A Unified Framework for Capacitated Covering Problems in Metric and Geometric Spaces

Sayan Bandyapadhyay, Santanu Bhowmick, Tanmay Inamdar, and Kasturi Varadarajan
Department of Computer Science, University of Iowa, Iowa City, USA†

Abstract

In this paper, we consider capacitated version of the set cover problem. Set cover has enormous applications in areas including wireless sensor networks, data mining, and machine learning. The capacity constraint arises naturally in practice. For example, in wireless networks the capacity of each antenna bounds the number of clients it can serve. Capacitated set cover admits an $O(\log n)$ approximation using a greedy algorithm ($n$ is the number of elements) and the approximation factor cannot be improved asymptotically under widely held complexity theoretic assumptions. However, the hard instances of this problem might not occur in practice, where it could be possible to improve the approximation factor. Motivated by the above direction, we consider several metric and geometric versions of the capacitated set cover problem. In one such variant, the elements are points and the sets are the balls in a metric space. We also assume that the capacities of the input balls are monotonic – for any two balls $B_i, B_j$ with radii $r_i$ and $r_j$, and capacities $U_i$ and $U_j$, respectively: $r_i > r_j$ implies that $U_i \geq U_j$. We refer to this problem as the Metric Monotonic Capacitated Covering (MMCC) problem. We note that this capacity model is reasonable in many applications including wireless networks. Also this model generalizes the uniform capacity model. The metric and geometric versions of set cover have been well-studied over the years. However, not much progress has been made for the capacitated variant of these versions. In this paper, we consider several variants of the MMCC problem. Unfortunately, MMCC remains as hard as the set cover problem even when the capacities are uniform, and 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 contribution is to design a unified algorithmic framework for obtaining bi-criteria constant approximations for several variants of the MMCC problem. Our framework is based on an iterative scheme based on LP rounding. As we show, only a constant ($\leq 9$) factor expansion of the balls is sufficient to obtain constant approximations for all these variants. In fact, for Euclidean metric, only $(1 + \epsilon)$ factor expansion is sufficient for any $\epsilon > 0$, where the approximation factor depends inversely on $\epsilon$. We also show that for a 1D variant of MMCC, one can get a constant approximation without needing any expansion of the balls. We believe that the framework has the potential of obtaining similar approximations for many other capacitated covering problems. We also obtained hardness of approximation results for some variants of MMCC. We prove that for any constant $c \geq 1$, MMCC is APX-hard, even when the capacity of each ball is 3 and $c$ factor expansion of the balls is allowed. We also show that for any constant $c \geq 1$, it is NP-hard to obtain a $o(\log n)$-approximation for the weighted version of MMCC with a very simple weight function (constant power of radius), even if $c$ factor expansion of the balls is allowed.

*This material is based upon work supported by the National Science Foundation under Grant CCF-1615845
†{sayan-bandyapadhyay, santanu-bhowmick, tanmay-inamdar, kasturi-varadarajan}@uiowa.edu
1 Introduction

In this paper, we consider the Capacitated Set Cover (CSC) problem, where we are given a set system \((X, F)\) with \(n = |X|\) elements and \(m = |F|\) subsets of \(X\), such that the elements of \(X\) are contained in the union of the subsets in \(F\). For each set \(F_i \in F\), we are also given an integer \(U_i\), which is referred to as its capacity. We are required to find a minimum size subset \(F' \subseteq F\) and compute an assignment of the elements in \(X\) to the sets in \(F'\), such that for each set \(F_i\), the number of points assigned to \(F_i\) is at most \(U_i\). Set cover, which is one of the most fundamental combinatorial optimization problems, is a special case of CSC where the capacity of each set is the same as the number of elements it contains. The greedy algorithm for set cover achieves an approximation ratio of \(O(\log n)\), and this approximation factor cannot be improved under widely held complexity theoretic assumptions [10]. However, it is possible to get improved approximation guarantees for set systems having an underlying geometric structure, like small VC dimension or union complexity [14, 17, 13, 6, 8].

Applications of set cover include placement of wireless sensors or antennas to serve clients, VLSI design, and image processing [4, 14]. 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 [22] 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 \(F\)), Chuzhoy and Naor [7] presented an algorithm with 3 approximation ratio, which was subsequently improved to 2 by Gandhi et al. [11]. 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 [15, 23].

Compared to the set cover problem, relatively fewer special versions of the CSC problem have been studied in the literature. In one such version, the set \(X\) consists of points in the plane, the set \(F\) comprises of all possible unit balls, and the capacity of each ball is a given integer \(L\). This problem appeared in the Sloan Digital Sky Survey project [20]. When \(L\) is at least \(|X|\), this is a regular covering problem, for which Hochbaum and Maass [14] gave a polynomial time approximation scheme (PTAS) using a grid shifting strategy. Their result was matched in the capacitated case by Ghasemi and Razzazi [12] using a similar technique. Berman et al. [4] have considered the “soft” capacitated version of the CSC problem that allows making multiple copies of input sets. To the best of our knowledge, there are no other results on capacitated geometric covering problems where the \(O(\log n)\) approximation guarantee is improved.

Motivated by the above line of work, we consider several variants of CSC where the elements are points and the sets are induced by objects in metric or geometric space. To define these variants we describe only the corresponding set systems and the assumptions made about the capacities.

Given a metric space \((X, d)\), we define a ball of radius \(r \geq 0\) centered at \(x \in X\) as \(B(p, r) = \{y \in X \mid d(x, y) \leq r\}\). In the Metric Monotonic Capacitated Covering (MMCC) problem, we are given two sets of points \(P\) and \(C\) in a metric space \((P \cup C, d)\), along with \(B\), a set of balls centered at the points of \(C\). The set system considered in this variant is \((P, B)\). Each ball \(B_i = B(c_i, r_i)\) has a capacity \(U_i\). We assume that the capacities of the given balls are monotonic w.r.t. the radii, i.e., for any two balls \(B_i, B_j \in B\), the following holds: \(r_i > r_j \implies U_i \geq U_j\). We note, that we consider the version of the problem with hard capacities, i.e., we are not allowed to make multiple copies of the balls in \(B\). In the case, where the metric is Euclidean we refer to the version of MMCC as the Euclidean Monotonic Capacitated Covering (EMCC) problem. The dimension of the Euclidean space is assumed to be a constant. We also consider a special version of the 1-dimensional EMCC problem, where the capacity of each interval is uniform (or same). We refer to this version as the Uniform Capacitated Interval Cover (UCIC) problem.
It is easy to see that MMCC is a generalization of many natural versions, such as where all balls have equal capacity (but may have different radii); or where all balls have equal radii (but may have different capacity). 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.

The best known approximation for the MMCC problem is via the greedy algorithm of Wolsey [22], which yields an $O(\log n)$ approximation. It is NP-hard to improve the approximation factor asymptotically beyond that, as demonstrated by the following reduction of set cover to MMCC. We take a ball of radius 1 corresponding to each set, and a point corresponding to each element. If an element is in a set, then the distance between the center of the corresponding ball and the point is 1. The capacity of each ball is the number of points contained in it. We consider the metric space induced by the centers and the points. It is easy to see that any solution for this instance of MMCC directly gives a solution for the input instance $(X, F)$ of the general set cover, implying that it is not possible to get any approximation guarantee better than the $O(\log n)$ bound for set cover.

Since getting a better approximation than $O(\log n)$ is seemingly impossible for the MMCC problem, we focus our efforts on finding a bicriteria 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 MMCC, 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 $B_i$, $d(c_i, p_j) \leq \beta \cdot r_i$), and the cost of the solution is at most $\alpha$ times that of an optimal solution (which does not expand the balls). From the reduction above, we can see that it is NP-hard to get an $(f(n), \lambda)$-approximation for any $\lambda < 3$ and $f(n) = o(\log n)$. We note that it is a common practice in wireless network setting to expand the radii of antennas at the planning stage to improve the quality of service. For example, Bose et al. [5] propose a scheme for replacing omni-directional antennas by directional antennas that expands the antennas by a constant factor.

Related Work  Though there are very few works in the domain of capacitated set cover, the capacitated version of facility location and clustering type problems have been well-studied over the years. One of such popular clustering problems is capacitated $k$-center. For the uniform capacitated version of this problem, $O(1)$ approximations were given in [3, 16]. The non-uniform capacitated version is addressed in [2, 9]. Notice that the problem dual to the non-uniform capacitated $k$-center is a special case of MMCC, where the set $B$ consists of balls of a common radius $r$ centered at each point of the capacitated $k$-center instance. For other related optimization problems such as metric facility location, $k$-median etc - see [1, 18, 19] for recent advances.

Our Results and Contributions. In this paper, we design an LP rounding based algorithmic framework for obtaining constant approximation (with constant expansion of the objects in some cases) for the capacitated covering problems we consider. We believe that the LP rounding framework may be of independent interest for obtaining similar results for related capacitated problems. We obtain the following results using the framework.

- An $(O(1), 9)$-approximation for the MMCC problem
- An $(O(1), 6.47)$-approximation for the MMCC problem with uniform capacities
- An $(O(\epsilon^{-4d} \log(1/\epsilon)), 1 + \epsilon)$-approximation for EMCC problem in $\mathbb{R}^d$
- An $(O(\epsilon^{-2d}), 1 + \epsilon)$-approximation for the unit radii version of EMCC problem in $\mathbb{R}^d$
• An $O(1)$-approximation for UCIC

Not much progress has been made in recent years towards understanding the status of capacitated covering problems. Hence we make significant progress by achieving $(O(1), O(1))$-approximations for several metric and geometric capacitated covering problems. We note that when the input balls have the same radii in case of MMCC, the results (in fact, even a $(1, O(1))$-approximation) follow from the results for capacitated k-center [9, 2]. Alternatively, consider the following simpler approach: solve the natural LP-relaxation and partition the balls into clusters such that for each cluster (a) the total opening cost that the LP pays for all the balls is at least $1/2$, and (b) any ball when expanded by $O(1)$ factor contains all other balls in the cluster. Then from each cluster select $t$ balls, where $t$ is the ceiling of the total opening cost of the cluster, and expand the selected balls by $O(1)$ factor. This results in a $(2, O(1))$ approximation. A partition into clusters with these properties is possible when all input balls have the same radii, but not when the input balls have widely different radii. The novelty in our work lies in handling the latter situation. Though the LP rounding technique is a standard tool that has been used in the literature of capacitated problems, our actual rounding scheme is different from the existing ones. In fact, the standard rounding schemes for facility location are not useful for our problems, as there a point can be assigned to any facility. But in our case the points 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.

Our approximation guarantee hinges on a classification of the balls into “heavy” and “light” (depending on the degree to which a ball is opened in the solution to the LP relaxation). We use a flow rerouting scheme that allows us to fully open a few light balls and close appropriately many other light balls, thereby giving us an integral solution that is within a constant factor of the cost of the LP solution. The framework and its applications are described in Section 3.

We also obtain some hardness of approximation results that give a better understanding of the problems we consider. Firstly, we show that for any constant $c \geq 1$, there exists a constant $c_c > 0$ such that it is NP-hard to obtain a $(1 + c_c, c)$-approximation for the MMCC problem, even when the capacity of all balls is 3. One strict requirement of the hardness construction is that all the balls in the hard instance cannot have the same radii. 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, 9]), to obtain a $(1, O(1))$-approximation. We also consider the weighted version of MMCC which is a natural extension of the unweighted version. We show that for any constant $c \geq 1$, there exists a constant $c' > 0$, such that it is NP-hard to obtain a $(c' \log |P|, c)$-approximation for the weighted version of MMCC with a very simple weight function (constant power of original radius). We describe these two hardness results in Section 4.

2 LP relaxation for CSC

First we consider an integer programming formulation of CSC. For each set $F_i \in F$, let $y_i = 1$ if the set $F_i$ is selected in the solution, and 0 otherwise. Similarly, for each element $e_j \in X$ and each set $F_i \in F$, let the variable $x_{ij} = 1$ if $e_j$ is assigned to $F_i$, and $x_{ij} = 0$ otherwise. We relax these integrality constraints, and state the corresponding linear program as follows:

$$\begin{align*}
\text{minimize} & \quad \sum_{F_i \in F} y_i \\
\text{s.t.} & \quad x_{ij} \leq y_i \quad \forall e_j \in X, \forall F_i \in F
\end{align*}$$ (CSC-LP) (1)
\[ \sum_{e_j \in X} x_{ij} \leq y_i \cdot U_i \quad \forall F_i \in \mathcal{F} \tag{2} \]
\[ \sum_{F_i \in \mathcal{F}} x_{ij} = 1 \quad \forall e_j \in X \tag{3} \]
\[ x_{ij} = 0 \quad \forall e_j \in X, \forall F_i \in \mathcal{F} \text{ such that } e_j \notin F_i \tag{4} \]
\[ x_{ij} \geq 0 \quad \forall e_j \in X, \forall F_i \in \mathcal{F} \tag{5} \]
\[ 0 \leq y_i \leq 1 \quad \forall F_i \in \mathcal{F} \tag{6} \]

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

3 The Algorithmic Framework

In this section, we describe the framework based on LP rounding. The framework consists of two major steps – Preprocessing and Rounding. The Rounding step is in turn divided into two smaller steps – Cluster Formation and Selection of Objects. For simplicity of exposition, at first we 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 and UCIC problems.

3.1 The Algorithm for the MMCC Problem

Before we describe the algorithm we introduce some definitions and notations which will heavily be used throughout this section. Recall, that in MMCC the set system is \((P, B)\), where \(P\) is a set of points and \(B\) is a collection of balls. Consider any solution \((x, y)\) of CSC-LP w.r.t MMCC, where \(X = P\) and \(\mathcal{F} = B\). For point \(p_j \in P\) and ball \(B_i \in B\), 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 \(B_i \in B\) 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 \(B'\) to a ball \(B_k \notin B''\)” means obtaining a new solution \((\hat{x}, \hat{y})\) from the current solution \((x, y)\) in the following way: (a) For all points \(p_j \in P'\), \(\hat{x}_{kj} = x_{kj} + \sum_{B_i \in B' \setminus B_k} x_{ij}\); (b) for all points \(p_j \in P'\) and balls \(B_i \in B'\), \(\hat{x}_{ij} = 0\); (c) the other \(\hat{x}_{ij}\) variables be 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 \(0 < \alpha \leq \frac{1}{2}\) be a parameter to be fixed later. A ball \(B_i \in B\) 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 \(\mathcal{H} = \{B_i \in B \mid y_i = 1\}\) denote the set of heavy balls, and \(\mathcal{L} = \{B_i \in B \mid 0 < y_i \leq \alpha\}\) 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 \(d(e_i, p_j)\) 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 $\tilde{\sigma}$ points might receive fractional amount of flow from the chosen balls. However, as the capacity of each ball is integral we can find, using network flow schemes, another solution with the same set of chosen balls, such that the new solution satisfies all the properties of $\tilde{\sigma}$ and the additional property, that for each point $p$, there is a single chosen ball that sends one unit of flow to $p$ [7]. We note that choosing an optimal LP solution as the input 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 $\tilde{\sigma} = (\tilde{x}, \tilde{y})$ from $\sigma$, such that each ball in $\tilde{\sigma}$ is either heavy or light, and for each point $p_j \in P$, the amount of flow $p_j$ receives from the set of 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 $\tilde{\sigma}$ 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 $AvCap(B_i)$, to be $U_i - \sum_{p_j \in P} x_{ij}$. We now describe the preprocessing step.

### 3.1.1 The Preprocessing Step

**Lemma 1.** Given a feasible LP solution $\sigma = (x, y)$, and a parameter $0 < \alpha \leq \frac{1}{2}$, there exists a polynomial time algorithm to obtain another LP solution $\tilde{\sigma} = (\tilde{x}, \tilde{y})$ that satisfies Constraints 1 to 6 except 4 of CSC-LP. Additionally, $\tilde{\sigma}$ satisfies the following properties.

1. Any ball $B_i \in \mathcal{B}$ with non-zero $\tilde{y}_i$ is either heavy ($\tilde{y}_i = 1$) or light ($0 < \tilde{y}_i \leq \alpha$).
2. For each point $p_j \in P$, we have that
   \[ \sum_{B_i \in \mathcal{L}, \tilde{x}_{ij} > 0} \tilde{y}_i \leq \alpha, \tag{7} \]
   where $\mathcal{L}$ is the set of light balls with respect to $\tilde{\sigma}$.
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. $cost(\tilde{\sigma}) \leq \frac{1}{\alpha} cost(\sigma)$.

**Proof.** The algorithm starts off by initializing $\tilde{\sigma}$ 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 $\alpha < \sum_{B_i \in \mathcal{L}_j} \tilde{y}_i \leq 2\alpha$. Note that such a set $\mathcal{L}_j$ always exists because the $\tilde{y}_i$ variables corresponding to light balls are at most $\alpha \leq \frac{1}{2}$. Let $B_k$ be a ball with the largest radius from the set $\mathcal{L}_j$. (If there are more than one balls 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 $B_m \in \mathcal{L}_j$, by the monotonicity assumption along with the above mentioned assumption, we have that $U_k \geq U_m$.

2. We set $\tilde{y}_k \leftarrow \sum_{B_i \in \mathcal{L}_j} \tilde{y}_i$, and $\tilde{y}_m \leftarrow 0$ for $B_m \in \mathcal{L}_j \setminus \{B_k\}$. Note that $\tilde{y}_k \leq 2\alpha \leq 1$. Let $A = \{p_l \in P \mid x_{jl} > 0 \text{ for some } B_i \in \mathcal{L}_j \setminus \{B_k\}\}$ 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 $\tilde{\sigma}$, all other $\tilde{x}_{ij}$ and $\tilde{y}_i$ 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 $\tilde{y}_k > \alpha$, $B_k$ is no longer a light ball.)
Finally, for all balls $B_i$ such that $\overline{y}_i > \alpha$, we set $\overline{y}_i = 1$, making them heavy. Thus $\text{cost}(\sigma)$ is at most $\frac{1}{\alpha}$ times $\text{cost}(\sigma)$, and $\sigma$ satisfies all the conditions stated in the lemma.

**Remark.** As a byproduct of Lemma 1, we get a simple $(4, 3)$-approximation algorithm for the soft capacitated version of our problem (see Section 3.4.2).

### 3.1.2 The Main Rounding Step

The main rounding step can logically be divided into two stages. The first stage, *Cluster Formation*, is the crucial step 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 so that the number of chosen balls is not “large”. The idea is to use the capacity of a selected light ball as much as possible to reroute the flow from other intersecting balls. This in turn frees up some capacity of 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 due to the rerouting of flow from the other balls to the heavy ball. In the second stage, referred to as *Selection of Objects*, we select exactly one ball (notably 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 $O$ (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 $O$ 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 of light balls that is not in $O$, and not yet clustered. 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 $B_i'$ is same as that in the preprocessed solution, i.e., the flow assignment of $p_j$ remains unchanged. 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 $\sum_{p_j \in P} \overline{\pi}_{tj}$ out of $B_i$:

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 $A_j$ denote the set of points currently being served by $B_j$. Also, for $B_j \in \Lambda$, let $k_j = \min\{U_j, |A_j|\}$, i.e. $k_j$ denotes the minimum of its capacity, and the number of active points that it serves. We select the ball $B_t \in \Lambda$ with the maximum value of $k_j$, and add it to the set $O$. 

6
There are following three cases depending on the value of $k_t$.

1. $k_t = |A_t| \leq U_t$. In this case, for each point $p_t$ in $B_t$ that gets served by $B_t$, we reroute the flow of $p_t$ from $B \setminus O$ to $B_t$. Note that $p_t$ 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 $B_t \in \mathcal{H}$, $\text{AvCap}(B_t)$ increases by $\sum_{p_t:B_t \text{ serves } p_t} x_{it}$.

2. $k_t = U_t < |A_t|$, but $k_t = U_t > 1$. Note that according to the LP solution, the ball $B_t$ has already used the following amount of capacity $-\sum_{p_t \in A_t} x_{it} \leq \alpha U_t = \alpha k_t$.

In this case, we select a point $p_j \in A_t$ arbitrarily, and reroute the flow of $p_j$ from $B \setminus O$ to $B_t$. This will increase the available capacity of other balls in $B \setminus O$ 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 $A_t$ until we encounter a point $p_t$ such that rerouting the flow of $p_t$ from $B \setminus O$ to $B_t$ violates the capacity of $B_t$. Thus the flow assignment of $p_t$ 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 $A_t$ in this manner, since $U_t > 1$ and $\alpha \leq 1/2$.

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

(i) Reroute the flow of $p_j$ from $\Lambda$ to $B_t$, and thus $p_j$ is no longer being served by a ball in $\Lambda$. Note that in this step, we reroute at most $\alpha$ amount of flow. 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 $O$. (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 $O$ or clustered. We set $\overline{y}_t \gets 1$ for each ball $B_t \in O$, thus making it heavy. For convenience, we also set $\text{cluster}(B_t) = \{B_t\}$ for each $B_t \in O$.

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 $C$. Let $C = \emptyset$ initially. For each heavy ball $B_t$, we consider $\text{cluster}(B_t)$ and perform the following steps.

(a) If $\text{cluster}(B_t)$ consists of only the heavy ball, we add $B_t$ to $C$.

(b) Otherwise, let $B_j$ be a largest ball in $\text{cluster}(B_t)$. If $B_j = B_t$, then we expand it by a factor of 3. Otherwise, $B_j$ is a light ball intersecting with $B_t$, 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 $C$, and we set $\overline{y}_s \gets 0$ for any other ball $B_s$ in the cluster.

After processing the clusters, we set $\overline{y}_t \gets 1$ for each ball $B_t \in C$. Finally, we return the current set of heavy balls as the solution.

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 $C$. 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 $\overline{\sigma} = (\overline{x}, \overline{y})$ at hand after preprocessing, as $\sigma = (x, y)$. Now we bound the size of the set $O$ computed during Cluster Formation. The basic idea is that each light ball added to $O$ 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 $O$. 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 $O$ plays a crucial role in the argument.

Let $H_1$ (resp. $L_1$) 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 $O$) 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 active points being served by $L_j$, and its capacity). Note that $k_1 \geq k_2 \geq \cdots \geq k_I$. For any $B_i \in H_1$, 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 same as the increase in $AvCap(B_i)$ when $L_t$ is added to $O$. Correspondingly, we define $Y(L_t, B_i)$, the “$y$-credit contributed by $L_t$ to $B_i$”, to be $\frac{F(L_t, B_i)}{k_i}$. Now, the increase in available capacity over all balls in $H_1$ is $F_t = \sum_{B_i \in H_1} F(L_t, B_i)$. 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 2.** Consider a ball $B_t \in O$ processed in the Cluster Formation stage, step c. For $0 < \alpha \leq 3/8$, $F_t \geq \frac{1}{5}k_t$.

**Proof.** The algorithm ensures that the flow assignment of each point in $A_t$ is same as that w.r.t. the preprocessed solution. Thus by property 2 of Lemma 1, each such point gets at most $\alpha$ amount of flow from the balls in $O$. Now there are three cases corresponding to the three substeps of step c.

1. When $k_t = |A_t| \leq U_t$, it is possible to reroute the flow of all points of $A_t$ from $B \setminus O$ to $B_t$. Therefore, we get that $F_t \geq (1 - \alpha)k_t \geq \frac{4}{5}k_t$, since $0 < \alpha \leq 3/8$.
2. When $1 < k_t = U_t < |A_t|$, it is possible to reroute the flow of at least $\lfloor (1 - \alpha)U_t \rfloor = \lfloor (1 - \alpha)k_t \rfloor$ points of $A_t$ from $B \setminus O$ 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 $\frac{1}{5}k_t$, again by using the fact that $0 < \alpha \leq 3/8$.
3. When $1 = k_t = U_t < |A_t|$, $F_t \geq (1 - 2\alpha) \geq \frac{3}{5}k_t$, as $0 < \alpha \leq 3/8$.

At any moment in the Cluster Formation stage, for any ball $B_i \in H_1$, define its $y$-accumulation as

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

The idea is that $B_i$ gets $y$-credit when a light ball is added to $O$, and loses $y$-credit when it adds a light ball to cluster($B_i$); thus, $\tilde{y}(B_i)$, a proxy for the available capacity of $B_i$, indicates the “remaining” $y$-credit. The next lemma gives a relation between the $y$-accumulation of $B_i$ and its available capacity.

**Lemma 3.** Fix a heavy ball $B_i \in H_1$, and an integer $1 \leq t \leq I$. Suppose that $L_1, L_2, \cdots, L_t$ have been added to $O$. Then $AvCap(B_i) \geq \tilde{y}(B_i) \cdot k_t$.

**Proof.** The proof is by induction on $t$. For this proof, we abbreviate $AvCap(B_i)$ 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 $L_1, \cdots, L_{t-1}$ to the set $O$, 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 $A_i \geq \tilde{y}(B_i) \cdot k_{t-1}$.

From this point, there is a phase where $B_i$ uses up some of its available capacity to add 0 or more balls to cluster($B_i$), after which we add $L_t$ to $O$.
Suppose that in the said phase, one or more balls is added to \( \text{cluster}(B_i) \). Note that the first such ball \( B_j \) serves \( k \leq k_{t-1} \) points, because of the order in which we add balls to \( O \). 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,
\[
A_i' \geq A_i - k \cdot y_j \geq (\tilde{y}(B_i) \cdot k_{t-1}) - k \cdot y_j \geq (\tilde{y}(B_i) - y_j) \cdot k_{t-1} = (\tilde{y}(B_i)' \cdot k_{t-1})
\]
where we use the fact that \( k \leq k_{t-1} \) in the third inequality. Therefore, the claim is true. Note that \( B_i \) may add multiple balls \( B_j \) 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 \( O \). 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 \( k_t \leq k_{t-1} \).

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 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 \( O \), suppose \( L_t \) was the ball whose addition to \( O \) 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, 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 \( y_t k_t \leq \alpha k_t \).

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 \( O \), which is a contradiction. \( \square \)

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

**Proof.** At the end of Cluster Formation stage,
\[
\sum_{B_i \in \mathcal{H}_1} \tilde{y}(B_i) \geq \sum_{B_i \in \mathcal{H}_1} Y(L_t, B_i) - \sum_{B_i \in \mathcal{H}_1} \sum_{B_j \in \text{cluster}(B_i)} y_j \geq \sum_{1 \leq t \leq I} \frac{F_t}{k_t} - \sum_{B_j \in \mathcal{L}_1} y_j \geq \frac{1}{5} \cdot |O| - \sum_{B_j \in \mathcal{L}_1} y_j
\]
(8)
where the second inequality follows, as \( F_t = \sum_{B_i \in \mathcal{H}_1} F(L_t, B_i) = k_t \cdot \sum_{B_i \in \mathcal{H}_1} Y(L_t, B_i) \), and we used Lemma 2 to get the last inequality. Now, adding the inequality of Lemma 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. \( \square \)

**Lemma 6.** The cost of the solution returned by the algorithm is at most a constant 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 \( |O| + |H_1| \). Using Lemma 5,

\[
|O| + |H_1| \leq 5 \cdot \left( (1 + \alpha) \cdot |H_1| + \sum_{B_j \in L_1} y_j \right) + |H_1| = (6 + 5\alpha) \cdot \sum_{B_j \in H_1} y_j + 5 \cdot \sum_{B_j \in L_1} y_j
\]

\[
\leq (6 + 5\alpha) \left( \sum_{B_j \in H_1} y_j + \sum_{B_j \in L_1} y_j \right) \leq (6 + 5\alpha) \cdot \text{cost}(\sigma) \leq \left( \frac{6 + 5\alpha}{\alpha} \right) \cdot \text{OPT} = 21 \cdot \text{OPT}
\]

The last equality follows by setting \( \alpha = 3/8 \). \qed

Lemma 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 light 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.

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 Balls stage (see Section 3.4.1). Lastly, from Lemma 6 and Lemma 7, we get the following theorem.

Theorem 1. There is a polynomial time \((O(1), 9)\)-approximation algorithm for the MMCC problem.

3.2 The Algorithm for the EMCC Problem

Overview of the Algorithm. For the EMCC problem, we can exploit the structure of \( \mathbb{R}^d \) to restrict the expansion 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 \( L_j \) of light balls. We select a largest ball from this set and reroute the flow of other balls in \( L_j \) 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 \( L_j \), we consider the bounding hypercube whose side is at most a constant times the maximum radius of any ball from \( L_j \), and subdivide it into multiple cells. The granularity of the cells is carefully chosen to ensure that (1) Selecting the maximum radius ball with centers lying in that cell, and expanding it by \((1 + \epsilon)\) factor is enough for rerouting the flow from all such balls to this ball, and (2) The total number of cells is \( \text{poly}(1/\epsilon) \). The Cluster Formation stage for the EMCC problem is exactly same as that for the MMCC problem. Finally, in the Selection of Balls stage, we use similar technique as 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 2. For any \( \epsilon > 0 \), there is a polynomial time \((O(\epsilon^{-4d} \log(1/\epsilon)), 1 + \epsilon)\)-approximation algorithm for the EMCC problem in \( \mathbb{R}^d \). Moreover, for the unit radii version there is a polynomial time \((O(\epsilon^{-2d}), 1 + \epsilon)\)-approximation algorithm.
Now we describe the algorithm in detail. For simplicity, at first we consider the \( d = 2 \) case. Our algorithm takes an additional input – a constant \( \epsilon > 0 \), and gives an \( O(\epsilon^{-8}\log(1/\epsilon)) \) 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 8.** Given a feasible LP solution \( \sigma = (x, y) \) corresponding to an EMCC instance in \( \mathbb{R}^2 \), and parameters \( 0 < \alpha \leq 2 \), and \( \epsilon > 0 \), there exists a polynomial time algorithm to obtain another LP solution \( \overline{\sigma} = (\overline{x}, \overline{y}) \) that satisfies Constraints 1 to 6 except 4 of CSC-LP. Additionally, \( \overline{\sigma} \) satisfies the following properties.

1. Any ball \( B_i \in \mathcal{B} \) with non-zero \( y_i \) is either heavy (\( y_i = 1 \)), or light \( 0 < y_i \leq \alpha \).
2. For each point \( p_j \in \mathcal{P} \), we have that
   \[
   \sum_{B_i \in \mathcal{L} : y_i > 0} y_i \leq \alpha
   \]
   where \( \mathcal{L} \) is the set of light balls with respect to \( \overline{\sigma} \).
3. For any heavy ball \( B_i \), and any point \( p_j \in \mathcal{P} \) 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 \( p_j \in \mathcal{P} \) served by \( B_i \), \( d(c_i, p_j) \leq r_i \).
5. \( \text{cost}(\overline{\sigma}) = O(\epsilon^{-2}\log(1/\epsilon)) \cdot \text{cost}(\sigma) \).

**Proof.** As in Lemma 1, in each iteration, we pick an arbitrary point \( p_j \in \mathcal{P} \) for which the Inequality (9) is not met, and consider a set \( \mathcal{L}_j \) of light balls. We select a subset of these balls, and for each such selected ball \( B_i \), we set \( y_i \leftarrow 1 \). For each ball \( B_i \) in \( \mathcal{L}_j \) which is not selected, we set \( y_i \leftarrow 0 \). We show that the corresponding solution satisfies the desired properties. Let \( r \) be the radius of a maximum radius ball from the set \( \mathcal{L}_j \). Now all balls from the set \( \mathcal{L}_j \) contain a common point \( p_j \). Thus any point that belongs to a ball with radius less than \( r\epsilon/2 \), is within distance \((1 + \epsilon) \cdot r \) from the center of a maximum radius ball. As we are going to select such a maximum radius ball and its capacity is larger than the capacity of any ball with radius less than \( r\epsilon/2 \), we discard balls with radius smaller than \( r\epsilon/2 \). We reroute all the flow from those balls to the selected ball.

Now we divide the balls into \( O(\log(1/\epsilon)) \) classes such that the \( i^{\text{th}} \) class contains balls of radii between \( 2^{i-1}r\epsilon \) and \( 2^i r\epsilon \) for \( 0 \leq i \leq O(\log(1/\epsilon)) \). We consider each class separately and select a subset of balls from each class. Consider the \( i^{\text{th}} \) class. Note that there exists an axis-parallel square of side \( 2^{i+2}r\epsilon \) such that the centers of the balls in \( i^{\text{th}} \) class are contained in it. We subdivide this square into smaller squares, by overlaying a grid of granularity \( 2^{i-2}r^2 \). Note that the number of smaller squares (henceforth referred to as a *cell*) in the larger square is \( O(\epsilon^{-2}) \). We show how to select at most one light ball from each cell.

Consider a cell from the subdivision, and let \( \mathcal{L}_j' \) be the balls in \( i^{\text{th}} \) class (with radius at least \( 2^{i-1}r\epsilon \)) whose centers belong to this cell. Now, we select a ball \( B_m \in \mathcal{L}_j' \) with the maximum radius \( r_m \) from the set \( \mathcal{L}_j' \), and reroute the flow from the other balls to \( B_m \). Since the center of \( B_m \) is within distance \( 2^{i-1}r^2 \) from the center of any ball \( B_l \in \mathcal{L}_j' \), all the points contained in any ball \( B_l \in \mathcal{L}_j \) are within distance \( 2^{i-1}r^2 + r_l \leq \epsilon r_m + r_m = (1 + \epsilon) \cdot r_m \) from the center \( c_m \) of the ball \( B_m \).

Note that the capacity \( U_m \) of the ball \( B_m \) is at least that of the capacity of any ball from the set \( \mathcal{L}_j' \), because of the *monotonicity* property. Furthermore, the ball \( B_m \) has enough capacity to receive all the redirected flow, since
\[
\sum_{p_j \in \mathcal{P}, B_l \in \mathcal{L}_j, B_l \text{ serves } p_j} x_{lj} \leq \sum_{B_l \in \mathcal{L}_j'} U_l \cdot y_l \leq U_m \cdot \sum_{B_l \in \mathcal{L}_j} y_l \leq U_m \cdot 2 \cdot \alpha \leq U_m.
\]
As \( \sum_{B_l \in \mathcal{L}_j} y_l > \alpha \) and we select at most \( O(\epsilon^{-2}\log(1/\epsilon)) \) balls in total from \( \mathcal{L}_j \) the increase in cost is by at most \( O(\epsilon^{-2}\log(1/\epsilon)) \) factor. 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 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 remains similar to that of Lemma 8.

**Lemma 9.** There exists a scheme for the Selection of Objects stage for the EMCC problem, such that for any $\epsilon > 0$,

1. From each cluster, we choose at most $O(\epsilon^{-6})$ balls.
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.** We show how to process each cluster $C_i$ by choosing a set of balls $B_i \subseteq C_i$ of size $O(\epsilon^{-6})$, such that each ball in $B_i$ is expanded by at most $1 + \epsilon$ factor. Finally, for each point $p_j \in P$ that is served by the heavy ball $B_h \in C_i$, we reroute the flow from $B_h$ to an arbitrary ball $B' \in B_i$ (possibly $B_h$) such that $p_j$ is contained in $B'$. We also set $\tau_i \leftarrow 0$ for all balls $B_i \in C_i \setminus B_i$, and $\overline{\tau}_i \leftarrow 1$ for all balls $B_i \in B_i$. The feasibility of this solution follows easily from the monotonicity property. Finally, we return $\bigcup B_i$ over all clusters $C_i$ as the solution. It only remains to describe how to choose the set $B_i$ for each cluster $C_i$.

If the cluster $C_i$ contains only the heavy ball $B_h$, we set $B_i = \{B_h\}$. In this case, we do not need any expansion. Otherwise, let $r_h$ be the radius of the heavy ball $B_h$ at the center of the cluster $C_i$, and let $r_m$ be the maximum radius of any ball from the cluster $C_i$.

If $r_m \leq r_h \cdot \epsilon/2$, then we expand $B_h$ by a factor of $1 + \epsilon$, and set $B_i = \{B_h\}$.

Otherwise, we consider one of the following three cases. In each case, we subdivide the enclosing square of side $4r_m$ into a grid, which is very similar to Lemma 8. Therefore, we discuss in brief the granularity of the grid and the balls that are added to the set $B_i$.

1. $r_h < r_m \cdot \epsilon/4$. In this case, $B_m$ can expand by a factor of $1 + \epsilon$ and can cover the points covered by the balls in $C_i$ that have radius smaller than $r_m \cdot \epsilon/4$, and discard them. Then, we overlay a grid of granularity $r_m \cdot \epsilon^2/8$, which adds $O(\epsilon^{-4})$ balls with radius at least $r_m \cdot \epsilon/4$, to the set $B_i$.

2. $r_m \cdot \epsilon/4 \leq r_h \leq r_m/c$, for some constant $c > 1$. In this case, we discard balls from $C_i$ with radii less than $r_h \cdot \epsilon/4$, and then overlay a grid of granularity $r_h \cdot \epsilon^2/8 \geq r_m \cdot \epsilon^3/32$, which adds $O(\epsilon^{-6})$ balls with radius at least $r_m \cdot \epsilon^2/(16)$, to the set $B_i$.

3. $r_h \geq r_m/c$ for some constant $c > 1$. In this case, we discard balls from $C_i$ with radii smaller than $r_m \cdot \epsilon/(2c)$, and then overlay a grid of granularity $r_m \epsilon^2/(4c)$, which adds $O(\epsilon^{-4})$ balls with radius at least $r_m \cdot \epsilon/(2c)$, to the set $B_i$.

In the second and the third cases above, we also add the heavy ball $B_h$ to the set $B_i$, if it is not added already. \hfill \Box

We note that Lemma 8, and Lemma 9 can be modified to work in $\mathbb{R}^d$. In this case, the increase in the cost of solution become $O(\epsilon^{-d} \log(1/\epsilon))$, and $O(\epsilon^{-3d})$, respectively (where the constants inside the Big-Oh may depend exponentially on the dimension $d$). If the radii of all balls are equal, then we can improve both the bounds to $O(\epsilon^{-d})$, since grids of granularity $O(\epsilon^{-1})$ suffice. Therefore, with suitable modifications to Lemma 8, the analysis of the Cluster Formation stage from the MMCC algorithm, and Lemma 9, Theorem 2 follows.

### 3.3 The Algorithm for the UCIC Problem

In this section, we consider the UCIC problem to demonstrate a variant of the general MMCC problem, where we do not need any expansion of the objects. In the Preprocessing step, in each iteration, we consider a set $L_j$ of light intervals, that serve a common point $p_j$. Let $I_l, I_r \in L_j$
be the intervals with leftmost left endpoint, and rightmost right endpoint respectively. It is easy to see that $I_L$ and $I_R$ together cover all points that are served by any interval in the set $L_j$. By using the arguments as in Lemma 1, we can show that they also have enough capacity, so that the solution remains feasible when we reroute the flow for these points from $L_j \setminus \{I_L, I_R\}$ to an appropriate interval $I_L$ or $I_R$. We also modify the $y_i$ values appropriately. Therefore, we increase the cost of the solution by at most $2/\alpha$ factor, without any expansion.

The Cluster Formation step and its analysis are same as that in the MMCC problem. As for the Selection of Objects stage, we use a similar idea as the Preprocessing stage. Note that in a cluster $C_i$, there is a heavy interval $I_h$, and a (possibly empty) set of light intervals that intersect $I_h$. Here, we can choose the intervals $I_L, I_R$ with leftmost left endpoint and rightmost right endpoint respectively, along with the heavy interval $I_h$. Again, it is easy to see that $I_L, I_h, I_R$ cover all the points covered by $C_i$, and they have enough capacity. In this way, we add at most 3 intervals per cluster, which implies a constant factor approximation.

**Theorem 3.** There is a polynomial time $O(1)$-approximation algorithm for the UCIC problem.

### 3.4 Variants of the MMCC Problem

In this section, we consider two variants of the MMCC problem – the version where all capacities are equal and the soft capacitated version.

#### 3.4.1 Uniform Capacitated Version of MMCC

**Lemma 10.** If the capacities of all balls are equal, then there exists an alternative procedure to the Selection of Balls 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 Balls 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 $C$ (initialized to $\emptyset$), without expansion.

Otherwise, let $r_l$ denote the radius of a largest 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 3 cases:

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

$\frac{1}{\sqrt{3}} \leq r_l < r_h$: Let $B = B_l$. We set its new radius to be $3r_l + 2r_h$.

$r_l < \frac{1}{\sqrt{3}} r_h$: 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 $C$ respectively. Finally, we set $y_i \leftarrow 1$ for all balls $B_i \in C$, and return $C$ 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 $3 + 2 \cdot \sqrt{3} < 6.47$. 


3.4.2 Soft capacitated version of MMCC

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 a capacity same as that of the original ball. However, we need to pay a cost of 1 for each copy. The LP corresponding to the soft capacitated version, is same as CSC-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 1, we can ensure that the flow that each point receives from the set of non-light balls $(B \setminus L)$ is at least $1 - \alpha$. Then, opening $\frac{1}{1-\alpha} [y_i]$ identical copies of each non-light ball $B_i$ ensures that at least one demand of each point 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 $\alpha = \frac{1}{3}$ yields a simple 4-approximation for this version, where each ball is expanded by a factor of at most 3.

4 Hardness of Approximation

4.1 Hardness of Metric Monotonic Capacitated Covering

In this section, we consider the Metric Monotonic Capacitated Covering (MMCC) problem, and 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. Contrast this result with the earlier result which states that it is NP-hard to obtain a $(o(\log n), c)$-approximation for the MMCC for $1 \leq c < 3$ — the following construction shows that even if we relax the expansion requirement above 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) 3-Dimensional Matching problem.

Consider the Maximum Bounded 3-Dimensional Matching (3DM-3) problem (defined in [21]). In this problem, we are given 3 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 exactly 1, 2 or 3 triples of $T$. A triple $t = (x, y, z) \in T$ is said to cover $x \in X, y \in Y, z \in Z$. The goal is to find a maximum cardinality subset $M \subseteq T$ of triples that does not agree in any coordinate. Here, the 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. We have the following result for the 3DM-3 problem from Petrank [21].

Lemma 4.1 (Restatement of Theorem 4.4 from [21]). 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.

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 the version of the MMCC problem, we are allowed to expand the balls in the input by a constant factor $c \geq 1$.

First, we show how to construct the metric space $(P \cup C, d)$ for the MMCC instance $I'$, that is induced by the shortest path metric on the following graph $G = (P \cup C, E)$. Recall that $C$ is the set of centers, and $P$ is the set of points that need to be covered by the balls centered at centers in $C$. Before describing this graph, we construct some objects that will be useful in the description.

Consider a vertex $c_1 \in C$, that is connected to 4 other vertices $p_1, \ldots, p_4 \in P$ (for convenience we refer to them as left, right, top, bottom vertices respectively) by an edge of weight 1. We also add a ball of radius 1 at the center $c_1$. For convenience, we refer to this object (the 5 vertices and the ball) as a small cluster, and the ball as a small ball. Similarly, if we the radius and the edge
weights of the ball are \( c \), then we refer to such an object as a large cluster, and the ball as a large ball.

Now, consider \( p = \left\lceil \frac{c(c+1)}{2} \right\rceil + 1 \) copies of small clusters, numbered \( \kappa_1, \cdots, \kappa_p \). 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 \( \kappa_i \). 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 its \( \kappa_1 \) as its leftmost small cluster, and to \( \kappa_p \) as its rightmost small cluster. Now, consider a big cluster that is “glued to” two small chains on two sides. That is, the left vertex of the big cluster is the same as the right vertex of the \( \kappa_p \) of a small chain (the left half), and the right vertex of the big cluster is the same as the left vertex of the \( \kappa_1 \) of another small chain (the right half). We call this object (which contains \( 2p \) small clusters and balls, and 1 large cluster and ball) a large chain.

We now describe the element gadget. In an element gadget we consider two large chains \( ch_1, ch_2 \) that are glued together, such that the they share a common small cluster. That is, the rightmost small cluster of the right half of \( ch_1 \) is same as the leftmost small cluster of the left half of \( ch_2 \). Denote this common cluster by \( \kappa \). The respective 3 bottom points of a) The cluster \( \kappa \), b) The cluster leftmost small cluster of the left half of \( ch_1 \), and c) The rightmost small cluster of the right half of \( ch_2 \) are referred to as ideal points. Note that each element gadget contains \( 4p + 1 \) balls and \( 3(4p + 1) + 1 \) vertices.

For each element \( w \in W \), we add an element gadget. Now we describe the triple gadget. This gadget is similar to a large cluster, the only difference is that in addition to the central point \( c_1 \), it contains only 3 other points \( p_1, p_2, p_3 \). We also add a triple gadget for each triple \( t = (x, y, z) \in T \). Now, for each such triple \( t = (x, y, z) \), we set \( p_1, p_2, p_3 \) same as one of the ideal points from the gadgets of \( x, y, z \) respectively. Here, we ensure that if an element is contained in multiple triples, then a different ideal point is assigned to each triple. Finally, we set the capacity of each ball to be 3. The total number of balls in all the element gadgets is \( B := 3N \cdot (4p + 1) \). 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. 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. It is easy to see that the instance \( I' \) has not only monotonic capacities, but even uniform capacities. It is also worth highlighting that there are only two distinct radii in the 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 \( c \geq 1 \).

**Claim 4.1.** Consider the instance \( I' \) of MMCC problem, that is obtained from an instance \( I \) of 3DM-3, using the above reduction. We can always convert a solution to \( I' \), where a selected ball may be expanded by a factor at most \( c \geq 1 \), to another feasible solution in polynomial time, where all element balls in the element gadgets are selected. Furthermore, we can ensure that in this assignment, every selected ball serves a point that is contained in it, (without any expansion).

**Proof.** First, notice that in each element gadget, the number of points is equal to the total capacity of all the balls in the element gadget, plus 1.

Now, consider a solution that does not satisfy the required property. Initially, we discard any balls in the solution if they do not use any of their capacity. Now, consider an element gadget from which a ball is not chosen in the solution. Note that this unchosen ball cannot be the central large ball of any of its chains, because no other ball can cover the top and bottom points corresponding to it, even after expanding by a factor of \( c \). We can also infer from this that, all such large balls have at most 1 capacity that can be used for serving other points. Without loss of generality, we
assume that this remaining capacity is assigned to either left or right point, since one can always find such an assignment.

Now, any unchosen ball has to be a small ball. Without loss of generality, we can assume that 3 of the 4 points that it contains, are not served by any other small ball, since we can always reassign capacities to ensure that that is not the case. Therefore, the points contained in it must be covered by triple balls, such that the corresponding triples cover the corresponding element. However, note that because of the existence of the large ball at the center of a small chain, two different triple balls cannot serve points from a common small ball. Therefore, all 3 points must be served by a single triple ball. But any triple ball also has a capacity of 3, and therefore, we can swap out the triple ball for this small ball, without increasing the cost of the solution.

By repeating this process, we can include all element balls, by swapping out some of the triple balls if necessary. Now at this point, all element balls, as well as enough triple balls are included in the solution, such that the solution is feasible. Now, we assign the capacities of the selected triple balls to the corresponding ideal points. It is easy to see that in each element gadget, at least one ideal point is served by the corresponding triple ball, and therefore, there exists a simple capacity assignment scheme to ensure that each element ball serves only the points contained in it.

**Lemma 4.2.** Consider an instance $I$ of 3DM-3, and let $I'$ be the reduced MMCC instance. In the 3DM instance $I$, $M \subseteq T$ is a minimum size set of triples that covers all the elements of $W$, with $|M| = K$, if and only if the minimum cost of a solution to $I'$, wherein the balls may be expanded by up to a factor $c \geq 1$, is $B + K$.

**Proof.** We first show that given a minimum size cover $M$ of size $K$, of the elements of $W$, how to select $B + K$ balls in the instance $I'$. Firstly, for all triples in $M$, include the corresponding balls in the triple gadget in the solution, which accounts for $K$ balls. Now, for such ball in a triple gadget, assign the capacity to serve the ideal points of the corresponding element gadgets. Now, the number of points in an element gadget that are not served is at most the total capacity of each element gadget. Therefore, by selecting all $B$ balls, each point can be covered by one of the selected balls that it is contained in. Now we prove that there is no solution with a lesser cost.

Consider a minimum cost solution to $I'$, wherein the balls may be expanded by a factor at most $c \geq 1$. Then, we use Claim 4.1 on to obtain another solution of at most the same size, in which all element balls are selected, and no ball is expanded. Now, this solution must have the same size, because the cost of an optimal solution where the balls may not be expanded, is at least the minimum cost of a solution where the balls may be expanded. Therefore, the costs of optimal solutions to the original and the relaxed version of the MMCC instance are equal for the instance $I'$.

In the new solution, each selected triple ball serves some of the ideal points. The number of selected triple balls must be at least $K$, because otherwise the set of selected triple balls corresponds to a cover of the instance $I$ of size less than $K$, which is a contradiction.

Now assume that the minimum cost solution to the instance $I'$, where the balls may be expanded by a factor $c \geq 1$, is $B + K$. Using the argument from the previous paragraph, we can obtain a cover of size $K$ for the 3DM-3 instance $I$. To show that it is optimal, assume for contradiction, a smaller size solution, and use the argument from the first paragraph to obtain a solution to the MMCC instance $I'$ of size smaller than $B + K$, but where the balls are not expanded. Now, this contradicts the optimality of initial solution, because as argued before, the optimal costs of the strict and relaxed versions of the MMCC problem are equal for the instance $I'$.

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 have.
Corollary 1. 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 $\mathcal{B} + N$.

Corollary 2. If in the 3DM-3 instance $I$, the maximum number of elements that can be matched is at most $3\alpha N$ ($0 < \alpha < N$), 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 $\mathcal{B} + \alpha N + \frac{3(1-\alpha)N}{2} = \left(1 + \frac{1-\alpha}{8(3p+1)}\right) \cdot (\mathcal{B} + N)$.

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 $|M| \geq \alpha N + \frac{3(1-\alpha)N}{2}$.

Let $M = M_1 \cup M_2$, where $M_1$ is the maximal set of triples such that each set covers 3 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 2 new elements each, $|M_2| \geq \frac{2}{3}(N - |M_1|)$. Therefore, since $M_1$ and $M_2$ are disjoint, $|M| = |M_1| + |M_2| \geq \alpha N + \frac{3(1-\alpha)N}{2}$. It is easy to verify that the previous quantity is exactly equal to $\left(1 + \frac{1-\alpha}{8(3p+1)}\right) \cdot (\mathcal{B} + N)$, recalling that $\mathcal{B} = 3N \cdot (4p + 1)$. \qed

Now, from Lemma 4.1, Corollary 1, and Corollary 2, we obtain the following APX-hardness result.

Theorem 4. 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 uniform capacitated version of MMCC.\footnote{$\epsilon_c$ depends inversely on $c^2$.}

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 $X_j$. Note that $\sum_{i=1}^{n} m_i = \sum_{j=1}^{m} n_j$. Given any instance $I$ of set cover we construct an instance $I' = (P', \mathcal{B})$ of MMCC-W. Let $[t] = \{1, \ldots, t\}$. Fix a constant $c$, which is the factor by which the balls in the solution are allowed to be expanded, and a real $\alpha > 0$. Let $N = \max\{m, n\}$ and $M = c^{1+1/\alpha} N^{2/\alpha}$. $P$ contains $M \cdot m_i$ points corresponding to each element $e_i$. To describe the
distances between the points we define a weighted graph $G$ whose vertex set is $P \cup C$, where $C$ is the centerpoints of the balls in $B$. The graph contains a set $V_i$ of $2M \cdot m_i - 1$ vertices corresponding to each element $e_i$. The subgraph of $G$ induced by the vertices of $V_i$ is a path of $2M \cdot m_i - 1$ vertices. We denote this path by $\pi_i$. Refer a degree 1 vertex on $\pi_i$ as its $1^{st}$ vertex, the vertex connected to it as the $2^{nd}$ vertex and in general for $i \geq 2$, the index of the vertex connected to the $i^{th}$ vertex other than the $(i - 1)^{th}$ vertex is $i + 1$. 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 $\pi$. For each $i$, 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 $1 \leq i \leq m_i$, we connect the $(i - 1) \cdot M + 1^{th}$ vertex of $\pi_i$ to the vertex corresponding to the set $f(i)$ by an edge of weight $R$. Note that for any set $X_j$, $u_j$ gets connected to $n_j$ vertices of $G$. This concludes the description of $G$.

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 $X_j \in F$, we add the ball $B(u_j, R)$ to $B$ and set its capacity to 1. We note that $B(p, cR/M)$ contains exactly two points. The balls in $B$ 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 $r^{1+\alpha}$.

**Lemma 4.3.** The elements in $X$ can be covered by $k$ sets of $F$ 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.** Let $X$ be covered by a collection $F'$ of $k$ sets. We construct a feasible solution $B' \subseteq B$ to MMCC-W whose weight is at most $(k + 1) \cdot R^{1+\alpha}$. For each set $X_j \in F'$, we add the ball $B(u_j, R)$ to $B'$. We assign the $n_j$ points in $B(u_j, R)$ to it. Now for each $e_i \in X$, we add balls in the following manner. Note that at least one point of $V_i \cap P$ has already been assigned to a ball in $B'$. Now for each point $p \in C$ on $\pi_i$, we add the ball $B(p, cR/M)$ to $B'$. As one point of $V_i \cap P$ is already assigned to a ball in $B'$, only $M \cdot m_i - 1$ points of $V_i \cap P$ are left for assigning. As we have chosen $M \cdot m_i$ balls each of capacity 1 corresponding to $\pi_i$ one can easily find a valid assignment of these points. Thus $B'$ is a feasible solution to MMCC-W. Now the weight of the balls selected w.r.t. the sets is $k \cdot R^{1+\alpha}$. The weight of the balls chosen w.r.t. each path $\pi_i$ is at most $M \cdot m_i (cR/M)^{1+\alpha}$. The total weight of the balls w.r.t. all such paths is at most

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

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

Now suppose there is a solution $B'$ to MMCC-W with weight at most $(k + 1) \cdot R^{1+\alpha}$. The total number of points in $P$ is $\sum_{i=1}^{n} M \cdot m_i > \sum_{j=1}^{n} n_j$. Now the total capacities of the balls w.r.t. the sets is $\sum_{j=1}^{n} n_j$. Thus there must be at least one ball in $B'$ which is w.r.t. a path. Also the weight of the ball w.r.t. each set is $R^{1+\alpha}$. Thus there must be at most $k$ balls w.r.t. the sets in $F$ that are in $B'$. We consider the collection $F' \subseteq F$ of sets corresponding to these balls (at most $k$ in number). We claim that $F'$ covers all elements of $X$. Consider any element $e_i \in X$ and the path $\pi_i$ corresponding to it. Note that $\pi_i$ contains $M \cdot m_i$ points of $P$. Consider one such point $p$ and any center $p'$ on the path $\pi_j$ where $i \neq j$. Now distance between the $p'$ and $p$ is at least $2R$ and thus even after $c$ factor expansion the ball $B(p', cR/M)$ cannot contain $p$. Now consider a center point $u_i$ such that $e_i \notin X_i$. Then due to the construction the distance between $p$ and $u_i$ is at least $3R + (cR/M) \cdot 2M > cR$. Thus even after $c$ factor expansion the ball $B(u_i, R)$ cannot contain $p$.
Hence $p$ must be assigned to either a ball corresponding to $\pi_i$ or a ball corresponding to a set $X_j$ that contains $e_i$. Now there are only $M \cdot m_i - 1$ balls w.r.t. $\pi_i$ in $B$ each of whose capacity is 1 and thus there must be a point $p \in P$ lying on $\pi_i$ that is assigned to a ball corresponding to a set $X_j$. It follows that $e_i \in X_j$. Thus $F'$ covers all the points of $X$.

As set cover is NP-hard to approximate within a factor of $o(\log n)$, from Lemma 4.3, we obtain the following theorem.

**Theorem 5.** For any constant $c \geq 1$, there exists a constant $c' > 0$, such that it is NP-hard to obtain a $(c' \log |P|, c)$-approximation for MMCC-W.

**References**

[1] H. An, M. Singh, and O. Svensson. Lp-based algorithms for capacitated facility location. In *FOCS*, pages 256–265, 2014.

[2] H. An, A. Bhaskara, C. Chekuri, S. Gupta, V. Madan, and O. Svensson. Centrality of trees for capacitated k-center. *Math. Program.*, 154(1-2):29–53, 2015. doi: 10.1007/s10107-014-0857-y. URL https://doi.org/10.1007/s10107-014-0857-y.

[3] J. Bar-Ilan, G. Kortsarz, and D. Peleg. How to allocate network centers. *J. Algorithms*, 15(3):385–415, 1993. URL http://dblp.uni-trier.de/db/journals/jal/jal15.html#Bar-IlanKP93.

[4] P. Berman, M. Karpinski, and A. Lingas. Exact and approximation algorithms for geometric and capacitated set cover problems. *Algorithmica*, 64(2):295–310, 2012.

[5] P. Bose, P. Carmi, M. Damian, R. Y. Flatland, M. J. Katz, and A. Maheshwari. Switching to directional antennas with constant increase in radius and hop distance. *Algorithmica*, 69(2):397–409, 2014.

[6] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete & Computational Geometry*, 14(4):463–479, 1995.

[7] J. Chuzhoy and J. Naor. Covering problems with hard capacities. *SIAM J. Comput.*, 36(2):498–515, 2006.

[8] K. L. Clarkson and K. R. Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete & Computational Geometry*, 37(1):43–58, 2007.

[9] M. Cygan, M. Hajiaghayi, and S. Khuller. LP rounding for k-centers with non-uniform hard capacities. In *FOCS*, pages 273–282, 2012.

[10] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[11] R. Gandhi, E. Halperin, S. Khuller, G. Kortsarz, and S. Aravind. An improved approximation algorithm for vertex cover with hard capacities. *J. Comput. Syst. Sci.*, 72(1):16–33, 2006.

[12] T. Ghasemi and M. Razzazi. A PTAS for the cardinality constrained covering with unit balls. *Theor. Comput. Sci.*, 527:50–60, 2014.

[13] S. Har-Peled and M. Lee. Weighted geometric set cover problems revisited. *JoCG*, 3(1):65–85, 2012.
[14] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985.

[15] M. Kao. Iterative partial rounding for vertex cover with hard capacities. In *SODA*, pages 2638–2653, 2017.

[16] S. Khuller and Y. J. Sussmann. The capacitated $K$-center problem. *SIAM J. Discrete Math.*, 13(3):403–418, 2000.

[17] N. Lev-Tov and D. Peleg. Polynomial time approximation schemes for base station coverage with minimum total radii. *Computer Networks*, 47(4):489–501, 2005.

[18] R. Levi, D. B. Shmoys, and C. Swamy. Lp-based approximation algorithms for capacitated facility location. *Math. Program.*, 131(1-2):365–379, 2012. doi: 10.1007/s10107-010-0380-8. URL https://doi.org/10.1007/s10107-010-0380-8.

[19] S. Li. On uniform capacitated $k$-median beyond the natural LP relaxation. In *SODA*, pages 696–707, 2015.

[20] R. Lupton, F. M. Maley, and N. E. Young. Data collection for the sloan digital sky survey - A network-flow heuristic. *J. Algorithms*, 27(2):339–356, 1998. doi: 10.1006/jagm.1997.0922. URL https://doi.org/10.1006/jagm.1997.0922.

[21] E. Petrank. The hardness of approximation: Gap location. *Computational Complexity*, 4:133–157, 1994. URL http://dblp.uni-trier.de/db/journals/cc/cc4.html#Patrank94;http://dx.doi.org/10.1007/BF01201244.

[22] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

[23] S. C. Wong. Tight algorithms for vertex cover with hard capacities on multigraphs and hypergraphs. In *SODA*, pages 2626–2637, 2017.