Approximate Covering with Lower and Upper Bounds via LP Rounding

Sayan Bandyapadhyay
Department of Informatics, University of Bergen, Norway
Sayan.Bandyapadhyay@uib.no

Aniket Basu Roy
IBISC, University of Evry, France
aniket.basu.roy@gmail.com

Abstract
In this paper, we study the lower- and upper-bounded covering (LUC) problem, where we are given a set $P$ of $n$ points, a collection $B$ of balls, and parameters $L$ and $U$. The goal is to find a minimum-sized subset $B' \subseteq B$ and an assignment of the points in $P$ to $B'$, such that each point $p \in P$ is assigned to a ball that contains $p$ and for each ball $B_i \in B'$, at least $L$ and at most $U$ points are assigned to $B_i$. We obtain an LP rounding based constant approximation for LUC by violating the lower and upper bound constraints by small constant factors and expanding the balls by again a small constant factor. Similar results were known before for covering problems with only the upper bound constraint. We also show that with only the lower bound constraint, the above result can be obtained without any lower bound violation.

Covering problems have close connections with facility location problems. We note that the known constant-approximation for the corresponding lower- and upper-bounded facility location problem, violates the lower and upper bound constraints by a constant factor.

2012 ACM Subject Classification Mathematics of computing → Approximation algorithms

Keywords and phrases lower- and upper-bounded covering, bicriteria approximation, LP rounding

Digital Object Identifier 10.4230/LIPIcs...

Funding This work is partly supported by the Research Council of Norway via the project “MULTIVAL”.

1 Introduction

A ball $B(c, r)$ with center $c$ and radius $r$, in a metric space $(X, d)$, is the set of points at a distance at most $r$ from $c$, i.e., $B(c, r) = \{p \in X \mid d(c, p) \leq r\}$. In this paper, we introduce a generic covering problem, which we refer to as lower- and upper-bounded covering (LUC). In LUC, we are given a set $P$ of $n$ points and a collection $B$ of balls, in a metric space. We are also given lower and upper bound parameters $L$ and $U$, respectively, such that $L \leq U$. The goal is to find a minimum-sized subset $B' \subseteq B$ and an assignment of the points in $P$ to $B'$, such that each point $p \in P$ is assigned to a ball that contains $p$ and for each ball $B_i \in B'$, at least $L$ and at most $U$ points are assigned to $B_i$. We note that LUC is a generalization of the well-studied metric capacitated covering (MCC) problem [4, 5] where $L = 1$. Note that if a ball is selected in a solution of LUC, then at least one point must be assigned to it. Thus, if $L$ is equal to 1, we do not really have a lower bound constraint. Similar to MCC, one can think of natural applications of LUC in wireless networks and facility location. We also study a restricted version of LUC, which we refer to as metric lower-bounded covering (MLC). In MLC, $U$ is equal to $n$. Note that at most $n$ points can be assigned to any ball in a solution of LUC, and thus in case of MLC, we do not really have an upper bound constraint.

Over the years, MCC has received a sufficient amount of attention. An $O(\log n)$-approximation follows for this problem from a classical greedy algorithm due to Wolsey [20].
This approximation is indeed tight, as one can find a reduction from set cover. Consider the following simple construction of a graph. In this graph, each element of the set cover instance is a vertex and also each set is a vertex. If an element is contained in a set, add an edge of distance 1 between them. Now, \( P \) is the set of element vertices, \( B \) is the set of radius-1 balls centered at the set vertices, \( U = n \), and the metric is the shortest path metric in this graph. Then, it is not hard to see that there is a set cover of size \( k \) if and only if there is a solution of MCC of size \( k \). Note that as \( U \) is set to \( n \), the reduction works even when we do not have any upper bound constraint. Thus, even MLC is as hard as set cover. However, to the best of our knowledge, no \( O(\log n) \)-approximation is known for MLC.

As there is no hope for a constant-approximation for MCC, researchers have focused on studying bicriteria approximation. [5] shows that it is possible to obtain a solution whose size is at most a constant times the optimal solution size if we are allowed to expand the balls in the solution by 6.47 factor. Note that as we are allowed to expand the balls in our solution, we have an added advantage over the optimal solution which does not expand the balls. One can show that, in the above reduction, if an element is not contained in a set, then the distance between the element and the set must be at least 3. Thus, in this construction, if a (radius-1) ball is expanded by a factor of \( \alpha < 3 \), it does not contain any extra point than before. It follows that with any \( \alpha < 3 \) expansion factor, one cannot achieve any \( o(\log n) \)-approximation for MCC. In a recent work based on [5], [4] improved the expansion factor for constant-approximation from 6.47 to 4.24. Closing the gap between 3 and 4.24 still remains an open question.

We note that covering problems have a close connection with facility location problems [3, 19, 11]. Indeed, they can be seen as variants of facility location where each point of \( P \) is a client and the center of each ball is a facility with the ball itself being the coverage area of the facility. Moreover, we have an additional constraint that an opened facility can serve a client if the client is within its coverage area. Covering problems are often considered to be much harder compared to facility location problems due to this additional constraint and as the literature proves, relaxation of this constraint leads to better guarantees. Note that true constant-approximations are known for capacitated facility location and lower-bounded facility location [15, 18, 19]. The problem corresponding to the LUC problem, in the facility location literature, is the lower- and upper-bounded facility location (LUFL) problem where each opened facility must be assigned with at least \( L \) and at most \( U \) clients. Friggstad et al. [11] showed that it is possible to obtain a solution for LUFL whose cost is at most a constant times the optimal cost, such that each opened facility in the solution is assigned at least \( L/\beta \) and at most \( \gamma \cdot U \) clients for some constants \( \beta, \gamma > 1 \). In fact, their result holds even for a more general version where each facility has an individual lower bound instead of the uniform lower bound.

1.1 Our Results and Techniques

Our first result is the following theorem.

> **Theorem 1.** There is a polynomial-time approximation algorithm for MLC that returns a solution with the following properties.

- The cost of the solution is at most the optimal cost.
- Each ball in the solution is assigned at least \( L \) points.
- Each ball is expanded by at most a factor of 5.83.

Our result should be compared with the existing results for MCC. Indeed, Theorem 1 shows that one can obtain an exact solution if the balls can be expanded by 5.83 factor. Note
that even a constant-approximation is not possible with any $\alpha < 3$ expansion factor. Our algorithm is much simpler than the algorithms used to obtain the results for MCC. As we argue later, the existing algorithms for MCC violate the lower bound constraint, and hence cannot be used to obtain a result such as Theorem 1.

For the more general LUC problem, we obtain the following result.

**Theorem 2.** There is a polynomial-time approximation algorithm for LUC that returns a solution with the following properties.

- The cost of the solution is at most 15 times the optimal cost.
- Each ball in the solution is assigned at least $L/3$ and at most $5U/3$ points.
- Each ball is expanded by at most a factor of 6.47.

We note that this result is different from the result when we have only lower (or upper) bound constraint, as in the latter case we do not have any lower (or upper) bound violation. However, as mentioned before, even for achieving the corresponding facility location result, constant factor violation of the lower and upper bounds is needed. Note that our violation factors are reasonably small. Our algorithm is built on the LP rounding algorithm in [5] for MCC. But, our analysis is vastly different, as we need to satisfy the lower and upper bounds simultaneously. Moreover, through a more careful analysis, we improve our approximation factor to 15 from their 21 factor. This also improves the best-known approximation factor for MCC from 21 to 15.

We also prove NP-hardness of LUC even in the special case of $L = U = s$ for any constant $s \geq 3$. This result essentially follows from the hardness of partitioning a graph into stars.

### 1.2 Related Work

Euclidean MCC is also a well-studied problem where $P \subseteq \mathbb{R}^d$ and the metric is the Euclidean distance. $d$ is usually considered to be a constant. Researchers have studied two versions of this problem: (i) continuous: one can use any ball in $\mathbb{R}^d$ for the purpose of covering, and (ii) discrete: a predefined set of balls is given from which we need to select the balls. The continuous version appears in the Sloan Digital Sky Survey project [16], and Ghasemi and Razzazi [12] gave a PTAS for this version. There is a constant bicriteria approximation for the discrete version that uses only $1 + \epsilon$ expansion of the balls [5].

The covering problem without lower and upper bounds has been studied in the literature. It follows that even this problem is as hard as set cover. The Euclidean version of this problem received a huge amount of attention. In a seminal work, Brönnimann and Goodrich [7] obtained an $O(1)$-approximation for this problem in the plane. Mustafa and Ray [17] obtained a local-search based PTAS for this planar version. However, no better than $O(\log n)$-approximation is known in dimension more than 2. Har-Peled and Lee [13] obtained a $(1 + \epsilon)$-approximation for this version using $1 + \epsilon$ expansion of the balls.

Facility location is another well-studied optimization problem that is closely related to covering problems. Constant-approximations are known for capacitated facility location using rounding of LP [3] and local search [11,9,15,18]. Lower-bounded facility location is also another well-studied problem for which constant-approximations are known [19,2].

**Organization.** In Section 2, we have some definitions and notation that we will use throughout the paper. In Section 3 and 4, we describe the algorithm for LUC and MLC, respectively. The NP-hardness result appears in Section 5. Finally, we conclude with some open questions.
Recall that in LUC, we are given a set $P$ of $n$ points and a set $B$ of balls. For a ball $B_i \in B$, let $c_i$ and $r_i$ be its center and radius, respectively. First, we describe the ILP of LUC. In this ILP, there are two types of variables $x$ and $y$. For each ball $B_i \in B$, there is a 1/0 variable $y_i$ that denotes whether $B_i$ is selected in the solution or not. For each ball $B_i \in B$ and point $p_j \in P$, there is a 1/0 variable $x_{ij}$ that denotes whether $p_j$ is assigned to $B_i$ or not. The LP relaxation of this ILP is shown in the following. The objective function is simply the sum of all $y$ values. Constraint 1 ensures that if a point is assigned to a ball, then the ball must be selected. Constraints 2 and 3 are to ensure that the total number of points assigned to $B_i$ is at most $U$ and at least $L$. Constraints 4 and 5 ensure that each point is assigned to a ball that contains the point. The remaining constraints specify the domain of the variables.

\[
\min \sum_{B_i \in B} y_i \quad \text{(LUC-LP)}
\]

\[
\begin{align*}
\text{s.t.} & \quad x_{ij} \leq y_i & \forall p_j \in P, \forall B_i \in B \\
& \sum_{p_j \in P} x_{ij} \leq y_i \cdot U & \forall B_i \in B \\
& \sum_{p_j \in P} x_{ij} \geq y_i \cdot L & \forall B_i \in B \\
& \sum_{B_i \in B} x_{ij} = 1 & \forall p_j \in P \\
& x_{ij} = 0 & \forall p_j \in P, \forall B_i \in B \text{ such that } p_j \notin B_i \\
& x_{ij} \geq 0 & \forall p_j \in P, \forall B_i \in B \\
& 0 \leq y_i \leq 1 & \forall B_i \in B
\end{align*}
\]

We denote a solution of the ILP by a tuple $\sigma = (x, y)$. Note that for any solution with integral values of $y$, the objective function correctly denotes the number of balls in the solution. We refer to such a solution with integral values of $y$ as a semi-integral solution, where $x$ can be fractional. We denote the cost $\sum_{B_i \in B} y_i$ of a solution $\sigma = (x, y)$ by $\text{cost}(\sigma)$.

Now, consider any LP solution $(x, y)$. For a point $p_j \in P$ and a ball $B_i$, $p_j$ is said to receive $x_{ij}$ flow from $B_i$. If $x_{ij} > 0$, $B_i$ is said to serve $p_j$. Similarly, for a set $S \subseteq B$, we say $p_j$ receives $\sum_{B_i \in S} x_{ij}$ amount of flow from $S$. For a ball $B_i$, the quantity $\sum_{p_j \in P} x_{ij}$ is called its flow. Next, we define the reroute operation. Consider any two balls $B_i$ and $B_\ell$. For a point $p_j$, rerouting of flow of amount $g$ from $B_i$ to $B_\ell$ means $x_{ij}$ is increased by $g$ and $x_{ij}$ is decreased by $g$. Rerouting of flow from $B_i$ to $B_\ell$ means for each point $p_j$ served by $B_i$, $x_{ij}$ amount of flow is rerouted from $B_i$ to $B_\ell$. Thus, the flow of $B_i$ becomes $0$ after this operation. Next, consider a set of balls $S$ and a ball $B_\ell \notin S$. Rerouting of flow from $S$ to $B_\ell$ means, rerouting of flow from each $B_i \in S$ to $B_\ell$. For a point $p_j$, rerouting of $g$ amount of flow from the balls in $S$ to $B_\ell$ means $x_{ij}$ is increased by $g$ and $x_{ij}$ is decreased by $g_i \geq 0$ for each $B_i \in S$ such that $\sum_{B_i \in S} g_i = g$.

A solution $S$ of LUC is said to violate the lower bound by at most a factor of $\beta > 1$, if for each ball $B_i$ in $S$, the number of points assigned to $B_i$ is at least $L/\beta$. Similarly, a solution $S$ is said to violate the upper bound by at most a factor of $\gamma > 1$, if for each ball $B_i$ in $S$, the number of points assigned to $B_i$ is at most $\gamma \cdot U$. 

2 Preliminaries
3 The Algorithm for LUC

In this section, we assume that $U \geq 2$. Note that if $U = 1$, then the problem becomes a bipartite matching problem between $P$ and $B$, which can be solved in polynomial time. Let OPT be the optimal cost. Fix $0 \leq \alpha < 1/2$. Let $\sigma = (x, y)$ be an optimal fractional LP solution of LUC-LP. Note that $\text{cost}(\sigma) \leq \text{OPT}$. A ball $B_i$ is called heavy w.r.t. $\sigma$ if $y_i = 1$. A ball $B_i$ is called light w.r.t. $\sigma$ if $0 < y_i \leq \alpha$. Let $\mathcal{H}$ and $\mathcal{L}$ be the set of heavy and light balls w.r.t. $\sigma$, respectively. Similarly, one can define sets of heavy and light balls w.r.t. any such LP solution. For simplicity, we do not use $\sigma$ in the notations $\mathcal{H}$ and $\mathcal{L}$ – the reference will be resolved from the context. Our algorithm is based on LP rounding. We start with the fractional solution $\sigma$ and round it to a semi-integral solution. It is sufficient to obtain such a solution, as one can make it fully integral by solving a minimum cost network flow problem. The existence of a feasible fractional flow follows by the semi-integral solution.

Due to integrality of flow and integer lower and upper bounds, we obtain a feasible integral problem. The existence of a feasible fractional flow follows by the semi-integral solution. For any ball $B_i$, do the following.

3.1 Preprocessing

In the first stage of our algorithm, we apply a preprocessing scheme on $\sigma$ to obtain a new fractional solution which we also denote by $\sigma$ for simplicity. The goal of Preprocessing stage is to ensure that each point receives at least $1 - \alpha$ amount of flow from $\mathcal{H}$. This is a crucial property needed in later stages. In this stage, our algorithm is same as the algorithm in [5], except we need to account for lower bound violation of the balls in the preprocessed solution. The algorithm is as follows.

While there is a point $p_j \in P$ that receives more than $\alpha$ flow from $\mathcal{L}$, do the following. Let $T \subseteq \mathcal{L}$ be the set of balls that serve $p_j$ and $S \subseteq T$ such that $\alpha \leq \sum_{B_i \in S} y_i \leq 2\alpha$. Also, let $B_r$ be the largest ball in $S$. Set $y_r$ to 1 and the $y$-value of all other balls in $S$ to 0.

Note that now $B_r$ is a heavy ball. Reroute the total amount of flow from $S \setminus \{B_r\}$ to $B_r$. Lastly, for all ball $B_i$ with $\alpha < y_i < 1$, set $y_i$ to 1.

We note that, in the above, the subset $S$ can be found by a linear scan on $T$, as $p_j$ receives more than $\alpha$ flow from $T$ and $y$-values of all balls in $T$ are at most $\alpha$. Next, we have the following lemma that states the guarantees achieved by the above algorithm.

\begin{itemize}
  \item \textbf{Lemma 3.} When the above algorithm terminates, the following are true.
  \item 1. $\sigma$ satisfies all constraints except 5 and 8.
  \item 2. For any ball $B_i$ with $y_i > 0$, $B_i$ is either heavy or light.
  \item 3. For each point $p_j \in P$, $p_j$ receives at least $1 - \alpha$ amount of flow from $\mathcal{H}$.
  \item 4. For each heavy ball $B_i$, $\sum_{p_j \in P} x_{ij} \geq \alpha L$.
  \item 5. For each light ball $B_i$, $\sum_{p_j \in P} x_{ij} \geq y_i L$.
  \item 6. For each heavy ball $B_i$ and a point $p_j$ that it serves, $d(c_i, p_j) \leq 3 \cdot r_i$.
  \item 7. For each light ball $B_i$ and a point $p_j$ that it serves, $d(c_i, p_j) \leq r_i$.
  \item 8. $\text{cost}(\sigma) \leq \text{OPT}/\alpha$.
\end{itemize}

\textbf{Proof.} First, we show that Constraints [1, 2, and 4] are satisfied. Note that the only times $x_{ij}$ values are changed are when flow is rerouted to a ball $B_r$. Constraint [4] is satisfied, as whenever we reroute flow from a ball to $B_r$, we set $y_r$ to 1.
Note that once $y_i$ is set to 1, it becomes a heavy ball from a light ball. Constraint 2 is satisfied, as after flow rerouting the total flow of $B_i$ is
\[ \sum_{B_i \in S} \sum_{p_j \in P} x_{ij} \leq \sum_{B_i \in S} (y_i \cdot U) \leq 2\alpha \cdot U \leq U. \]
The second inequality follows, as $\sum_{B_i \in S} y_i \leq 2\alpha$. The last inequality follows, as $\alpha \leq 1/2$. Constraint 4 is satisfied as we only reroute flow and in this process no flow gets lost. It is easy to see that Constraint 5 and 7 are also satisfied. This completes the proof of Item 1.

Item 2 follows from the last step of the algorithm. Item 3 follows due to the termination condition of the while loop and the last step.

Note that the total flow assigned to $B_i$ is at least
\[ \sum_{B_i \in S} \sum_{p_j \in P} x_{ij} \geq \sum_{B_i \in S} (y_i \cdot L) \geq \alpha \cdot L. \]
Also, in the last step when a ball $B_i$ is made heavy, its flow must be more than $\alpha \cdot L$, as its $y$-value was more than $\alpha$. Hence, Item 4 follows.

Item 5 follows, as the flow of the light balls that are in the final solution do not change from the initial solution. For the same reason, Item 7 follows.

Item 6 follows, as we reroute flow from other balls in $S$ to the largest ball $B_i$ and all balls in $S$ contain a common point.

Note that whenever we set $y_i$ to 1, we also set $y$-value of all other balls in $S$ to 0. Moreover, $\sum_{B_i \in S} y_i \geq \alpha$. Thus, we can charge the cost of 1 against the $y$-values of the balls in $S$ in the optimal LP solution. Similarly, in the last step, for each ball whose $y$ value is set to 1, we can charge $\alpha$ amount in the optimal LP solution. Hence, Item 8 follows.

### 3.2 Cluster Formation

The input to this step is the preprocessed solution, which we rename to $\sigma = (\overline{x}, \overline{y})$. Note that $\overline{x}$ contains only heavy and light balls. Let $\mathcal{H}_1$ and $\mathcal{L}_1$ be the sets of heavy and light balls. We expand each heavy ball by a factor of 3, to ensure that all the points it serves are in the expanded ball. The $y$-value of heavy balls are already 1. Thus, we need to decide which light balls to select in the solution. We will apply an iterative greedy rounding algorithm to make this decision. If a light ball is selected, we will reroute flow from heavy balls to this ball to utilize its capacity. This, in turn, will make more capacity of heavy balls available, and they can potentially absorb flow from other light balls that are not yet selected. Note that if the flow of a light ball is absorbed by a heavy ball, we can simply remove it from consideration.

During this stage, we maintain a solution $\sigma = (x, y)$ which is initialized to $\overline{\sigma}$. Let $\mathcal{O}$ be the subset of $\mathcal{L}_1$ that the algorithm decides to select, which is initialized to $\emptyset$. For each $B_i \in \mathcal{H}_1$, initialize the cluster of $B_i$, cluster($B_i$) to $\{B_i\}$. Each ball in $\mathcal{L}_1$ is eventually added to either $\mathcal{O}$ or to the cluster of a heavy ball. If it is added to a cluster, then its total flow is assigned to the heavy ball. At any moment in the algorithm, let $\Lambda \subseteq \mathcal{L}_1$ be the set of balls that are not yet added to $\mathcal{O}$ or to the cluster of a heavy ball. During the course of the algorithm, we maintain the invariant that if a point is served by a ball in $\Lambda$, then it gets at least $1 - \alpha$ flow from $\mathcal{H}_1$. The invariant follows from Lemma 3 in the beginning, as initially $\Lambda = \mathcal{L}_1$. For each $B_i \in \mathcal{H}_1$, define the available capacity of $B_i$, $AC(B_i) = (1 + 2\alpha) \cdot U - \sum_{p_j \in P} x_{ij}$. For each $B_j \in \mathcal{L}_1$, $AC(B_j) = U - \sum_{p_j \in P} x_{ij}$. Note that the available capacities might change throughout the algorithm, as the $x$-values might get updated.

While there is a ball in $\Lambda$, apply the following steps.
(1) While there is a ball \( B_j \) in \( \Lambda \) and a ball \( B_i \) in \( \mathcal{H}_1 \) such that \( B_j \cap B_i \neq \emptyset \) and the flow of \( B_j \) is at most \( AC(B_j) \), reroute total flow of \( B_j \) to \( B_i \). Add \( B_j \) to cluster(\( B_i \)). If \( \Lambda \) becomes empty after this step, terminate the algorithm.

(2) For each ball \( B_j \in \Lambda \), let \( A_j \) be the number of points in \( B_j \). Define \( k_j = \min\{A_j, U\} \).

Let \( B_t \) be a ball in \( \Lambda \) having the maximum \( k_j \) value over all \( B_j \in \Lambda \).

(3) Add \( B_t \) to \( \mathcal{O} \). Next, we assign more flow to \( B_t \) to utilize its capacity. There are two cases.

(a) \( k_t = A_t \leq U \). In this case, for each point \( p_t \) in \( B_t \), we reroute its total flow from \( B \setminus \mathcal{O} \) to \( B_t \). Note that after this rerouting, \( p_t \) does not get served by a ball in \( \Lambda \), and thus the invariant is maintained.

(b) \( k_t = U < A_t \). Note that the flow of \( B_t \) is at most \( \alpha \cdot U \), as \( B_t \) is a light ball. Thus, \( AC(B_t) \geq (1 - \alpha) \cdot U \). We consider a subset of points in \( B_t \) of size \( \lfloor AC(B_t) \rfloor \). For each point in this subset, we reroute the total flow from \( B \setminus \mathcal{O} \) to \( B_t \). Note that none of the points in this subset is now being served by the balls in \( \Lambda \), and the invariant is maintained.

When the outermost while loop terminates, \( \Lambda \) is empty. For each ball \( B_l \in \mathcal{O} \), set \( y_l = 1 \) and cluster(\( B_l \)) = \{ \( B_l \) \}.

We note that in contrast to our different definitions of available capacity for heavy and light balls, [5] has only one definition. Indeed, they define it in the same way as we define w.r.t. the light balls. Due to our definition, a ball in \( \mathcal{H}_1 \) can absorb as large as \( (1 + 2\alpha) \cdot U \) amount of flow from balls in \( \mathcal{L}_1 \). Thus, in contrast to [5], we allow violation of upper bounds in this stage. We need this to ensure, when flow is rerouted from a heavy ball to a selected light ball, the flow of the heavy ball does not become too small. Also, for rerouting of flow to \( B_t \), they have three cases (a separate case for \( U = 1 \)), as they allowed upper bound to be non-uniform up to certain extent. We show that we can manage with only two cases. Additionally, in Case (b), we slightly modify their flow rerouting scheme to ensure the desired lower bound violation.

### 3.3 Selection of Balls

The algorithm in this stage is same as the one in [5]. In this stage, we decide which balls to actually select in our solution. For each cluster, we select exactly one ball. If the cluster contains a single ball, we readily select that ball. Otherwise, the cluster must be the cluster of a heavy ball. In this case, we have to be careful to guarantee the desired expansion factor. Let \( B_h \) be the heavy ball and \( B_t \) be the largest ball of \( \mathcal{L}_1 \) that has been added to this cluster. Note that \( B_h \) might already be expanded by 3 factor. There are two cases.

1. \( r_t \geq r_h/\sqrt{3} \). In this case, we select \( B_t \) and set its radius to \( 2r_h + 3r_t \).
2. \( r_t < r_h/\sqrt{3} \). In this case, we select \( B_h \) and set its radius to \( r_h + 2r_t \).

If from a cluster \( B_h \) is not selected, we reroute flow from \( B_h \) to \( B_t \).

It is not hard to see that any selected ball contains all the points it serves. Let \( \mathcal{O}' \) be the set of balls selected in this stage. Note that \( |\mathcal{O}'| = |\mathcal{H}_1| + |\mathcal{O}| \).

### 3.4 The Analysis

Here we analyze our three stage algorithm. Although, our algorithm is similar to the one in [5], the analysis is significantly different. For example, [5] does not have any upper bound violation. But, we need to prove that the violations are bounded by small constant factors in
Approximate Covering with Lower and Upper Bounds via LP Rounding

Thus, such a ball $B_i$ may violate the upper bound. Now, consider a ball $B_i$ in $\mathcal{H}_1$. In the Cluster Selection stage, we set the available capacity $AC(B_i)$ to $(1 + 2\alpha) \cdot U - \sum_{p_j \in P} x_{tj}$. Note that the only step when flow is assigned to $B_i$ is the first step in Cluster Selection. Moreover, we reroute flow from a ball in $\Lambda$ to $B_i$ if the flow of $B_i$ is at most $AC(B_i)$. Thus when the algorithm terminates, the total flow assigned to $B_i$ is at most $(1 + 2\alpha) \cdot U$. Now, in the Selection of Balls stage, all balls in $\mathcal{O}$ are selected, and for each $B_i \in \mathcal{H}_1$, one ball from cluster($B_i$) is selected. If $B_i$ is not selected, a largest ball $B_i$ in cluster($B_i$) is selected, and the flow of $B_i$, which is of amount at most $(1 + 2\alpha) \cdot U$, is rerouted to $B_i$. As the upper bounds are same for all balls, $B_i$ violates the upper bound by at most $1 + 2\alpha$ factor. Hence, no selected ball violates the upper bound by more than $1 + 2\alpha$ factor.

Lemma 4. $O'$ violates the upper bound by at most $1 + 2\alpha$ factor.

Proof. First, note that the balls in the preprocessed solution do not violate the upper bound. Now, consider a ball $B_i$ in $\mathcal{O}$. There can be two cases. In iteration $t$ when $B_i$ is added to $\mathcal{O}$, either (i) $k_t = \mathcal{A}_t \leq U$ or (ii) $k_t = U < \mathcal{A}_t$. In the first case, we assign $\mathcal{A}_t \leq U$ flow to $B_i$. In the second case, the flow assigned to $B_i$ is

$$\sum_{p_j \in P} x_{tj} + \left\lfloor AC(B_i) \right\rfloor = \sum_{p_j \in P} x_{tj} + \left\lfloor U - \sum_{p_j \in P} x_{tj} \right\rfloor \leq U.$$

Here, $x_{tj}$ is the flow received by $p_j$ from $B_i$ in the $t$-th iteration before rerouting of flow. Thus, such a ball $B_i$ does not violate the upper bound. Now consider any ball $B_i$ in $\mathcal{H}_1$. In the Cluster Formation stage, we set the available capacity $AC(B_i)$ to $(1 + 2\alpha) \cdot U - \sum_{p_j \in P} x_{tj}$. Note that the only step when flow is assigned to $B_i$ is the first step in Cluster Selection. Moreover, we reroute flow from a ball in $\Lambda$ to $B_i$ if the flow of $B_i$ is at most $AC(B_i)$. Thus when the algorithm terminates, the total flow assigned to $B_i$ is at most $(1 + 2\alpha) \cdot U$. Now, in the Selection of Balls stage, all balls in $\mathcal{O}$ are selected, and for each $B_i \in \mathcal{H}_1$, one ball from cluster($B_i$) is selected. If $B_i$ is not selected, a largest ball $B_i$ in cluster($B_i$) is selected, and the flow of $B_i$, which is of amount at most $(1 + 2\alpha) \cdot U$, is rerouted to $B_i$. As the upper bounds are same for all balls, $B_i$ violates the upper bound by at most $1 + 2\alpha$ factor. Hence, no selected ball violates the upper bound by more than $1 + 2\alpha$ factor.

Lemma 5. $O'$ violates the lower bound by at most $1/\alpha$ factor.

Proof. Note that in the preprocessed solution $\pi$, no balls in $\mathcal{L}_1$ violate the lower bound. Also, a ball in $\mathcal{H}_1$ may violate the lower bound by at most $\alpha$ factor in $\pi$. Now, during the Cluster Formation stage, balls in $\mathcal{L}_1$ are added to $\mathcal{O}$. Consider any such ball $B_i$ in $\mathcal{O}$. Again, there are two cases: (i) $k_t = \mathcal{A}_t \leq U$ and (ii) $k_t = U < \mathcal{A}_t$. In the first case, for each of the $\mathcal{A}_t$ points in $B_i$, flow is rerouted from $B_i \setminus O$ to $B_i$. By the invariant maintained, each of the points in $B_i$ was getting a flow of at least $1 - \alpha$ from $\mathcal{H}_1$. Thus at least $(1 - \alpha) \cdot \mathcal{A}_t \geq (1 - \alpha) \cdot L$ flow is assigned to $B_i$. In the second case, the amount of flow assigned to $B_i$ is at least

$$\sum_{p_j \in P} x_{tj} + \left\lfloor AC(B_i) \right\rfloor = \sum_{p_j \in P} x_{tj} + \left\lfloor U - \sum_{p_j \in P} x_{tj} \right\rfloor.$$

$x_{tj}$ is again the flow received by $p_j$ from $B_i$ in the $t$-th iteration before rerouting of flow. Now, if $U - \sum_{p_j \in P} x_{tj} \geq 1$, $\sum_{p_j \in P} x_{tj} \geq (U - \sum_{p_j \in P} x_{tj})/2$. Thus, the total flow assigned to $B_i$ is at least

$$\sum_{p_j \in P} x_{tj} + (U - \sum_{p_j \in P} x_{tj})/2 \geq U/2 \geq L/2 \geq \alpha \cdot L.$$

Otherwise, $U - \sum_{p_j \in P} x_{tj} < 1$ and the total flow assigned to $B_i$ is at least

$$\sum_{p_j \in P} x_{tj} > U - 1 \geq U/2 \geq L/2 \geq \alpha \cdot L.$$
The second last inequality follows, as by our assumption $U \geq 2$. Next, we consider a ball $B_i \in H_1$. Before the start of Cluster Formation, $B_i$ may violate the lower bound by at most $\alpha$ factor. Now, if no flow is rerouted from $B_i$ to balls added to $O$ during Cluster Formation, then we are done. Otherwise, let $t$ be the largest iteration number of this stage such that in $t$-th iteration non-zero amount of flow is rerouted from $B_i$ to $B_i'$, which decreases the flow of $B_i$. It follows that $B_i \cap B_i' \neq \emptyset$. Let $f$ and $f'$ be the respective flow of $B_i$ and $B_i'$ before the rerouting. Also, let $A = \text{AC}(B_i)$ at that moment. Note that in the worst case, at most $U$ flow is rerouted to $B_i'$. Thus, after rerouting, the flow of $B_i$ must be at least $f - U$. Now, as $B_i'$ did not join the cluster of $B_i$ even though $B_i \cap B_i' \neq \emptyset$, it must be the case that the flow of $B_i'$ was more than the available capacity of $B_i$, i.e., $f' > A$. But, $f' \leq \alpha \cdot U$, as $B_i' \in L_1$. It follows that, $A = (1 + 2\alpha) \cdot U - f < \alpha \cdot U$. Thus the flow $f$ of $B_i$ was at least $(1 + \alpha) \cdot U$ and after rerouting, it becomes at least $\alpha \cdot U \geq \alpha \cdot L$. Now, $B_i$ itself might not be selected. But, the ball selected from cluster($B_i$) must be assigned with the flow of $B_i$. As the lower bound is same for all balls, the selected ball violates the lower bound by at most $1/\alpha$ factor.

Now the maximum violation factor is $\max\{1/\alpha, 1/(1 - \alpha)\} = 1/\alpha$, as $\alpha \leq 1/2$. Hence, the lemma follows.

Next, we move on towards the analysis of the approximation factor. Note that the total number of balls selected in the solution is $|H_1| + |O|$. Now, $|H_1|$ is at most $\text{OPT}/\alpha$, as the cost of the preprocessed solution, $\text{cost}(\pi) \geq |H_1|$ and $\text{cost}(\pi) \leq \text{OPT}/\alpha$. Thus it is sufficient to give a bound on $|O|$. To do this we are going to introduce a quantity called $y$-accumulation of heavy balls. Roughly, we show that in each iteration $t$, $B_i'$ contributes a constant amount to this quantity. Thus, $y$-accumulation is $\Omega(1) \cdot |H_1|$. To show this we use the argument that in each iteration sufficient amount of flow is rerouted to $B_i'$ and the heavy balls total available capacities get increased by this amount. However, we show that only a bounded amount of $y$-accumulation is possible at each heavy ball, as otherwise it can use the $y$-accumulation to absorb flow of balls in $\Lambda$. Thus, $y$-accumulation is $O(1) \cdot |H_1|$. It follows that, $|O|$ is at most a constant factor of $|H_1|$, and thus a constant approximation follows.

To start with, we define some notations regarding flow rerouting in iteration $t$. For a heavy ball $B_i \in H_1$, let $F(B_i', B_i)$ be the flow rerouted from $B_i$ to $B_i'$. Also, let $F_i = \sum_{B_i \in H_1} F(B_i', B_i)$ be the total flow rerouted from $H_1$ to $B_i'$. Our first observation is the following, which is similar to Lemma 3.2 in [5]. However, we obtain stronger bound due to more careful analysis.

**Lemma 6.** For $1 \leq t \leq I$ and $0 < \alpha \leq 1/3$, $F_i \geq k_i/3$.

**Proof.** Again consider the two cases considered in Cluster Formation. In the first case $k_i = A_i$. In this case, for $A_i = k_i$ points, we reroute flow from $B \setminus O$ to $B_i'$. By the invariant maintained, each of the points in $B_i'$ was getting a flow of at least $1 - \alpha$ from $H_1$. Thus, $F_i \geq (1 - \alpha) \cdot k_i \geq 2k_i/3 \geq k_i/3$.

In the second case, we reroute flow from $B \setminus O$ to $B_i'$ for $|\text{AC}(B_i')|$ points. Note that the flow of $B_i'$ is at most $\alpha \cdot U$, as $B_i' \in L_1$. Thus, $\text{AC}(B_i') \geq (1 - \alpha) \cdot U$. Again, by the invariant, $F_i \geq (1 - \alpha) \cdot (1 - \alpha) \cdot U \geq U/3 = k_i/3$. The last inequality follows, as $U \geq 2$ and $\alpha \leq 1/3$.

Next, we define $y$-accumulation of a heavy ball. First, for each ball $B_i \in H_1$, define its $y$-credit in iteration $t$, $Y(B_i', B_i) = F(B_i', B_i)/k_i$. The $y$-accumulation of $B_i$ at any moment during Cluster Formation, $y(B_i) = \sum_{B_i' \in O} Y(B_i', B_i) - \sum_{B_i' \in \text{cluster}(B_i)} y_{j'}$.

Intuitively, whenever flow is rerouted from $B_i$ to $B_i'$, $B_i$ gains some normalized credit, and whenever it absorbs flow from a ball in $\Lambda$, its credit gets used up. To bound the size of $O$, we obtain a lower and upper bound on the sum of $y$-accumulation of all balls in $H_1$. 

Lemma 7. Suppose $0 < \alpha \leq 1/3$. At the end of Cluster Formation, $\sum_{B_i \in H_1} \tilde{y}(B_i) \geq (|O|/3) - \sum_{B_j \in L_1} \overline{y}_j$.

Proof.

\[
\sum_{B_i \in H_1} \tilde{y}(B_i) = \sum_{B_i \in H_1} \sum_{B'_t \in \mathcal{O}} Y(B'_t, B_i) - \sum_{B_i \in H_1} \sum_{B_j \in L_1 \cap \text{cluster}(B_i)} \overline{y}_j \\
\geq \sum_{B_i \in H_1} \sum_{B'_t \in \mathcal{O}} (F(B'_t, B_i)/k_i) - \sum_{B_j \in L_1} \overline{y}_j \\
= \sum_{i=1}^t (F_i/k_i) - \sum_{B_j \in L_1} \overline{y}_j \\
\geq (|O|/3) - \sum_{B_j \in L_1} \overline{y}_j \quad (F_i \geq k_i/3 \text{ by Lemma 6})
\]

We need the following lemma from [5] for giving the upper bound on $y$-accumulation of heavy balls. The lemma continues to hold even though we define $AC(B_i)$ in a different way.

Lemma 8 (Lemma 3.4, [5]). At any point, for any ball $B_i \in H_1$, $\tilde{y}(B_i) < 1 + \alpha$.

To see this bound, suppose the $y$-accumulation of $B_i$ is at least $1 + \alpha$. Note that initially $y$-accumulation is 0, and whenever flow is rerouted from $B_i$, this quantity gets increased. Thus, when the first time it exceeds the bound of $1 + \alpha$, it must be due to selection of a ball $B'_t$. However, the maximum credit $Y(B'_t, B_i)$ it can get from $B'_t$ is at most 1. Thus, it already had a credit of $\alpha$ which it could have used to absorb the flow from $B'_t$, as the $y$-value of $B'_t$ is at most $\alpha$. Hence, we obtain a contradiction and the bound follows. We note that we need the fact that $k_1 \geq k_2 \geq \ldots \geq k_I$ for proving this lemma. The next lemma shows the desired bound on approximation factor.

Lemma 9. The number of balls selected by the algorithm, $|O| + |H_1| \leq 15 \cdot \text{OPT}$.

Proof. From Lemmas 7 and 8 it follows that $(1 + \alpha) \cdot |H_1| \geq (|O|/3) - \sum_{B_i \in L_1} \overline{y}_i$. Thus,

\[
|H_1| + |O| \leq |H_1| + 3((1 + \alpha) \cdot |H_1| + \sum_{B_i \in L_1} \overline{y}_i) \\
\leq (4 + 3\alpha)(|H_1| + \sum_{B_i \in L_1} \overline{y}_i) \\
\leq (4 + 3\alpha) \cdot \text{cost}(\sigma) \\
\leq (4 + 3\alpha) \cdot \text{OPT}/\alpha \\
= 15 \cdot \text{OPT}. \quad \text{(setting } \alpha = 1/3) \]

The expansion factor again follows from [5], as our algorithm for Selection of Balls is same as the one in [5]. With $\alpha = 1/3$, from Lemmas 4, 5, and 9, Theorem 2 follows. We note that one can use a similar analysis for MCC to achieve the same approximation that does not violate the capacity constraint. Hence, we have the following lemma.

Lemma 10. There is a 15-approximation for MCC that expands the balls by 6.47 factor.

4 The Algorithm for MLC

In this section, we consider the metric lower-bounded covering (MLC) problem. Recall that in MLC, the goal is to find a minimum-sized subset $B' \subseteq B$ and an assignment of the points
in $P$ to $B'$, such that each point $p \in P$ is assigned to a ball that contains $p$ and for each ball $B_i \in B'$, at least $L$ points are assigned to $B_i$. We assume that each ball contains at least $L$ points, as otherwise it is not possible to satisfy its lower bound.

We design a simple LP rounding based exact algorithm for MLC that expands the balls by at most 5.83 factor. It is interesting to note that such a simple algorithm is not known for metric capacitated covering.

Naturally, the ILP formulation of MLC is the same as that of LUC, except here Constraint 2 is absent. We compute an optimal fractional solution $\sigma = (x, y)$ of the LP relaxation of this ILP. Similar to, in the case of LUC, here also we round this fractional solution to a semi-integral solution. One can obtain a fully integral solution by solving a similar minimum cost network flow problem. In the following, we describe our rounding algorithm.

Again, let OPT be the optimal cost. We say a ball $B_j$ is in the 1-neighborhood of another ball $B_i$ if $B_i \cap B_j \neq \emptyset$. We say a ball $B_k$ is in the 2-neighborhood of another ball $B_i$ if there is a ball $B_j$ such that $B_j$ is in the 1-neighborhood of both $B_i$ and $B_k$.

Note that here we consider open neighborhoods, i.e., $B_i$ is not in its 1- and 2-neighborhoods. Also, it is not hard to see that the 1-neighborhood of a ball $B_i$ is a subset of its 2-neighborhood. Our algorithm has two steps. The first step is the coloring step where we color each ball by either red or green. The set of green balls will determine our solution. In the second step, we assign points to these green balls. Now, we describe the details of the two steps.

**First step.** Let $T$ be a set which is initialized to the set of all balls with non-zero $y$ value in $\sigma$. Also, let $R$ and $G$ be the set of red and green balls, respectively, both of which are initially empty. While $T$ is not empty, do the following.

- Remove the largest ball $B$ from $T$ and add it to $G$. Remove all the balls from $T$ that are in the 2-neighborhood of $B$ and add them to $R$.

Set the $y$ value of a ball to 1 if it is in $G$ and to 0 if it is in $R$.

**Second step.** For each ball $B_i \in G$, consider any subset of $L$ points in $B_i$ and fully assign them to $B_i$. (set the $x$ values to 1). Let $P'$ be the set of points assigned to the balls in $G$ in this process. Now, for each ball $B_k \in R$, do the following.

- Set the $y$ value of each such ball is 1 and $x$ values can be at most 1, Constraint 1 is satisfied.

Clearly, we obtain a semi-integral solution. Let us denote it by $\overline{\sigma}$. Next, we analyze our algorithm. We have the following lemma.

> **Lemma 11.** $\overline{\sigma}$ satisfies all the LP constraints except Constraint 2. Moreover, it has the following properties: (i) $\cost(\overline{\sigma}) \leq \OPT$, and (ii) If a ball $B_i$ serves a point $p$ such that $p \not\in B_i$, then $p$ is contained in a ball $B_k$ in the 2-neighborhood of $B_i$, such that $r_k \leq r_i$.

**Proof.** Note that only the balls in $G$ serve the points in $\sigma$. As $y$ value of each such ball is 1 and $x$ values can be at most 1, Constraint 1 is satisfied.

In the second step, the algorithm selects a set of $L$ points in each ball $B_i \in G$ and assigns them to $B_i$. As the balls in $G$ are pairwise disjoint, these sets of points are also pairwise disjoint. Thus, for each ball in the solution, Constraint 3 is satisfied.

For points in $P'$, Constraint 4 is trivially satisfied. For points in $P \setminus P'$, as we only reroute flow from balls in $R$ to the balls in $G$, Constraint 5 is satisfied. It is trivial to verify that the domain constraints are also satisfied.

Now, we prove the moreover part. Note that for each ball $B_i \in G$, there is a point $p_i$ in $P'$ that is fully assigned to $B_i$. Let $T_i$ be the set of balls in the fractional solution $\sigma = (x, y)$ that serve $p_i$. Note that $\sum_{B_k \in T_i} y_k \geq 1$. Now, consider two balls $B_i, B_j \in G$ and
the corresponding sets of balls $T_i$ and $T_j$. We claim that $T_i \cap T_j = \emptyset$. Otherwise, there is a ball $B_k \in T_i \cap T_j$. It follows that $p' \in B_i \cap B_k$ and $p' \in B_j \cap B_k$, and thus $B_i$ is in the 2-neighborhood of $B_j$, and vice versa. But, this is not possible by the definition of $G$. Hence, $T_i \cap T_j$ must be empty. Now,

$$\text{cost}(\sigma) = |G| = \sum_{B_i \in G} 1 \leq \sum_{B_i \in G} \sum_{B_k \in T_i} y_k \leq \sum_{B_i \in G} y_i \leq \text{OPT}$$

The first inequality follows, as $\sum_{B_k \in T_i} y_k \geq 1$. The second inequality follows, as the sets in $\{T_i\}$ are pairwise disjoint.

Finally, consider a ball $B_i$ that serves a point $p$ such that $p \notin B_i$. Note that if $p \in P'$, then $d(c_i, p) \leq r_i$. Thus, $p \in P \setminus P'$. It follows that in the second step of the algorithm, flow was rerouted for $p$ from a ball $B_k$ in $R$ to $B_i \in G$. But, then it must be the case that $B_i$ is the ball in $G$ because of which $B_k$ was forced to join $R$. It follows that $B_k$ is a ball in the 2-neighborhood of $B_i$, and $B_k$ was present in $T$ when $B_i$ was added to $G$. Now, at that moment, $B_i$ was the largest ball in $T$. Hence, $r_k \leq r_i$. This completes the proof of our claim, and hence this lemma follows.

Note that if a ball $B_i$ serves a point $p$ in $\sigma$, $d(c_i, p)$ can still be very large and thus the expansion factor of this solution might not be bounded. In the next lemma, we show how to modify this solution to obtain a new solution with bounded expansion factor.

**Lemma 12.** Given the solution $\sigma$, it is possible to find another LP solution $\hat{\sigma}$ that satisfies all the constraints except Constraint 5 and has the following additional properties: (i) $\text{cost}(\hat{\sigma}) \leq \text{OPT}$, and (ii) If a ball $B_i$ serves a point $p$ in $\hat{\sigma}$, then $d(c_i, p) \leq 5.83 \cdot r_i$.

**Proof.** In the beginning, set $\sigma$ to be $\hat{\sigma}$. We will modify $\hat{\sigma}$ so that it has the desired properties. For each ball $B_i \in G$, consider the largest ball $B_k$ in the 1-neighborhood of $B_i$. If $r_k > \sqrt{2} \cdot r_i$, reroute flow from $B_i$ to $B_k$, and set $\hat{y}_i$ to 0 and $\hat{y}_k$ to 1.

As we just take one ball in the solution $\hat{\sigma}$ for every ball in $G$ and each ball has the same lower bound $L$, it is not hard to see that $\hat{\sigma}$ satisfies all the LP constraints satisfied by $\sigma$. Also, $\text{cost}(\hat{\sigma}) \leq |G| \leq \text{OPT}$.

Next, we argue about the distance between a point $p$ and the center of a ball that serves $p$. Consider any ball $B_i \in G$. From Lemma 11, we know that if $B_i$ serves a point $p$ in $\sigma$ and $p \notin B_i$, then $p$ must be contained in a ball $B_k$ in the 2-neighborhood of $B_i$, such that $r_k \leq r_i$. Now, there can be two cases. In the first case, $r_k \leq \sqrt{2} \cdot r_i$, and thus $B_i$ is chosen in the solution $\hat{\sigma}$. Hence, in the worst case, $d(c_i, p) \leq r_i + 2r_i + 2r_k \leq 3r_i + 2r_k \leq (3 + 2\sqrt{2}) \cdot r_i < 5.83 \cdot r_i$.

In the second case, $r_k > \sqrt{2} \cdot r_i$ and $B_k$ is chosen in the solution. Thus, in the worst case, $d(c_i, p) \leq r_i + 2r_i + 2r_k + 2r_k \leq 3r_i + 4r_k < (3 + 4\sqrt{2}) \cdot r_i < 5.83 \cdot r_i$.

Lemmas 11 and 12 complete the proof of Theorem 1.

## 5 NP-hardness of a Restricted Version of LUC

We consider a special case of LUC when $U = L = c$, where $c$ is a constant and show that even this version is NP-hard. We reduce the Star Partition problem to this special case.

**Star Partition**

Input: A graph $G = (V, E)$ with $n$ vertices and a positive integer $s$ such that $n$ is a multiple of $s + 1$.

Question: Does there exist a partition of $V$ into $V_1 \cup \cdots \cup V_t$ such that $t = n/(s + 1)$ and $G[V_i]$ contains $K_{1,s}$ as a subgraph for $1 \leq i \leq t$?
Note that in the special case of \( s = 1 \), STAR PARTITION boils down to computation of a perfect matching, which can be solved in polynomial time. However, the problem is NP-hard for any \( s \geq 2 \). Next, we describe our reduction.

Given an unweighted graph \( G = (V, E) \), we define a metric \((V, d)\) where \( d \) is the shortest path distance function on \( V \). For every vertex \( v_i \in V \), we define a metric ball \( B_i \) of radius 1 centered at \( v_i \), thus \( B_i \) contains every vertex in the closed neighborhood of \( v_i \). We refer to the set of these \( n \) balls as \( B \). Finally, we set \( P = V \) and \( L = U = s + 1 \). For our convenience, we will use the terms vertex and point interchangeably.

We claim that there exists a partition of \( V \) into \( V_1 \uplus \cdots \uplus V_t \) such that \( K_{1,s} \subseteq G[V_i] \) iff there exists a feasible solution of LUC on the constructed instance with \( t \) balls. Consider the forward direction. Suppose there is such a partition of \( V \). For each \( 1 \leq i \leq t \), consider the set \( V_i \). As \( K_{1,s} \subseteq G[V_i] \), there are at least \( s + 1 \) vertices in \( V_i \) for \( 1 \leq i \leq t \). Additionally, as \( n \) is a multiple of \( s + 1 \), \( V_i \) contains exactly \( s + 1 \) vertices. Select the unit ball \( B_i \) centered at the center of the star \( K_{1,s} \) in \( G[V_i] \), in the solution. Assign the \( s + 1 \) vertices of \( V_i \) to \( B_i \). By the definition of the metric, \( B_i \) contains all the points in \( V_i \). Thus we obtain a feasible solution of LUC with exactly \( t \) balls.

The other direction is very similar. Suppose we are given a feasible solution of LUC with \( t \) balls \( B_1, \ldots, B_t \). Note that \( B_i \) contains exactly \( s + 1 \) points for \( 1 \leq i \leq t \) and the balls in the solution contain all the points in \( P = V \). Thus, these balls define a partition of \( V \) into \( t \) parts each containing exactly \( s + 1 \) vertices. By the definition of the metric, every vertex in \( B_i \) is in the closed neighborhood of the center of \( B_i \) in \( G \). Hence, there exists a \( K_{1,s} \) in \( G \) induced over the vertices corresponding to the points in \( B_i \).

In the light of the above discussion, we obtain the following theorem.

\textbf{Theorem 13.} The LUC problem is NP-hard even if \( L = U \geq 3 \).

\section{Conclusions and Open Questions}

In this paper, we obtained constant bicriteria-approximations for LUC and MLC by expanding the balls by small constant factors. Several questions remain open. One interesting question is to find a constant-approximation for LUC that does not violate any lower and upper bound constraints and expands the balls by a constant factor. Also, one can try to close the gap of 3 and 5.83 for MLC, and 3 and 4.24 for MCC. It would also be interesting to see if MLC admits a true \( O(\log n) \)-approximation. The non-uniform version of covering problems are not well-studied. One can try to find constant bicriteria-approximations for this version.

\begin{thebibliography}{10}

1. Ankit Aggarwal, Anand Louis, Manisha Bansal, Naveen Garg, Neelima Gupta, Shubham Gupta, and Surabhi Jain. A 3-approximation algorithm for the facility location problem with uniform capacities. \textit{Math. Program.}, 141(1-2):527–547, 2013.

2. Sara Ahmadian and Chaitanya Swamy. Improved approximation guarantees for lower-bounded facility location. In Thomas Erlebach and Giuseppe Persiano, editors, \textit{Approximation and Online Algorithms - 10th International Workshop, WAOA 2012, Ljubljana, Slovenia, September 13-14, 2012, Revised Selected Papers}, volume 7846 of \textit{Lecture Notes in Computer Science}, pages 257–271. Springer, 2012.

3. Hyung-Chan An, Mohit Singh, and Ola Svensson. \( L_p \)-based algorithms for capacitated facility location. \textit{SIAM J. Comput.}, 46(1):272–306, 2017.

4. Sayan Bandyapadhyay. Improved bounds for metric capacitated covering problems. \textit{CoRR}, abs/2006.12454, 2020.

\end{thebibliography}
Approximate Covering with Lower and Upper Bounds via LP Rounding

Sayan Bandyapadhyay, Santanu Bhowmick, Tanmay Inamdar, and Kasturi Varadarajan. Capacitated covering problems in geometric spaces. *Discrete & Computational Geometry*, pages 1–31, 2019.

Manisha Bansal, Naveen Garg, and Neelima Gupta. A 5-approximation for capacitated facility location. In Leah Epstein and Paolo Ferragina, editors, *Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, volume 7501 of *Lecture Notes in Computer Science*, pages 133–144. Springer, 2012.

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

Jérémie Chalopin and Daniël Paulusma. Packing bipartite graphs with covers of complete bipartite graphs. In *International Conference on Algorithms and Complexity*, pages 276–287. Springer, 2010.

Fabián A. Chudak and David P. Williamson. Improved approximation algorithms for capacitated facility location problems. *Math. Program.*, 102(2):207–222, 2005.

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

Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Approximating connected facility location with lower and upper bounds via LP rounding. In *15th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2016, June 22-24, 2016, Reykjavik, Iceland*, pages 1:1–1:14, 2016.

Taha Ghasemi and Mohammadreza Razazazi. A PTAS for the cardinality constrained covering with unit balls. *Theor. Comput. Sci.*, 527:50–60, 2014.

Sariel Har-Peled and Mira Lee. Weighted geometric set cover problems revisited. *JoCG*, 3(1):65–85, 2012.

David G. Kirkpatrick and Pavol Hell. On the complexity of general graph factor problems. *SIAM Journal on Computing*, 12(3):601–609, 1983.

Madhukar R. Korupolu, C. Greg Plaxton, and Rajmohan Rajaraman. Analysis of a local search heuristic for facility location problems. *J. Algorithms*, 37(1):146–188, 2000.

Robert Lupton, F. Miller Maley, and Neal E. Young. Data collection for the sloan digital sky survey - A network-flow heuristic. *J. Algorithms*, 27(2):339–356, 1998.

Nabil H. Mustafa and Saurabh Ray. Improved results on geometric hitting set problems. *Discrete & Computational Geometry*, 44(4):883–895, 2010.

Martin Pál, Éva Tardos, and Tom Wexler. Facility location with nonuniform hard capacities. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 329–338. IEEE Computer Society, 2001.

Zoya Svitkina. Lower-bounded facility location. *ACM Transactions on Algorithms (TALG)*, 6(4):1–16, 2010.

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