  are referred to as *endpoints* of the long chain. Note that a long chain has  $2p + 1$  balls and  $6p + 4$  points from  $P$ . See Fig. 2 for an illustration.

Now we combine three long chains  $\psi_1$ ,  $\psi_2$ , and  $\psi_3$  and a large cluster to obtain the element gadget. This is done by identifying an endpoint of  $\psi_1$  (resp.  $\psi_2$ ,  $\psi_3$ ) with the left (resp. right, bottom) vertex of the large cluster. The other endpoints of the three chains that have not been identified are referred to as the *ideal points* of the element gadget. Note that an element gadget has  $6p + 4$  balls, and  $18p + 13 = 3(6p + 4) + 1$  points from  $P$ .

**Triple Gadget.** For each element  $w \in W$ , we add an element gadget. Now we describe the triple gadget for triple  $t = (x, y, z)$ . This gadget is similar to a *large cluster*, the only difference is that in addition to the central point  $c_t$ , it contains only three other points  $p_x, p_y, p_z$ . Recall that the edges from  $c_t$  to these three points have weight  $c + 1$ , which is also the radius of the corresponding ball. We add a triple gadget for each triple  $t = (x, y, z) \in T$ . Now, for each such triple  $t = (x, y, z)$ , we identify the point  $p_x$  (resp.  $p_y, p_z$ ) in its gadget with an ideal point from the element gadget of  $x$  (resp.  $y, z$ ); see Fig. 3. Here, we ensure that if an element is contained in multiple triples, then a different ideal point from its gadget is assigned to each triple. The total number of balls in all the element gadgets is  $\mathcal{E} := 3N \cdot (6p + 4)$ . We refer to these balls as *element balls*. Similarly, the total number of balls in all the triple gadgets is  $|T|$ , which we refer to as *triple balls*. Finally, we set the capacity of all the balls to be 3. The total number of points in  $P$  is  $3N \cdot (18p + 13)$ . As mentioned above, the metric is induced by the graph  $G = (P \cup C, E)$  as described above.

This completes the description of the instance  $I'$  of the MMCC problem. By construction, the instance  $I'$  satisfies the monotonicity property because all ball capacities are the same. It is also worth highlighting that there are only two distinct radii in the





**Fig. 3** Three triple gadgets (identified by dashed red balls) corresponding to triples  $t_1, t_2, t_3$  are attached to three different ideal points (identified by squares) of an element gadget

instance  $I'$ . We are able to show that such a restricted version of the MMCC problem remains APX hard, even when we are allowed to expand the balls by a constant  $c \geq 1$  factor.

#### 4.1.1 Properties of the Constructed Instance

Given a ball with radius  $r$  and  $c \geq 1$ , we refer to the concentric ball of radius  $c \cdot r$  as its  $c$ -expansion. We begin with an observation about the points that a  $c$ -expansion of a ball in instance  $I'$  can cover. This follows easily from the length  $p$  of a short chain and the radii of the small and large balls.

**Lemma 4.2** *Let  $B$  be any input ball in instance  $I'$  and let  $p \in P$  be a point that is contained in the  $c$ -expansion of  $B$ . Then (a) if  $B$  is a large (element or triple) ball, then either  $p \in B$  or  $p$  is a point in a small chain incident to  $B$ ; (b) if  $B$  is small, then  $p$  is a point in the small chain containing  $B$ .*

A large element ball has either one or two points that are not part of any short chain. Thus, Lemma 4.2 implies that any feasible solution to  $I'$  must choose all the large element balls. Lemma 4.2 is also useful in showing that in any feasible solution, at least one triple ball incident to each element gadget must be chosen.

**Lemma 4.3** *Suppose we have a subset  $\mathcal{B}'$  of input balls in  $I'$  such that for some element  $w \in W$ ,  $\mathcal{B}'$  includes none of the triple balls containing an ideal point of  $w$ 's gadget. There is no feasible solution to  $I'$  with  $\mathcal{B}'$  as the chosen set of balls.*

**Proof** By Lemma 4.2, the only balls whose  $c$ -expansion contains a point from the element gadget for  $w$  are the balls in  $w$ 's element gadget, and the triple balls containing an ideal point of  $w$ 's gadget. If none of these triple balls are in  $\mathcal{B}'$ , there is no feasible assignment with  $\mathcal{B}'$ , as the total capacity of the balls in  $w$ 's gadget is one less than the number of points in it.  $\square$

The next lemma says that the above necessary condition is sufficient.

**Lemma 4.4** *Let  $\mathcal{B}'$  be a subset of the balls in  $I'$  such that for every element  $w \in W$ ,  $\mathcal{B}'$  includes at least one triple ball containing an ideal point of  $w$ 's gadget. Furthermore, suppose  $\mathcal{B}'$  also contains all element balls. There is a feasible solution to  $I'$  with  $\mathcal{B}'$  as the set of chosen balls.*

**Proof** For each element  $w \in W$ , pick one ideal point  $q$  from its gadget that is contained in a triple ball in  $\mathcal{B}'$ , and assign  $q$  to that triple ball. For each element ball in  $w$ 's gadget, among the four points it contains, assign to it the three points that are furthest from  $q$  (in the subgraph corresponding to  $w$ 's gadget). In this assignment, each ball (element or triple) in  $\mathcal{B}'$  is assigned at most three points, and all points in  $P$  are assigned. Thus, the solution is feasible, and no input ball is expanded.  $\square$

Now, we establish a technical lemma that is needed subsequently.

**Lemma 4.5** *For some element  $w \in W$  and a feasible solution to  $I'$  with chosen balls  $\mathcal{B}'$ , suppose that each triple ball in  $\mathcal{B}'$  containing an ideal point of  $w$  is assigned at most two points belonging to the gadget for  $w$ . Then  $\mathcal{B}'$  contains every element ball from  $w$ 's gadget.*

**Proof** As we observed already, all large element balls belong to  $\mathcal{B}'$ . Each small ball in  $w$ 's gadget belongs to one of six short chains. There are two types of short chains. The first type we consider is incident to a large element ball  $B_1$  at one end and a (large) triple ball  $B_2$  at the other. The short chain  $\psi$  has  $p$  small balls and  $3p + 1$  points from  $P$ . The large element ball  $B_1$  incident to it has two points that are not in any short chain. From Lemma 4.2, it follows that these two points are assigned to  $B_1$ . Thus,  $B_1$  can be assigned at most one point from  $\psi$ . By assumption,  $B_2$  is assigned at most two points from  $\psi$ . This leaves at least  $3(p - 1) + 1$  points from  $\psi$  that must be assigned to the balls in  $\psi$ . Thus, all  $p$  balls in  $\psi$  must be in  $\mathcal{B}'$ .

The second type of short chain  $\psi$  we consider is incident to two large element balls, a large element ball  $B_1$  that is part of the long chain that  $\psi$  is part of, and the central large element ball  $B_2$ . As before,  $B_1$  can be assigned at most one point from  $\psi$ . The ball  $B_2$  has one point not on any short chain that must be assigned to it; hence it can be assigned at most two points from  $\psi$ . Arguing as above, we conclude that in this case too, all  $p$  balls in  $\psi$  must be in  $\mathcal{B}'$ . This concludes the proof. For future reference, we note that in the second type of short chain, we did not use the assumption in the lemma about each triple ball being assigned at most two points.  $\square$

Finally, we argue that in an optimal solution to  $I'$ , we may assume without loss of generality that all element balls are chosen.

**Lemma 4.6** *There is an optimal solution to  $I'$  in which every element ball is chosen.*

**Proof** Fix an optimal solution that maximizes the number of element balls. We claim that such a solution must choose every element ball. Suppose that this is not the case; we obtain a contradiction. Suppose an element ball  $B$  belonging to the gadget for element  $w \in W$  is not chosen. The ball  $B$  must be small. From the proof of Lemma 4.5,  $B$

must belong to a short chain  $\psi$  that is incident to a large element ball  $B_1$  at one end and a (large) triple ball  $B_2$  at the other. Furthermore, the triple ball  $B_2$  must be chosen and assigned three points from  $\psi$ . As the chain  $\psi$  has  $3p + 1$  points from  $P$ , it must be that the  $p - 1$  other small balls in it are chosen, and the large element ball  $B_1$  is assigned exactly one point from  $\psi$ .

Now, from the optimal solution, we swap out  $B_2$  and swap in  $B$ . Assign to the large element ball  $B_1$  the endpoint  $q$  of chain  $\psi$  that it contains. Assign to each small ball in  $\psi$  the three points (among the four it contains) that are furthest from  $q$ . This modified assignment is feasible. Thus we have computed another optimal solution that has one more element ball, a contradiction.  $\square$

#### 4.1.2 Hardness Guarantee

We can now begin relating an instance  $I$  of 3DM-3 with the corresponding instance  $I'$  of MMCC.

**Lemma 4.7** *Consider an instance  $I$  of 3DM-3, and let  $I'$  be the reduced MMCC instance. In the 3DM-3 instance  $I$ , there is a set  $M \subseteq T$  with at most  $K$  triples that covers all the elements of  $W$  if and only if there is a solution to  $I'$ , wherein the balls may be expanded by up to the factor  $c \geq 1$ , with cost at most  $\mathcal{E} + K$ . Here,  $\mathcal{E}$  is the number of element balls in  $I'$ .*

**Proof** Suppose that  $M \subseteq T$  is a set of at most  $K$  triples that covers all the elements of  $W$ . In the instance  $I'$ , let  $\mathcal{B}'$  include the  $\mathcal{E}$  element balls and the triple balls corresponding to  $M$ . By Lemma 4.4, there is a feasible solution with  $\mathcal{B}'$  as the chosen balls. The cost of this solution is  $|\mathcal{B}'| \leq \mathcal{E} + K$ .

For the reverse direction, suppose that there is a solution to  $I'$  with cost at most  $\mathcal{E} + K$ . Such a solution may expand the input balls in  $I'$  by up to a factor of  $c$ . Then, we use Lemma 4.6 to obtain another solution of size at most  $\mathcal{E} + K$ , in which all element balls are selected. This solution uses at most  $K$  triple balls. By Lemma 4.3, this set of triple balls includes, for each element  $w \in W$ , at least one triple ball containing an ideal point in  $w$ 's gadget. Thus the set of triples corresponding to these at most  $K$  triple balls covers  $W$ .  $\square$

Using this lemma, we obtain the following two corollaries, that show the gap between the instances that have a perfect matching, and those that do not.

**Corollary 4.8** *If there exists a perfect matching in the 3DM-3 instance  $I$ , then the corresponding MMCC instance  $I'$  has an optimal solution of size exactly  $\mathcal{E} + N$ .*

**Corollary 4.9** *If in the 3DM-3 instance  $I$ , the maximum number of elements that can be matched is at most  $3\alpha N$  ( $0 < \alpha < 1$ ), then the minimum cost of any solution where the balls may be expanded by a factor  $c \geq 1$ , in the corresponding MMCC instance  $I'$ , is at least  $\mathcal{E} + \alpha N + \frac{3(1-\alpha)N}{2} = \left(1 + \frac{1-\alpha}{36p+26}\right) \cdot (\mathcal{E} + N)$ .*

**Proof** In the 3DM-3 instance  $I$ , the maximum number of elements that can be matched is at most  $3\alpha N$ , for some  $0 < \alpha < 1$ . If  $M \subseteq T$  is a minimum size set of triples that covers all the elements in  $W$ , then we first show that  $|M| \geq \alpha N + 3(1 - \alpha)N/2$ .

Let  $M = M_1 \cup M_2$ , where  $M_1$  is the maximal set of triples such that each set covers three distinct elements, and  $M_2$  is the remaining triples. By assumption, the number of matched elements is at most  $3\alpha N$ , and therefore,  $|M_1| \leq \alpha N$ . The number of elements left to be covered by  $M_2$  is  $3N - 3|M_1|$ , and since each triple in  $M_2$  covers at most two new elements each,  $|M_2| \geq 3(N - |M_1|)/2$ . Therefore, since  $M_1$  and  $M_2$  are disjoint,  $|M| = |M_1| + |M_2| \geq N - |M_1|/2$ , which is minimized when  $|M_1| = \alpha N$ . Therefore,  $|M| \geq \alpha N + 3(1 - \alpha)N/2$ .

Now, using Lemma 4.7, we conclude that the cost of an optimal solution to  $I'$  is at least  $\mathcal{E} + \alpha N + \frac{3(1-\alpha)N}{2}$ . It is easy to verify that the previous quantity is exactly equal to  $(1 + \frac{1-\alpha}{2(18p+13)}) \cdot (\mathcal{E} + N)$ , recalling that  $\mathcal{E} = 3N \cdot (6p + 4)$ .  $\square$

Now, from Lemma 4.1, and Corollaries 4.8 and 4.9, we obtain the following APX-hardness result.

**Theorem 4.10** *For any constant  $c \geq 1$ , there exists a constant  $\epsilon_c > 0$  such that it is NP-hard to obtain a  $(1 + \epsilon_c, c)$ -approximation for Metric Capacitated Covering, the special case of MMCC where all input balls have the same capacity.<sup>2</sup>*

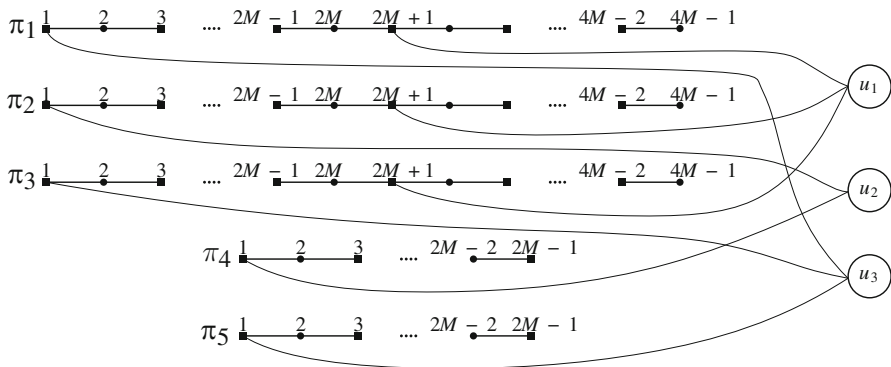
## 4.2 Hardness of Metric Monotonic Capacitated Covering with Weights

We consider a generalization of the Metric Monotonic Capacitated Covering (MMCC) problem. Like in the MMCC problem, here also we are given a set of balls  $\mathcal{B}$  and a set of points  $P$  in a metric space. Each ball has a capacity, and the capacities of the balls are monotonic. Additionally, each ball has a non-negative real number associated with it which denotes its weight. The weight of a subset  $\mathcal{B}'$  of  $\mathcal{B}$  is the sum of the weights of the balls in  $\mathcal{B}'$ . The goal is to find a minimum weight subset  $\mathcal{B}'$  of  $\mathcal{B}$  and compute an assignment of the points in  $P$  to the balls in  $\mathcal{B}'$  such that the number of points assigned to a ball is at most its capacity. We refer to this problem as Metric Monotonic Capacitated Covering with Weights (MMCC-W). In the case where all balls have the same radius and the same capacity one can get an  $(1, O(1))$ -approximation for MMCC-W by using a constant approximation algorithm for the Budgeted Center problem [2]. However, as we prove, there are instances of MMCC-W that consist of balls of only two distinct radii for which it is NP-hard to obtain a  $(o(\log |P|), c)$ -approximation for any constant  $c$ .

The reduction is from the Set Cover problem. Recall that in Set Cover we are given a set system  $(X, \mathcal{F})$  with  $n = |X|$  elements and  $m = |\mathcal{F}|$  subsets of  $X$ . For each element  $e_i \in X$ , let  $m_i$  be the number of sets in  $\mathcal{F}$  that contain  $e_i$ . Also for each set  $X_j \in \mathcal{F}$ , let  $n_j$  be the number of elements in  $X_j$ . Without loss of generality, we assume that  $n_j \geq 2$ . Note that  $\sum_{i=1}^n m_i = \sum_{j=1}^m n_j$ . Let  $[t] = \{1, \dots, t\}$ . Fix a constant  $c$ , which is the factor by which the balls in the solution are allowed to be expanded, and a real  $0 < \alpha \leq 1$ , which determines the weights of the balls. Let  $N = \max\{m, n\}$  and  $M = c^{1+1/\alpha} N^{2/\alpha}$ .

Given any instance  $I$  of Set Cover we construct an instance  $I' = (P, \mathcal{B})$  of MMCC-W. Intuitively, in a reduction from Set Cover to a metric covering problem, one should take a point for each element and a ball for each set. Also, if an element is contained

<sup>2</sup>  $\epsilon_c$  depends inversely on  $c^2$ .



**Fig. 4** An example illustrating the graph construction for the set system  $(X, \mathcal{F})$ , where  $X = \{e_1, e_2, e_3, e_4, e_5\}$  and  $\mathcal{F} = \{X_1 = \{e_1, e_2, e_3\}, X_2 = \{e_2, e_4\}, X_3 = \{e_1, e_3, e_5\}\}$ . The point (resp. center) vertices are shown using squares (resp. disks)

in a set, the corresponding point should be contained in the respective ball. This construction shows an  $o(\log n)$  inapproximability when we are not allowed to expand the balls. However, we would like to prove the same inapproximability even when we are allowed to expand each ball by factor of  $c$ . To achieve this result, we will tweak the above mentioned construction and take a collection of points for each element, which together form a pathlike structure. Now, we formally describe our construction.

Our instance  $I'$  is built on top of an underlying weighted graph  $G = (P \cup C, E)$  that we construct below. The set  $P$  will be the set of points that need to be covered in  $I'$ , and each point in  $C$  will be the center of a single ball in the instance. We have  $P \cap C = \emptyset$ . The gadget for each element  $e_i \in X$  consists of a path  $\pi_i$  with  $2M \cdot m_i - 1$  vertices, and we refer to this set of vertices as  $V_i$ . Refer a degree 1 vertex on  $\pi_i$  as its 1st vertex, the vertex connected to it as the 2nd vertex and in general for  $\ell \geq 2$ , the index of the vertex connected to the  $\ell$ th vertex other than the  $(\ell - 1)$ th vertex is  $\ell + 1$ . The odd indexed vertices on this path belong to  $P$ , and the even indexed vertices belong to  $C$ . Thus  $M \cdot m_i$  (resp.  $M \cdot m_i - 1$ ) vertices of the path are in  $P$  (resp.  $C$ ). The weight of each path edge is set to be  $cR/M$ , where  $R$  is a positive real. For  $1 \leq \ell \leq m_i$ , we refer to the  $((\ell - 1) \cdot 2M + 1)$ th vertex of  $\pi_i$  as the  $\ell$ 'th *ideal vertex* of  $\pi_i$  (or  $e_i$ ). Note that all ideal vertices belong to  $P$ , and the weighted distance between two consecutive ideal vertices (along  $\pi_i$ ) is  $(cR/M) \cdot 2M = 2cR$ .

Corresponding to each subset  $X_j \in \mathcal{F}$ ,  $G$  contains a vertex  $u_j$  that belongs to  $C$ . Now for each  $e_i \in X$ , consider a one-to-one mapping  $f$  from the set  $[m_i]$  to the set of  $m_i$  subsets of  $X$  that contain  $e_i$ . For  $1 \leq \ell \leq m_i$ , we connect the  $\ell$ th ideal vertex of  $e_i$  to the vertex corresponding to the set  $f(\ell)$  by an edge of weight  $R$ . That is, we connect the first ideal vertex of  $e_i$  to the (vertex for the) first set containing  $e_i$ , the second ideal vertex of  $e_i$  to the (vertex for the) second set containing  $e_i$ , and so on. Note that for any set  $X_j \in \mathcal{F}$ ,  $u_j$  gets connected to  $n_j$  vertices of  $G$ . This concludes the description of  $G$ . See Fig. 4 for an illustration.

We consider the metric space  $(P \cup C, d)$  for  $I'$ , where  $d$  is the shortest path metric on  $G$ . Now we describe the set of balls in  $I'$ . For each  $X_j \in \mathcal{F}$ , we add the ball  $B(u_j, R)$  to  $\mathcal{B}$  and set its capacity to  $n_j$ . Note that  $B(u_j, R)$  contains exactly  $n_j$  points

of  $P$ . Each such ball is called a *subset ball*. For each  $e_i \in X$ , now consider the set of vertices  $V_i$  on path  $\pi_i$ . For each point  $p$  of  $C \cap V_i$ , we add the ball  $B(p, cR/M)$  to  $\mathcal{B}$  and set its capacity to 1. We note that  $B(p, cR/M)$  contains two points from  $P$ . Each such ball is called an *element ball*. The balls in  $\mathcal{B}$  have only two distinct radii. It is not hard to see that the capacities of these balls are monotonic w.r.t their radii. We set the weight of each ball  $B(p, r)$  to  $r^{1+\alpha}$ . This completes the description of instance  $I'$ .

We say that a subset ball corresponding to  $u_j$  is *incident* on path  $\pi_i$  if the ball contains an ideal point of  $\pi_i$ , which happens precisely when  $e_i \in X_j$ .

#### 4.2.1 Properties of the Constructed Instance

We begin with an observation about the points that a  $c$ -expansion of a ball in instance  $I'$  can cover.

**Lemma 4.11** *Let  $B$  be any input ball in instance  $I'$  and let  $q \in P$  be a point that is contained in the  $c$ -expansion of  $B$ . We have: (a) if  $B$  is an element ball corresponding to element  $e_i$  (and therefore centered at a point on  $\pi_i$ ), then  $q$  belongs to path  $\pi_i$ ; and (b) if  $B$  is a subset ball corresponding to set  $u_t$ , then  $q$  belongs to path  $\pi_i$  for some  $e_i \in X_t$ .*

**Proof** For (a), let  $p \in C$  denote the center of  $B$ . Consider any  $p'$  on path  $\pi_j$  where  $i \neq j$ . Now the distance between  $p'$  and  $p$  is at least  $2R$  and thus even after  $c$  factor expansion the ball  $B(p, cR/M)$  cannot contain  $p'$ , as  $M \geq c^2$ . For (b), consider a point  $p'$  on path  $\pi_j$  such that  $e_j \notin X_t$ . By construction, the shortest path from  $u_t$  to  $p'$  has to traverse at least three edges of weight  $R$  plus the distance between two consecutive ideal points in some element gadget. Thus, the distance between  $u_t$  and  $p'$  is at least  $3R + (cR/M) \cdot 2M > cR$ . Thus even after  $c$  factor expansion the ball  $B(u_t, R)$  cannot contain  $p'$ .  $\square$

The next lemma states a necessary condition for a feasible solution for  $I'$ .

**Lemma 4.12** *Suppose we have a subset  $\mathcal{B}'$  of input balls in  $I'$  such that for some element  $e_i \in X$ ,  $\mathcal{B}'$  includes none of the subset balls incident on path  $\pi_i$ . There is no feasible solution to  $I'$  with  $\mathcal{B}'$  as the chosen set of balls.*

**Proof** By Lemma 4.11, the only balls whose  $c$ -expansion contains a point from  $\pi_i$  are the element balls centered at points in  $\pi_i$ , and the subset balls incident on  $\pi_i$ . If none of these subset balls are in  $\mathcal{B}'$ , there is no feasible assignment with  $\mathcal{B}'$ , as the total capacity of the element balls centered at points in  $\pi_i$  is one less than the number of points from  $P$  on  $\pi_i$ .  $\square$

The next lemma says that the above necessary condition is sufficient.

**Lemma 4.13** *Let  $\mathcal{B}'$  be a subset of the balls in  $I'$  such that for every element  $e_i \in X$ ,  $\mathcal{B}'$  includes at least one subset ball incident on  $\pi_i$ . Furthermore, suppose  $\mathcal{B}'$  also contains all element balls. There is a feasible solution to  $I'$  with  $\mathcal{B}'$  as the set of chosen balls.*

**Proof** For each element  $e_i \in X$ , pick one ideal point  $q$  from  $\pi_i$  that is contained in a subset ball in  $\mathcal{B}'$ , and assign  $q$  to that subset ball. For each element ball centered at a point in  $\pi_i$ , among the two points of  $P$  it contains, assign to it the point further from  $q$  (along  $\pi_i$ ). This assignment respects the capacity constraint for all balls in  $\mathcal{B}'$ . Thus, the solution is feasible.  $\square$

#### 4.2.2 The Hardness Guarantee

**Lemma 4.14** *The elements in  $X$  can be covered by at most  $k$  sets of  $\mathcal{F}$  iff there is a solution to MMCC-W for the instance  $I'$  with weight at most  $(k+1) \cdot R^{1+\alpha}$  where the balls in the solution can be expanded by a factor of  $c$ .*

**Proof** Suppose  $X$  can be covered by a collection  $\mathcal{F}'$  of at most  $k$  sets. We construct a set of balls  $\mathcal{B}' \subseteq \mathcal{B}$  whose weight is at most  $(k+1) \cdot R^{1+\alpha}$ . For each set  $X_j \in \mathcal{F}'$ , we add the subset ball  $B(u_j, R)$  to  $\mathcal{B}'$ . We also add all element balls to  $\mathcal{B}'$ . Observe that for each element  $e_i \in X$ ,  $\mathcal{B}'$  includes a subset ball that is incident on  $\pi_i$ . By Lemma 4.13, there is feasible solution to  $I'$  with  $\mathcal{B}'$  as the set of chosen balls. Now the total weight of the subset balls is at most  $k \cdot R^{1+\alpha}$ . The weight of all the element balls centered at points in path  $\pi_i$  is at most  $M \cdot m_i (cR/M)^{1+\alpha}$ . The total weight of all element balls is then at most

$$n \cdot M \cdot m \cdot (cR/M)^{1+\alpha} \leq R^{1+\alpha}.$$

Thus the weight of  $\mathcal{B}'$  is at most  $(k+1) \cdot R^{1+\alpha}$ .

Now suppose there is a solution, with  $\mathcal{B}'$  as the chosen balls, to MMCC-W with  $c$  factor expansion of the balls, and the weight of the balls in  $\mathcal{B}'$  is at most  $(k+1) \cdot R^{1+\alpha}$ . The total number of points in  $P$  is  $\sum_{i=1}^n M \cdot m_i > \sum_{j=1}^m n_j$ . Now the total capacities of the subset balls is  $\sum_{j=1}^m n_j$ . Thus there must be at least one element ball in  $\mathcal{B}'$ . Also the weight of any subset ball is  $R^{1+\alpha}$ . Thus there must be at most  $k$  subset balls in  $\mathcal{B}'$ . We consider the collection  $\mathcal{F}' \subseteq \mathcal{F}$  of sets corresponding to these balls (at most  $k$  in number). By Lemma 4.12, for each element  $e_i \in X$ , there is a subset ball in  $\mathcal{B}'$  centered at some  $u_t$  that is incident on path  $\pi_i$ . Thus,  $e_i \in X_t$  and  $X_t \in \mathcal{F}'$ . Thus  $\mathcal{F}'$  covers all the points of  $X$ .  $\square$

As Set Cover is NP-hard to approximate within a factor of  $o(\log n)$ , from Lemma 4.14, we obtain the following theorem.

**Theorem 4.15** *There exists a constant  $c' > 0$ , such that for any constant  $c \geq 1$ , it is NP-hard to obtain a  $(c' \log |P|, c)$ -approximation for MMCC-W. This result holds even for the particular weight function where the weight of a ball is equal to a constant power of its original radius.*

## References

1. Aggarwal, A., Chakaravarthy, V.T., Gupta, N., Sabharwal, Y., Sharma, S., Thakral, S.: Replica placement on bounded treewidth graphs. In: Algorithms and Data Structures: 15th International Symposium (WADS 2017). Lecture Notes in Computer Science, vol. 10389, pp. 13–24. Springer, Cham (2017)



2. An, H.-C., Bhaskara, A., Chekuri, C., Gupta, S., Madan, V., Svensson, O.: Centrality of trees for capacitated  $k$ -center. *Math. Program.* **154**(1–2), 29–53 (2015)
3. An, H.-C., Singh, M., Svensson, O.: LP-based algorithms for capacitated facility location. In: 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014), pp. 256–265. IEEE Computer Society, Los Alamitos (2014)
4. Aronov, B., Ezra, E., Sharir, M.: Small-size  $\epsilon$ -nets for axis-parallel rectangles and boxes. *SIAM J. Comput.* **39**(7), 3248–3282 (2010)
5. Bar-Ilan, J., Kortsarz, G., Peleg, D.: How to allocate network centers. *J. Algorithms* **15**(3), 385–415 (1993)
6. Becker, A.: Capacitated dominating set on planar graphs. In: Approximation and Online Algorithms: 15th International Workshop (WAOA 2017), pp. 1–16. (2017)
7. Berman, P., Karpinski, M., Lingas, A.: Exact and approximation algorithms for geometric and capacitated set cover problems. *Algorithmica* **64**(2), 295–310 (2012)
8. Bose, P., Carmi, P., Damian, M., Flatland, R.Y., Katz, M.J., Maheshwari, A.: Switching to directional antennas with constant increase in radius and hop distance. *Algorithmica* **69**(2), 397–409 (2014)
9. Brönnimann, H., Goodrich, M.T.: Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.* **14**(4), 463–479 (1995)
10. Chan, T.M., Grant, E., Könnemann, J., Sharpe, M.: Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In: Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012), pp. 1576–1585. ACM, New York (2012)
11. Chuzhoy, J., Naor, J.: Covering problems with hard capacities. *SIAM J. Comput.* **36**(2), 498–515 (2006)
12. Clarkson, K.L., Varadarajan, K.R.: Improved approximation algorithms for geometric set cover. *Discrete Comput. Geom.* **37**(1), 43–58 (2007)
13. Cygan, M., Hajiaghayi, M., Khuller, S.: LP rounding for  $k$ -centers with non-uniform hard capacities. In: IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS 2012), pp. 273–282. IEEE Computer Society, Los Alamitos (2012)
14. Dinur, I., Steurer, D.: Analytical approach to parallel repetition. In: Proceedings of the 2014 ACM Symposium on Theory of Computing (STOC 2014), pp. 624–633. ACM, New York (2014)
15. Gandhi, R., Halperin, E., Khuller, S., Kortsarz, G., Aravind, S.: An improved approximation algorithm for vertex cover with hard capacities. *J. Comput. Syst. Sci.* **72**(1), 16–33 (2006)
16. Ghasemi, T., Razzazi, M.: A PTAS for the cardinality constrained covering with unit balls. *Theor. Comput. Sci.* **527**, 50–60 (2014)
17. Govindarajan, S., Raman, R., Ray, S., Roy, A.B.: Packing and covering with non-piercing regions. In: 24th Annual European Symposium on Algorithms (ESA 2016). LIPIcs. Leibniz International Proceedings in Informatics, vol. 57, pp. 47:1–47:17. Schloss Dagstuhl. Leibniz-Zentrum für Informatik, Wadern (2016)
18. Har-Peled, S., Lee, M.: Weighted geometric set cover problems revisited. *J. Comput. Geom.* **3**(1), 65–85 (2012)
19. Hochbaum, D.S., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM* **32**(1), 130–136 (1985)
20. Kao, M.-J.: Iterative partial rounding for vertex cover with hard capacities. In: Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017), pp. 2638–2653. SIAM, Philadelphia (2017)
21. Khuller, S., Sussmann, Y.J.: The capacitated  $K$ -center problem. *SIAM J. Discrete Math.* **13**(3), 403–418 (2000)
22. Levi, R., Shmoys, D.B., Swamy, C.: LP-based approximation algorithms for capacitated facility location. *Math. Program.* **131**(1–2), 365–379 (2012)
23. Lev-Tov, N., Peleg, D.: Polynomial time approximation schemes for base station coverage with minimum total radii. *Comput. Netw.* **47**(4), 489–501 (2005)
24. Li, S.: On uniform capacitated  $k$ -median beyond the natural LP relaxation. In: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015), pp. 696–707. SIAM, Philadelphia (2015)
25. Lupton, R., Maley, F.M., Young, N.E.: Data collection for the sloan digital sky survey—a network-flow heuristic. *J. Algorithms* **27**(2), 339–356 (1998)
26. Mahdian, M., Pál, M.: Universal facility location. In: European Symposium on Algorithms. Lecture Notes in Computer Science, vol. 2832, pp. 409–421. Springer, Berlin (2003)

27. Mustafa, N.H., Ray, S.: Improved results on geometric hitting set problems. *Discrete Comput. Geom.* **44**(4), 883–895 (2010)
28. Pál, M., Tardos, É., Wexler, T.: Facility location with nonuniform hard capacities. In: 42nd IEEE Symposium on Foundations of Computer Science, pp. 329–338. IEEE Computer Society, Los Alamitos (2001)
29. Petrank, E.: The hardness of approximation: gap location. *Comput. Complex.* **4**, 133–157 (1994)
30. Varadarajan, K.R.: Weighted geometric set cover via quasi-uniform sampling. In: Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010), pp. 641–647. ACM, New York (2010)
31. Wolsey, L.A.: An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica* **2**(4), 385–393 (1982)
32. Wong, S.C.: Tight algorithms for vertex cover with hard capacities on multigraphs and hypergraphs. In: Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017), pp. 2626–2637. SIAM, Philadelphia (2017)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

**Sayan Bandyapadhyay<sup>1</sup> · Santanu Bhowmick<sup>1</sup> · Tanmay Inamdar<sup>1</sup> · Kasturi Varadarajan<sup>1</sup>**

Sayan Bandyapadhyay  
sayan-bandyapadhyay@uiowa.edu

Santanu Bhowmick  
santanu-bhowmick@uiowa.edu

Tanmay Inamdar  
tanmay-inamdar@uiowa.edu

Kasturi Varadarajan  
kasturi-varadarajan@uiowa.edu

<sup>1</sup> Department of Computer Science, University of Iowa, Iowa City, USA